



HAL
open science

Territorial Prospective by Agent-Based Simulation

Daniel David

► **To cite this version:**

Daniel David. Territorial Prospective by Agent-Based Simulation. Informatique [cs]. Université de la Réunion, 2010. Français. NNT : . tel-00613002v1

HAL Id: tel-00613002

<https://theses.hal.science/tel-00613002v1>

Submitted on 2 Aug 2011 (v1), last revised 18 Oct 2011 (v2)

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

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

Prospective Territoriale par Simulation Orientée Agent

***Daniel
DAVID***

***financement : Région Réunion
encadrement : Rémy COURDIER
contact : daniel.david@univ-reunion.fr***



Laboratoire d'Informatique et de Mathématiques - Université de La Réunion

THESE

pour l'obtention du

Doctorat de l'Université de La Réunion

(spécialité : informatique)

par

Daniel DAVID

Prospective Territoriale par Simulation Orientée Agent

Thèse présentée et soutenue publiquement le 4 décembre 2010

Rapporteurs : Bernard MOULIN - Professeur
Université Laval, Quebec, Canada
Jean-Pierre MÜLLER - Directeur de Recherche
CIRAD/LIRMM, Montpellier

Examineurs : Pierre CHEVAILLIER - Maître de Conférences HDR
LISyC, ENIB, Université Européenne de Bretagne
Michel DIMOU - Professeur
PIMENT, Université de La Réunion
Gilles LAJOIE - Professeur
UMR ESPACE-DEV, Université de La Réunion

Directeur : Rémy COURDIER - Professeur
LIM/IREMIA, Université de La Réunion

Remerciements

Merci à Jean-Pierre Müller et Bernard Moulin d'avoir accepté de prendre sur leur temps précieux pour être les rapporteurs de cette thèse. Merci aussi et surtout pour les échanges productifs que j'ai eu la chance d'avoir avec vous, de visu ou par voie électronique, pré-soutenance ou post-soutenance.

Merci à Pierre Chevaillier d'avoir accepté de participer au jury de cette thèse. C'est toujours un plaisir d'échanger des idées avec toi depuis notre rencontre lorsque tu nous avais accueillis à Brest pour les JFSMA en 2008.

Merci à Gilles Lajoie et Michel Dimou d'avoir également contribué à l'évaluation de cette thèse. Au delà de votre professionnalisme et de votre sympathie, vous avez donné à mon jury une dimension pluridisciplinaire qui me tient particulièrement à coeur. Merci pour ça.

Merci à Rémy Courdier, mon directeur de thèse, de m'avoir encadré durant ces quelques années. Mais surtout merci parce qu'au fil du temps j'ai découvert que j'avais la chance d'avoir un directeur dont les qualités humaines dépassent de loin celles d'un simple encadrant, et ça, franchement, ça aide beaucoup.

Merci à Denis Payet, sans qui une grande partie de mes travaux n'aurait jamais pu voir le jour. C'est là un modèle d'esprit scientifique - à la fois têtu, ouvert, farfelu et brillant - que j'ai la chance de côtoyer depuis quatre ans.

Merci à Tiana Ralambondrainy, Nicolas Sébastien et Yassine Gangat qui ont partagé mon bureau et sont devenus mes amis. Tiana, merci pour ton calme et ta sérénité à toute épreuve ; Nicolas, merci pour bien plus de choses que tu ne l'imagines ; et Yassine, merci surtout pour avoir apporté un peu de fraîcheur après mes années passées enfermée avec les individus précédemment nommés !

Merci à Aurélie Botta, William's Daré, Sigrid Aubert et à tous les membres du projet DOMINO d'avoir été les premiers à me donner ma chance dans le monde du travail en m'accueillant au CIRAD en tant que VCAT au sortir de mon Master 2. Je n'imaginai alors pas à quel point cette expérience allait s'avérer marquante pour la suite donnée à ma jeune "carrière".

Merci à tous les personnels de l'université de La Réunion qui m'ont apporté leur aide et leur soutien au cours des dernières années, essentiellement les membres du Laboratoire d'Informatique et de Mathématiques et toute l'équipe du Pôle Recherche.

Merci aux amis de mon quartier, la Grande-Montée : Cédric, Laurent, Patrick, Johan, Claude-Michel, Henri, Stan, Alexandre, Yann, Ludovic et tous les autres, avec une pensée particulière pour les membres "historiques" du CVSP. Vous êtes là depuis toujours et vu comment vous êtes j'ai la vague impression que vous serez toujours là. Je vous dois beaucoup, merci.

Merci aux amis qui me sont proches depuis le collège et le lycée : Alexandrine, Teddy, Samuel, Evelyne et Virginie. En particulier, merci à Alex d'avoir joué la psy de service pendant les inévitables moments de doute que j'ai pu avoir durant cette thèse, mais surtout merci de m'aider depuis tant d'années à mieux comprendre ce que c'est que l'amitié.

Merci à ceux qui sont restés mes amis depuis nos premières années de fac : Christophe, Loïc et Gil, avec une mention spéciale pour ce dernier. Un jour il y a bien longtemps j'avais rigolé en parlant de toi comme étant "Gil Payet, mon ami exceptionnel". Aujourd'hui je crois que je ne mesure toujours pas à quel point il y avait du vrai là dedans. Alors merci d'être mon Ami, tout simplement.

Merci à tous ceux dont j'ai eu la chance que la vie les ait mis sur mon chemin, essentiellement : Lilian, Romain, Emmanuelle, Aurélie, Julie, Tony, Sébastien, Georges, Safia, Nirmal, Karine, Alice, Pascale, Gaëlle, Audrey, Marie, Mayeul, Ulrich, Didier, Mickael, Laetitia, Indira, Magali, Géraldine, les Charrier, les Payet, les Chouzier, les Ralchy... et tant d'autres que j'oublie de citer aujourd'hui mais qui, je l'espère, ne m'en tiendront pas trop rigueur.

Merci à Zine, Félin, Puce, Pilou et tous les autres pour avoir été de merveilleux compagnons ; merci à Olivier et Angelo pour m'avoir appris à rêver ; et merci à Simba, Nala, Toulouse, Marie et Berlioz pour m'avoir si souvent apaisé.

Enfin, tout simplement, un immense merci à ma famille et en particulier à mes parents. Pour tout, depuis toujours.

- Et où t'installerais-tu, avec ta compagnie de rêveurs ?

- Là où il y a de quoi semer du grain, de quoi se bâtir un abri. Sur une terre libre pour qu'on ne nous la dispute pas. Et facile à défendre car nous ne plairons pas, c'est sûr, à tout le monde. Une île peut-être, et point trop vaste. C'est comme un grand bateau une île, ça doit pouvoir se diriger de pareille façon.

D. Vaxelaire - Les Mutins de la Liberté.

Table des matières

Introduction générale	1
Contexte et motivations	3
La recherche dans l'équipe SMART du LIM	4
Problématique et objectifs	5
Organisation et chronologie des travaux	6
I Etat de l'art et réflexions sur la simulation et la notion d'émergence	11
1 Modélisation et simulation pour la prospective territoriale	13
1.1 Introduction	13
1.2 Modéliser et simuler des systèmes complexes	14
1.2.1 Quelques définitions	14
1.2.2 Modéliser et simuler, pour quoi faire ?	16
1.2.3 Comment faire ?	18
1.3 Modélisation et Simulation Orientée Agent	19
1.3.1 Méthodologies de conception de simulations	19
1.3.2 Les Systèmes Multi-Agents	24
1.3.3 La Simulation Orientée Agent	25
1.4 Prospective territoriale et Simulation Orientée Agent	27
1.4.1 La prospective territoriale	27
1.4.2 Quelques réalisations de l'équipe SMART du LIM	29
1.4.3 Quelques travaux et réalisations connexes	32
1.5 Conclusion	36
2 La notion d'émergence	39
2.1 Introduction	39
2.2 L'émergence ?	40
2.2.1 Positionnement et précisions	41
2.2.2 Quelques exemples simples et visuels	41

2.2.3	Histoire "récente" de l'émergence	45
2.3	Définitions et classifications	46
2.3.1	Définitions célèbres et concepts transversaux	46
2.3.2	Oppositions et classifications	49
2.4	Emergence et simulation	53
2.4.1	Emergence et approches de modélisation multi-niveaux	54
2.4.2	Emergence et outils de simulation	57
2.5	Conclusion	59
II	Propositions conceptuelles, méthodologiques et architecturales pour la prise en compte de l'émergence dans les SOA	61
3	Voir l'émergence comme une méta-connaissance	63
3.1	Introduction	63
3.2	L'émergence, une méta-connaissance	64
3.2.1	L'importance du contexte	64
3.2.2	La méta-connaissance	65
3.2.3	Méta-connaissance et processus d'ingénierie des connaissances	67
3.2.4	Définition	69
3.3	Prendre en compte l'émergence dans une méthodologie de conception de SOA	69
3.3.1	Vers une prise en compte de l'émergence sur le plan méthodologique	69
3.3.2	Retour sur les méthodologies de conception de SOA	70
3.3.3	Quelle est la place de l'émergence dans ces méthodologies ?	71
3.3.4	Méthodologie de conception de simulations	73
3.4	Conclusion	75
4	Méthodologie de réification des phénomènes émergents dans une SOA	77
4.1	Introduction	77
4.2	Réifier l'émergence : la détecter et la matérialiser	78
4.2.1	Détection, matérialisation, réification	79
4.2.2	Trois question complémentaires	79
4.3	Phase de détection des phénomènes émergents	80
4.3.1	Veut-on détecter un phénomène émergent ?	80
4.3.2	Comment est déclenchée cette détection ?	80
4.3.3	A quel niveau doit se faire cette détection ?	82
4.3.4	Bilan et analyse	83
4.4	Phase de matérialisation des phénomènes émergents	85
4.4.1	Le phénomène émergent doit-il être matérialisé ?	85
4.4.2	A quel niveau doit se faire cette matérialisation éventuelle ?	86
4.4.3	Comment est déclenchée cette matérialisation éventuelle ?	87

4.4.4	Bilan et analyse	89
4.5	Conclusion	91
5	Cadre formel et architectural de réification des phénomènes émergents	95
5.1	Introduction	95
5.2	Vers une formalisation de la réification des phénomènes émergents	97
5.2.1	Cadre formel pour la réification des phénomènes émergents	98
5.2.2	Introspection et intercession	98
5.3	Phase d'introspection	99
5.3.1	Faits et connaissances	99
5.3.2	Révélateurs d'émergence et méta-connaissance	102
5.4	Phase d'intercession	104
5.4.1	Agents d'émergence et éléments d'interposition	105
5.4.2	Lois d'émergence et méta-structures émergentes	105
5.5	Construire une SOA adaptée à l'émergence	108
5.5.1	Le rôle des thématiciens dans la conception et l'utilisation d'une SOA	108
5.5.2	Une plate-forme de SOA adaptée	109
5.5.3	Faisabilité et spécificités liées à la plateforme GEAMAS-NG	110
5.6	Conclusion	113
III	Applications et outils pour la prospective territoriale	115
6	Le modèle DS : un outil pour la prospective territoriale à La Réunion	117
6.1	Introduction	117
6.2	La prospective territoriale : une nécessité à La Réunion	118
6.2.1	La Réunion : un territoire aux multiples contraintes	118
6.2.2	Des espaces naturels et agricoles mités par l'urbanisation	120
6.2.3	L'importance des documents de planification	120
6.3	L'ATP DOMINO du CIRAD	124
6.3.1	L'ATP DOMINO à La Réunion	124
6.3.2	Réalisations issues de l'ATP DOMINO	124
6.4	Le modèle DS	126
6.4.1	DS et les atouts de GEAMAS-NG	126
6.4.2	L'outil DS côté utilisateur	128
6.4.3	Sorties et résultats	133
6.4.4	Quelques résultats exploités régionalement	136
6.5	Conclusion	137
7	L'initialisation : charger et utiliser des données cartographiques	143
7.1	Introduction	143
7.2	Des cartes colorées comme plans d'informations	144

7.2.1	L'initialisation : un point clé dans les simulations	144
7.2.2	Les cartes, un vecteur de transmission idéal	145
7.2.3	Quelques cartes plans d'informations	145
7.3	Le langage XELOC	149
7.3.1	Les bases du Langage	150
7.3.2	Configurer agents et environnements	152
7.3.3	Charger les cartes pour le modèle DS	155
7.4	Conclusion	157
8	Le modèle DS : dynamiques et émergence de nouvelles zones urbaines	159
8.1	Introduction	159
8.2	Une modélisation adaptée à la simulation territoriale	160
8.2.1	La modélisation orientée dynamique	160
8.2.2	Vue d'ensemble du modèle	161
8.2.3	Couplage de dynamiques via l'environnement	164
8.2.4	Couplage et retour méthodologique	168
8.3	L'émergence de nouvelles zones urbaines	171
8.3.1	Un phénomène bien particulier	171
8.3.2	Analyse et mise en place architecturale	173
8.3.3	Détection dans le modèle DS	174
8.3.4	Matérialisation dans le modèle DS	179
8.4	Conclusion	187
	Conclusion générale	191
	Synthèse des travaux	191
	Contributions	193
	Perspectives proches et lointaines	195
	Bibliographie	199
	Table des figures	209
	Table des définitions	213
	Résumé & Abstract	215

Introduction générale

Prospective Territoriale par Simulation Orientée Agent

Préambule

Nous commencerons ce manuscrit en précisant que le "nous" employé tout au long des différents chapitres n'est pas simplement académique et, qu'en ce sens, il ne réfère pas toujours aux travaux d'une seule personne. Cette précision s'impose de par la coutume, mais elle s'impose aussi (et surtout !) en raison de l'implication exemplaire des membres de toute une équipe dans la production des travaux présentés ici qui sont souvent issus de projets impliquant de nombreux acteurs, et qui sont quasiment toujours le fruit de réflexions menées à plusieurs.

Contexte et motivations

L'île de la Réunion, qui possède un fort développement sur un territoire limité, doit gérer de nombreux phénomènes de croissance et les conséquences qui en découlent. Au cours des prochaines années, nous devons en effet faire face à une augmentation de la population estimée comme atteignant les 30% sur 20 ans ! Cette simple évolution démographique ouvre la porte à de nombreuses questions, notamment celle du logement : au vu de l'existant, et en tenant compte de 200 000 habitants supplémentaires d'ici 2030, le besoin de création en logements est estimé à plus de 100 000. Même avec une hypothèse de densification forte, ceci se traduirait par une demande de foncier urbain de plusieurs milliers d'hectares alors que seuls 1700 hectares sont actuellement réservés à l'urbanisation !

L'évolution du sol réunionnais doit donc se faire selon une politique d'urbanisation claire et les documents de planification réglementant l'évolution de l'urbanisation de l'île devront prendre en compte au mieux ces projections, car il est clair que l'urbanisation à tout va n'est pas un scénario viable à long terme.

Sur un territoire aussi restreint que le nôtre, différents enjeux sont effectivement à préserver au mieux relativement aux évolutions inhérentes à l'évolution démographique. Ce ne sont en effet pas moins de 100 000 hectares d'espaces naturels toujours protégés et maîtrisés qui ne

demandent qu'à être conservés au mieux afin de favoriser la préservation d'une biodiversité qui fait la richesse de l'île. Si nous voulons admirer encore longtemps tuit-tuits, pieds de faham et autres pétrels – trois espèces locales menacées – il convient donc de tenir compte du mieux possible dans nos choix d'urbanisation de cet objectif de conservation de la biodiversité.

Par ailleurs, quelques 40 000 hectares du territoire réunionnais sont aujourd'hui occupés par une activité agricole qui constitue, avec ses 15 000 emplois générés, un des réels poumons économiques de l'île. Une telle occupation de l'espace par l'agriculture constitue un formidable outil d'aménagement du territoire et protéger ces espaces agricoles, notamment les espaces canniers, constitue donc également une priorité, et l'on se doit de veiller à freiner leur consommation par l'urbanisation en limitant les extensions urbaines et en empêchant le grignotage des meilleures terres agricoles par le mitage.

Comme relevé dans [Agorah 2006], en matière d'aménagement du territoire, La Réunion doit donc relever le défi d'accueillir une population de plus en plus importante tout en valorisant son terroir agricole et en protégeant ses espaces naturels et ses paysages exceptionnels.

La recherche dans l'équipe SMART du LIM

Les travaux de recherche présentés dans ce manuscrit ont été réalisés au LIM, le Laboratoire d'Informatique et de Mathématiques de l'université de La Réunion, et plus particulièrement au sein de l'équipe SMART¹ de l'IREMIA².

Cette équipe rapproche deux problématiques informatiques liées autour de la notion de communication en réseau en considérant d'une part la vue système technologique de télécommunication s'attachant aux services sur les réseaux et d'autre part la vue applicative au travers d'outils de simulations de systèmes complexes en proposant et en utilisant des techniques du monde des systèmes multi-agents.

En particulier, l'équipe s'intéresse à la problématique d'ingénierie de conception de systèmes de simulation par approche multi-agents avec une démarche d'intégration de cette technologie comme réponse à des besoins émanant de partenaires scientifiques. La Simulation Orientée Agent (SOA) permet de construire de véritables micro-mondes artificiels dont on peut contrôler différents paramètres (quantitatifs ou qualitatifs) et cela à tous les niveaux du système, qu'il s'agisse de l'entité, du groupe, de la société ou au niveau des effets externes sur l'environnement.

Notre activité dans ce domaine est traditionnellement structurée selon trois axes :

- le développement d'une plateforme multi-agents dédiée à la simulation de systèmes complexes naturels et sociaux, nommée GEAMAS [Soulié *et al.* 1998], actuellement en version NG (New Generation) [Payet *et al.* 2006a];

¹ Systèmes Multi-Agents et Réseaux de Télécommunication

² Institut de REcherche en Mathématiques et Informatique Appliquées

- la réalisation d’expérimentations et développement d’applications partenaires réalisées en collaboration avec des chercheurs de différents horizons (universités, CIRAD, IRD, IFREMER, etc.) au sein de projets pluridisciplinaires [Courdier *et al.* 2002, David *et al.* 2007, Gangat *et al.* 2010];
- la proposition de démarches méthodologiques en ingénierie de conception de SMA pour la simulation [Marcenac & Giroux 1998, Vally & Courdier 1999, Caldéroni 2002, Payet *et al.* 2006b, Payet *et al.* 2009].

Dans le cadre de notre "jeune" recherche, les travaux de Tiana Ralambondrainy portent sur l’observation dans les simulations large échelle [Ralambondrainy 2009], ceux de Nicolas Sébastien portent sur la distribution et la parallélisation des simulations [Sébastien 2009, Sébastien *et al.* 2009], et ceux, naissants, de Yassine Gangat portent sur la problématique de la modélisation comportementale. De façon générale, notre plateforme de simulation GEAMAS-NG nous sert de base d’expérimentation à double titre : elle est utilisée pour réaliser des outils d’aide à la décision et nous l’utilisons pour confronter nos propositions aux utilisateurs.

Problématique

Le contexte réunionnais fait qu’il est aujourd’hui important de développer des outils qui soient dédiés à la prospective territoriale. En tant qu’informaticiens, nous nous proposons de mettre le potentiel des Simulations Orientées Agent (SOA) au service d’une telle problématique afin d’obtenir des résultats qui fournissent des visualisations prédictives de l’avenir des territoires réunionnais. En espérant que ces outils puissent aider à mettre en place une organisation des territoires qui soit la plus cohérente possible en donnant des pistes de réflexions aux décideurs confrontés aux choix de demain, comme l’élaboration des SCoT (Schémas de Cohérence Territoriale) et du SAR (Schéma d’Aménagement Régional) et leur mise en œuvre effective au fil des années.

Dans une telle démarche, la conception des outils de simulation nécessite de prendre en compte, à une échelle suffisamment fine, de nombreux paramètres comme l’état des lieux, les réglementations en vigueur, la forte croissance de la population, ou encore les différentes volontés de conservation ou d’urbanisation. Ceci implique généralement la présence dans la SOA d’un grand nombre d’entités ayant des comportements et des interactions riches et variées. Et au niveau de la modélisation, processus qui amorce la conception de nos outils de simulation, cela implique d’avoir recours à des connaissances qui proviennent de thématiciens dont les domaines d’expertises varient énormément (géographes, économistes, sociologues, ...) et qu’il est donc difficile de réunir et d’exploiter.

Si les méthodologies existantes détaillent bien le rôle de chacun des intervenants (thématiciens, modélisateurs, informaticiens) aux différentes étapes d’une démarche de conception de simulations, elles insistent moins sur la façon dont l’expertise des différents thématiciens est utilisée et injectée dans les modèles ; et elles n’abordent pas la façon dont doivent être pris

en compte les phénomènes qui émergent dans une simulation et dont l'observation constitue pourtant l'un des résultats les plus intéressants. Ceci étant d'autant plus dommageable que les SOA, lorsqu'elles sont dédiées à la prospective territoriale, sont souvent constituées d'agents évoluant au sein d'environnements spatialisés et les simulations sont alors particulièrement propices à l'émergence de phénomènes.

Au vu de cet ensemble d'éléments, nous pensons qu'il est essentiel de se poser les deux questions suivantes : comment améliorer la prise en compte de l'émergence dans des SOA ? et comment améliorer l'intégration dans nos modèles et outils de connaissances provenant de différents experts ?

Objectifs des travaux

Relativement à la problématique que nous venons d'énoncer, les travaux réalisés dans le cadre de cette thèse ont pour objectif principal de proposer un système qui soit capable de manipuler les phénomènes émergents susceptibles d'apparaître dans une SOA.

Dans notre communauté, plusieurs travaux s'intéressent à ce que nous appelons la "réification" des phénomènes émergents. Mais ils concernent souvent des phénomènes spécifiques et les mécanismes qui sont mis en place ne sont pas facilement réutilisables dans d'autres contextes. Notre approche devrait permettre d'aller un peu plus loin : allant de la théorie à la pratique, nous avons l'ambition d'identifier et de définir les concepts clés liés à l'émergence de phénomènes afin de proposer une formalisation qui leur soit adaptée.

L'idée étant d'implémenter dans notre plateforme de simulation GEAMAS-NG des mécanismes génériques capables d'utiliser cette formalisation pour améliorer la prise en compte – par les thématiciens observant la simulation ou par le système s'observant lui-même – des éventuels phénomènes émergents. Notre but applicatif étant de pouvoir utiliser ces mécanismes pour accroître les fonctionnalités de DS, un modèle développé pour simuler l'évolution de la population et des territoires à La Réunion, et d'apporter ainsi de nouveaux éléments utiles aux réflexions liées à la prospective territoriale.

Enfin, nous avons pour objectif de proposer des méthodes et outils facilitant la conception de modèles qui intègrent (et qui produisent !) des connaissances qui n'ont du sens que pour certains de leurs multiples co-concepteurs et utilisateurs.

Organisation et chronologie des travaux

Comme souvent dans une démarche de recherche scientifique, l'ordre dans lequel les travaux sont présentés dans ce manuscrit ne suit pas l'ordre chronologique dans lequel ils ont été réalisés. De façon originale, certains éléments présentés dans la troisième partie de cette thèse, "Applications et outils pour la prospective territoriale", font même partie des premières contributions à avoir vu le jour chronologiquement parlant.

En particulier, les premières versions du modèle DS ont été réalisées en 2007 dans le cadre

de l'ATP DOMINO conduit par le CIRAD Réunion, et le contexte de leur réalisation a servi de genèse à une partie des travaux qui ont été réalisés au cours des années suivantes.

Les propositions conceptuelles et méthodologiques présentées dans la deuxième partie de ce manuscrit sont donc apparues chronologiquement après certains des outils présentés dans la troisième partie, mais le lien entre elles existe bel et bien, puisqu'en quelque sorte les éléments méthodologiques de la deuxième partie sont les fruits des réflexions qui ont suivi la création des outils de la troisième partie, qu'elles ont ensuite permis d'enrichir *a posteriori*.

Ce document est organisé en 8 chapitres qui sont répartis dans 3 grandes parties. Cette organisation est représentée sur la figure 1 sur laquelle apparaissent également les liens logiques qui existent entre les différents chapitres.

La première partie regroupe des éléments d'état de l'art et des réflexions sur la simulation et sur la notion d'émergence. En particulier, le chapitre 1 présente un ensemble d'éléments relatifs à la modélisation et à la simulation de systèmes complexes à des fins de prospective territoriale, et le chapitre 2 prend pour cible l'émergence, une notion que nous avons identifiée comme étant un élément sur lequel les méthodologies de conception de simulations existantes étaient peu focalisées.

La deuxième partie regroupe un ensemble de propositions sémantiques, méthodologiques, conceptuelles et architecturales pour améliorer la prise en compte de l'émergence dans les SOA. En particulier, le chapitre 3 permet de définir l'émergence comme étant une méta-connaissance, définition sur laquelle nous nous appuyons par la suite pour définir un cadre méthodologique de conception de simulations dans lequel les notions de connaissance et de méta-connaissance sont représentées explicitement. Le chapitre 4 étend cette méthodologie puisqu'il permet de cadrer de façon spécifique les choix à opérer qui permettent de distinguer différentes façons de prendre en compte les phénomènes qui peuvent émerger dans une Simulation Orientée Agent. Ceci se fait *via* la définition du concept de réification des phénomènes émergents, un processus qui voit se succéder une phase de détection et une phase de matérialisation. La présentation des possibilités de prise en compte des phénomènes émergents aboutit à une classification des différents types d'émergences réifiables. Le chapitre 5 propose quant à lui un cadre formel et architectural permettant de mettre en oeuvre certaines des propositions précédentes dans le cadre d'une SOA.

La troisième partie regroupe la description d'un ensemble d'outils et applications pour la prospective territoriale. En particulier, le chapitre 6 présente le modèle DS qui permet de simuler, selon différents scénarios de simulation, l'évolution de la population et l'évolution des trois grandes classes d'occupation du sol (espaces naturels, espaces agricoles, et espaces urbains) sur le territoire de l'île de La Réunion. Le chapitre 7 revient sur un point clé souvent négligé dans les simulations, l'importance de la phase d'initialisation, et présente une méthode d'initialisation utilisée dans DS qui est basée sur l'utilisation de cartes colorées sémantiques et qui présente l'avantage de faciliter à la fois les interactions homme-machine et les interactions

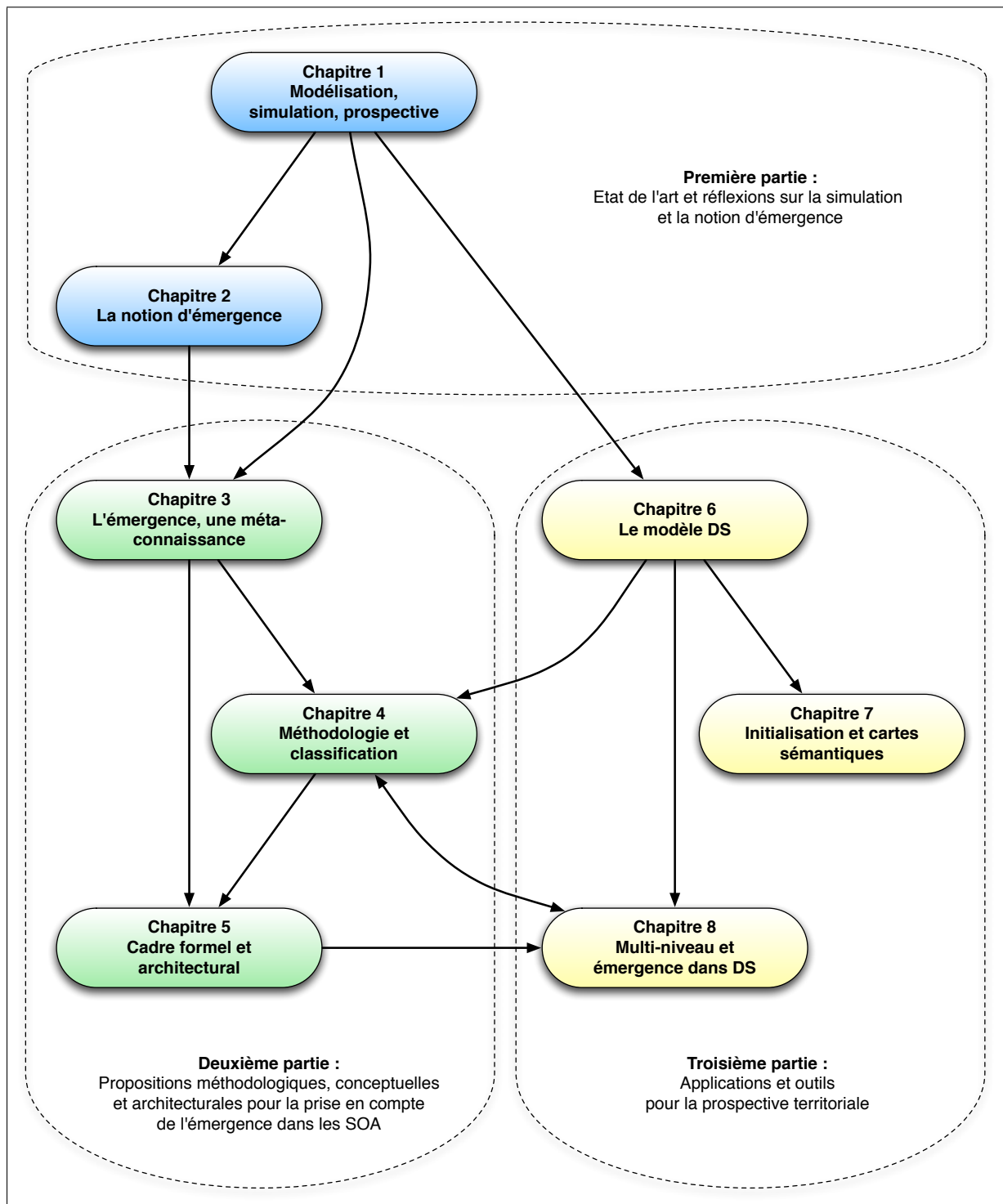


FIG. 1 – Organisation des parties et chapitres de la thèse

thématicien-informaticien. Enfin le chapitre 8 présente le fonctionnement interne de DS, et notamment la façon dont sont articulées les deux dynamiques qui y coexistent, une dynamique d'évolution de la population et une dynamique d'évolution de l'espace. Ces dynamiques étant l'une à un niveau microscopique et l'autre à un niveau macroscopique, le lien est alors fait avec les chapitres de la seconde partie et les éléments relatifs à la prise en compte de l'émergence dans les SOA.

L'organisation des chapitres de cette partie peut paraître surprenante, mais, dans la lignée des premiers travaux effectués sur le modèle DS durant l'ATP DOMINO, cette thèse a eu la chance de voir la réalisation d'un projet d'ampleur. D'où notre volonté de présenter d'une part DS en tant qu'outil tel qu'il est utilisé et valorisé régionalement (dans le chapitre 6), et de présenter d'autre part le modèle, son fonctionnement sous-jacent, et les expérimentations liées aux phénomènes émergents qui s'y produisent (dans le chapitre 8). Le chapitre 7 venant s'intercaler entre ces deux chapitres pour donner une place plus cohérente à l'initialisation dans la démarche générale.

Ces parties et chapitres peuvent être lus de façon relativement indépendante mais, regroupés ensemble, ils forment les Parties d'un Tout qui n'en prend que plus de sens et qui n'en devient que plus cohérent.

Première partie

Etat de l'art et réflexions sur la simulation et la notion d'émergence

Modélisation et simulation pour la prospective territoriale

Sommaire

1.1 Introduction	13
1.2 Modéliser et simuler des systèmes complexes	14
1.2.1 Quelques définitions	14
1.2.2 Modéliser et simuler, pour quoi faire?	16
1.2.3 Comment faire?	18
1.3 Modélisation et Simulation Orientée Agent	19
1.3.1 Méthodologies de conception de simulations	19
1.3.2 Les Systèmes Multi-Agents	24
1.3.3 La Simulation Orientée Agent	25
1.4 Prospective territoriale et Simulation Orientée Agent	27
1.4.1 La prospective territoriale	27
1.4.2 Quelques réalisations de l'équipe SMART du LIM	29
1.4.3 Quelques travaux et réalisations connexes	32
1.5 Conclusion	36

1.1 Introduction

Dans nos travaux, nous nous intéressons essentiellement à la modélisation et à la simulation de systèmes complexes naturels et sociaux à des fins de prospective territoriale. Dans ce premier chapitre, nous allons donc évoquer un ensemble d'éléments qui serviront à poser les bases des réflexions et propositions qui seront développées dans les parties et chapitres suivants. Nous nous focaliserons essentiellement sur les notions de modélisation et de simulation, puis nous nous attarderons sur la technique de simulation qui est la nôtre, la Simulation

Orientée Agent (SOA). Enfin, après avoir présenté différentes méthodologies de conception de simulations, nous ferons le lien entre nos travaux d'informaticiens et leur champ d'application, la prospective territoriale, un domaine localement en plein essor dont nous décrirons rapidement les principes généraux. En guise de conclusion nous présenterons des problèmes et manques que nous avons identifiés et qui sont ceux qui ont guidé nos pas durant la réalisation des travaux présentés dans la suite de ce manuscrit.

1.2 Modéliser et simuler des systèmes complexes

Lorsque l'on énonce le titre de cette section "Modéliser et simuler des systèmes complexes" à un ami complètement néophyte en la matière, cette personne soulève généralement trois questions majeures :

- qu'est-ce que ça veut dire exactement ?
- à quoi ça sert ?
- comment est-ce qu'on fait ?

A propos de l'une de ces questions, Epstein tient les propos suivants dans [Epstein 2008] :

The first question that arises frequently — sometimes innocently and sometimes not — is simply, "Why model?" Imagining a rhetorical (non-innocent) inquisitor, my favorite retort is, "You are a modeler." Anyone who ventures a projection, or imagines how a social dynamic — an epidemic, war, or migration — would unfold is running "some" model.

Cette réponse, non dénuée d'humour, mais surtout pleine de bon sens, donnera une bonne vision des choses aux personnes néophytes en la matière puisque, dans sa réponse à la question "pourquoi modéliser ?", Epstein leur offre le sens commun de ce qu'est en réalité un modèle. Pour aller au bout des choses, nous allons dans la suite présenter quelques définitions afin de mieux cerner les concepts de "modèle" et de "simulation", mais également ce qu'est un "système complexe" puisque c'est justement ce qu'il s'agit de modéliser et de simuler.

1.2.1 Quelques définitions

Pour parvenir à modéliser un système complexe, puis à le simuler, il convient de se fixer au préalable sur le sens précis de ces notions. La définition classique d'un modèle est celle établie par Minsky qui précise dans [Minsky 1965] que, pour un observateur B, un objet A* est un modèle d'un objet A dans la mesure où B peut utiliser A* pour répondre aux questions qui l'intéressent à propos de A.

Suivant une vision similaire, mais énoncée de façon moins formelle, nous nous conformerons à la définition suivante établie par Hagget dans [Hagget 1965] :

Définition I.1 (Modèle) *Un modèle est une représentation schématique de la réalité, élaborée en vue de la comprendre et de la faire comprendre.*

Dans le contexte de nos travaux, il convient de distinguer différents types de modèles, ou plutôt de raffiner la définition précédente d'un modèle en fonction du contexte dans lequel l'on se trouve. Trois types de modèles sont ainsi définis dans [Ralambondrainy 2009] : le modèle du domaine, le modèle informatique, et le modèle simulé.

Définition I.2 (Modèle du domaine) *Modèle réalisé à partir du monde réel, contenant les connaissances des spécialistes de la partie du monde réel considérée.*

Définition I.3 (Modèle informatique) *Modèle correspondant à l'implémentation informatique du modèle du domaine.*

Définition I.4 (Modèle simulé) *Modèle qui est une instance du modèle informatique initialisée avec des paramètres définis et qui peut évoluer au cours d'un temps virtuel.*

L'enchaînement de ces définitions, et notamment la notion de modèle simulé, permet d'aboutir à la définition de ce qu'est une simulation :

Définition I.5 (Simulation informatique) *Une simulation informatique correspond à l'évolution d'un modèle simulé dans un temps virtuel.*

Mais d'une façon générale on pourra se conformer pour cette dernière à une définition un peu plus vulgarisée, qui fait abstraction de la notion explicite de modèle, qui est donnée par Banks dans [Banks 2000] :

Définition I.6 (Simulation) *Simulation is the imitation of the operation of a real-world process or system over time.*

Maintenant que sont posées les définitions d'un modèle et d'une simulation, nous pouvons donner la définition suivante, reprise dans [Ralambondrainy 2009] et [Sébastien 2009], pour décrire un système complexe :

Définition I.7 (Système Complexe) *Un système complexe est une sous-partie du monde réel composée d'éléments en interactions, qui fait l'objet de modélisation et de simulation à des fins de compréhension du monde réel, la complexité pouvant provenir du nombre important de parties du système qui interagissent d'une manière non simple, et conduisant à de l'imprévisibilité possible, de l'émergence du nouveau et du sens plausible.*

Maintenant que ces notions de bases sont explicitées, et qu'il dispose d'une vue plus rigoureuse du contexte et des termes utilisés lorsque l'on parle de modélisation et de simulation, nous pouvons à présent expliquer à notre ami néophyte l'intérêt qu'il y a à se lancer dans une telle démarche.

1.2.2 Modéliser et simuler, pour quoi faire ?

Depuis la deuxième moitié du XX^{ème} siècle, les techniques, les recherches et les moyens scientifiques et technologiques qui sont à notre disposition ont connu des avancées prodigieuses. Ceci est particulièrement vrai dans le domaine des Technologies de l'Information et de la Communication.

Lorsqu'Alan Turing, dans les années 30, présenta sa fameuse machine éponyme [Turing 1937], premier calculateur universel programmable, il ne se doutait pas qu'il donnait ainsi l'élan qui permettrait à notre discipline, l'Informatique, d'apparaître et de décoller comme une flèche encore jamais retombée depuis ; et dont on peut sérieusement se demander si elle le fera un jour. Nos grands-parents ont connu une époque à laquelle sont apparus les premiers ordinateurs que furent le Colossus, l'ENIAC, ou le Bull Gamma, tous complètement inaccessibles (voire même inconnus !) pour le grand public. Nos parents ont connu l'apparition de l'Apple II et du Commodore 64. Et aujourd'hui, alors que nous avons connu l'époque des premiers PC Pentium, des PowerMac Apple, et des premiers supercalculateurs, nous vivons déjà dans un monde où l'iPad et l'iPhone deviennent des objets courants, et où l'accès à des fermes de calcul est à la portée du premier quidam venu en quelques clics sur Internet !

Ces ressources prodigieuses font que la puissance technologique est aujourd'hui disponible à des coûts de développement raisonnables. Et ceci pose la modélisation et la simulation informatique, auxquels on ne pensait même pas autrefois, comme étant à présent des outils performants et accessibles pour représenter des systèmes complexes. Epstein, toujours lui, résume alors l'utilité de modéliser et simuler un système complexe en listant une quinzaine d'objectifs qui peuvent chacun justifier de se lancer dans une telle démarche [Epstein 2008] :

- expliquer (ce qui est bien différent de prédire) ;
- aiguiller la collecte des données ;
- mettre en avant des dynamiques principales ;
- suggérer des analogies dynamiques ;
- découvrir de nouvelles questions ;
- promouvoir des techniques scientifiques ;
- réduire l'ensemble des possibles ;
- mettre en avant des incertitudes fondamentales ;
- fournir des solutions en temps réel pour des problèmes critiques ;
- démontrer ou suggérer l'efficacité de méthodes ;
- tester la robustesse de théories lorsqu'elles sont confrontées à des perturbations ;
- mettre en avant des incompatibilités dans des jeux de données ;
- entraîner les praticiens ;
- aider à discipliner les politiques ;
- éduquer le grand public ;
- montrer la complexité (resp. simplicité) de choses simples (resp. complexes).

Nous étendrons cette liste générale d'intérêts en ajoutant qu'il est des cas pour lesquels la

modélisation et la simulation s'avèrent être des outils idéaux : toutes les situations pour lesquelles l'expérimentation directe demeure difficile voire impossible ! Nous avons ainsi identifié trois types d'expérimentations pour lesquelles la modélisation et la simulation s'avèrent être l'une des seules approches raisonnables :

- les expérimentations impossibles à mener de façon directe (souvent pour des raisons techniques) ;
- les expérimentations qu'il est possible, mais irréaliste, de mener de façon directe (essentiellement pour des raisons de coût ou d'éthique) ;
- les expérimentations directes pour lesquelles l'observation perturbe l'expérience en elle-même.

En effet, l'expérimentation directe atteint parfois des limites insurmontables, comme c'est le cas lorsque l'on cherche à faire des expériences sur des plages temporelles trop petites ou trop grandes. Lorsque l'on cherche à observer et à comprendre des phénomènes qui se produisent très rapidement, on se retrouve confrontés aux limites des dispositifs d'instrumentation qui ne permettent pas d'observer les processus avec précision. Et lorsque l'on cherche à observer et à comprendre des phénomènes qui se produisent très lentement (par exemple les mécanismes de l'évolution Humaine), on est bien souvent confrontés à nos propres limites de temps et de vie ce qui ne permet pas de mener l'expérience à terme, si ce n'est en la faisant durer sur plusieurs générations de scientifiques se relayant au fil du temps. Ce problème d'échelle, lié d'abord à l'aspect temporel, se retrouve également dans d'autres dimensions, et notamment sur le plan spatial. Ainsi il est impossible aujourd'hui de réaliser des expérimentations directes lorsque l'on touche aux domaines de l'infiniment petit (par exemple pour étudier certaines interactions qui se passent dans des réactions chimiques, biologiques, ou particulaires) ou de l'infiniment grand (par exemple pour étudier des phénomènes comme l'expansion de l'Univers).

Par ailleurs, il existe aujourd'hui un large ensemble de situations qui, sans être impossibles à expérimenter, sont pourtant irréalisables. C'est par exemple le cas des expérimentations qui deviendraient trop coûteuses à être mises en place. En considérant que l'on veuille faire des expériences visant à tester différents types d'accroissement urbain (comme ce sera l'objet, en simulation, d'une partie des travaux décrits dans la suite de ce manuscrit), il devient vite impossible de réaliser des expériences intéressantes de façon directe. En admettant que l'on dispose d'un terrain suffisamment grand (par exemple pour construire ne serait-ce qu'un quartier d'une grande ville), il n'est pas imaginable de pouvoir se permettre d'aller construire, puis démolir, puis reconstruire, puis redémolir, puis reconstruire encore et encore des bâtiments entiers afin de tester différents scénarios d'évolution ! Et pour peu que l'on veuille expérimenter différents scénarios d'évacuation d'un stade pour comprendre les problèmes qui peuvent se produire en terme de gestion des foules, il n'est pas réalisable de faire faire l'expérience de façon directe à des sujets humains qui seraient donc confrontés aux risques potentiels que l'expérimentation vise à supprimer !

Enfin, n'oublions pas qu'il existe un certain nombre d'expérimentations qui ne sont pas réellement impossibles à mener, mais plutôt impossibles à observer car l'observation, si elle existait, perturberait l'expérience en elle-même. C'est essentiellement le cas de phénomènes qui peuvent se produire dans le domaine du quantique, où la présence d'un observateur, *a priori* extérieur au système, modifie pourtant les résultats des expérimentations.

Nous concluons à présent cette section comme nous l'avions entamée, par une citation extraite de [Epstein 2008] :

Models can surprise us, make us curious, and lead to new questions. This is what I hate about exams. They only show that you can answer somebody else's question, when the most important thing is : Can you ask a new question ? It's the new questions that produce huge advances, and models can help us discover them.

Cette citation permet de mettre en avant les raisons profondes qui font qu'une démarche de modélisation peut devenir passionnante puisque elle entraîne l'apparition de nouveaux questionnements, et qu'elle génère donc au fil du temps la production de nouvelles connaissances.

Cet ensemble de raisons que nous avons décrites (celles listées par Epstein et celles que nous avons ajoutées) suffisent amplement à mettre en avant l'intérêt que l'on trouve à modéliser et simuler des systèmes complexes. Nous pouvons à présent expliquer à notre ami néophyte comment est-ce que l'on procède pour modéliser un système complexe et concevoir une simulation.

1.2.3 Comment faire ?

Différentes approches permettent de concevoir des simulations telles que nous les avons décrites à la section précédente. Il convient donc de choisir l'approche adaptée en fonction du contexte dans lequel l'on se trouve, ou plutôt en fonction des spécificités du modèle du domaine que l'on souhaite réaliser afin que l'implémentation qui s'en suivra puisse offrir un maximum d'avantages (ou du moins entraîne un minimum d'inconvénients à même de perturber la modélisation initialement souhaitée). Les techniques les plus répandues sont les suivantes :

- la simulation basée sur des approches mathématiques qui consistent à modéliser les systèmes à étudier sous la forme d'équations différentielles, ordinaires ou partielles, qui donnent ainsi des représentations dans lesquelles le temps et l'espace sont continus, et dans lesquelles la dimension spatiale n'a qu'une existence implicite [May 1976] ;
- la simulation à événements discrets, dans laquelle le système évolue en fonction des événements qui sont générés au fur et à mesure de l'évolution de la simulation, comme c'est le cas dans les approches formelles connues telles que DEVS [Zeigler et al. 2000] ;
- la simulation par automates cellulaires, dans lesquels le système est composé de cellules ayant chacune leurs propres contraintes et qui évoluent et interagissent via des

règles de transition qui sont fonction de leur état et de celui des cellules de leur voisinage [Benenson & Torrens 2004];

- la Simulation Orientée Agent (SOA), dans laquelle le système est modélisé par l'ensemble des entités qui le composent, les agents, qui interagissent entre eux directement ou indirectement *via* le monde dans lequel elles sont généralement plongées, l'environnement [Ferber 1995, Wooldridge & Jennings 1995].

Chacune de ces approches possède son lot d'avantages et d'inconvénients. Mais dans la section suivante, nous ne nous attarderons que sur la dernière technique listée, la SOA, qui est celle que nous avons choisi d'utiliser dans nos travaux. Mais avant cela nous détaillerons quelques-unes des méthodologies de modélisation existantes car, quelle que soit l'approche de simulation choisie, le processus de conception de simulations dans son ensemble demande à être particulièrement réfléchi et d'autant plus cadré que les projets, systèmes et autres phénomènes que l'on souhaite étudier sont importants.

1.3 Modélisation et Simulation Orientée Agent

Dans cette section, nous allons donner quelques éléments permettant de décrire une méthodologie de modélisation, puis nous présenterons les Systèmes Multi-Agents (SMA) qui constituent le support de base de la Simulation Orientée Agent (SOA), et nous détaillerons les éléments qui nous ont orientés vers le choix de cette technique de simulation.

1.3.1 Méthodologies de conception de simulations

Les notions de base (modèles, simulation, ...) qui ont été définies précédemment apparaissent en tant qu'éléments de ce que nous appelons une méthodologie de conception de simulation, un processus que nous définissons de la manière suivante :

Définition I.8 (Méthodologie de conception de simulation) *Une méthodologie de conception de simulation consiste à définir la suite d'opérations à réaliser pour obtenir une simulation d'une partie du monde réel.*

Nous emploierons souvent ces termes en utilisant le pluriel pour "simulation" puisque les modèles que nous concevons sont la plupart du temps destinés à générer de nombreuses simulations. De nombreux travaux ont été consacrés à définir de telles méthodologies de conception et nous allons en présenter rapidement quelques unes.

Méthodologie de conception de simulation selon Fishwick

Fishwick considère que la conception de simulation se fait suivant trois étapes principales qui sont décrites dans [Fishwick 1997] :

- la conception du modèle du système que l'on souhaite étudier ;

- l'exécution de ce modèle ;
- l'analyse des résultats produits par cette exécution.

Ces trois étapes sont représentées sur la figure 1.1 et les concepteurs qui adoptent une telle méthodologie les suivent de façon cyclique pour s'arrêter dès lors que les simulations qu'ils obtiennent leur semblent conformes au système réel ou théorique étudié.

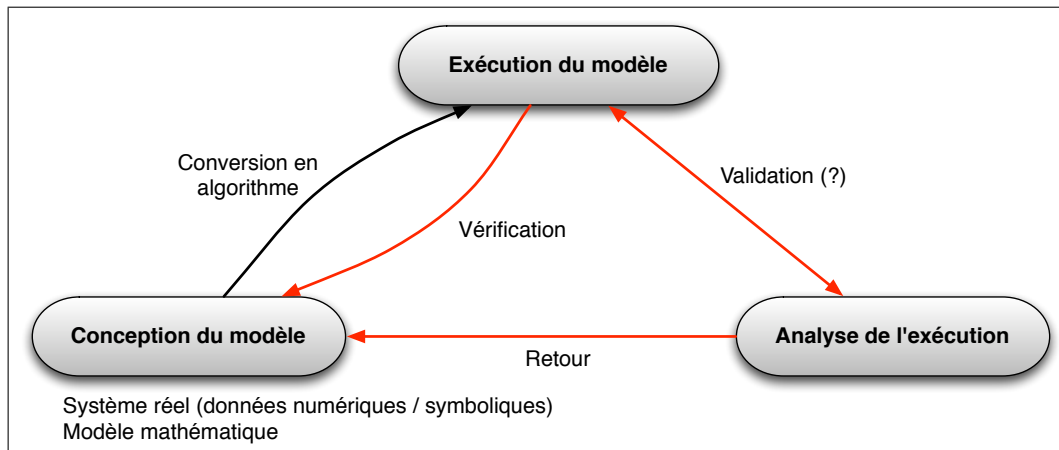


FIG. 1.1 – Méthodologie de conception de simulations selon Fishwick (adapté d'après [Fishwick 1997])

Méthodologies de conception de simulation selon Gilbert et Troizch, et selon Ralambondrainy et Sébastien

Ralambondrainy et Sébastien [Ralambondrainy 2009] et [Sébastien 2009] décrivent un processus de conception de simulations en quatre étapes principales, représentées sur la figure 1.2 :

- la conception du modèle du domaine, étape à laquelle se fait la modélisation du monde réel (ce qui inclut la délimitation des parties que l'on souhaite modéliser) sous la forme du modèle du domaine que l'on souhaite étudier ;
- la conception du modèle informatique, étape à laquelle le modèle du domaine est transposé en un modèle informatique exécutable ;
- l'exécution de la simulation, étape d'exécution à laquelle le modèle informatique est mis en œuvre dans un temps virtuel ;
- l'analyse et l'interprétation des résultats produits par le modèle informatique, ou plutôt par les différentes exécutions de celui-ci.

Les étapes d'un tel processus de conception de simulation sont cycliques : les productions et résultats obtenus à chacune des étapes peuvent entraîner des retours aux étapes antérieures, dont les éléments se retrouvent alors améliorés. Ainsi la production de connaissance est à la fois itérative et cyclique, conduisant alors notre processus de simulation à un cycle de développement en spirale similaire à celui décrit dans [Courdier 2003].

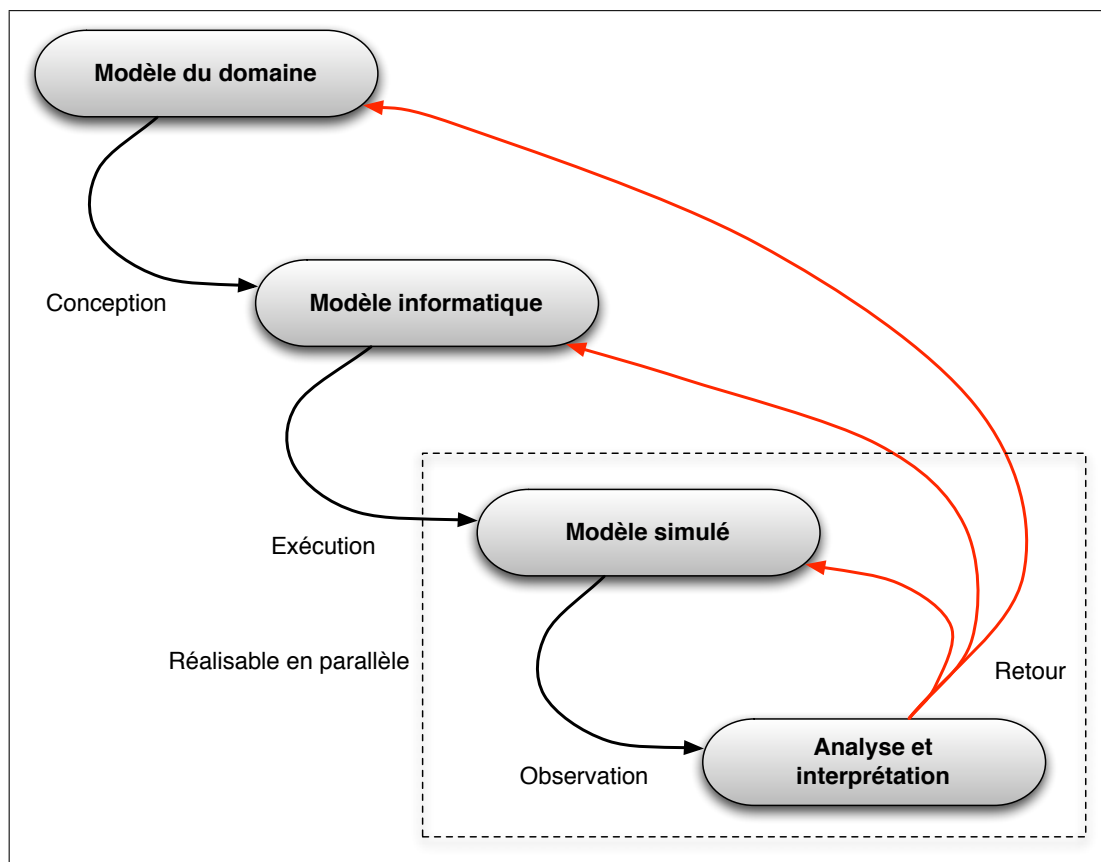


FIG. 1.2 – Méthodologie de conception de simulations selon Ralambondrainy et Sébastien (adapté d'après [Ralambondrainy 2009, Sébastien 2009])

Dans l'esprit cette méthodologie est assez proche de celle proposée par Gilbert et Troitzch [Gilbert & Troitzsch 1999], qui est représentée sur la figure 1.3, et dans laquelle on note de façon intéressante l'apparition de la connaissance comme élément explicite.

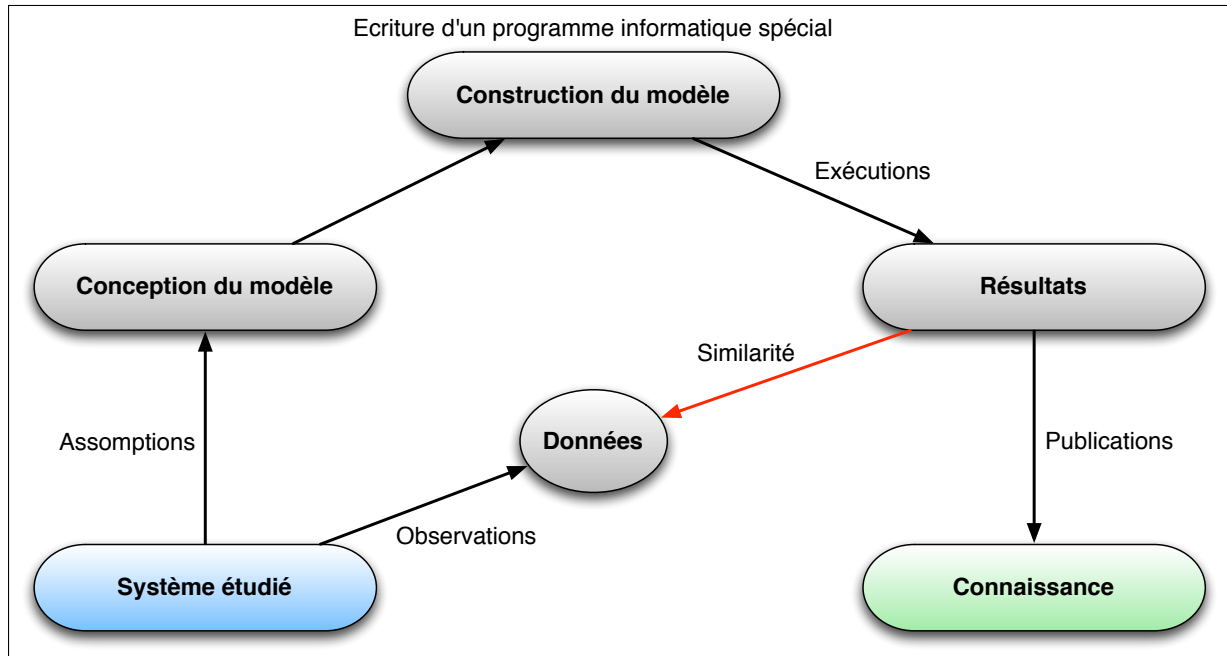


FIG. 1.3 – Méthodologie de conception de simulations selon Gilbert et Troitzch [Gilbert & Troitzsch 1999] (adapté d'après [Drogoul *et al.* 2003])

Ces deux méthodologies diffèrent principalement de celle de Fishwick dans le sens où elles introduisent une étape supplémentaire, la construction du modèle informatique. Comme souligné dans [Drogoul *et al.* 2003], ceci raffine donc la vision un peu trop naturelle qui veut que la traduction d'un modèle initial en modèle informatique se fasse de façon complètement transparente.

Méthodologie de conception de simulation selon Drogoul, Meurisse et Vanbergue

Drogoul, Meurisse et Vanbergue proposent dans [Drogoul *et al.* 2003] l'une des méthodologies de conception de simulations les plus complètes. Représentée sur la figure 1.4, elle reprend les étapes de conception des différents modèles – modèle (conceptuel) du domaine, modèle informatique – que nous avons vues dans les deux méthodologies précédentes, mais elle met également (et surtout) l'accent sur les liens qui existent entre les étapes de la conception et les différents intervenants (le thématique, le modélisateur, et l'informatique) en charge de leur réalisation.

De façon intéressante, micro-connaissances et macro-connaissances sont représentées explicitement dans une telle méthodologie. Elles reflètent la connaissance du thématique et les liens entre ces deux niveaux de connaissance sont ici considérés comme étant ceux qui

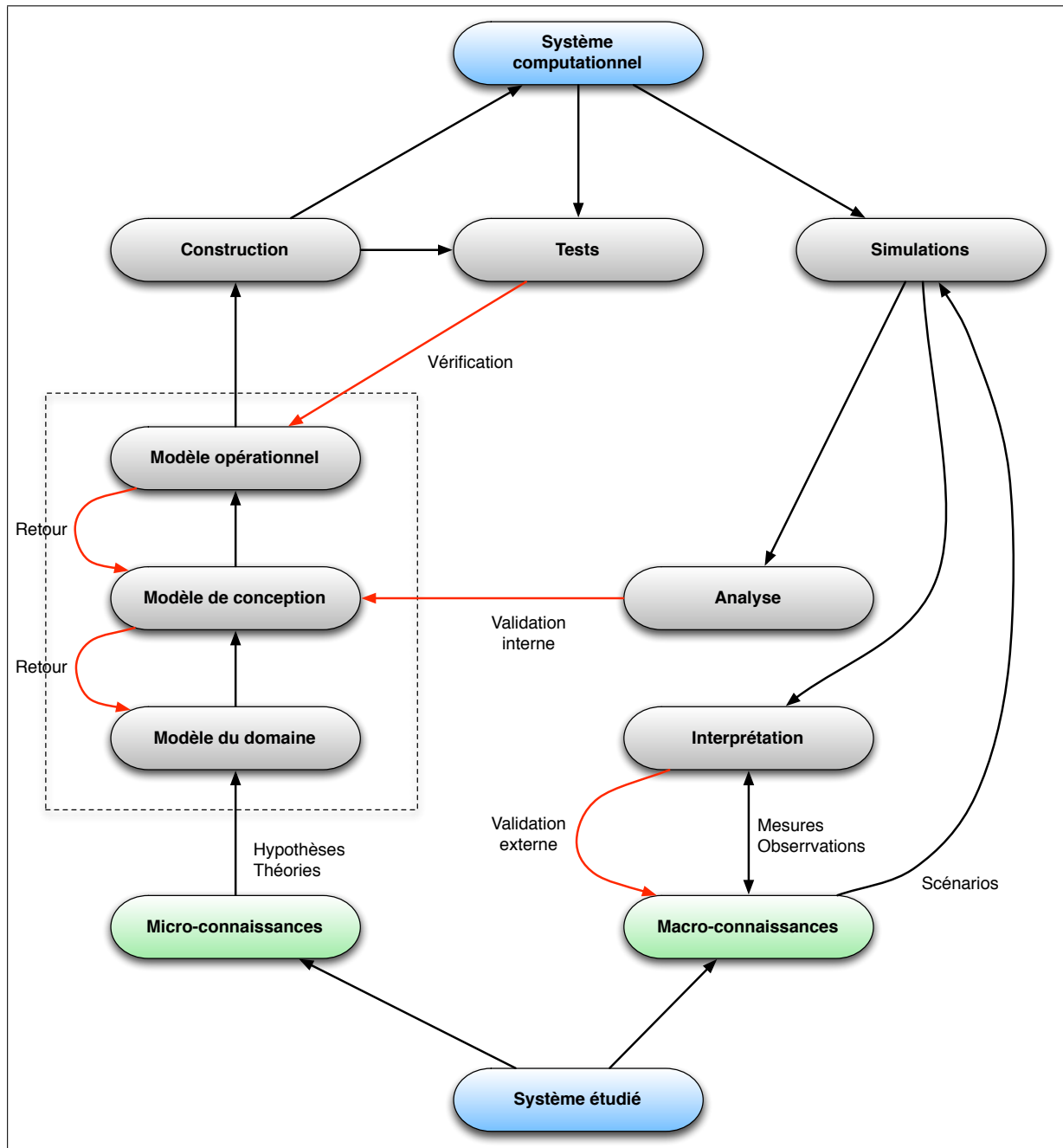


FIG. 1.4 – Méthodologie de conception de simulations selon Drogoul, Vanbergue et Meurisse (adapté d'après [Drogoul *et al.* 2003])

garantissent la bonne compréhension des phénomènes qui se produisent dans les simulations et qui permettent donc de juger, lors de l'analyse des résultats, si l'objectif de modélisation et de simulation est atteint ou si des retouches sont à prévoir. Pour les mettre en avant, nous avons représenté en vert ces éléments liés à la "connaissance" sur les différents schémas, car il est important de ne pas oublier que le but de tout ceci n'est autre que de découvrir et d'acquérir de nouvelles connaissances sur les systèmes que l'on étudie.

Cette dernière méthodologie est considérée comme étant une référence dans le domaine de la conception de simulations. Mais il est très intéressant de noter que sur celle-ci, comme sur toutes celles que nous avons présentées, une étape reste négligée dans sa description : les différents "retours", que nous avons représentés en rouge sur les différents schémas, qui se passent dans les cycles finis d'étapes qui se répètent pour parvenir à un modèle convenable. Et ceci est dommageable lorsque l'on sait que ces "retours" consistent bien souvent à trouver les bons ajustements, ou la bonne façon d'injecter de la nouvelle connaissance (celle-là même que nous avons représentée en vert) dans les modèles afin de les améliorer. C'est là l'un des points principaux sur lequel nous avons orienté les travaux que nous présenterons dans les chapitres 4 et 5 de la deuxième partie de ce manuscrit.

1.3.2 Les Systèmes Multi-Agents

La thématique des SMA se focalise sur l'étude des comportements collectifs et sur la répartition de l'intelligence sur des agents plus ou moins autonomes, capables de s'organiser et d'interagir pour résoudre des problèmes. Nous définissons un SMA de la façon générale suivante :

Définition I.9 (Système Multi-Agents) *Un Système Multi-Agents (SMA) est un système composé d'agents, qui peuvent être situés dans un certain environnement, et qui peuvent interagir selon certaines relations.*

En France, leur utilisation a été popularisée depuis la parution de [Ferber 1995] et de nombreuses définitions plus complexes ou plus précises que celle présentée ci-dessus ont été établies depuis. Et il en va évidemment de même pour ce qui est de définir la principale notion qui apparaît dans la définition précédente d'un SMA : les agents. Ainsi, Ferber considère qu'un agent est "une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents". Alors que Wooldridge [Wooldridge & Jennings 1995] en parle comme d'un "système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu" ; que Shoham le définit comme étant "une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent" ; et que Russel évoque encore plus simplement "une entité qui perçoit son environnement et agit sur celui-ci".

De ces différentes définitions ressort une idée globale, un consensus, qui s'articule autour de notions fortes comme l'autonomie et la pro-activité, et qui est assujéti à quelques variations (présence de connaissances ? de croyances ? de buts ? rôle de l'environnement vis à vis des interactions ?). Ces différences ont suscité l'apparition de typologies d'agents diverses. Ainsi on distingue par exemple aujourd'hui les agents réactifs, qui fonctionnent sur un mode stimulus-réponse, et les agents cognitifs, qui possèdent des notions de mémoire, de représentation de leur environnement, et dont le comportement suit des processus décisionnels plus poussés. De la même façon on distingue les agents situés, qui sont représentés et positionnés dans un environnement spatialisé qu'ils utilisent pour communiquer entre eux de façon indirecte, et les agents dits purement communicants qui n'ont pas forcément d'ancrage physique et qui communiquent directement entre eux. Selon les classifications et les contextes sont ainsi également reconnus les agents téléonomiques, réflexes, pulsionnels, intentionnels, coopératifs, individualistes, etc.

Tous ces points de vue sur les SMA en général, et sur les notions d'agent et d'environnement en particulier, résultent de la relative jeunesse de ce domaine de l'informatique qui est toujours à un croisement où se mêlent la Recherche et la Science : si la première est un processus dialectique, qui avance, recule, se contredit, dit tout et son contraire, la seconde est le résultat du filtrage et de la sédimentation des recherches passées [Drogoul 2005].

Ainsi, de tous ces points de vue, il ressort que ces caractéristiques principales des SMA sont basées sur la vision locale et décentralisée des problèmes :

- vision locale, car chaque agent est responsable de ses connaissances (encapsulation) et de ses actions (autonomie), mais également de l'organisation qu'il met en place avec d'autres agents, aucun d'entre eux n'ayant une vue globale du SMA ;
- vision décentralisée, car on s'efforce d'éliminer tout contrôle central, les tâches à réaliser et les compétences pour le faire étant distribuées sur les agents du système.

De ceci découle le fait que tous les agents d'un système ont un point de vue partiel : chacun d'entre eux a des informations et/ou des capacités de résolution de problèmes limitées. Par ailleurs, il est bon de préciser que l'un des intérêts découle de la possibilité de faire cohabiter, dans un même système, des agents complètement différents ; on parle alors d'agents (et de systèmes) hétérogènes.

1.3.3 La Simulation Orientée Agent

La SOA est basée sur l'utilisation des Systèmes Multi-Agents pour faire de la simulation. Elle permet d'offrir des solutions à des problèmes pour lesquels des méthodes de simulation plus classiques montrent leurs limites. Elle est ainsi particulièrement indiquée dans les cas où l'on souhaite s'intéresser à la représentation des comportements qui se passent dans un système plutôt qu'aux résultats de ces comportements. Elle l'est aussi dès que l'on souhaite examiner avec attention des systèmes dans lesquels cohabitent des dynamiques dans lesquelles le passage de l'échelle microscopique à l'échelle macroscopique joue un rôle prépondérant. Et

elle est donc particulièrement indiquée lorsque l'on présuppose que les dynamiques présentes dans les systèmes que l'on veut simuler vont conduire à l'émergence de structures spatio-temporelles.

Afin de ne pas repartir toujours de zéro dans leurs développements, les chercheurs travaillant dans le domaine de la SOA ont mis en place différents outils, qui sont souvent regroupés au sein de ce que nous appelons des plateformes de simulation. Citons pêle-mêle CORMAS [Bousquet *et al.* 1998], MadKit [Gutknecht & Ferber 2000], GEAMAS [Soulié *et al.* 1998], MIMOSA [Müller 2004], Swarm [Johnson & Lancaster 2002, Daniels 1999], ou encore NetLogo [Wilenski 1999], bien connue pour ses vertus pédagogiques. Ces plateformes ont toutes leurs spécificités, qui sont souvent le reflet des principes adoptés et des théories proposées par les équipes qui les mettent en place et qui les maintiennent. Modèle d'agents, gestion du temps, gestion des organisations, et types de relations avec l'environnement sont ainsi bien différentes d'une plateforme à l'autre. Mais leur utilisation simplifie bien souvent la vie des développeurs qui veulent réussir à construire des outils de simulation ayant une certaine ampleur.

L'une des spécificités de la SOA par rapport aux SMA au sens général réside dans les relations qui unissent les agents et leur environnement. Ainsi l'environnement, qui est parfois absent dans le monde des agents logiciels, est presque toujours présent dans les plateformes de simulations que nous avons listées. Par ailleurs, dans un contexte de simulation, les agents sont souvent considérés comme étant des entités possédant un "esprit" (dans lequel se tiennent les processus décisionnels de l'agent) et un "corps" (qui est sa représentation dans l'environnement). Le lien entre un agent et son environnement se fait alors via l'utilisation de mécanismes dits d'influences et de perception, comme représenté sur la figure 1.5.

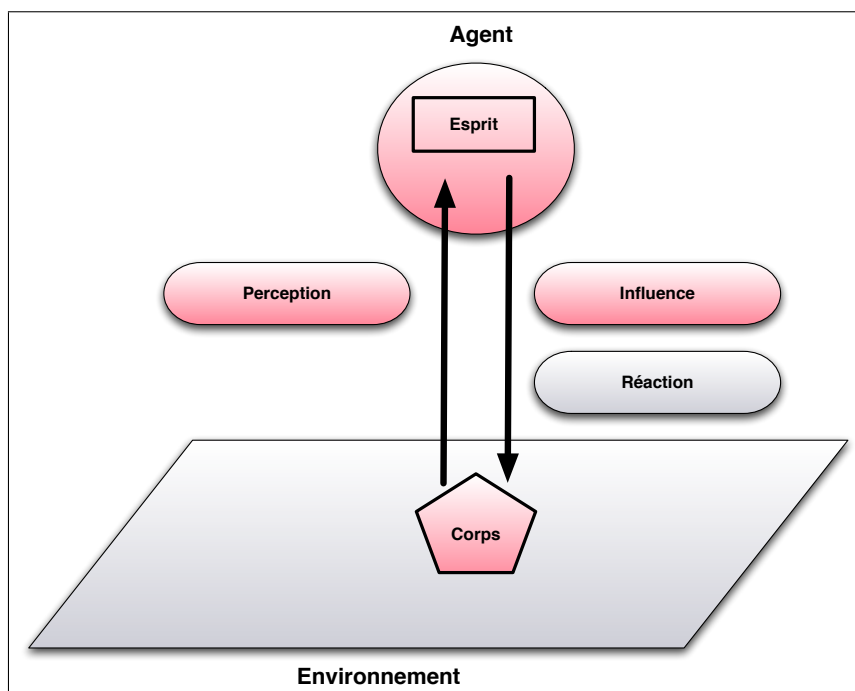


FIG. 1.5 – Les mécanismes d'influence et perception dans une SOA

De tels mécanismes sont les garants de la cohérence du système puisque seule leur utilisation garantit que des modifications inopportunes n'ont pas lieu (par exemple si deux agents veulent tous les deux se saisir d'un même objet de l'environnement). Ce point des relations agent/environnement nous a paru important à préciser car les mécanismes d'influence et de perception auront un rôle à jouer dans les mécanismes de prise en compte de l'émergence dans les SOA que nous décrivons dans le chapitre 5 de ce manuscrit.

La Simulation Orientée Agent ainsi décrite est la technique que nous avons choisi d'utiliser dans nos travaux, notamment, comme nous le verrons dans la troisième partie de ce manuscrit, pour réaliser le modèle informatique du modèle conceptuel de DS qui est le principal outil que nous avons mis en place pour faire de la prospective territoriale.

1.4 Prospective territoriale et Simulation Orientée Agent

Dans cette section, nous allons donner une rapide vue d'ensemble du domaine auquel se rattachent la plupart des applications développées dans le cadre de cette thèse : la prospective territoriale. Nous allons donc définir en quoi consistent ces termes, avant de faire une rapide revue des travaux existants qui mêlent déjà prospective territoriale et Simulation Orientée Agent. Nous en profiterons pour présenter quelques travaux de l'équipe SMART du LIM qui entrent dans ce cadre.

1.4.1 La prospective territoriale

Comme nous l'avons énoncé précédemment dans ce chapitre, le fait de pouvoir prévoir de la façon la plus précise possible un certain nombre d'évènements est aujourd'hui devenu primordial dans de nombreux domaines, et en particulier dans celui de la géographie. En effet, si l'on considère les notions d'aménagement du territoire, de répartition des populations ou tout simplement de gestion des ressources humaines et/ou naturelles, il paraît évident que les impacts sur l'environnement des besoins, des attentes, des demandes, ou des projets (à petite comme à grande échelle) doivent être suffisamment réfléchis.

La notion même de *prospective* a été "inventée" et introduite par Gaston Berger au milieu du XX^{ème} siècle. Comme l'expliquent de nombreux auteurs [Berger 1959, Loinger 2005], elle consiste à penser le temps long pour agir avec plus d'efficacité sur les mécanismes de prise de décision du court et du moyen terme, d'où une relation itérative entre les temps longs des processus et le temps court de l'action et de la décision stratégique.

Naturellement, la *prospective territoriale* constitue la version localisée et spatialisée de la *prospective*. Ainsi considérée elle est vue comme étant un outil de réflexion qui participe à la construction de stratégies territoriales basées sur des raisonnements à long terme. Elle participe ainsi à la construction contemporaine des territoires ; construction qui s'effectue à travers les significations, dans la structuration et dans la mobilisation des acteurs [Fourny & Denizot 2007]. Pour les politiques, les décideurs, les aménageurs, et pour tous ceux

qui sont amenés à intervenir dans les processus qui entraînent des modifications de nos espaces de vie, elle est une vraie aide à la planification spatiale dont une définition plus complète est établie dans [Destatte & Van Doren 2003] :

Définition I.10 (Prospective territoriale) *La prospective territoriale est une démarche rigoureuse – généralement réalisée de façon transdisciplinaire et en réseau – qui permet de déceler les tendances d'évolution, d'identifier les continuités, les ruptures et les bifurcations des variables (acteurs et facteurs) de l'environnement, ainsi que de déterminer l'éventail des futurs possibles. Elle permet ainsi d'élaborer des stratégies cohérentes et d'améliorer la qualité de la décision à prendre. Elle est une des techniques nécessaires à la pro-activité, attitude de celui qui anticipe l'évènement dans ses réflexions, agit pour provoquer les changements souhaités et saisit les opportunités nouvelles.*

On retrouve dans cette définition les éléments qui permettent de décliner la prospective territoriale selon cinq dimensions majeures, qui sont définies de façon plus approfondie dans [Loinger 2005] :

- une dimension cognitive, car lorsque l'on veut prétendre à réfléchir sur l'avenir d'un territoire, il faut commencer par le connaître et par le comprendre, lui et ses dynamiques, son histoire, sa complexité ;
- une dimension participative, car pour connaître, comprendre, et ensuite prévoir, il faut réfléchir à plusieurs, partager ses analyses, et ainsi créer idéalement un débat collectif qui puisse aboutir à une représentation collective des avènements possibles désirés ;
- une dimension liée à l'existence de valeurs, car réfléchir sur l'avenir de territoires implique de s'interroger sur ce qui a du sens pour ces territoires et pour leurs habitants ;
- une dimension stratégique, car c'est ce qui débouchera sur l'apport d'une aide à la décision, pour que les réflexions entamées puissent se concrétiser dans le réel ;
- une dimension liée à la faisabilité, car faire de la prospective inclut le fait de s'interroger sur les conditions de réalisation et d'accompagner la mise en œuvre des choix stratégiques.

Les travaux qui ont été menés dans le cadre de cette thèse, et essentiellement ceux qui sont décrits dans la troisième partie de ce manuscrit, ont donc été réalisés afin d'apporter des avancées locales à ce domaine de la prospective territoriale et d'apporter ainsi des contributions en rapport avec les différentes dimensions que nous venons de lister.

Les liens entre prospective territoriale d'une part, et modélisation et simulation d'autre part, sont évidents et réciproques. En effet, dans un sens la modélisation et la simulation constituent des techniques de premier choix pour participer à la réalisation d'outils au service de la prospective territoriale. Et dans l'autre sens la prospective territoriale constitue un terrain d'expérimentation rêvé pour toute personne travaillant dans le domaine de la modélisation et simulation de systèmes complexes. Et au final les interactions n'en deviennent que plus intéressantes entre informaticiens et thématiciens.

Pour un géographe, les dimensions de la prospective territoriale qui sont listées ci-dessus sont présentées dans un ordre qui est parfaitement cohérent puisque l'existence de chaque di-

mension est justifiée par la présence des précédentes. Mais, lorsque l'on examine les choses avec l'œil d'un modélisateur informaticien, la dimension participative devient rapidement celle qui va influencer la façon dont les travaux et outils seront réalisés. En effet, le point crucial qui se situera à chaque réalisation sera l'interaction entre les différents acteurs : thématiciens, modélisateurs, et informaticiens. Deux points clés, liés au contexte participatif et aux échanges pluridisciplinaires, ressortent de ce constat et servent de guide à la suite des travaux que nous présenterons :

- l'importance du contexte dans une démarche de modélisation (que ce soit le contexte de l'étude ou le contexte des connaissances) ;
- l'importance des techniques d'échanges (que ce soient les échanges entre les thématiciens et les modélisateurs-informaticiens ou les échanges entre les thématiciens-utilisateurs et les outils de simulation mis à leur disposition).

Comme nous l'avons précisé en introduction, le contexte local de l'île de La Réunion fait qu'il est aujourd'hui nécessaire de faire des efforts dans ce domaine de la prospective territoriale. Ainsi de nombreux chercheurs locaux, essentiellement des spécialistes de l'urbanisation, y consacrent une partie de leurs travaux [Lajoie 2005, Lajoie 2007, Dimou 2008].

Mais ce n'est que récemment que des liens se sont tissés pour faire émerger des projets pluridisciplinaires incluant des informaticiens locaux capables d'apporter leurs connaissances en modélisation et capables de produire des outils de simulation performants pour aider à imaginer les avenir possibles pour les territoires réunionnais.

Dans la suite de ce chapitre, nous décrivons donc un petit panel des travaux et outils de simulation qui ont été effectués dans le domaine de la prospective territoriale, que ce soit par des membres de l'équipe SMART ou par d'autres chercheurs.

1.4.2 Quelques réalisations de l'équipe SMART du LIM

L'équipe SMART du LIM entretient depuis plusieurs années des collaborations intensives dans le but de créer des outils de simulation avec des laboratoires et organismes dont les chercheurs, venant de différents horizons, font office de thématiciens experts de leurs domaines respectifs. Citons essentiellement :

- le CREGUR (Centre de Recherche En Géographie de l'Université de la Réunion) ;
- le CIRAD Réunion (Centre de Coopération Internationale en Recherche Agronomique pour le Développement) ;
- l'IRD de La Réunion (Institut de Recherche pour le Développement) ;
- l'IFREMER de La Réunion (Institut Français pour la Recherche sur la MER).

Cet ensemble de collaborations a permis la réalisation de plusieurs modèles et outils de simulation tous liés aux systèmes complexes naturels et sociaux, et participant, à leur manière, à fournir une aide à la prospective territoriale, que celle-ci soit directement liée à des territoires terrestres ou à des territoires marins dont l'évolution dépend de la présence sur site de certaines espèces. Par ailleurs, au delà de la portée qu'ils peuvent avoir dans leurs domaines

d'application, ces modèles et outils sont d'une aide précieuse dans la "simple" réalisation de nos travaux d'informaticiens. En effet, au fil du temps, ils nous permettent de développer notre plateforme de simulation GEAMAS, aujourd'hui devenue GEAMAS-NG (New Generation) à la suite d'évolutions majeures.

Nous présentons rapidement deux de ces projets, tout en sachant que le modèle DS, qui est l'une des productions directes de cette thèse, et qui sera décrit longuement dans la troisième partie de ce manuscrit, est lui-même une résultante d'un projet, l'ATP DOMINO, conduit par le CIRAD Réunion et dans lequel sont intervenus de nombreux acteurs tels le Comité de pilotage de la canne, l'Association pour la Promotion en milieu Rural et la Chambre d'Agriculture de la Réunion.

Prospective territoriale... terrestre !

Une des collaborations avec les chercheurs du CIRAD Réunion a permis de créer des outils ayant pour but de mettre en relation des exploitants agricoles producteurs de matières organiques (élevages) avec des exploitants consommateurs de matières organiques (cultures) de manière à éviter les rejets de matières organiques dans l'environnement afin de réduire les risques de pollution. Un même modèle appelé BIOMAS a ainsi été créé, enrichi et complexifié au cours de projets consécutifs [Courdier *et al.* 2002] afin de valoriser au mieux ces matières organiques. Pour l'exemple une vue d'ensemble obtenue en simulation est représentée sur la figure 1.6. On y distingue les bassins versants d'une zone agricole pour lesquels la simulation a abouti à des situations de débordement (zones de couleur rouge).

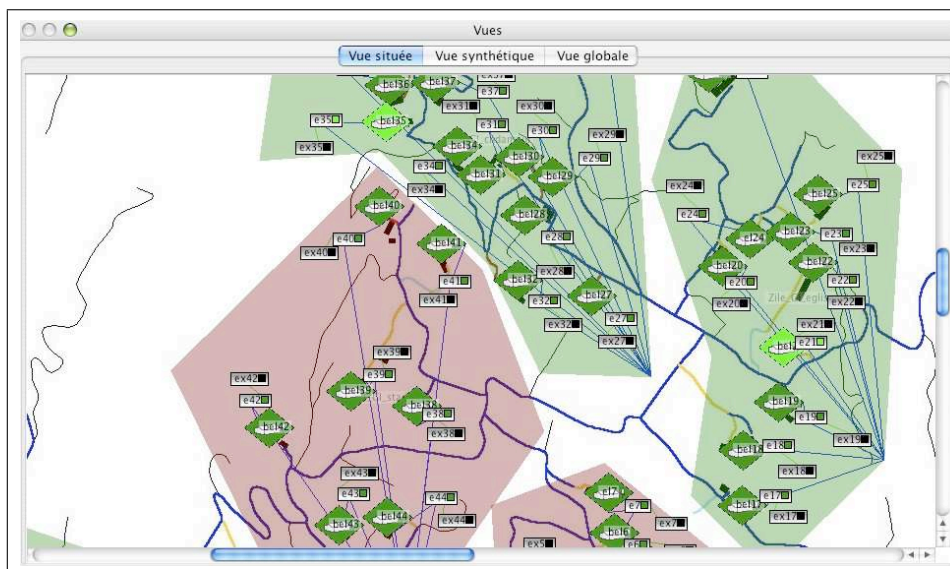


FIG. 1.6 – Vue sur l'évolution des bassins versants d'une zone agricole avec le modèle BIOMAS

Ce modèle BIOMAS, que l'on retrouve ainsi décrit dans [Ralambondrainy 2009], met en jeu plusieurs centaines d'agents hybrides, réactifs et cognitifs qui ont pour particularité d'être largement différents les uns des autres (éleveurs, élevages, moyens de transport, unités de

traitement, cultures, groupements d'exploitants...). De plus, ces agents mènent de nombreuses négociations entre effluents, qui sont particulièrement difficiles à analyser.

Ce type d'outil de simulation permet de mener des réflexions stratégiques sur l'avenir de certains territoires agricoles réunionnais et peut être utilisé en tant qu'outil d'aide à la décision, par exemple pour simuler et déterminer les emplacements idéaux pour les installations de futures unités centrales de traitements.

Et prospective territoriale... marine ?

Les collaborations avec des chercheurs de l'IFREMER de La Réunion ont été entamées depuis une dizaine d'années, afin de travailler sur des modèles permettant de simuler l'évolution de grands pélagiques comme les espadons [Soulié 2001].

Plus récemment, nous avons entamé une nouvelle collaboration afin de participer ensemble à la construction d'outils de simulation permettant d'observer l'évolution de tortues vertes marines (*Chelonya midas*) dans la zone Sud-Ouest de l'océan Indien. La pêche, le braconnage ou la pollution constituent autant de menaces anthropiques directes ou indirectes qui peuvent à terme affecter la survie de cette espèce et il est donc nécessaire de co-construire des outils de simulation pour mieux comprendre le comportement des tortues et ainsi pouvoir évaluer plus précisément l'impact des différentes menaces qui pèsent sur elles, et à terme pour les contrôler en prenant les décisions qui s'imposent, en s'aidant pourquoi pas d'outils d'aide à la décision que nous développerons, en poursuivant la voie ouverte par le prototype représenté sur la figure 1.7 et décrit dans [Gangat *et al.* 2010].

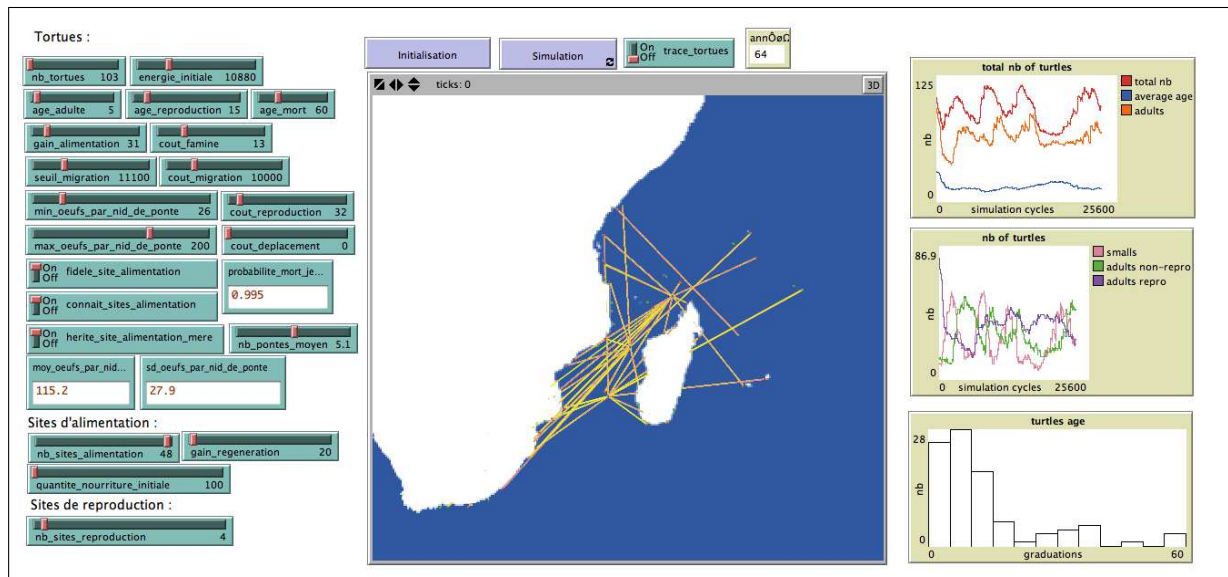


FIG. 1.7 – Un prototype de simulation des populations de tortues vertes dans la zone Sud-Ouest de l'océan Indien

Ces outils, qui sont plutôt centrés sur la simulation de populations animales, sont égale-

ment des outils de prospective territoriale à leur façon. En effet, en travaillant sur des zones spatialisées d'océan, et en apportant de nouvelles connaissances qui aident à la compréhension des espèces animales qui y vivent, ils peuvent à terme permettre d'identifier des aires spatiales qu'il sera bon de faire cibler, sur un plan économique (par exemple si on veut pêcher beaucoup de poissons), ou au contraire d'interdire, sur un plan législatif (par exemple si on veut s'assurer que des espèces menacées comme les tortues vertes ne se feront pas prendre par inadvertance dans des filets de pêche utilisés de façon légale), au travers des futurs documents de planification.

1.4.3 Quelques travaux et réalisations connexes

L'approche agent est utilisée par de nombreux autres chercheurs ayant cette même volonté de simuler des scénarios d'évolution d'espèces ou de territoires [Le Page *et al.* 2000, Li & Liu 2008] qui peuvent souvent être rattachés au domaine de la prospective territoriale. De nombreux travaux et outils de simulation ont donc été réalisés, souvent liés à l'évolution d'écosystèmes [Bousquet & Le Page 2004, Ratzé *et al.* 2007, Bonneaud *et al.* 2007], mais aussi – et c'est ce qui nous intéressera particulièrement – à l'évolution de zones naturelles [Bone & Dragicevic 2010], à l'évolution de territoires agricoles [Bousquet *et al.* 1998, Bonnefoy *et al.* 2001, Hill *et al.* 2000], ou encore à l'évolution de zones urbaines [Franc & Sanders 1997].

Nous allons donc brièvement présenter quelques outils conséquents qui sont utilisés pour simuler l'évolution de zones urbaines, l'évolution de zones agricoles ou l'évolution de zones naturelles. Nous présentons ces outils en particulier puisqu'ils sont souvent réalisés et utilisés dans des contextes comparables à celui de DS, notre outil de simulation de l'évolution des espaces réunionnais que nous présenterons au chapitre 6.

Evolutions urbaines

La simulation a depuis longtemps été identifiée comme étant un élément de choix pour travailler sur des éléments de prospective urbaine, que ce soit directement pour simuler la croissance des villes elles-mêmes [Allen & Sanglier 1979] ou que ce soit dans un contexte plus global pour simuler l'émergence de réseaux urbains [Guérin-Pace *et al.* 1996] en utilisant l'approche agent.

Fruit de la collaboration des équipes de géographes de Géographie-Cités¹ et des informaticiens du Lip6², le modèle Simpop2 (et ses dérivés tels EUROSIM) fait suite aux travaux réalisés dans le cadre du projet Simpop [Sanders *et al.* 1996b] qui décrit de manière théorique le fonctionnement d'un système de villes sur 800 ans. L'évolution d'un ensemble de villes (sous des hypothèses de concurrence, formalisées par le fonctionnement et la spatialité d'un marché

¹UMR CNRS, Université Paris I, Université Paris VII

²Laboratoire d'Informatique de Paris 6, Université de Paris VI

d'échanges) est simulée par un système multi-agents réalisé en Objective-C et basé sur la plateforme multi-agents Swarm [Johnson & Lancaster 2002] dans lequel l'ensemble des villes, les agents du système, sont hétérogènes de par la diversité de leurs fonctions.

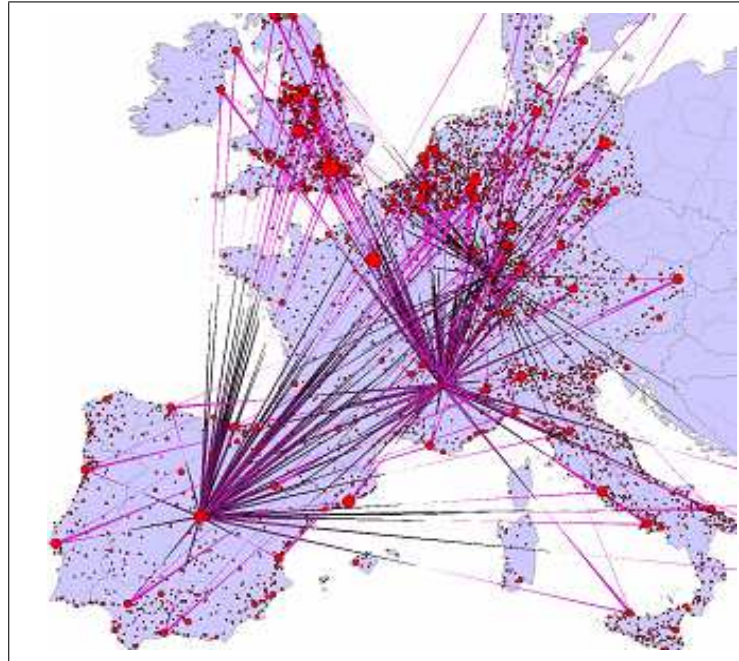


FIG. 1.8 – L'évolution d'un système de villes avec le Modèle Simpop2/EUROSIM (extrait de [Sanders *et al.* 1996a])

Simpop2 [Bretagnolle *et al.* 2007] vise essentiellement à simuler l'émergence et l'évolution du système des villes européennes de la fin du Moyen Age à nos jours et EUROSIM, visible sur la figure 1.8, simule l'évolution des villes européennes de 1950 à 2050. A partir d'un semis de peuplement rural relativement régulier et uniforme, le modèle simule l'apparition et l'évolution de villes qui acquièrent au cours du temps des fonctions de niveau de plus en plus élevé. Les règles d'évolution sont établies au niveau local et reposent sur les interactions entre les unités de peuplement. Comme on le devine sur la figure, les principaux effets observés sont la mise en place d'une organisation rang-taille et d'une organisation spatiale obéissant à certaines régularités d'espacement entre les villes suivant leur niveau hiérarchique.

D'autres travaux s'intéressent au développement des villes et zones urbaines, par exemple ceux menés à l'Austrian Research Center qui s'intéressent depuis longtemps [Loibl & Toetzer 2003, Loibl *et al.* 2007, Loibl & Walz 2010] à la modélisation des processus de densification. Leurs différentes expérimentations permettent de simuler l'évolution de la population dans la région de Vienne en prenant en compte un nombre impressionnant de critères (comme le montre d'ailleurs bien la figure 1.9 illustrant l'une de leurs applications) : migrations intra-urbaines, attractivité et coût des terrains, attractivité commerciale, desserte par les réseaux, etc. Ils utilisent pour cela de façon couplée des automates cellulaires et des agents spa-

tialisés : ces derniers servent à simuler les migrations et les décisions d’allocations des terres qui influent ensuite sur la façon dont seront effectués les changements en termes de densification et d’occupation du sol.

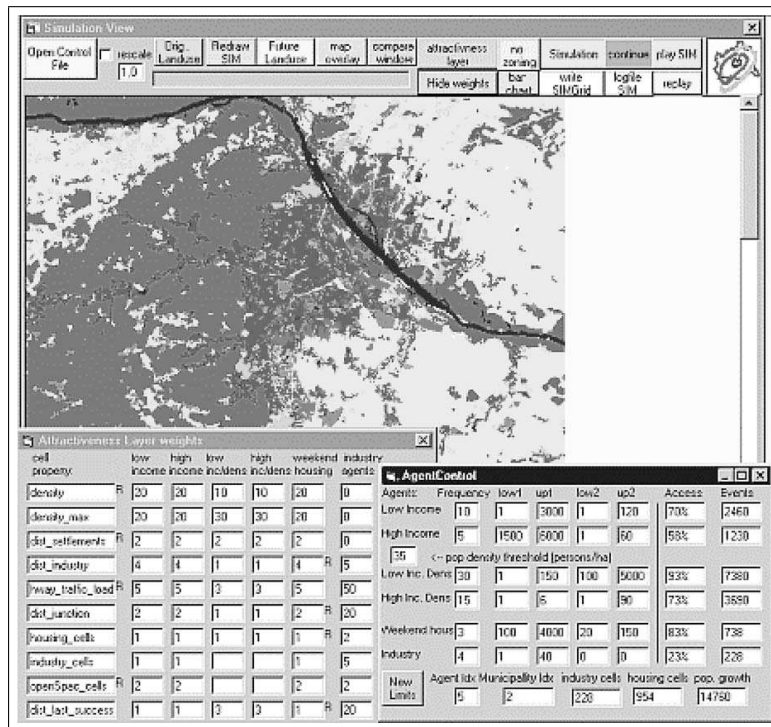


FIG. 1.9 – Un outil de simulation de l’évolution urbaine dans la région de Vienne (extrait de [Loibl & Toetzer 2003])

Ces travaux sont particulièrement intéressants pour nous car l’échelle à laquelle la densification et l’urbanisation sont étudiées s’approche de ce que nous avons mis en place dans DS (pour cette dynamique particulière), et ce bien que la zone d’étude ne soit pas aussi vaste que l’est La Réunion, et également bien que les contraintes associées aux types des territoires soient naturellement différentes.

Evolutions agricoles et naturelles

De nombreux travaux se consacrent à l’étude de l’évolution des surfaces agricoles. Le modèle PAYSAGRI, réalisé par les chercheurs d’AgroParisTech [Dépigny 2008], propose ainsi une représentation de l’effet des différentes sensibilités au paysage des agriculteurs sur les évolutions paysagères d’une portion de territoire rural et apporte la démonstration du rôle de la sensibilité au paysage dans les évolutions d’un paysage rural.

Comparativement aux travaux que nous menons dans l’équipe SMART, un tel modèle s’approche en partie de ce qui peut se faire avec le modèle BIOMAS, même s’il ne prend pas en compte un aspect spatialisé permettant de tester un plus grand nombre de choix d’aménagement. Mais au niveau usage, il s’approche plutôt d’une partie du modèle DS

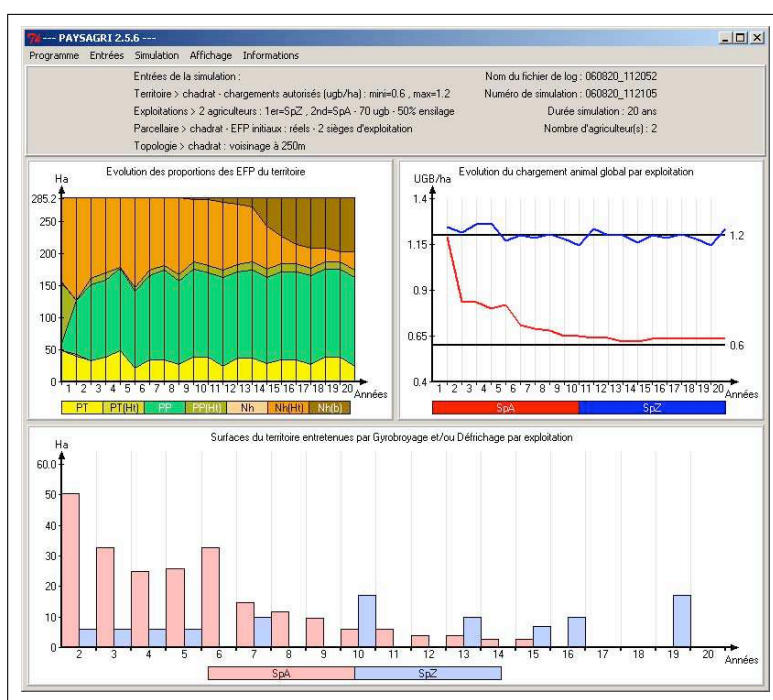


FIG. 1.10 – L'évolution de surfaces agricoles avec PAYSAGRI (extrait de [Dépigny 2008])

puisqu'il permet de suivre l'évolution de surfaces agricoles, chose que nous faisons également dans DS en l'associant à l'évolution des autres grandes classes d'occupation du sol.

Plus proches de ce que nous verrons dans DS, les travaux menés conjointement par des équipes américaines³ et équatoriennes⁴ portent sur l'évolution des surfaces agricoles en Amazonie Equatorienne [Walsh *et al.* 2008, Mena *et al.* 2010]. Ils utilisent l'approche agent pour simuler les processus décisionnels au niveau des exploitations agricoles, et leurs résultats permettent de mettre en avant les changements en terme d'occupation du sol sous une forme spatialisée. Les outils qu'ils développent permettent donc d'observer à une échelle globale les répercussions qui sont dues aux comportements des exploitants agricoles. On peut par exemple voir sur la figure 1.11 les résultats d'une simulation qui montre essentiellement l'importante évolution des zones pastorales (en jaune).

Ces travaux sont particulièrement intéressants puisqu'ils ont été réalisés, en ce qui concerne l'évolution des espaces agricoles, dans le même esprit que celui que nous avons eu en développant DS. Ils offrent, en outre, une modélisation relativement rigoureuse des phénomènes sociaux qui ont lieu dans les exploitations agricoles.

Par ailleurs, il est intéressant de noter que les dynamiques qui sont modélisées dans ces travaux sont parmi celles permettant, de manière assez avancée, de simuler de façon conjointe

³ Obermann Center for Advanced Studies (University of Iowa), Department of Geography and Center for Global Change and Earth Observation (Michigan State University), Department of Geography and Carolina Population Center (University of North Carolina)

⁴ Department of Life & Environmental Sciences (Universidad San Francisco de Quito)

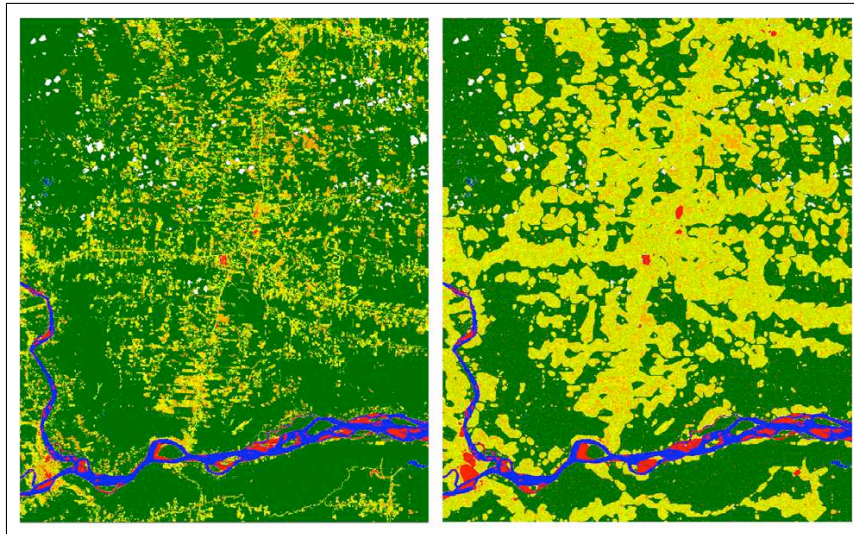


FIG. 1.11 – L'évolution de surfaces pastorales (en jaune) en Equateur. A gauche l'état initial, à droite l'état final (extrait de [Walsh *et al.* 2008])

l'évolution de différents types d'espaces. En effet, l'évolution des exploitations agricoles se fait notamment au détriment des zones de forêt primaire dont l'évolution est prise en compte.

1.5 Conclusion

Dans ce premier chapitre, nous avons défini un ensemble d'éléments qui permettent de poser les bases des travaux qui seront décrits dans la suite de ce manuscrit. En particulier, nous avons donné une vue d'ensemble du champ de la modélisation et de la simulation des systèmes complexes naturels et sociaux, en nous attardant sur ceux qui constitueront la majeure partie de nos travaux appliqués : les outils de simulation orientée agent dédiés au domaine de la prospective territoriale.

En guise de conclusion nous présentons ici des éléments que nous avons identifiés et qui sont ceux qui ont guidé les travaux menés dans la suite de cette thèse. Deux points principaux ont attiré notre attention et seront développés de façon transversale dans les chapitres à venir dans la suite de ce manuscrit :

1. un manque au niveau méthodologique pour tout ce qui touche à la prise en compte de la notion d'émergence qui apparaît souvent dès que l'on touche à des modèles répondant à des problématiques spatiales dans lesquelles il est courant de voir coexister plusieurs niveaux de descriptions et d'entités ;
2. la nécessité de créer des outils de simulation qui permettent de développer le domaine de la prospective territoriale à La Réunion, ceci passant par l'identification de points cruciaux qu'il sera important de prendre en compte pour améliorer les interactions entre thématiciens, modélisateurs et informaticiens.

Relativement au point 1, on se rappelle, comme nous l'avions souligné en citant Epstein dans la section 1 de ce chapitre, que les démarches de modélisation et de simulation auxquelles nous nous intéressons ont pour but de conduire à de nouveaux questionnements qui entraînent la découverte et la production de connaissances nouvelles. Or on constate que la plupart du temps ces nouvelles connaissances sont relatives à des phénomènes qui apparaissent ou à des entités qui se forment au cours d'expériences de simulation alors qu'ils n'étaient pas explicitement représentés au moment de la conception des systèmes virtuels ; on parle alors de phénomènes émergents. Ces phénomènes sont parfois vus comme étant les simples résultats attendus du modèle simulé, mais ils sont aussi vus comme étant les meilleurs représentants des connaissances nouvelles qui peuvent être produites, et ce même s'ils sont peu évoqués dans les différentes méthodologies de conception de simulations existantes. C'est donc pourquoi nous consacrerons le prochain chapitre à la notion d'émergence dans sa globalité, et que nous proposerons dans la deuxième partie de ce manuscrit un ensemble d'éléments conceptuels, méthodologiques et architecturaux visant à améliorer sa prise en compte dans les SOA.

Et relativement au point 2, et pour revenir de façon plus concrète sur la prospective territoriale, on constate à ce stade que les décideurs réunionnais manquent d'outils pouvant leur servir d'aide à la décision, et que la plupart des outils qui existent par ailleurs sont évolués et pertinents (et la plupart du temps performants), mais sont souvent centrés sur des champs d'expertise uniques : urbanisation, évolutions agricoles, conservation naturelle et préservation d'espèces, etc. Les travaux que nous présenterons par la suite ont donc également pour but de permettre aux thématiciens et informaticiens réunionnais de contribuer, ensemble, à la construction d'outils dans lesquels différentes dynamiques peuvent coexister, comme nous le verrons dans la troisième partie de ce manuscrit centrée sur le modèle DS.

La notion d'émergence

Sommaire

2.1 Introduction	39
2.2 L'émergence ?	40
2.2.1 Positionnement et précisions	41
2.2.2 Quelques exemples simples et visuels	41
2.2.3 Histoire "récente" de l'émergence	45
2.3 Définitions et classifications	46
2.3.1 Définitions célèbres et concepts transversaux	46
2.3.2 Oppositions et classifications	49
2.4 Emergence et simulation	53
2.4.1 Emergence et approches de modélisation multi-niveaux	54
2.4.2 Emergence et outils de simulation	57
2.5 Conclusion	59

2.1 Introduction

La notion d'émergence, que les philosophes Grecs définissaient en utilisant la phrase désormais célèbre "le Tout est plus que la somme des Parties" [[Aristote](#)], est présente dans l'ensemble des domaines scientifiques. Les physiciens, biologistes, chimistes, géographes ou autres sociologues sont ainsi tous confrontés à l'apparition de phénomènes dits *émergents* (comportements collectifs chez les fourmis et les termites, apparition d'un langage commun dans un groupe social humain, présence d'yeux chez des espèces radicalement différentes mais vivant toutes dans un environnement muni d'une source lumineuse, etc.) qu'ils tentent de comprendre et d'expliquer.

Dans cette optique, leur réflexion les amène à essayer de trouver les réponses à un ensemble de questions récurrentes [[Di Marzo Serugendo et al. 2006](#), [Fromm 2005](#)] parmi lesquelles :

- qu'est ce qui émerge ?
- quelles sont les propriétés du phénomène qui a émergé ?
- peut-on comprendre l'émergence de ce phénomène ?

Ces trois questions étant incluses dans celle, plus générale, mais qui se pose finalement à tous : au fond, *qu'est-ce que l'émergence ?*

Dans ce chapitre nous allons donc présenter un état de l'art des travaux axés autour de cette notion, passant ainsi en revue quelques unes des définitions et classifications qui en ont été établies. Nous présenterons également un certain nombre de notions transversales qui sont souvent rattachées au concept même d'émergence, et, enfin, nous donnerons quelques exemples de modélisations et d'outils de simulations qui intègrent l'émergence, le plus souvent comme élément découlant d'une modélisation multi-niveaux.

2.2 L'émergence ?

Cette question, qui se pose dès que l'on s'intéresse à des phénomènes du monde réel, se pose aussi dans leur pendant virtuel. En effet, et comme nous l'avons détaillé au chapitre précédent, dans le monde des systèmes complexes, et en particulier dans celui des Systèmes Multi-Agents (SMA) dédiés à la représentation de systèmes naturels et sociaux, nous nous attachons à reproduire des phénomènes qui se produisent dans des systèmes ou processus du monde réel [Banks 2000] pour améliorer nos connaissances sur celui-ci. L'émergence étant présente dans nombre de ces systèmes ou processus réels, elle se retrouve *de facto* présente dans les systèmes virtuels que nous créons pour les représenter. Travailler sur la notion d'émergence, sa représentation et son intégration au sein de tels systèmes est donc essentiel pour améliorer notre compréhension du système lui-même et par là même celle des phénomènes réels qui y sont représentés.

De nombreux travaux ont été menés en ce sens par les membres de la communauté SMA [Moncion 2008] et l'on dispose aujourd'hui d'une foison de définitions et de classifications [Dessalles *et al.* 2008, Deguet *et al.* 2007]. Mais qu'importe qu'elle soit considérée comme unilatérale ou bilatérale [Castelfranchi 1998], faible ou forte [Dessalles *et al.* 2007], synchrone ou diachronique [Stephan 2002], intrinsèque ou causale [Boschetti & Gray 2007], l'émergence trouve dans les SMA un formidable cadre d'expression.

Ainsi, comme énoncé dans [M.R.Jean 1997], les SMA constituent pour l'émergence, de par leur centre d'intérêt, à savoir des ensembles d'entités en interactions, un moyen privilégié de mise en œuvre et d'étude. Particulièrement adaptés pour appréhender ce concept, ils permettent de mettre en avant les phénomènes qui émergent dans le système.

Ce faisant, il devient intéressant de concevoir l'émergence comme pouvant prendre place en tant qu'entité à part entière dans une SOA. Si l'on se place dans le cadre de la simulation d'un lagon et de ses poissons, le fait de pouvoir d'une part détecter les éventuels bancs de poissons qui peuvent se former, et d'autre part de pouvoir les injecter dans le système en tant que tels présenterait un intérêt notable, ne serait-ce qu'en terme de chute de complexité

du système, ce qui est une des caractéristiques de l'émergence [Bonabeau & Dessalles 1997], et c'est ce vers quoi nous essayerons de faire tendre les travaux qui seront présentés dans la deuxième partie de ce manuscrit.

2.2.1 Positionnement et précisions

L'émergence est une notion tellement vaste et controversée qu'une simple thèse qui y serait complètement dédiée ne suffirait pas à couvrir l'étendue des travaux qui ont déjà été effectués. Aussi nous ne sommes pas là pour faire une revue exhaustive de tout ce qui touche à l'émergence, mais simplement pour donner quelques pistes de réflexion quant à la façon de parvenir à intégrer un peu mieux tout ce qui y est relatif dans nos modèles et dans nos outils de simulation. Les définitions et exemples listés dans la suite de ce chapitre sont donc là pour donner un éclairage qui permet de bien situer le contexte qui a conduit à l'émergence (eh oui !) des réflexions et propositions décrites dans les chapitres suivants.

Avant d'aller plus loin, précisons également que nous distinguons évidemment "l'émergence", en tant que concept, des "phénomènes émergents", en tant que faits qui occurrent dans un système, et que nous ne limitons donc pas notre étude à ces seuls "phénomènes émergents" au sens strict. Dans l'esprit, et par confort de lecture dans la suite de ce manuscrit lorsque la confusion des genres n'est pas trop importante, nous emploierons souvent invariablement l'un ou l'autre de ces termes. Mais il est très important de noter que dans notre démarche, et comme nous le verrons au prochain chapitre, nous considérons l'émergence comme étant au cœur du processus de création de connaissance, que ce soit au sein d'un système simulé, ou que ce soit dans l'esprit des concepteurs et utilisateurs de ce système. Il est donc bon de noter que, dans une démarche de conception de simulations, l'émergence la plus intéressante n'est pas celle de phénomènes dans la simulation, mais bien celle d'idées nouvelles, car ces idées, qui viennent alimenter les réflexions et analyses, sont souvent la source des plus grandes découvertes et avancées (quand elles ne constituent pas une avancée à elles seules).

Enfin, il va de soi que ce que nous considérons et présentons comme étant des phénomènes émergents pourraient tout aussi bien, selon l'angle (ou plutôt le niveau) de lecture choisi, n'être parfois considérés que comme relevant de la simple apparition de phénomènes, ou agrégation d'entités. Mais c'est dans un souci d'ouverture la plus large possible que nous les incluons à travers nos propos comme relevant de "l'émergence" même s'ils ne "méritent" parfois pas d'y être au regard de certaines définitions plus rigoureuses.

2.2.2 Quelques exemples simples et visuels

De prime abord, "l'émergence" est un concept un peu vague. Comme il est bien connu qu'une image vaut mille mots [Confucius], nous allons présenter et illustrer ici quelques exemples de phénomènes émergents bien connus et facilement observables qui permettront au lecteur néophyte de mieux intégrer cette notion. Les figures sont des résultats de simulations effectuées avec la plateforme de simulation NetLogo [Wilenski 1999] ou avec la plateforme de

simulation GEAMAS-NG.

Sugarscape

L'exemple de Sugarscape est une des références pour l'observation de dynamiques sociales dans les systèmes multi-agents. Il a été proposé par Epstein et Axtell dans [Epstein & Axtell 1996].

Les agents, initialement dispersés au sein de l'environnement, se rassemblent rapidement autour des deux "montagnes" de sucre [Peterson 1996] comme on peut le voir sur la figure 2.1 ; parmi eux, quelques-uns finissent par accumuler de grandes quantités de sucre, interprétables comme autant de richesses personnelles. Ces individus "fortunés" sont, inmanquablement, ceux qui combinent un large champ de vision à un métabolisme lent. D'autres, combinant un lent métabolisme à un champ de vision restreint, se maintiennent en bordure des "montagnes" où ils collectent juste assez de sucre pour survivre, sans voir les réserves de sucre qui s'amoncellent par-delà l'horizon.

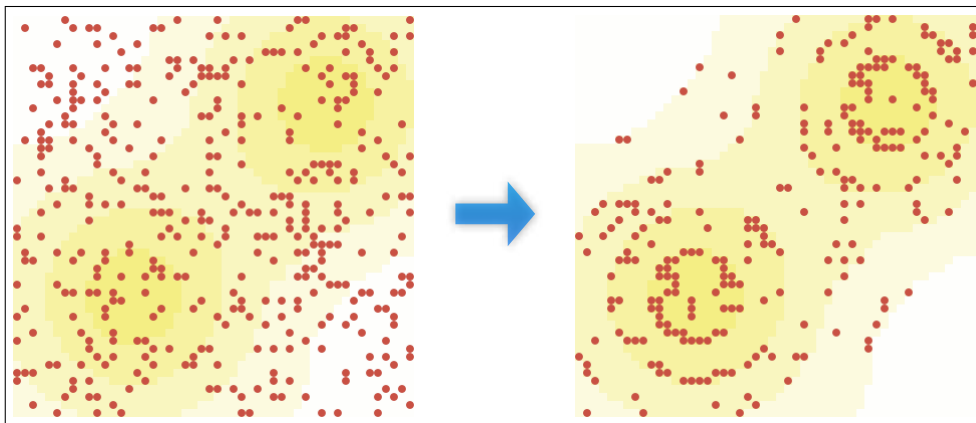


FIG. 2.1 – La répartition des agents sur les montagnes de sucre de SugarScape

Ce modèle se démarque comme étant un laboratoire dans lequel des sociétés virtuelles peuvent être créées et examinées. Bien que le lien qui puisse exister entre ces mondes virtuels et le monde réel reste discutable, les comportements sociaux complexes qui en émergent sont clairement reconnaissables, et ce malgré la simplicité des règles et des agents qui les composent.

Les HeatBugs

Dans cet exemple, décrit de façon plus complète dans [Daniels 1999], on suit l'évolution d'une population d'insectes, des "HeatBugs", dont chacune a une température idéale et dégage une certaine quantité de chaleur. La population partage un espace torique permettant justement la diffusion de chaleur. Lorsque les entités sont satisfaites de leur environnement calorifique elles restent immobiles ou se stabilisent et, dans le cas contraire, elles continuent à se déplacer.

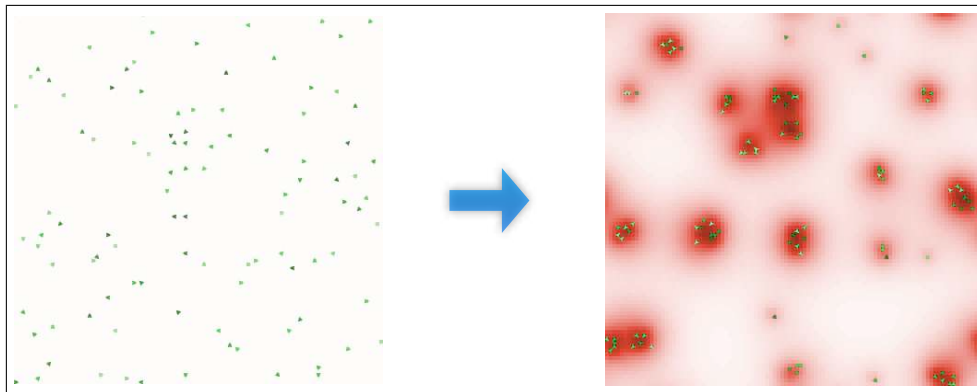


FIG. 2.2 – Les HeatBugs et la chaleur

Avec le temps on constate l'émergence de regroupements d'insectes qui volent tous dans une zone dans laquelle la chaleur s'est stabilisée autour d'une température qui leur paraît idéale.

Les Fourmis

L'exemple des agents fourmis qui cherchent à relier leur nid à une source de nourriture est sûrement l'un des plus célèbres lorsque l'on parle de phénomènes émergents car il est à la fois l'un des plus simples et des plus parlants.

Dans le système (décrit de façon plus complète dans [Bonabeau *et al.* 1997]), les fourmis sont dotées d'un comportement des plus simples (se déplacer aléatoirement ; suivre les éventuelles directions vers lesquelles elles perçoivent un maximum de phéromones ; déposer des phéromones sur leur propre passage) mais au bout d'un moment l'ensemble des agents fourmis empruntera le plus court chemin entre leur nid et la source de nourriture. Ce plus court chemin, visible sur la figure 2.3 sur laquelle la trace des phéromones est représentée en dégradé de noir, a "émergé" du comportement collectif des agents fourmis et ne doit son existence qu'à la considération du système dans sa globalité.

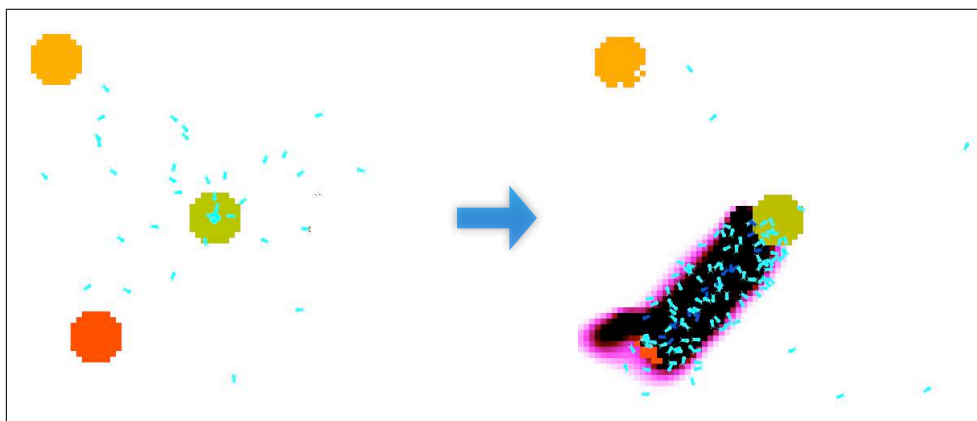


FIG. 2.3 – Le plus court chemin de la colonie de fourmis

Le Flocking

Cet exemple, illustré sur la figure 2.4, correspond à la reproduction des schémas de déplacement de bancs de poissons, vols d'oiseaux et autres troupes d'entités autonomes à partir de prises de décisions individuelles. L'une de ses descriptions les plus célèbres est le modèle Boids décrit dans [Reynolds 1986]. Il repose sur trois règles de comportement qui sont suivies par chacune des entités du modèle :

- une règle de séparation ;
- une règle d'alignement ;
- une règle de cohésion.

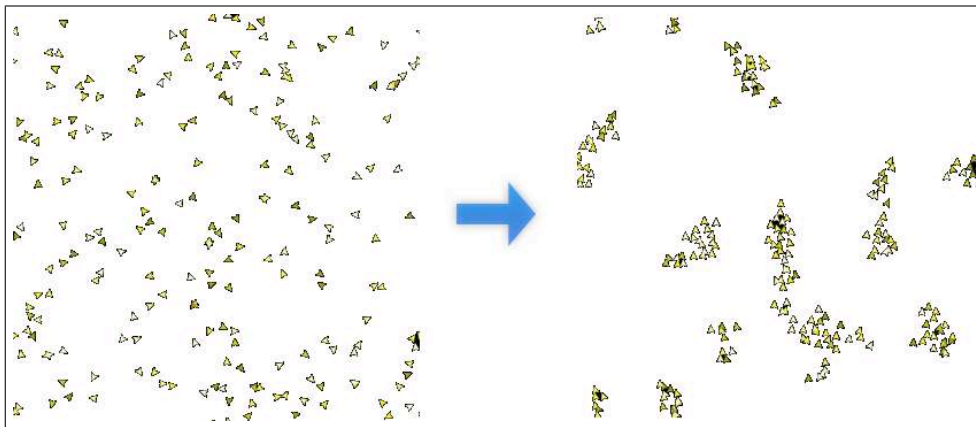


FIG. 2.4 – Le phénomène de flocking observé chez les oiseaux migrateurs

Cet exemple démontre de façon explicite dans quelle mesure des prises de décisions individuelles peuvent aboutir à l'émergence de mouvements de groupe, sans pour autant nécessiter l'élection d'un chef.

Le jeu de la vie

Le jeu de la vie est un exemple célèbre qui a été imaginé par John Conway en 1970. Il est décrit dans [Berlekamp *et al.* 2002]. C'est un automate cellulaire dont les cellules n'ont que deux états, vivantes ou mortes, et qui permet, malgré des règles très simples de "décès" et de "survie" pour chacune des cellules, le développement de motifs extrêmement complexes.

Trois types de structures émergentes sont ainsi couramment identifiées :

- les structures stables ;
- les structures périodiques (comme la structure oscillante résultant de la succession d'étapes représentée sur la figure 2.5) ;
- les glisseurs, ou vaisseaux, petites structures qui se "déplacent" dans l'environnement.

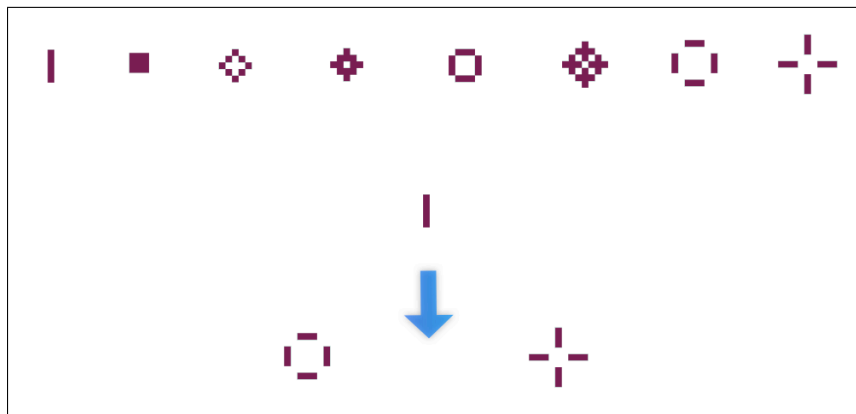


FIG. 2.5 – Les oscillateurs dans le jeu de la vie

2.2.3 Histoire "récente" de l'émergence

La perception et la compréhension de la notion d'émergence, que l'on devinait à travers les phrases antiques "le tout est plus que la somme des parties" et "le tout avant les parties", ont évidemment évolué au fil du temps et des époques. Ainsi ce concept, enrichi par les réflexions sur la "Gestalt" de Goethe, a donné naissance, au début du XXème siècle, à un courant appelé le "proto-émergentisme".

Dans le proto-émergentisme, les phénomènes liés à l'émergence sont vus comme faisant partie d'une boîte noire possédant des entrées et des sorties multiples. Ces entrées, de bas niveau, et ces sorties, de haut niveau, sont connues mais on ignore tout des processus qui ont permis de passer des unes aux autres. Cette vision des choses permettait de parvenir à identifier des phénomènes relevant de l'émergence, mais ne permettait en aucune façon d'expliquer le comment du pourquoi de leur apparition ou de leur fonctionnement. Il faut dire que le but n'était alors pas tant de comprendre les phénomènes émergents que de les reconnaître afin de pouvoir différencier ces phénomènes de ceux qui pourraient être expliqués au moyen d'autres modèles ou d'autres théories (qu'elles soient à dominante mathématiques, physiques, ou autres). Les travaux de [Lewes 1875] résument bien cette vision des choses puisqu'ils expliquent que la séquence d'étapes qui produisent un phénomène est traçable alors que, pour ce qui relève de l'émergence, il n'est pas possible de tracer les étapes du processus.

C'est donc naturellement qu'un courant dit du "néo-émergentisme" est apparu pour succéder au proto-émergentisme puisque le manque d'explications se devait naturellement d'être comblé. Il s'agissait alors de se plonger au cœur des choses et (de tenter) de comprendre ce qui se passe dans la boîte noire précédente afin de comprendre les mécanismes internes qui mènent à l'émergence de phénomènes et qui permettent donc d'obtenir un éclairage nouveau sur leur fonctionnement et par là même sur celui des systèmes qui les englobent dans lesquels ils sont apparus. Comme bien résumé dans [Georgé 2004], ce courant se base essentiellement sur :

- la théorie des systèmes complexes adaptatifs (Santa Fe Institute : Holland, Kauffman, Langton) : l'émergence fait référence aux patterns de niveau macro apparaissant dans

- des systèmes d'agents en interaction ;
- la théorie des systèmes dynamiques non linéaires issue de la théorie du chaos ;
 - l'école synergétique fondée par Hermann Haken qui étudie l'auto-organisation dans les systèmes physiques ;
 - la thermodynamique Loin et autour de l'Equilibre introduite par Ilya Prigogine : les phénomènes émergents sont les structures dissipatives apparaissant à partir de conditions Loin de l'Equilibre.

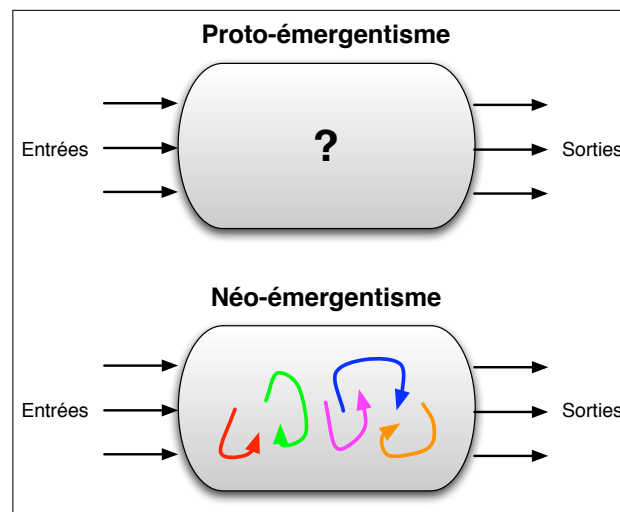


FIG. 2.6 – Proto-émergentisme et Néo-émergentisme : de la boîte noire à la compréhension des processus d'émergence

Proto-émergentisme et néo-émergentisme sont classiquement et simplement représentés comme sur la figure 2.6. Les nombreux travaux qui sont ressortis de ces deux courants ont ensuite conduit à la publication d'un large panel de définitions centrées sur la notion d'émergence.

2.3 Définitions et classifications

2.3.1 Définitions célèbres et concepts transversaux

Les définitions sur l'émergence et les phénomènes émergents sont pléthores et nous ne les reproduirons pas toutes, mais nous avons jugé utile, au vu de la suite des travaux qui seront présentés dans ce manuscrit, de faire apparaître ici certaines des plus célèbres d'entre elles. Ceci permet de mieux mesurer la vaste taille du contexte auquel on se retrouve confronté dès lors que l'on s'intéresse à l'émergence. En effet, si la plupart des définitions présentées ont un socle commun, elles n'en diffèrent pas moins par de nombreux détails qui illustrent les points de vue parfois radicalement différents des auteurs.

Emergence 1 (Arkin) *Emergence is often invoked in an almost mystical sense regarding the capabilities of behavior-based systems. Emergent behavior implies a holistic capability where the sum is considerably greater than its parts.*

Emergence 2 (Atlan) *L'émergence fait référence à l'apparition durant le processus d'auto-organisation dans un système complexe de structures ou de schémas (patterns) ou de propriétés nouvelles et cohérentes. Ces structures et/ou des fonctions apparaissent à un niveau macroscopique sans que la seule observation des propriétés des constituants permette de les prédire. [Atlan 2000]*

Emergence 3 (Bedau) *There are a variety of notions of emergence, and they are contested. We can provide some order to this controversy by distinguishing two hallmarks of how macro-level emergent phenomena are related to their micro-level bases : emergent phenomena are dependent on underlying processes, emergent phenomena are autonomous from underlying processes.*

Emergence 4 (Churchland) *In scientific contexts, one frequently hears it used to apply to what might be called a "network property", a property that appears exactly when the elements of some substrate are suitably organized, a property that consists in the elements of that substrate standing in certain relations to one another, a set of relations that collectively sustain the set of causal powers ascribed to the "emergent" property. In this innocent sense of "emergent", there are a great many emergent properties, and quite probably the qualia of our sensations should be numbered among them. But in philosophical contexts one more often encounters a different sense of "emergent", one that implies that an "emergent" property does not consist in any collective or organizational feature of its substrate. The first sense positively implies reducibility ; the second implies irreducibility.*

Emergence 5 (Darley) *A true emergent phenomenon is one for which the optimal means of prediction is simulation.*

Emergence 6 (Van de Vijver) *D'une part, l'émergence présuppose qu'il y a apparition de nouveauté (propriétés, structures, formes, fonctions) et d'autre part elle implique qu'il est impossible de décrire, d'expliquer ou de prédire ces nouveaux phénomènes en termes physiques à partir des conditions de base définies aux niveaux inférieurs. [Van de Vijver 1997]*

Emergence 7 (Drogoul) *L'émergence est un concept flou, mal défini et souvent décrié. Il dénote d'une manière générale, dans les systèmes qualifiés de complexes ou de non-linéaires, l'apparition d'un effet global qui n'est pas déductible de la connaissance des causes locales. Dans un système multi-agents, il dénote donc l'apparition d'un phénomène global qui n'est pas explicitement programmée dans le comportement des agents [Drogoul 1993].*

Emergence 8 (Langton) *L'émergence se définit en termes de relations de feedback entre les niveaux dans un système dynamique. Les micro-dynamiques locales causent les macro-dynamiques et les macro-dynamiques globales contraignent les locales. [Langton 1990]*

Emergence 9 (Lewes) *L'émergence est une théorie selon laquelle la combinaison d'unités d'un certain ordre réalise une entité d'ordre supérieur dont les propriétés sont entièrement nouvelles. [Lewes 1875, Grumbach 1997]*

Emergence 10 (Müller) *Unfortunately, the notion of emergence itself is problematic for the modeller as well as for the designer, being most often defined by an absence of "something" as composability, predictability, and so on when the notion itself is not criticised. [...] A phenomenon is emergent if and only if we have :*

- *a system of entities in interaction whose expression of the states and dynamics is made in an ontology or theory D ;*
- *the production of a phenomenon, which could be a process, a stable state, or an invariant, which is necessarily global regarding the system of entities ;*
- *the interpretation of this global phenomenon either by an observer or by the entities themselves via an inscription mechanism in another ontology or theory D'.*

Analyse et concepts transversaux

Ce panel de définitions est volontairement large et disparate, mais il est centré sur des définitions qui sont la plupart du temps établies par des informaticiens, et plus particulièrement par des chercheurs dans le champ de la modélisation et simulation de systèmes complexes. Aussi l'émergence est la plupart du temps définie en fonction des propriétés des systèmes ou des entités qui sont simulés. Et, vue par ces informaticiens et modélisateurs, l'émergence est en quelque sorte réduite aux seuls phénomènes émergents, même si l'on sent dans quelques définitions (par exemple dans celles de Drogoul ou de Müller) les traces de la notion même d'émergence au sens large. Il est donc important de rappeler qu'étant dans une démarche de découverte et de production de connaissance nous nous intéressons dans nos travaux aussi bien à l'émergence de phénomènes dans des systèmes qu'à celle de nouvelles connaissances dans l'esprit des concepteurs et utilisateurs de ces systèmes.

Si l'on se focalise sur l'émergence de phénomènes, on se rend compte qu'un grand nombre de propriétés, dont la plupart sont énoncées dans [Georgé 2004], se rattachent à la notion d'émergence et transparaissent dans certaines des définitions que nous venons de lister :

- la nouveauté radicale, que l'on retrouve dans les définitions de Van de Vijver et de Lewes ;
- l'interdépendance des niveaux, que l'on retrouve dans les définitions de Langton et de Atlan, et qui est reprise dans de nombreux travaux [Yamins 2005a, Yamins 2005b] ;
- l'irréductibilité, que l'on retrouve dans la définition de Churchland et dans les travaux de [Ali & Zimmer 1997] ;
- l'auto-organisation, que l'on retrouve dans la définition de Atlan.

De plus, n'oublions pas que l'étude des relations "Tout / Parties" est en lien étroit avec la notion d'émergence. Et ce d'autant plus que de nombreux domaines littéraires ou philoso-

phiques s'y intéressent. Ceci peut paraître surprenant de prime abord mais, lorsque l'on se rappelle que le fameux "Le Tout est plus que la somme des Parties" date de l'antiquité et des philosophes grecs, cela paraît plus naturel.

Ainsi des travaux menés dans des champs comme celui de la méronymie ou de l'holonymie sont intéressants à bien des égards [Keet 2006]. La méronymie et l'holonymie étant des relations sémantiques entre mots d'une même langue qui sont des relations partivistes hiérarchisées. On dit qu'un méronyme A d'un mot B est un mot dont le sens désigne une sous-partie du sens de B, la relation inverse étant l'holonymie. Par exemple, "Pied" est un méronyme de "Corps", et "Oiseau" est un holonyme de "Bec".

Voici par exemple quelques relations Tout/Parties à travers lesquelles on retrouve la notion d'émergence [Aurnague & Plenat 1996] :

- "composant / assemblage" : la roue de la voiture
- "morceau / tout" : le haut de la montagne
- "portion / tout" : une tranche de gâteau
- "substance / tout" : le rhum de ce punch
- "élément / collection" : une brebis du troupeau
- "sous-collection / collection" : les DOM font partie de l'Europe

Certes, certains de ces exemples sont très proches de la simple agrégation d'objets, en particulier celui relevant de la relation "composant / assemblage" ; mais considérons l'exemple (très réunionnais !) de la relation "substance / tout : le rhum de ce punch". Cet exemple est remarquable car dans ce cas le Tout est bien plus que la somme des Parties (il est bien meilleur de boire un punch que d'ingurgiter directement du sucre, des fruits et du rhum !) mais aussi puisqu'il illustre même une propriété qui ressort de plusieurs définitions et que nous venons de lister : le caractère d'irréductibilité. En effet, une fois que le rhum (la partie) a été utilisé pour produire le punch (le tout), il devient impossible de le récupérer en tant que tel, le processus étant non-réversible !

2.3.2 Oppositions et classifications

Au-delà des quelques définitions et concepts transversaux que nous venons de parcourir, certains auteurs proposent des classifications de l'émergence, ou plutôt des classifications des différents types d'émergences possibles. Ceci permet de dépasser les limites imposées par des définitions trop strictes et permet de montrer que l'émergence n'est pas une chose figée. Ainsi, vu au sens large, ce concept englobe bon nombre de phénomènes de types et d'ampleurs bien différents.

Quelques oppositions

Nous avons choisi de lister trois des oppositions classiques qui sont faites lorsque l'on parle d'émergence :

- émergence unilatérale et émergence bilatérale ;

- émergence synchronique et émergence diachronique ;
- émergence faible et émergence forte.

Ces trois oppositions sont particulièrement intéressantes car elles sont chacune le reflet de certains des concepts transversaux détaillés à la section précédente.

Emergence unilatérale et Emergence bilatérale

Dans le cas de l'émergence unilatérale, ce qui se passe dans un système à l'échelle microscopique explique directement ce qui se passe à l'échelle macroscopique. Ceci est illustré sur la figure 2.7. En considérant que l'on ait connaissance du bon ensemble de lois, on peut donc prédire de façon directe l'évolution des phénomènes émergents (de haut niveau).

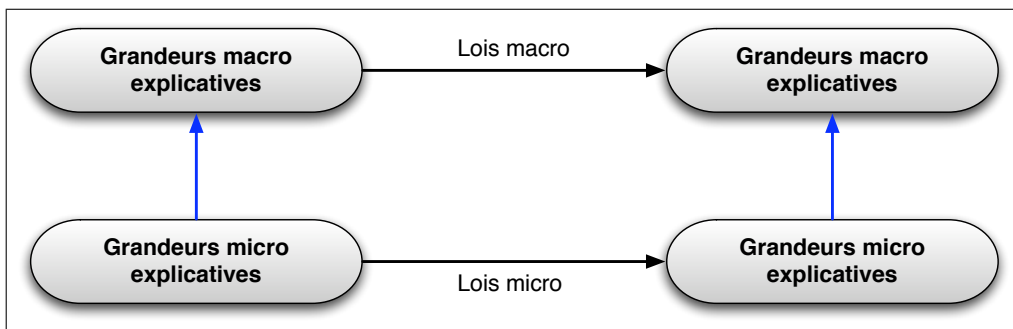


FIG. 2.7 – L'émergence unilatérale : le micro explique directement le macro

En revanche, dans le cas de l'émergence bilatérale, illustrée sur la figure 2.8, il y a présence d'interactions dans le système entre les différents niveaux qui le constituent. Les influences entre les niveaux microscopique et macroscopique sont à la fois ascendantes et descendantes.

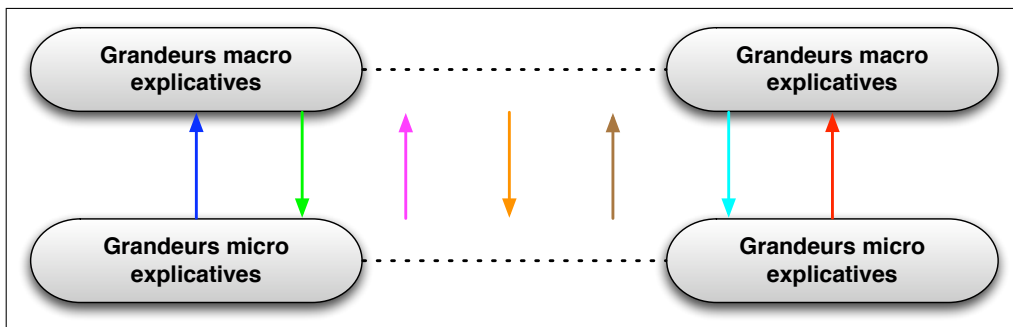


FIG. 2.8 – L'émergence bilatérale : influences ascendantes et descendantes entre micro et macro

Le concept transversal qui ressort de cette opposition est celui de la séparabilité qui existe dans un cas mais pas dans l'autre.

Emergence synchronique Vs Emergence diachronique

Dans le cas de l'émergence synchronique et de l'émergence diachronique, qui sont illustrées sur la figure 2.9, l'opposition principale est liée à la façon dont est prise en compte l'aspect

temporel.

Ainsi, dans le cas de l'émergence synchronique, le lien qui existe entre les entités du niveau microscopique et les phénomènes qui émergent au niveau macroscopique est instantané : l'émergence est un résultat, par exemple l'occurrence d'une structure particulière, qui est dû aux comportements des entités microscopiques à un moment donné. Ces entités microscopiques continuent d'ailleurs d'exister en même temps que le phénomène qu'elles ont fait émerger.

En revanche, dans le cas de l'émergence diachronique, la notion de temps qui s'écoule devient importante, et l'émergence est le résultat d'un ensemble d'influences et d'interactions qui se sont déroulées sur du (relativement) long terme. Les phénomènes qui apparaissent au niveau macroscopique ont émergé de façon progressive et on peut même envisager que les entités microscopiques qui ont contribué à faire émerger ces phénomènes puissent disparaître sans que ceux-ci en soient affectés. Evidemment il arrive que des phénomènes émergents relèvent de l'émergence diachronique, mais qu'en se stabilisant progressivement ils apparaissent comme relevant de l'émergence synchronique [Walliser 2009].

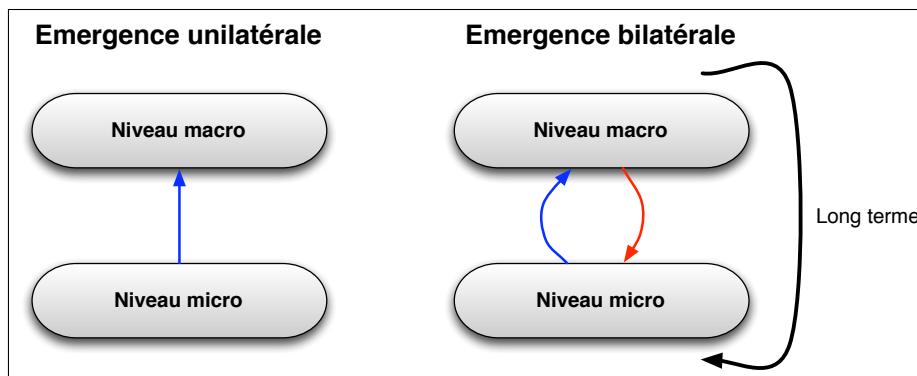


FIG. 2.9 – L'émergence synchronique et l'émergence diachronique

Plusieurs concepts transversaux ressortent de cette opposition, notamment ceux d'irréductibilité et d'interdépendance des niveaux.

Emergence faible Vs Emergence forte

On parle d'émergence faible lorsque les structures ou propriétés émergentes sont identifiées par un observateur extérieur au système dans lequel le phénomène a émergé. Elle peut bien entendu être identifiées dans le système par l'emploi de techniques de détection de phénomènes émergents, mais elles ne jouent aucun rôle en son sein.

En revanche, on parle d'émergence forte lorsque les structures ou propriétés émergentes sont intrinsèques au système et qu'elles sont indépendantes de l'observation qui pourrait être faite d'un point de vue extérieur.

Le concept transversal qui ressort de cette opposition est celui de l'irréductibilité, lui aussi déjà détaillé à la section précédente (voir l'exemple du "Rhum de ce punch").

Classification de Stephan

Dans [Stephan 2002], Achim Stephan reprend de façon croisée plusieurs des définitions et concepts présentés précédemment, et les utilise pour faire ressortir une classification relativement formelle des types d'émergence qui peuvent se manifester dans un système. Cette classification a depuis été reprise et adaptée dans de nombreux travaux, notamment ceux de Dessalles, Müller et Phan [Dessalles *et al.* 2007].

Comme illustré sur la figure 2.10, elle est pratique pour montrer comment s'accroissent les propriétés d'irréductibilité et de nouveauté radicale en fonction des situations d'émergence dans lesquelles l'on se trouve.

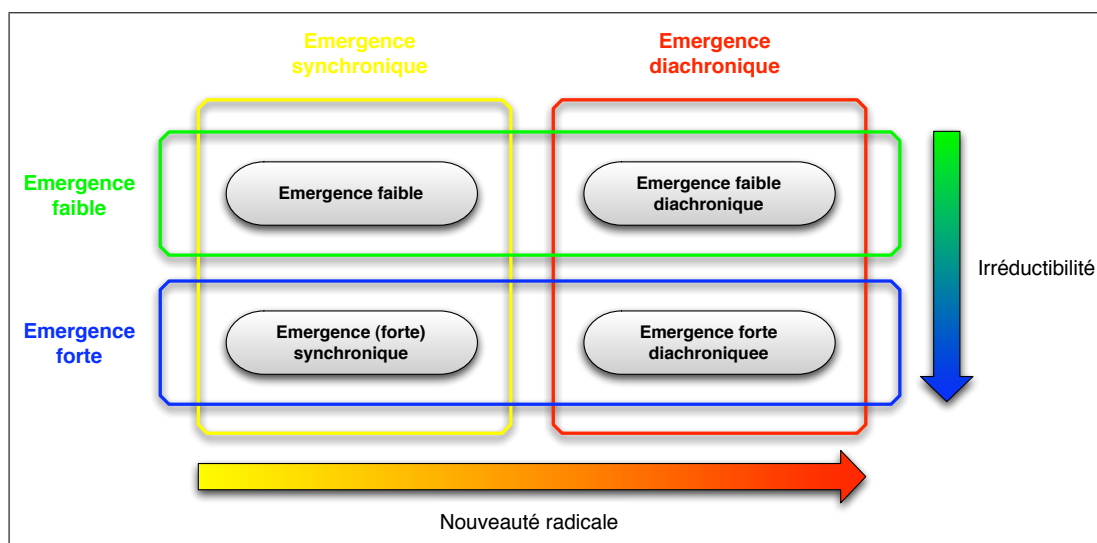


FIG. 2.10 – Une classification selon Stephan (adapté de [Stephan 2002] et [Dessalles *et al.* 2007])

A ce titre, cette classification s'avère être intéressante mais elle est difficile à mettre en pratique pour un néophyte confronté à l'émergence d'un phénomène. En effet, les termes utilisés pour décrire les types d'émergence étant explicites mais techniques, il est difficile de parvenir à classifier *a priori* le type de phénomène émergent auquel on est confronté.

Classification de Boschetti

Dans [Boschetti & Gray 2007] Boschetti reprend une classification qui est moins formelle que les précédentes, essentiellement car elle ne se base pas sur les différents concepts énoncés plus hauts, mais plutôt sur des éléments de bon sens. Elle est donc plus intuitive et plus facile à appliquer de prime abord.

Dans cette classification, illustrée sur la figure 2.11, les phénomènes émergents sont regroupés en trois catégories principales :

- les phénomènes relevant de la formation et de la détection de patterns ;
- une émergence dite "intrinsèque" ;

- une émergence dite "causale".

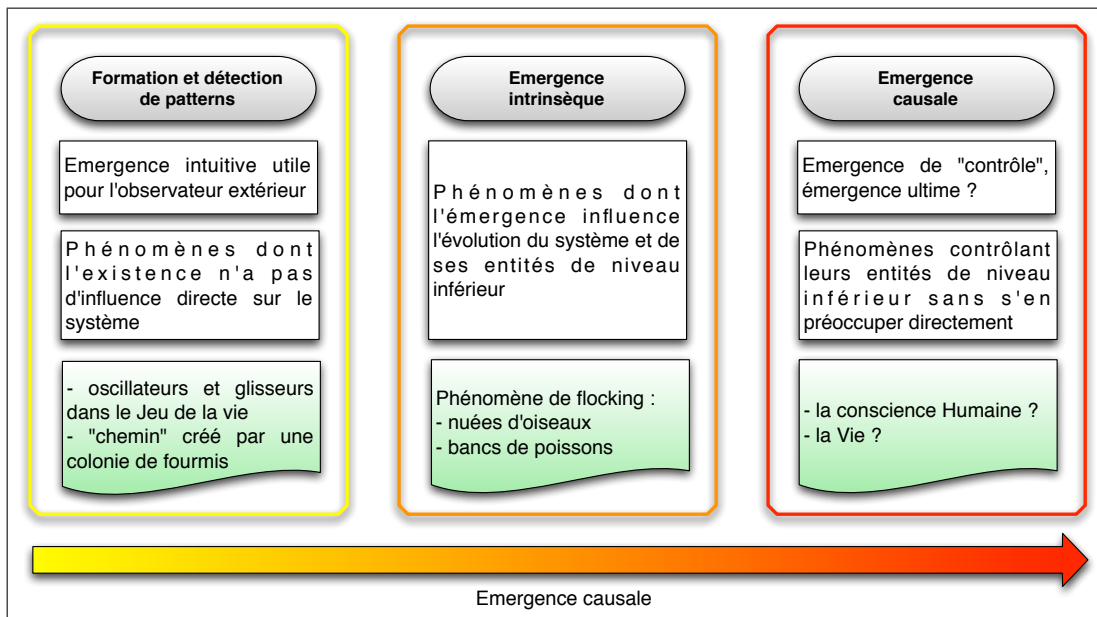


FIG. 2.11 – Une classification selon Boschetti [Boschetti & Gray 2007]

Ces trois catégories d'émergence réfèrent aux différents degrés de contrôle que peut avoir l'entité de haut niveau sur celles de bas niveau, liant ainsi l'intérêt d'un phénomène émergent à l'ampleur dont il est la cause [Deguet 2008]. La dernière catégorie d'émergence est la plus difficile à cerner, mais aussi, et évidemment, la plus intéressante, et il n'est pas sûr qu'on parvienne à l'appréhender à l'heure actuelle dans des systèmes informatiques. Cette classification est pleine de bon sens et a alimenté nos réflexions sur la prise en compte de l'émergence dans les méthodologies de conception de simulations que nous présenterons par la suite.

Ces éléments se retrouvent dans la vision de Crutchfield qui considère qu'il est bon de distinguer trois notions :

- l'émergence intuitive ("quelque chose est apparu");
- la formation de patterns ou d'organisations qui peuvent être observés par un observateur extérieur ;
- l'émergence intrinsèque dans laquelle le système lui-même capitalise sur les patterns qui sont apparus.

2.4 Emergence et simulation

Lorsque l'on évoque le concept d'émergence dans un contexte de simulation, on se rend compte que cette problématique est souvent liée au contexte méthodologique de la modélisation multi-niveaux. Il convient donc de s'intéresser à des modèles dans lesquels sont présentes plusieurs échelles de représentation de connaissances, et, typiquement, pour ce qui est des

Systèmes Multi-Agents, plusieurs niveaux d'agents entre lesquels existent des structures de groupes ou des liens hiérarchiques. Dans cette section, nous détaillons rapidement quelques méthodologies de modélisation multi-niveaux et quelques exemples d'outils de simulation qui permettent de prendre en compte – ou du moins qui tendent vers une prise en compte – du phénomène d'émergence dans les simulations.

2.4.1 Emergence et approches de modélisation multi-niveaux

Approche GEAMAS

L'étude de l'émergence occupe une place de choix dans les travaux de l'équipe SMART depuis une dizaine d'années. Ainsi dans [Marcenac *et al.* 1998] est proposée une façon de prendre en compte la gestion de structures de niveaux différents, en distinguant les spécificités des agents au niveau plateforme et au niveau simulation comme on peut le voir sur la figure 2.12. Dans cette approche, tout système est composé d'entités qui sont regroupées au sein de structures intermédiaires. Selon leur niveau d'existence sont donc distingués micro-agents, medium-agents, et macro-agents. Les premiers servent toujours à représenter les entités les plus basiques du système, tandis que les derniers peuvent servir à représenter des groupes, des sociétés, des organisations.

Cette approche est assez classique, mais elle est intéressante, surtout sur le plan de la conception d'applications, et donc dans notre contexte sur le plan de la conception du modèle informatique pour lequel elle cadre les possibilités de mise en œuvre qui sont faisables avec la plateforme GEAMAS.

Mais ces travaux ne s'accompagnent malheureusement pas d'une approche méthodologique globale qui permettrait de cadrer le processus complet de conception de simulation, et en particulier l'ensemble des étapes relevant de la modélisation car, comme nous l'avons souligné au chapitre précédent, c'est là l'un des points clés lors duquel la façon de prendre en compte l'émergence doit être améliorée.

Approche de Ratzé *et al*

Dans leurs travaux, Ratzé, Gillet, Müller et Stoffel font le point sur les problématiques d'échelles et de multi-niveaux dans le contexte de la modélisation et simulation de hiérarchies écologiques [Ratzé *et al.* 2007].

Ils distinguent, pour un même niveau d'organisation, des interactions faibles et des interactions fortes entre les différentes entités d'un système, et créent ainsi des frontières autour des entités qui ont des interactions fortes, ce qui est une façon à la fois fonctionnelle, spatiale, et temporelle de délimiter les sous-systèmes qui existent à un certain niveau par rapport à un niveau hiérarchique donné. Ils présentent ensuite une approche basée sur l'utilisation de "holons", qui sont considérés comme étant des entités pouvant être à la fois un composé et un composant, pour gérer l'articulation entre plusieurs niveaux d'organisation. Cette approche est représentée sur la figure 2.13.

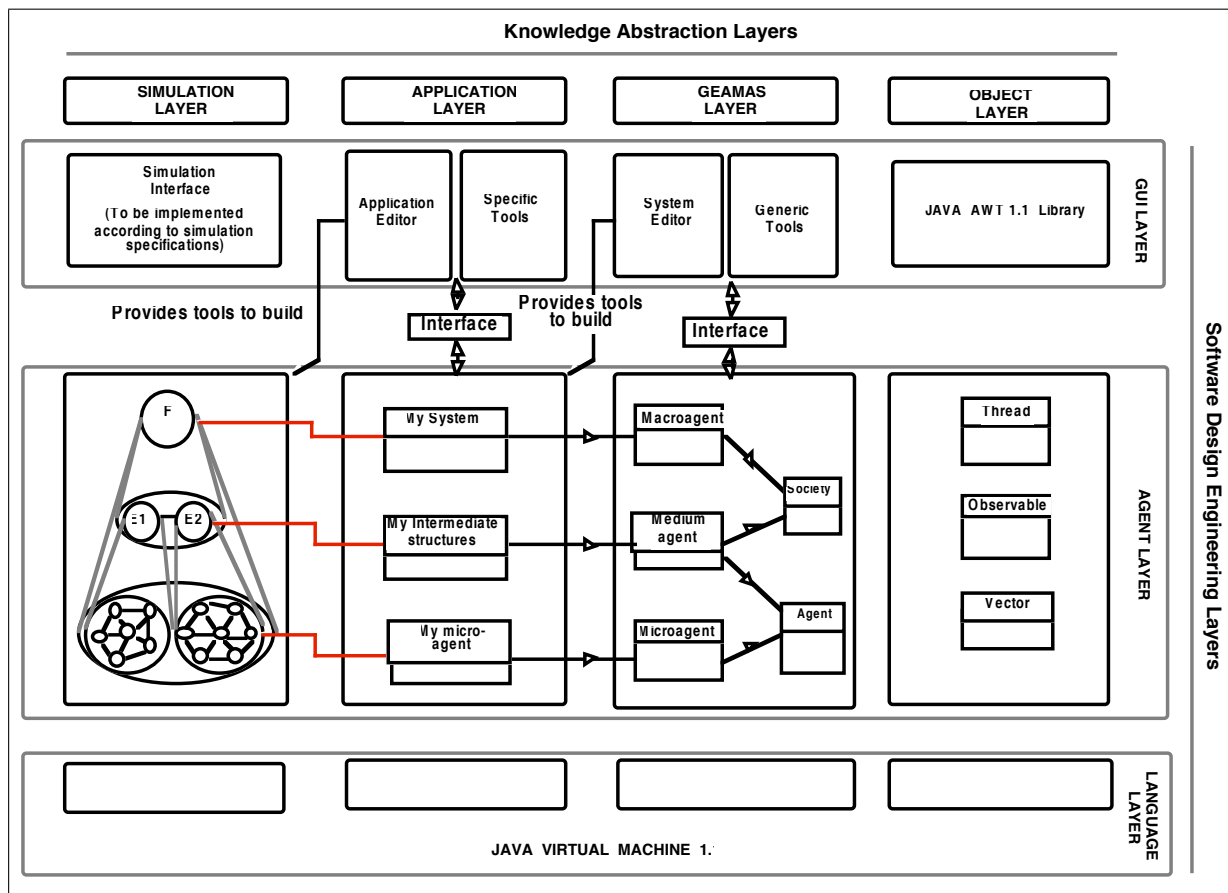


FIG. 2.12 – Modélisation multi-niveaux en SOA selon l'équipe SMART (extrait de [Marcenac *et al.* 1998])

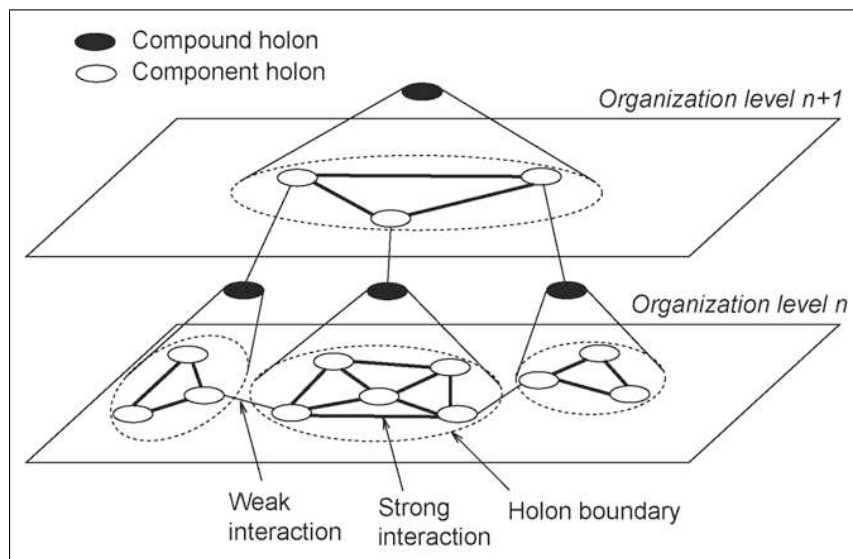


FIG. 2.13 – Les holons utilisés pour gérer l'articulation des organisations entre niveaux (extrait de [Ratzé *et al.* 2007])

Approche de Morvan *et al*

Dans des travaux récents [Morvan *et al.* 2008, Morvan 2009], et dans un contexte où l'objectif est la modélisation du comportement thermique d'un agrégat de larves afin de créer des applications SMA pour l'entomologie médico-légale, Gildas Morvan avance l'idée que la modélisation d'un système doit se faire selon deux points de vue :

- du point de vue des agents sont modélisées les conditions de naissance d'une organisation ;
- du point de vue de l'organisation sont modélisés les contraintes et émergences imposées aux agents, son comportement, et ses conditions de survie.

Dans une telle vision des choses, assez générique, une organisation n'est pas une structure mais est aussi un agent, comme on peut le voir sur la figure 2.14.

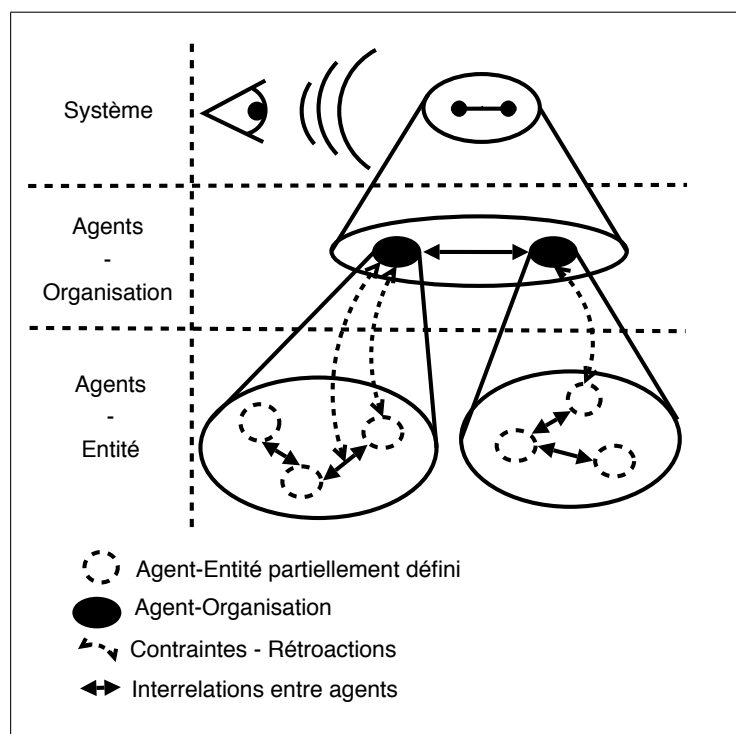


FIG. 2.14 – Modélisation multi-niveaux selon Morvan (extrait de [Morvan *et al.* 2008])

Le système comprend donc des Agents-Entités et des Agents-Organisations, qui sont liés entre eux, au sein d'un même niveau, par des interactions supposées classiques, et entre niveaux organisationnels différents par des mécanismes de contraintes et de rétroaction. Assez générique, cette approche reprend dans l'esprit les mêmes principes que les démarches précédentes, et synthétise donc d'une manière relativement classique la façon dont sont abordées les modélisations multi-niveaux dans la communauté SMA.

2.4.2 Emergence et outils de simulation

Les idées de base qui sont contenues dans les méthodologies de modélisation multi-niveaux que nous venons de présenter sont les mêmes que l'on retrouve mis en œuvre dans bon nombre d'applications dans lesquelles les systèmes étudiés font intervenir plusieurs échelles de représentation des connaissances. Nous détaillons deux de ces outils à titre d'exemples.

Travaux de Servat : agrégation de gouttes d'eaux

Les travaux de David Servat [Servat *et al.* 1998] dans le domaine des applications de simulation dédiées à l'hydrologie (cf. figure 2.15) sont bien connus. Leur but était de parvenir à prédire le ruissellement des eaux dans un environnement complexe et il convenait de trouver des approches capables de réduire les temps de calculs afin de pouvoir avoir un domaine spatial d'applicabilité qui soit suffisamment intéressant.

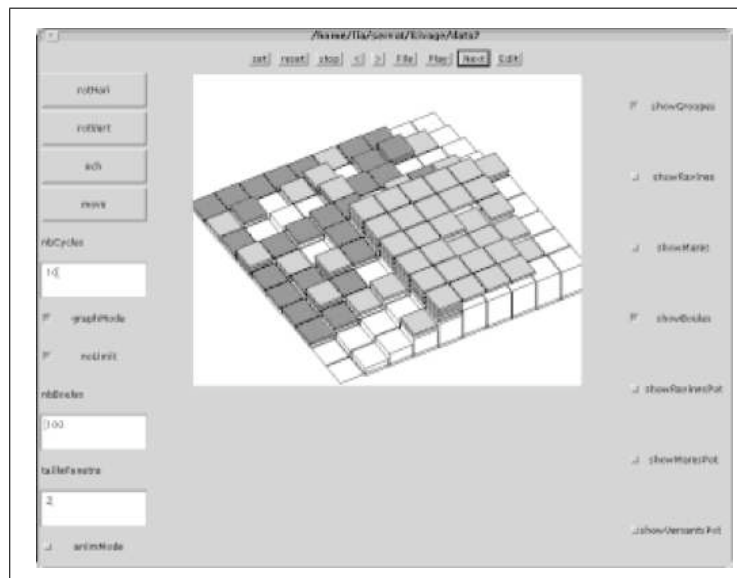


FIG. 2.15 – Modélisation multi-niveaux selon Servat (extrait de [Servat *et al.* 1998])

Dans un même système cohabitent des agents "gouttes d'eau" et des agents "flaques d'eau". A l'échelle microscopique les agents "gouttes d'eau" peuvent se grouper pour former les agents "flaques d'eau" qui apparaissent ainsi dans le système. Et dans l'autre sens à l'échelle macroscopique chaque agent "flaque d'eau" peut se scinder pour former des agents "gouttes d'eau" de niveau inférieur.

Lors de l'exécution de la simulation on peut donc observer l'émergence de structures d'eau qui sont dues à l'évolution de structures d'eau de niveau inférieur. Dans une telle approche cette émergence des structures est directement liée aux comportements qui ont été programmés dans le système, par exemple la simple relation de proximité entre plusieurs agents gouttes d'eau.

Cela n'était clairement pas le but des travaux, qui sont très centrés sur l'aspect applicatif, mais on constate que dans une telle approche peu de place est ainsi faite à la nouveauté, à la découverte de phénomènes émergents considérés comme étant de nouvelles connaissances, et donc à la prise en compte de l'existence de ces phénomènes dans le modèle. Le modèle est donc conçu pour permettre de simuler une chose bien précise – ce qu'il fait plutôt bien – mais n'est pas prévu pour être étendu ou modifié selon les connaissances nouvellement découvertes.

Travaux de Tranouez : réification de vortex

Dans ses travaux Pierrick Tranouez utilise la Simulation Orientée Agent pour simuler l'écoulement de fluides. Le but avoué des travaux est de parvenir à détecter et réifier des organisations qui pourraient apparaître lors des simulations. Comme expliqué dans [Tranouez 2005, Tranouez *et al.* 2006], le modèle de base est basé sur des particules vortex élémentaires dans un écoulement de fluides en deux dimensions, et l'intérêt est porté sur la détection d'organisations de même nature mais d'un niveau de description plus élevé.

En utilisant une méthode faisant suivre triangulation, calcul d'enveloppe convexe, puis calcul d'ellipse associée, les vortex de niveau supérieur sont détectés et sont réifiés de par la création d'agents les représentant, qui forment ainsi des vortex ayant un plus grand volume mais ayant les mêmes règles internes (par exemple le sens de rotation) que ceux du niveau inférieur. Cet exemple est représenté sur la figure 2.16 sur laquelle les points représentent les particules vortex et les ellipses représentent les structures de fluides réifiées et détectées, et sur laquelle les couleurs vert et rouge font référence aux sens de rotation des vortex.

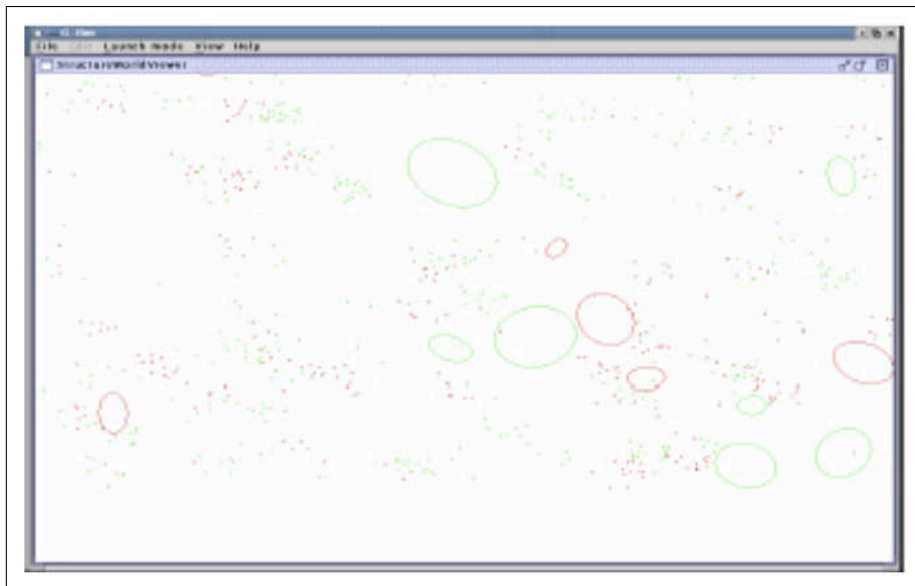


FIG. 2.16 – Détection et réification de Vortex dans l'écoulement de fluides (extrait de [Tranouez 2005])

Comme dans les travaux de Servat, ces travaux sont très centrés sur l'aspect applicatif, et dans la démarche rien ne permet de prendre en compte l'apparition de nouvelles connais-

sances autres que celles pré-supposées comme acquises.

Il est dommage qu'une démarche méthodologique de conception de simulations générique ne soit pas associée à ces travaux. En effet, on constate qu'il y a l'émergence de structures, qui sont ensuite détectées, et qui sont enfin réifiées en étant agentifiées dans le système. En ce sens ces travaux sont particulièrement intéressants, et ce d'autant plus que les règles de détection (triangulation, équations de Navier Stokes, etc.) des organisations émergentes sont relativement complexes, de mêmes que celles liées à leur réification (mécanismes de fuite et de satisfaction).

2.5 Conclusion

Dans ce chapitre, nous avons présenté un état des connaissances actuelles qui sont axées autour de la notion d'émergence. Différentes définitions et classifications ont été proposées et, ensemble, elles permettent de cerner le contexte des travaux réalisés dans la deuxième partie de cette thèse et que nous présentons dans la suite de ce manuscrit.

En particulier, cet état de l'art sur la notion d'émergence va nous permettre de définir par la suite un ensemble d'éléments ayant pour but d'améliorer sa prise en compte dans les simulations, et permettant donc *in fine* d'améliorer la façon dont sont traitées les nouvelles connaissances qui peuvent apparaître dans une simulation. On constate en effet, au vu des méthodologies de conception de simulations présentées au chapitre 1, qu'il est important de mener des travaux sur la façon de prendre en compte les phénomènes émergents dans les simulations car d'une part il existe des typologies et des méthodologies conceptuelles, et d'autre part il existe des implémentations et des applications liées à ces méthodologies, mais il n'existe pas d'approche générique conceptuelle et formelle permettant de mieux cadrer les choses.

Dans la deuxième partie de ce manuscrit nous présenterons donc un ensemble d'éléments sémantiques, conceptuels, formels et architecturaux qui permettent de tendre vers des simulations qui auront été réalisées en s'appuyant sur des éléments conceptuels, formels et architecturaux qui facilitent la prise en compte de l'émergence (au sens large) et la représentation des phénomènes émergents dans chacune des étapes de la conception d'une SOA. Ces propositions étant évidemment là dans le but de produire des modèles qui conduiront à l'émergence de nouvelles connaissances, que ce soit sous la forme de phénomènes émergents dans les systèmes simulés ou que ce soit sous la forme d'idées nouvelles dans l'esprit de leurs concepteurs et utilisateurs.

Deuxième partie

Propositions conceptuelles, méthodologiques et architecturales pour la prise en compte de l'émergence dans les SOA

Voir l'émergence comme une méta-connaissance

Sommaire

3.1	Introduction	63
3.2	L'émergence, une méta-connaissance	64
3.2.1	L'importance du contexte	64
3.2.2	La méta-connaissance	65
3.2.3	Méta-connaissance et processus d'ingénierie des connaissances	67
3.2.4	Définition	69
3.3	Prendre en compte l'émergence dans une méthodologie de conception de SOA	69
3.3.1	Vers une prise en compte de l'émergence sur le plan méthodologique	69
3.3.2	Retour sur les méthodologies de conception de SOA	70
3.3.3	Quelle est la place de l'émergence dans ces méthodologies?	71
3.3.4	Méthodologie de conception de simulations	73
3.4	Conclusion	75

3.1 Introduction

Dans la première partie de ce manuscrit, nous nous sommes intéressés de façon générale à la modélisation et à la simulation de systèmes complexes. Ce faisant, nous nous sommes focalisés sur l'émergence, car nous avons constaté que les éléments se rattachant à cette notion ne sont pas pris en compte de façon explicite dans les méthodologies de conception de simulations existantes.

Dans cette deuxième partie, nous allons donc proposer des solutions permettant de donner corps à l'émergence dans les Simulations Orientées Agent, que ce soit sur un plan méthodologique, sur un plan conceptuel, ou sur un plan architectural.

Ainsi, dans le présent chapitre, nous allons commencer par rattacher l'émergence à une notion qui apparaît dans de nombreux domaines scientifiques ayant trait à la conception de systèmes et qui est donc bien connue des chercheurs travaillant dans ce que nous nommons l'Intelligence Artificielle [Paquette 1998] : la méta-connaissance. Nous nous appuyerons ensuite sur cette notion pour faire ressortir la place qu'occupe l'émergence dans les méthodologies existantes. Et nous terminerons ce chapitre en proposant une démarche globale qui permet de mieux cadrer la façon dont les phénomènes émergents doivent être pris en compte dans une méthodologie de conception de simulations.

3.2 L'émergence, une méta-connaissance

3.2.1 L'importance du contexte

Lorsque l'on travaille sur la conception de modèles de simulations et que l'on réfléchit autour de la notion d'émergence dans les simulations (et ce quelle que soit l'approche de simulation choisie), on se rend compte que beaucoup de choses sont liées à la découverte, à l'analyse, ou à la simple compréhension des phénomènes qui ont pu se produire et dépendent donc directement d'un certain degré d'interprétation. Le contexte de l'étude et le niveau d'expertise des intervenants jouent ici un rôle prépondérant puisqu'il arrive bien souvent que certains trouvent du sens aux choses là où d'autres n'en trouvent aucun : les informaticiens qui créent les simulations n'ont (*a priori*) pas l'expertise suffisante relative aux systèmes complexes qu'ils tentent de simuler, et les thématiciens qui interviennent dans les processus de modélisation sont chacun experts de domaines spécifiques – il est donc fort logique qu'un phénomène émergent particulier puisse sauter aux yeux d'un expert géographe alors qu'il n'aura pas été remarqué par un expert économiste, ou que suite aux expériences de simulation une idée nouvelle émerge dans la tête de l'expert économiste mais pas dans celle de l'expert géographe.

Ces considérations sur l'importance du contexte d'interprétation des choses sont reprises dans de nombreux travaux ayant trait à la conception de Simulations Orientées Agent. Ainsi, sur la figure 3.1, extraite de [Dessalles & Phan 2006], on voit apparaître un ensemble de questions, toutes fondamentalement liées à la modélisation multi-niveaux, qui sont axées autour de la notion d'observation et des niveaux d'organisation, et parmi lesquelles on voit ressortir celle-ci : "For whom does this make sense?". Dans notre contexte de SOA de systèmes complexes naturels et sociaux, cette question revient à considérer l'aspect fondamental qui se niche dans les différences d'interprétation qui existent selon la nature des regards qui se posent sur le système et selon leur degré d'expertise.

Cet ensemble d'éléments nous permet d'avancer l'existence (et l'importance) d'un certain "contexte de connaissance", qui fait qu'un phénomène émergent donné non pas puisse émerger (puisque – comme nous le détaillerons par la suite – ce sera de toute façon toujours le cas dans la simulation, peu importe qu'elle soit observée ou pas), mais puisse être remarqué, appréhendé, caractérisé. Ceci nous met sur la voie du concept de méta-connaissance.

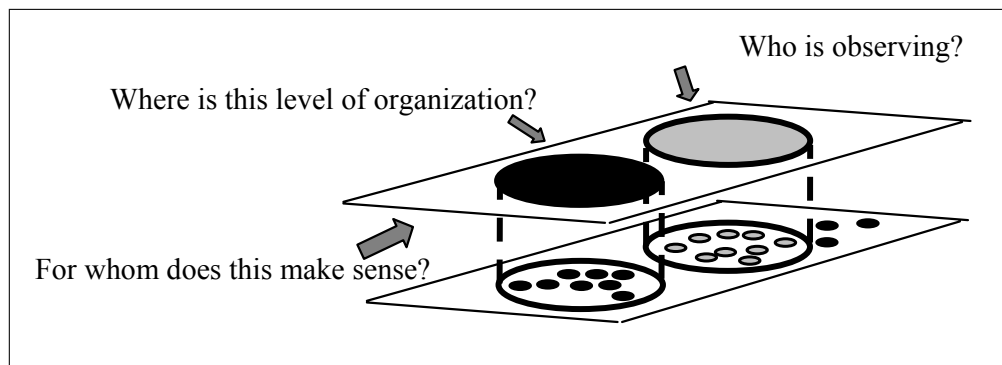


FIG. 3.1 – Les grandes questions de la modélisation multi-niveau (extrait de [Dessalles & Phan 2006])

3.2.2 La méta-connaissance

La méta-connaissance est au cœur du processus qui consiste à transformer l'information en connaissance. Ainsi vue en tant qu'outil permettant de travailler sur la connaissance, elle a été définie par Jacques Pitrat comme étant "une connaissance sur la connaissance plutôt qu'une connaissance sur un domaine en particulier comme les mathématiques, la médecine ou la géologie" [Pitrat 1990].

Définition II.1 (Méta-connaissance) *Une méta-connaissance est une connaissance sur la connaissance.*

D'une façon générale, la méta-connaissance offre une plus grande variété de comportements et une meilleure façon de s'adapter aux changements d'environnement [Luzeaux 1997]. Mais il ne faut pas perdre de vue que d'une certaine façon la méta-connaissance n'est que de la connaissance supplémentaire : la distinction tient au fait du niveau d'abstraction qui pourra être obtenu par exemple grâce au regard d'un observateur externe (qui peut être subjectif, expert d'un domaine, etc.) aux choses, aux phénomènes, au système, et au final à la partie du monde (réel ou théorique) qui est en train d'être observée et analysée.

Très large dans sa définition de base, la notion de méta-connaissance peut évidemment être affinée. On distingue ainsi entre autres quatre déclinaisons importantes :

- les méta-connaissances décrivant les connaissances ;
- les méta-connaissances sur l'utilisation des connaissances ;
- les méta-connaissances pour découvrir des connaissances ;
- les méta-connaissances pour manipuler des connaissances.

A la simple lecture de ces quatre types de méta-connaissance on remarque qu'un parallèle peut s'établir avec le concept d'émergence puisque la façon dont on le perçoit dans la communauté SMA entre parfaitement dans ce cadre : voir ainsi la méta-connaissance comme un outil permettant de découvrir, décrire, utiliser et manipuler des connaissances est en plein accord avec les concepts fondamentaux qui se dégagent autour de l'émergence, notamment celui de nouveauté radicale (découverte de connaissances) et celui d'interdépendance des

niveaux (utilisation, manipulation de connaissances) qui sont cités dans les travaux relatifs à l'émergence [Stephan 2002] et que nous avons évoqués au chapitre précédent. Il y a donc là existence d'un lien sémantique fort entre les notions d'émergence et de méta-connaissance.

Ce lien perdure au-delà des simples définitions. En effet, en listant et en classifiant de façon moins abstraite les types de méta-connaissance et leurs domaines d'applications présents dans la littérature [Kalfoglou *et al.* 2000], on distingue une séparation entre les méta-connaissances computationnelles et non-computationnelles qui sont représentées sur la figure 3.2.

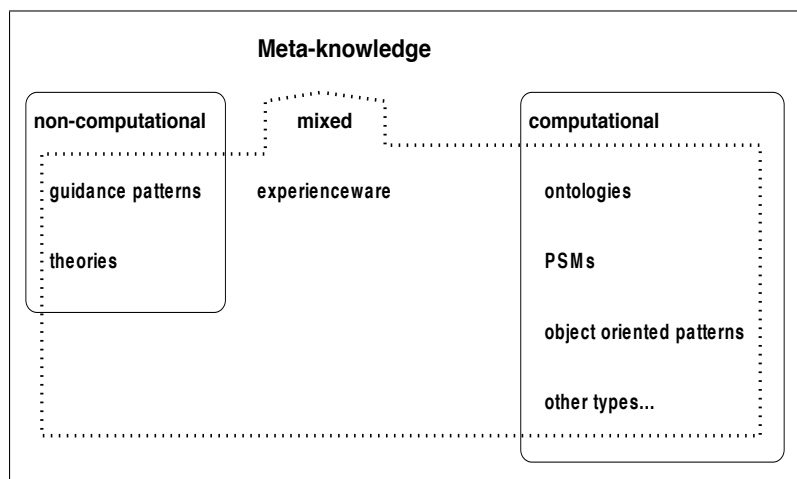


FIG. 3.2 – Quelques types de méta-connaissance listés dans la littérature (extrait de [Kalfoglou *et al.* 2000])

Ici aussi un lien fort avec la notion d'émergence se dégage : l'émergence peut être considérée comme appartenant à chacun des deux mondes, computationnel et non computationnel, et elle se retrouve aussi à la croisée d'entre eux, dans le champ de la "méta-connaissance mixte" qui fait ressortir le côté expérimental pour lequel la SOA est particulièrement appréciée. En effet, de part et d'autre de cette classification on retrouve des éléments auxquels se raccroche l'émergence dans nos systèmes complexes, à un niveau conceptuel d'une part (théories diverses, modèles de configurations orientées objet), et au niveau système informatique d'autre part (configurations orientées objet, modèles plateforme-spécifiques apparentables à nos "modèles informatiques" décrits au chapitre 1).

De plus, il est bon de noter que les théories citées en exemple sont directement reliées à la notion d'émergence présente en philosophie. C'est le cas de la méréologie, domaine d'étude bien connu des philosophes et des linguistes, que nous avons évoquée dans le chapitre précédent, et qui traite des relations entre le Tout et ses Parties. Et quand on parle du Tout et de ses Parties l'émergence n'est jamais bien loin ; citons pour rappel l'exemple "le rhum de ce punch", dans lequel le punch est plus que la somme des parties le constituant, celles-ci ne pouvant plus être dissociées dès lors que le punch a été créé (exemple dans lequel on retrouve le caractère d'irréductibilité propre à certaines formes d'émergence [Stephan 2002]).

Enfin, notons que l'on retrouve la présence de la méta-connaissance dans le cadre des systèmes experts dans lesquels les connaissances du domaine du système (qui peuvent par exemple être des connaissances médicales pour apporter une aide au diagnostic) sont généralement fournies de façon déclarative, tandis que les méta-connaissances (qui sont dans ce cas liées à la façon d'utiliser les connaissances) sont directement programmées dans le moteur d'inférences.

Cet ensemble d'éléments continue d'appuyer à nos yeux le lien qui existe entre la notion d'émergence telle qu'on la connaît dans nos simulations et celle de méta-connaissance.

3.2.3 Méta-connaissance et processus d'ingénierie des connaissances

Comme nous venons de le voir, la méta-connaissance est un concept qu'il est important de prendre en compte dès lors que l'on s'intéresse à la façon dont des connaissances sont découvertes, acquises, et manipulées au fil du temps. Elle occupe donc une place importante dans ce que nous appelons les processus d'ingénierie des connaissances, qui permettent, entre autres, de cadrer les processus de transformation de l'information en connaissance. Afin d'appuyer le lien entre émergence et méta-connaissance dont l'aspect statique ressort des éléments que nous avons avancés à la section précédente, nous avons choisi d'illustrer cette section en menant une rapide réflexion autour de deux processus d'ingénierie des connaissances dans lesquels apparaît la méta-connaissance. En ce sens cette façon de voir les choses rejoint la manière dont la "connaissance" est représentée en tant qu'élément explicite dans certaines des méthodologies de conception de simulations décrites au chapitre 1.

Dans le premier processus, inspiré de [Turban & Aronson 1997] et représenté sur la figure 3.3, la méta-connaissance apparaît sous plusieurs des formes que nous avons décrites. Elle est ainsi présente aux étapes qui concernent la validation, la vérification, et la représentation des connaissances qui auront été établies par des experts afin de faire de cette connaissance experte (auparavant inconnue) une connaissance générale (à présent admise). De façon implicite, c'est donc une fois de plus au travers du regard de l'expert d'un domaine que des éléments trouvés à l'intérieur de fichiers de résultats, de documents historiques, ou de livres prennent du sens.

Dans le second processus, inspiré de [Benchimol 2005] et représenté sur la figure 3.4, la méta-connaissance est directement présentée (et représentée) comme étant l'élément clé qui permet de faire le lien entre les connaissances existantes, actuelles, que l'on possède déjà sur un système donné, et les connaissances potentielles, que nous n'avons pas encore, mais qu'il nous reste à découvrir, à appréhender, et à comprendre. Evidemment, ceci vaut pour une personne donnée, un expert donné, ou un ensemble de gens (par exemple le grand public) dès lors que les connaissances potentielles nouvelles découvertes sous l'éclairage d'une méta-connaissance sont incorporées dans l'ensemble des connaissances existantes.

Dans ces deux processus d'ingénierie des connaissances, il ressort que la méta-connaissance est un élément de base sans lequel il est difficile, voire impossible, d'imaginer

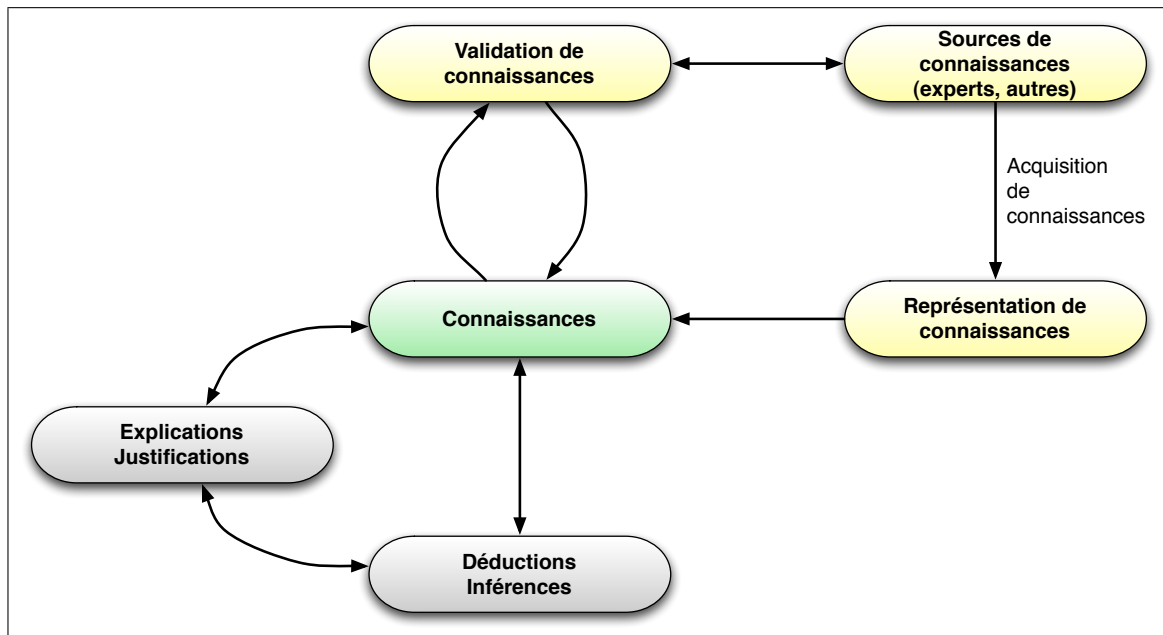


FIG. 3.3 – Processus d'ingénierie des connaissances (adapté d'après [Turban & Aronson 1997])

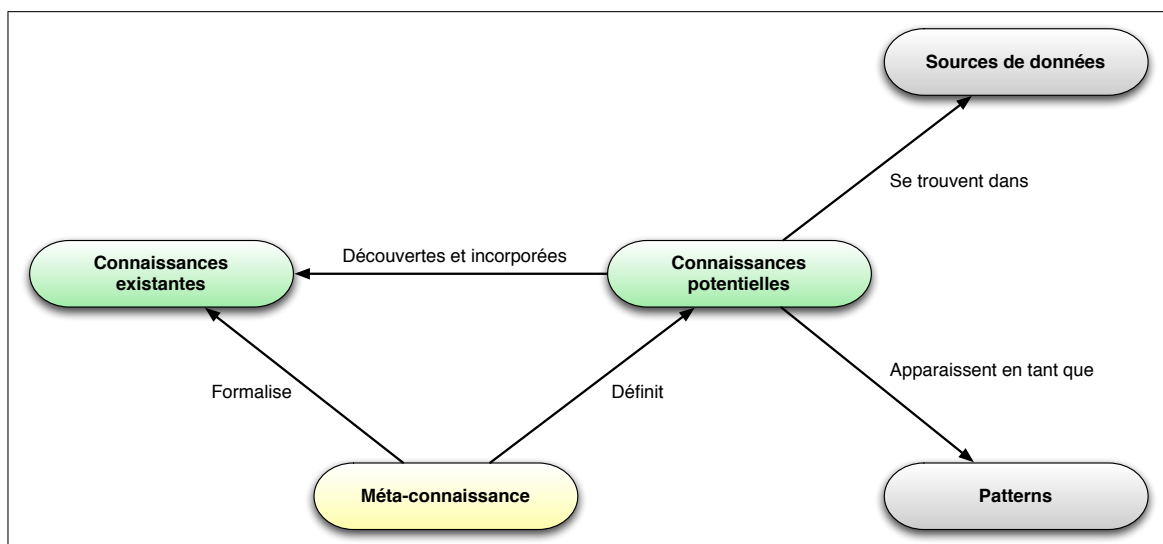


FIG. 3.4 – Metaknowledge thinking (adapté d'après [Benchimol 2005])

pouvoir enrichir un ensemble initial de connaissances avec des éléments appartenant à l'ensemble des connaissances potentiellement découvrables. Comme à la section précédente où le lien entre émergence et méta-connaissance nous a paru évident sur le plan sémantique, nous avons là un lien fort sur le plan méthodologique. En effet, dans le contexte de la conception de simulations de systèmes complexes, les phénomènes émergents qui peuvent apparaître dans un système sont justement ceux qui n'avaient pas été intégrés parmi les connaissances initiales qui étaient incluses dans le modèle du domaine du système complexe étudié. Et une démarche de modélisation et de simulation a justement pour but, à travers l'observation, et à travers l'analyse des résultats par l'ensemble des thématiciens experts, de faire ressortir de nouvelles connaissances ; nouvelles connaissances parmi lesquelles les plus intéressantes concernent bien souvent les phénomènes émergents qui sont apparus au cours de la simulation.

3.2.4 Définition

Au vu des arguments présentés précédemment, tant sur la plan sémantique que sur le plan méthodologique, nous posons la définition suivante :

Définition II.2 (Emergence) *L'émergence est une méta-connaissance.*

Cette définition a été introduite dans [David & Courdier 2008] et développée dans [David & Courdier 2009]. Elle pose de fait l'importance du contexte dans la reconnaissance et la caractérisation d'un phénomène émergent, et elle procure en outre l'avantage d'être compatible avec toutes les définitions de l'émergence qui ont été présentées dans le chapitre 2 qui était entièrement dédié à cette notion. Les différences de point de vue ne disparaissent évidemment pas, et certaines visions de l'émergence resteront toujours éloignées, mais cette définition permet de les uniformiser d'une certaine façon.

Par ailleurs, voir l'émergence comme étant une méta-connaissance nous permet d'aborder sous un nouvel angle la façon de concevoir des simulations et ouvre ainsi la porte à de nouveaux axes de méthodologie, comme nous allons le voir dans la section suivante.

3.3 Prendre en compte l'émergence dans une méthodologie de conception de SOA

3.3.1 Vers une prise en compte de l'émergence sur le plan méthodologique

Dans un contexte de modélisation ayant pour but la production d'outils de simulation, les SMA ne permettent pas spécialement d'étudier les connaissances que l'on a d'un système, mais permettent plutôt de valider (ou d'invalider), de compléter, ou de découvrir de nouvelles connaissances sur le système étudié. Ainsi ce n'est pas nos connaissances en tant que telles que nous étudions, mais le système que l'on modélise et pour lequel, précisément, nous n'avons

pas ou peu de connaissance, ou sur lequel nous n'avons donc qu'une connaissance approximative.

En quelque sorte, la simulation est un outil d'étude empirique qui, idéalement, devrait nous permettre de comprendre un système (ou du moins de "mieux" le comprendre) en enrichissant notre ensemble de connaissances initiales particulières sur celui-ci.

De façon générale, lorsque nous élaborons un modèle pour un système donné, nous entamons le processus de modélisation et de simulation en établissant une représentation grossière du système. Puis nous réalisons un ensemble d'expérimentations et, au fur et à mesure des expériences de simulation réalisées, nous élaborons de nouvelles hypothèses qui nous permettent d'affiner le modèle de départ. C'est, en résumé, ce qui ressort des méthodologies de conception de simulations que nous avons présentées dans le premier chapitre de ce manuscrit.

Au final, après parfois de nombreux cycles de modifications par rapport au modèle (ou par rapport "aux modèles" dans le cas de l'existence d'un modèle domaine et d'un modèle informatique) initialement produit, si on arrive à un stade où les expériences de simulation (qui correspondent à l'évolution des différents modèles simulés) donnent des résultats "réalistes", on peut alors supposer que la modélisation du système est satisfaisante. Ceci ne veut pas spécialement dire que la modélisation qui a été obtenue correspond en tous points à la véritable nature du système réel, mais logiquement cela permet de prouver que l'on a acquis un niveau de connaissance sur le système étudié qui est suffisamment bon relativement à ce que l'on espérait obtenir au commencement de notre démarche.

3.3.2 Retour sur les méthodologies de conception de SOA

Au regard de notre définition de l'émergence comme étant une méta-connaissance, nous pensons que cette méta-connaissance constitue justement le substrat qui est manipulé au cours de la démarche de conception de simulation et qui permettra de nous faire progresser vers le but final du processus : une modélisation "idéale" du système étudié.

Dans la première partie de ce manuscrit nous avons défini ce qu'était une méthodologie de conception de simulation, et nous avons vu qu'il existe différentes approches pour parvenir à réaliser, au bout du compte, des modèles et simulations qui soient les meilleures possibles. De ces différentes démarches nous pouvons dégager une trame commune, qui reprend de façon simplifiée les méthodologies de conception de simulations de [Ralambondrainy 2009] ou de [Drogoul *et al.* 2003].

Cette trame peut être représentée comme correspondant à un cycle (forcément fini, puisqu'il faut bien s'arrêter à un moment) d'étapes qui sont illustrées sur la figure 3.5 :

1. une phase de modélisation (réalisation d'un modèle du domaine puis sa traduction en un modèle informatique) : réalisation d'une modélisation initiale, grossière lors de la première itération du cycle, et de plus en plus pertinente par la suite.
2. une phase de simulation : réalisation d'une succession d'expériences qui à chaque étape produisent des données à analyser.

3. une phase d'analyse : confrontation des résultats produits par les expériences de simulation avec la réalité (ou avec d'autres connaissances supposées "réelles" ou supposées correctes).

Cette dernière étape permet soit d'établir de nouvelles hypothèses, soit de vérifier que les résultats sont conformes à ceux espérés ou attendus. Elle débouche donc sur l'une des étapes suivantes :

4. une phase de modifications : les nouvelles hypothèses conduisent à modifier la modélisation du système faite à la phase 1 et à reprendre le cycle d'étapes à la phase 2.
5. on arrive à un stade de modélisation où tous les résultats des simulations sont conformes et où l'on ne peut rien apprendre de plus que ce que l'on sait déjà (situation utopique). La démarche de simulation prend fin ici.

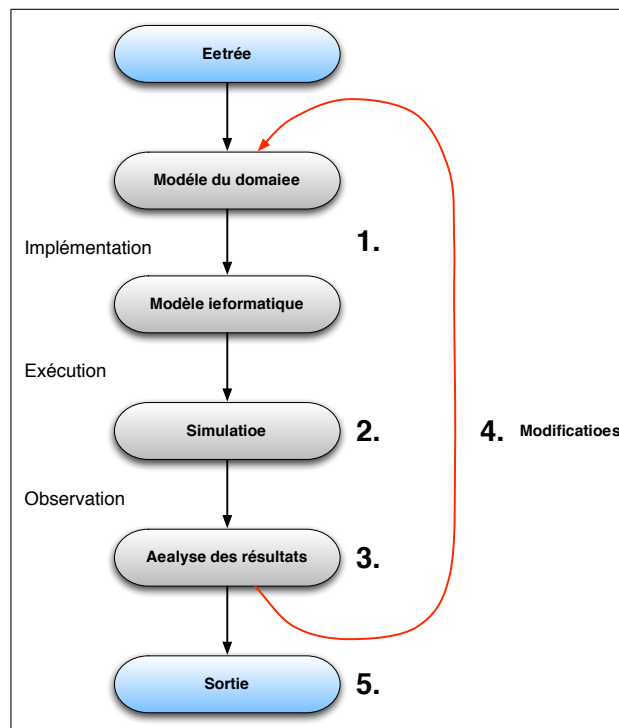


FIG. 3.5 – La trame commune des méthodologies de conception de SOA

Quelle est donc la place de l'émergence dans une telle vision des choses ? Pour donner à l'émergence une place dans cette trame méthodologique de conception de simulations il convient de raffiner cette trame afin d'y faire apparaître les couches relevant de la connaissance et de la méta-connaissance.

3.3.3 Quelle est la place de l'émergence dans ces méthodologies ?

Prendre en compte l'émergence dans une méthodologie de conception de simulations revient à déterminer la place que doit occuper cette prise en compte dans la trame méthodolo-

gique que nous venons de présenter. Cette tâche s'avère complexe puisque l'émergence semble *a priori* faire partie de la démarche globale et n'intervient donc pas forcément à une étape en particulier. Mais les choses se retrouvent facilitées si l'on se ramène aux différentes déclinaisons de la méta-connaissance que nous avons présentées précédemment.

Il est ainsi possible de rattacher l'émergence à différents points de notre trame méthodologique :

- lors de la phase de modélisation (phase 1) et surtout lors de la phase de simulation (phase 2), l'émergence apparaît sous la forme la plus naturelle puisque c'est justement là que d'éventuels phénomènes émergents peuvent... émerger du fait des interactions qui se produisent entre les entités qui sont présentes dans la simulation. L'émergence vue comme étant une méta-connaissance est ici à prendre en tant que "connaissance pour produire de la connaissance". Mais cette connaissance là, qui émerge, et qui est donc produite, ne sera pas forcément acquise par les observateurs qui doivent avoir une certaine expertise pour être capables de l'appréhender.
- lors de la phase d'observation et d'analyse (phase 3), l'émergence apparaît puisque c'est justement à cette étape que certains phénomènes émergents apparus lors de la phase précédente peuvent éventuellement être reconnus par les thématiciens ou par tout observateur extérieur au système simulé qui serait là pour apporter son expertise et ainsi tirer des conclusions des analyses sur les données produites par les expériences de simulation. L'émergence vue comme étant une méta-connaissance est ici à prendre en tant que "connaissance pour décrire de la connaissance".
- lors de la phase de modifications (phase 4), l'émergence apparaît également puisque c'est au cours de cette phase que seront pris en compte (dans le modèle du domaine et/ou dans le modèle simulé) les éléments qui auront émergé lors de la phase précédente, que ce soit d'éventuels phénomènes émergents (au sens strict) auxquels on veut faire prendre corps dans la simulation, ou "simplement" les nouvelles idées qui ont émergé dans l'esprit des experts (au vu des expériences produites et des résultats précédemment analysés) et que l'on souhaite désormais intégrer au(x) modèle(s). L'émergence vue comme étant une méta-connaissance est ici à prendre en tant que "connaissance pour produire de la connaissance".
- de la phase 1 à la phase 5 (soit l'ensemble du processus, cycles de modifications inclus), l'émergence apparaît dans sa globalité puisque nous venons de voir qu'il était possible de la rattacher (à chacun des cycles) aux phases d'analyse et de modification. L'émergence vue comme étant une méta-connaissance est ici à prendre en tant que "connaissance pour manipuler de la connaissance".

Notons bien la distinction qui existe entre la façon de voir l'émergence lors de la phase de simulation (phase 2) et lors de la phase de modifications (phase 4). Dans les deux cas l'émergence est vue comme étant une méta-connaissance à prendre en tant que "connaissance pour produire des connaissances", mais dans le premier cas cette production de connaissances est

l'œuvre du système (et on peut donc facilement imaginer que l'extérieur au système ne s'en rende pas compte... ce qui ne veut pour autant pas dire que le système lui-même s'en rende compte), alors que dans le second cas elle est l'œuvre des modificateurs (souvent les concepteurs) du système (et on peut donc facilement imaginer que les entités du système ne s'en rendent pas compte... ce qui ne veut pour autant pas dire que les concepteurs eux-mêmes s'en rendent compte).

Enfin, notons que l'aspect "méta-connaissance pour l'utilisation de la connaissance" sort du cadre de la démarche de simulation en elle-même, et concerne plutôt ici l'exploitation de cette démarche par ses utilisateurs en termes d'objectif et de finalité. Ainsi, on peut lier à l'émergence et replacer dans une trame de méthodologie de conception de simulations les quatre déclinaisons de la méta-connaissance que nous avons présentées plus tôt dans ce chapitre.

Il convient à présent de voir comment cette méta-connaissance peut s'intégrer de façon synthétique dans une méthodologie de conception de simulations.

3.3.4 Méthodologie de conception de simulations

Nos considérations sur le fait de voir l'émergence comme étant une méta-connaissance nous ont permis de mieux cerner les étapes d'une méthodologie de conception de simulations auxquelles la notion d'émergence était à considérer. Mais elles nous ont aussi permis de faire ressortir, *in fine*, une méthodologie de conception de simulation générique dans laquelle les notions de connaissance et de méta-connaissance sont à prendre au sens large, hors du cadre précis de l'émergence.

De façon classique, cette méthodologie de conception de simulations est cyclique, et les différentes étapes de ce cycle sont reprises sur la figure 3.6 sur laquelle les flèches noires symbolisent l'enchaînement des étapes lors de la conception initiale du système à simuler (c'est à dire lors de la première itération du cycle de conception), et les flèches rouges symbolisent les différents retours qui sont opérés lors de la phase de modifications qui fait suite à l'analyse des résultats (et qui permettent de passer d'une itération du cycle de conception à la suivante).

On retrouve évidemment dans notre cycle méthodologique les grandes étapes qui sont souvent communes aux méthodologies classiques, que ce soit les phases de modélisation, de simulation, et d'analyse que nous avons représentées sur la trame commune de la section précédente, ou les étapes de conception du modèle du domaine, du modèle informatique et du modèle simulé que nous avons représentés directement sur la figure.

Notre point de vue dans cette méthodologie est que la méta-connaissance se place au carrefour de l'ensemble des données qui constituent la connaissance sur le système complexe qui est étudié, et de l'ensemble de celles qui sont issues de l'analyse des résultats des simulations et qui constituent donc de la connaissance sur la version simulée de ce système complexe. C'est grâce à une certaine méta-connaissance qu'il est possible de transformer des connaissances potentielles en connaissance nouvelles, de même que c'est grâce à une certaine méta-connaissance qu'il était possible, au début de la démarche de conception, d'extraire de la connaissance, en observant le système complexe que l'on souhaite étudier, afin d'en

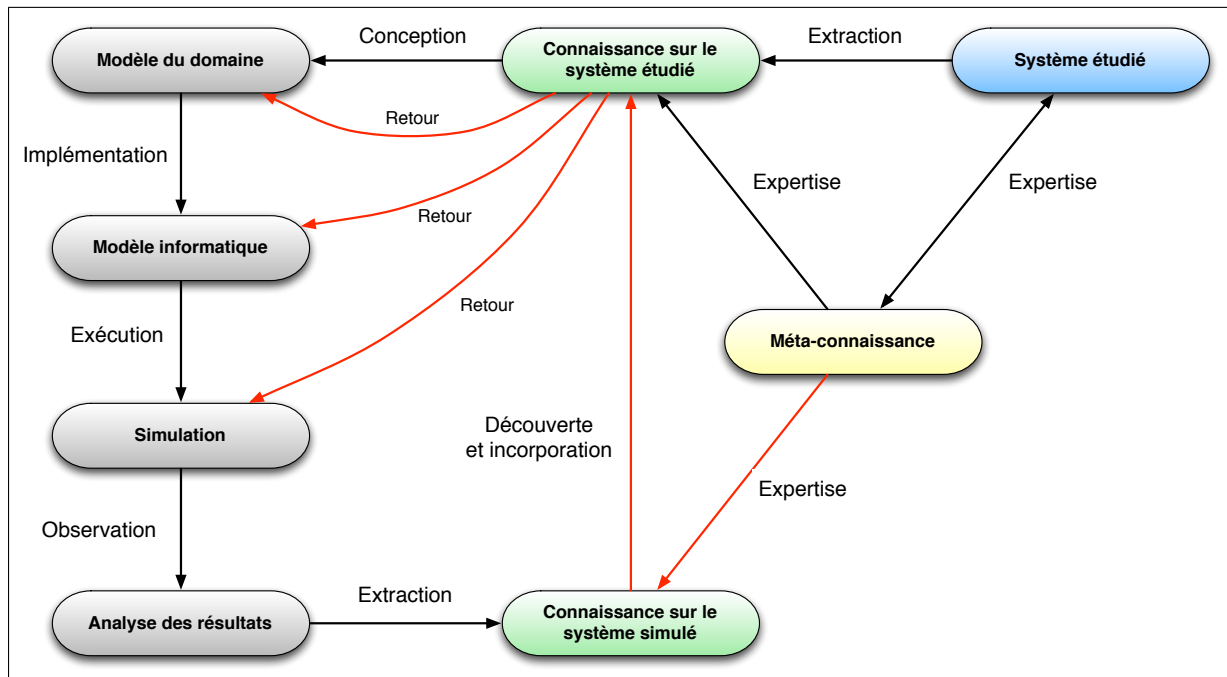


FIG. 3.6 – Connaissance et méta-connaissance dans une méthodologie de conception de SOA

commencer la modélisation. Elle joue ainsi son rôle au sein du processus de transformation de l'information en connaissance puisqu'elle permet de transformer (ou non) de la connaissance acquise sur un système virtuel simulé en connaissance supplémentaire sur le système réel qui est étudié.

Si on revient à notre problématique de la prise en compte de l'émergence, ce sont les connaissances obtenues au sortir des simulations, et les méta-connaissances qui dépendent du contexte d'analyse des résultats de ces simulations – qui diffère selon le regard (par exemple celui d'un expert de tel ou tel domaine) qui se pose sur les résultats – qui permettent de définir d'éventuels phénomènes émergents. Rappelons que la méta-connaissance concernant les phénomènes émergents est directement incluse dans la connaissance car il ne faut pas oublier que, dans chaque contexte d'observation et d'analyse, dès qu'une méta-connaissance existe elle devient par nature une connaissance.

Cette vision des choses nous permet de fondre la prise en compte de l'émergence, en tant que méta-connaissance, dans un processus de modélisation classique. Mais cela nous montre aussi que si une réflexion doit avoir lieu pour injecter des nouvelles connaissances dans le ou les modèles créés, cela doit se faire (pour et par les concepteurs de ces modèles) lors du passage de la phase d'analyse à la phase de modifications au moment où la démarche de modélisation va reboucler sur elle-même.

3.4 Conclusion

Dans ce chapitre, nous avons défini l'émergence comme étant une méta-connaissance, et nous avons montré que la voir ainsi permet d'aborder de façon plus claire la façon dont elle est présente dans les méthodologies de conception de simulations et dont elle est prise en compte dans les simulations elles-mêmes.

Nous avons donc proposé une trame méthodologique dans laquelle nous avons fait figurer la méta-connaissance comme étant un élément clé permettant de faire l'articulation entre les connaissances existantes sur un système et les nouvelles connaissances qui pouvaient être découvertes sur celui-ci. Dans cette trame méthodologique nous avons mis l'accent sur l'importance de la phase de modifications, la phase de "retour" du cycle de conception : c'est là le point au moment duquel les concepteurs doivent s'intéresser à la façon de prendre en compte l'émergence dans les systèmes qu'ils veulent modéliser et simuler. Ceci est à ne pas confondre avec l'émergence (et là nous parlons des phénomènes émergents) qui surviendra ou non en cours de simulation, ceci se passant évidemment toujours à la phase de simulation du processus que nous avons décrit. Il s'agit donc, une fois de plus, de bien distinguer ces deux points¹ :

1. l'émergence de phénomènes (souhaitée ou non, détectés ou non, ...);
2. et la capacité à détecter, identifier, caractériser ces phénomènes pour ensuite avoir la volonté (ou non) de prendre en compte leur émergence dans le système.

Le point 1, l'émergence de phénomènes, a toujours lieu lors de la phase de simulation du processus méthodologique que nous avons décrit, au sein de chaque modèle simulé qui aura été instancié et exécuté. Et si la conception et la réalisation des différents modèles a été cohérente, on peut avancer que cette émergence de phénomènes se situe aussi bien, à scénarios de simulation équivalents (si l'on peut dire ainsi), dans le modèle du domaine que dans le modèle simulé.

Le point 2, qui concerne la façon dont sont éventuellement pris en compte les phénomènes ayant émergé, a lui en revanche lieu à des moments différents selon que l'on considère que cette prise en compte est effectuée par les thématiciens et les concepteurs, qui sont des observateurs extérieurs au système, ou par le système lui même. Dans le premier cas les phénomènes émergents sont pris en compte lors des phases d'analyse et de modifications. Et dans le second cas ils sont naturellement pris en compte, dans le modèle simulé, lors de la phase de simulation. Mais pour autant ils ne font pas forcément partie du modèle du domaine, tout dépendant ici des choix qui auront été faits lors de la conception.

¹A titre de remarque il est intéressant de faire ici un parallèle entre le processus que nous avons décrit et ce qui se passe dans une SOA au cours de l'exécution du modèle simulé. En effet, on peut voir chaque cycle de notre processus comme étant ce qui se passe lors d'un pas de simulation. Nos premières étapes correspondent à ce qui se passe lorsque les agents d'une simulation sont en train de dérouler leur comportement et faire leurs différentes actions, percevant et influençant ainsi leur environnement. Et la phase de retour correspond à la mise à jour de l'environnement une fois que tous les agents du système ont terminé leur exécution, ceci pouvant se faire selon des lois d'évolution, et décidant de la façon dont seront traitées les différentes influences reçues.

Ces deux points seront développés dans la suite de cette partie, puisque nous proposerons au chapitre 4 une méthodologie décisionnelle relative au point 2 et que nous présenterons au chapitre 5 un cadre conceptuel formel et des éléments architecturaux spécifiques aux SMA relatifs aux répercussions du point 2 sur le point 1.

Méthodologie de réification des phénomènes émergents dans une SOA

Sommaire

4.1	Introduction	77
4.2	Réifier l'émergence : la détecter et la matérialiser	78
4.2.1	Détection, matérialisation, réification	79
4.2.2	Trois question complémentaires	79
4.3	Phase de détection des phénomènes émergents	80
4.3.1	Veut-on détecter un phénomène émergent ?	80
4.3.2	Comment est déclenchée cette détection ?	80
4.3.3	A quel niveau doit se faire cette détection ?	82
4.3.4	Bilan et analyse	83
4.4	Phase de matérialisation des phénomènes émergents	85
4.4.1	Le phénomène émergent doit-il être matérialisé ?	85
4.4.2	A quel niveau doit se faire cette matérialisation éventuelle ?	86
4.4.3	Comment est déclenchée cette matérialisation éventuelle ?	87
4.4.4	Bilan et analyse	89
4.5	Conclusion	91

4.1 Introduction

Dans la première partie de ce manuscrit nous avons identifié un manque au niveau méthodologique relatif à la prise en compte de l'émergence, une notion qui apparaît souvent dès que l'on touche à des modèles répondant à des problématiques spatiales et dans lesquels il est courant de voir coexister plusieurs niveaux de descriptions et d'entités. Afin de combler ce

manque, nous avons donc défini au chapitre précédent l'émergence comme étant une méta-connaissance et nous nous sommes appuyés sur cette définition pour proposer une méthodologie de conception de simulations dans laquelle la méta-connaissance en général, et donc l'émergence en particulier, sont représentées explicitement. Dans cette méthodologie, nous avons mis en avant l'importance de différentes phases, et nous avons souligné la nécessité, pour les concepteurs des modèles, de bien distinguer d'une part l'émergence de phénomènes et d'autre part la capacité à détecter, caractériser et manipuler ces mêmes phénomènes afin de décider ensuite de la façon dont leur émergence sera prise en compte.

Dans ce chapitre nous proposons donc un ajout à la méthodologie de conception de simulations décrite précédemment qui permet de cadrer de façon spécifique les choix à opérer qui permettent de distinguer différentes façons de prendre en compte les phénomènes qui peuvent émerger dans une Simulation Orientée Agent. Pour cela nous nous attardons sur le concept de réification des phénomènes émergents, qui est un processus que nous définirons comme étant la succession de deux phases : une phase de détection et une phase de matérialisation.

L'intérêt et la façon de réifier ou non un phénomène émergent sont évidemment discutés en tenant compte d'un ensemble d'avantages et d'inconvénients qui dépendent directement des contextes de modélisation et d'utilisation du système.

De façon transversale, cette méthodologie de prise en compte de l'émergence débouche sur une classification des différents types de réifications possibles des phénomènes émergents dans une Simulation Orientée Agent. Pour chacun de ces types de réifications possibles des phénomènes émergents, nous décrivons les changements qu'impose dans le système une réification éventuelle, en distinguant si ces changements doivent être effectués de façon endogène (par des fonctionnalités du système ou par certains agents eux-mêmes) ou s'ils doivent être effectués de façon exogène (par le thématicien-modélisateur ou l'utilisateur). Nous montrons aussi que ces changements peuvent être vus comme étant des modifications effectuées sur le modèle du domaine ou le modèle informatique, dont ils sont alors partie intégrante, ou simplement sur le modèle simulé.

Enfin, en terme d'usage, nous montrons que les types de réifications de l'émergence que nous avons identifiés se placent à des niveaux différents en terme de niveau d'Intelligence Artificielle du système informatique et des entités qui le composent.

4.2 Réifier l'émergence : la détecter et la matérialiser

Dans le cadre d'une démarche globale de conception de simulations, les questionnements et propositions qui vont suivre se positionnent clairement comme étant ceux qui doivent avoir lieu, pour les concepteurs d'un outil de simulation, lors du point clé identifié au chapitre précédent qui correspond à la phase de retour du cycle méthodologique et qui se situe donc à la phase de modifications de la méthodologie de conception de simulations que nous avons décrite au chapitre précédent. C'est en effet à ce stade de la démarche de modélisation, au sortir de la phase d'analyse, que se posent les questions relatives à la prise en compte de l'émergence

dans une démarche de modélisation.

4.2.1 Détection, matérialisation, réification

Nous considérons que la prise en compte de l'émergence se fait au travers de deux étapes dissociables, la détection des phénomènes émergents qui sont apparus au cours des simulations, et leur matérialisation éventuelle au sein du système étudié et modélisé. Pour un phénomène émergent donné, lorsque ces deux étapes sont effectuées, elles forment le processus complet de sa réification. Nous posons donc les définitions suivantes :

Définition II.3 (Phase de détection) *La phase de détection est un processus qui consiste à détecter les phénomènes émergents qui peuvent apparaître pendant l'exécution d'une simulation via la création et l'expression de connaissance et de méta-connaissance sur cette simulation.*

Définition II.4 (Phase de matérialisation) *La phase de matérialisation est un processus qui consiste à injecter dans une simulation des phénomènes émergents qui auront été détectés pendant la phase de détection.*

Ces deux définitions nous permettent de proposer une définition de la réification, un concept souvent évoqué dans les travaux liés de près ou de loin à l'émergence [Hofstadter 1985, Drogoul 1993, Tranouez 2005], mais qui est rarement définie.

Définition II.5 (Réification) *La réification est un processus qui se fait au travers de deux étapes dissociables, la détection des phénomènes émergents qui sont apparus dans les simulations au cours de la phase de détection, et leur matérialisation au sein du système étudié et modélisé au cours de la phase de matérialisation.*

Les stratégies de prise en compte de l'émergence dans une SOA sont directement liées à chacune de ces deux phases, la détection des phénomènes émergents, suivie de leur matérialisation éventuelle. Il faut donc réussir à décomposer chacune de ces phases afin de déterminer les étapes qui les composent et les actions à effectuer pour les réaliser.

4.2.2 Trois questions complémentaires

Nous pensons que cette réflexion, de laquelle ressort un schéma global qui pourra servir de guide lors de la conception de SOA dans lesquelles l'émergence doit être prise en compte, peut être menée en se posant, en tant que concepteurs, trois questions majeures pour chacune des deux phases.

Pour chaque phénomène émergent potentiel, les trois questions qui se posent lors de la phase de détection sont les suivantes :

- le phénomène émergent doit-il être détecté ?
- comment est déclenchée cette détection éventuelle ?
- à quel niveau doit se faire cette détection éventuelle ?

Se poser ces trois questions et essayer d’y répondre permet d’établir une stratégie d’ensemble de détection des phénomènes émergents dans un processus de simulation.

Ensuite, lors de la phase de matérialisation, les trois questions qui se posent pour chacun des phénomènes émergents ayant été détecté lors de la phase de détection sont les suivantes :

- le phénomène émergent doit-il être matérialisé ?
- à quel niveau doit se faire cette matérialisation éventuelle ?
- comment est déclenchée cette matérialisation éventuelle ?

Se poser ces trois questions et essayer d’y répondre permet d’établir une stratégie d’ensemble de matérialisation des phénomènes émergents dans un processus de simulation.

Dans la suite de ce chapitre nous allons donc discuter de ces deux phases et des questionnements qui leur sont liés. Nous listerons ainsi un ensemble de choix qui conduisent à différentes configurations possibles qui, au bout du compte, aboutissent à une classification de ce que nous appellerons les *types d’émergences réifiables* et qui permettent de définir des stratégies de réification des phénomènes émergents dans un processus de simulation.

4.3 Phase de détection des phénomènes émergents

Les trois questions qui se posent lors de la phase de détection des phénomènes émergents peuvent paraître évidentes, mais elles ont chacune leur importance.

4.3.1 Veut-on détecter un phénomène émergent ?

La première des questions à se poser est évidemment de savoir si l’on souhaite détecter les phénomènes émergents qui sont éventuellement apparus lors des simulations. La réponse à cette question dépend directement des choix des experts thématiques, car ce sont eux les experts du modèle, et dans la plupart des cas ce sont aussi eux qui seront ses utilisateurs principaux. Dans l’absolu, on peut sans mal imaginer que la réponse qu’ils apporteraient à cette question sera toujours affirmative, car les utilisateurs experts n’ont rien à perdre à espérer la détection de phénomènes émergents. Mais les ressources (qu’elles soient humaines ou logicielles) ne sont pas extensibles à l’infini, et il faut bien à un moment mettre fin à la démarche globale de modélisation et de conception de simulation si l’on estime que les résultats obtenus sont suffisamment cohérents.

4.3.2 Comment est déclenchée cette détection ?

Cette deuxième question est celle qui permettra de déterminer la façon concrète dont la détection sera réalisée. Trois options sont possibles :

- une détection manuelle ;
- une détection mécanique ;

- une détection automatique.

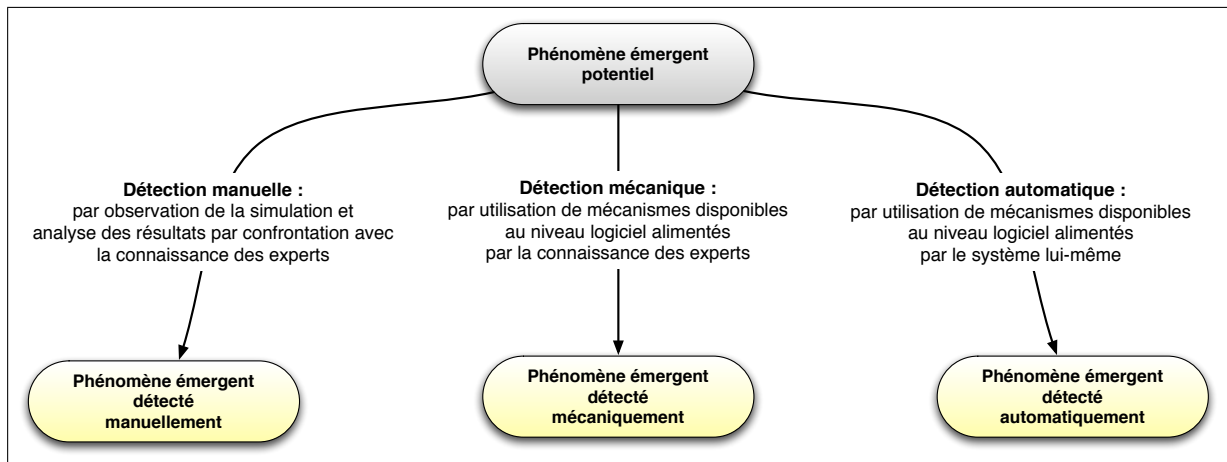


FIG. 4.1 – Comment est déclenchée la détection d'un phénomène émergent

Par rapport au modèle, la détection d'un phénomène émergent est alors effectuée de façon exogène dans le premier cas, et de façon endogène dans les deux cas suivants. Nous allons détailler chacune de ces possibilités que nous avons représentées sur la figure 4.1.

Détection manuelle

La technique de détection dite "manuelle" est la plus évidente, et aussi la plus facile à mettre en œuvre. C'est la détection traditionnelle *via* l'observation par le ou les thématiciens qui avec leurs yeux d'experts sont capables, chacun dans son propre contexte d'expertise, de reconnaître telle forme, tel regroupement, tel pattern, tel enchaînement de communications, telle suite logique d'instructions, etc. Les concepteurs doivent donc se tourner vers les bons experts qui apportent chacun leur connaissance particulière sur les systèmes étudiés et qui ont donc un niveau de méta-connaissance suffisant sur les systèmes simulés pour pouvoir en extraire de la nouvelle connaissance.

Détection mécanique

La technique de détection dite "mécanique" repose sur l'utilisation, par les thématiciens qui souhaitent détecter des phénomènes émergents, de mécanismes disponibles au niveau logiciel, par exemple au niveau de la plateforme de simulation sur laquelle le modèle informatique est implémenté. De tels mécanismes, tels ceux décrits formellement dans [David & Courdier 2008] et dans [David & Courdier 2009] qui sont présentés au chapitre 5, peuvent permettre de détecter directement les phénomènes qui émergent dans une simulation en se basant sur l'existence de révélateurs d'émergence et de lois d'émergence fournies par les thématiciens.

Détection automatique

La technique de détection dite "automatique" est la technique de détection des phénomènes émergents la plus évoluée. Mais elle est aussi la plus difficile à mettre en œuvre, car l'idée est ici d'avoir une détection complètement automatisée, sans que les thématiciens experts du modèle aient à fournir d'informations sur les éventuels phénomènes émergents recherchés. Les éléments nécessaires à la mise en place d'une telle détection automatique peuvent actuellement être mis en œuvre dans des cas simples, voire simplistes, mais il serait aujourd'hui utopique de penser pouvoir les généraliser. En effet, il paraît faisable de mettre en place des algorithmes d'apprentissage, de reconnaissance des formes, permettant par exemple de détecter certaines formes émergentes dans le Jeu de la vie de Conway (oscillateurs, glisseurs) [Berlekamp *et al.* 2002] sans fournir au modèle une source d'informations exogène. Mais il est également possible d'imaginer une telle détection automatique pour des phénomènes émergents un peu plus complexes. On rejoint d'ailleurs ici l'ensemble des questionnements liés à la présence d'une émergence causale au sens de Boschetti que nous avons évoquée au chapitre 2.

4.3.3 A quel niveau doit se faire cette détection ?

Cette dernière question fait référence au niveau auquel la détection sera effectuée, et, proche de la notion de global/local, on peut penser que ceci pourra être fait soit au niveau du système, soit au niveau des agents (du modèle ?). Nous allons détailler ces deux possibilités que nous avons représentées sur la figure 4.2.

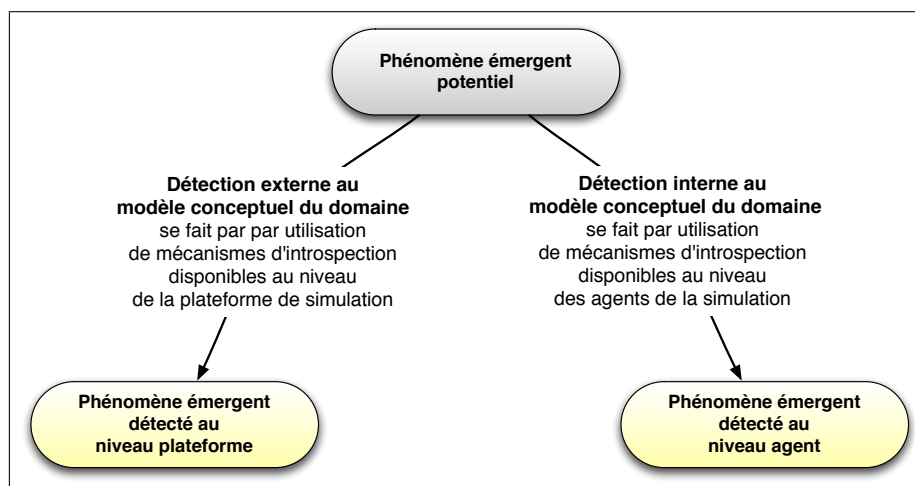


FIG. 4.2 – A quel niveau est déclenchée la détection d'un phénomène émergent

Niveau système

La technique de détection niveau système est la plus facile à mettre en œuvre. Elle suppose une vision globale du modèle, et peut être liée à l'utilisation de mécanismes plateforme tels ceux décrits précédemment. On a là une vision centrale du modèle, un "œil qui voit tout", qui

permet d'avoir une connaissance illimitée sur ce qui se passe. C'est comme si le système dans sa globalité faisait sa propre introspection pour découvrir les phénomènes qui émergent dans le modèle qu'il simule.

La détection des éventuels phénomènes émergents se fait ici de façon indépendante relativement au comportement des agents qui constituent le modèle informatique correspondant au modèle du domaine conçu avec les thématiciens. En ce sens cette technique de détection des phénomènes émergents est considérée comme étant extérieure au modèle conceptuel du domaine.

Niveau agent

La technique de détection niveau agent présuppose quant à elle que la détection des phénomènes émergents relève des agents eux-mêmes. Dans cette technique de détection on considère ainsi que les agents du système sont dotés de facultés de perception, de déduction, ou de réflexion qui leur sont propres et qui leur permettent donc de faire leur propre introspection. Ceci leur permet, dans la mesure de leurs connaissances, de se rendre compte qu'un phénomène émergent donné a eu lieu dans l'environnement dont ils ont la perception ou si ils ont eux-mêmes pris part à l'émergence d'un phénomène.

La détection des éventuels phénomènes émergents étant directement liée aux capacités des entités constituant le modèle, cette technique de détection des phénomènes émergents est donc considérée comme étant interne au modèle conceptuel du domaine.

4.3.4 Bilan et analyse

Les différentes étapes de la phase de détection des phénomènes émergents sont regroupées sur la figure 4.3. Ces étapes, et les questions et réponses qui y sont rattachées, permettent de dresser un arbre dont les feuilles constituent l'ensemble des types de détection possibles pour un phénomène émergent donné. Une flèche est représentée en dessous des différentes feuilles de l'arbre décisionnel ainsi créé : elle montre la progression de gauche à droite en terme d'Intelligence Artificielle en place dans le système. Suivant ce critère nous considérons logiquement que la technique de détection manuelle se situe en deçà des techniques de détection mécanique et automatique, car la connaissance et l'intelligence sont là dans l'esprit des observateurs du système et non pas dans le système lui-même. Et, faisant partie de la communauté SMA, et donc partisans de l'Intelligence Artificielle Distribuée, nous considérons que les techniques de détection internes aux modèles, qui mettent en œuvre des mécanismes au niveau des agents du système, sont – potentiellement – plus performantes que celles qui se situent au niveau système.

Il est possible de mettre en œuvre la plupart des stratégies mises en avant sur cette figure,

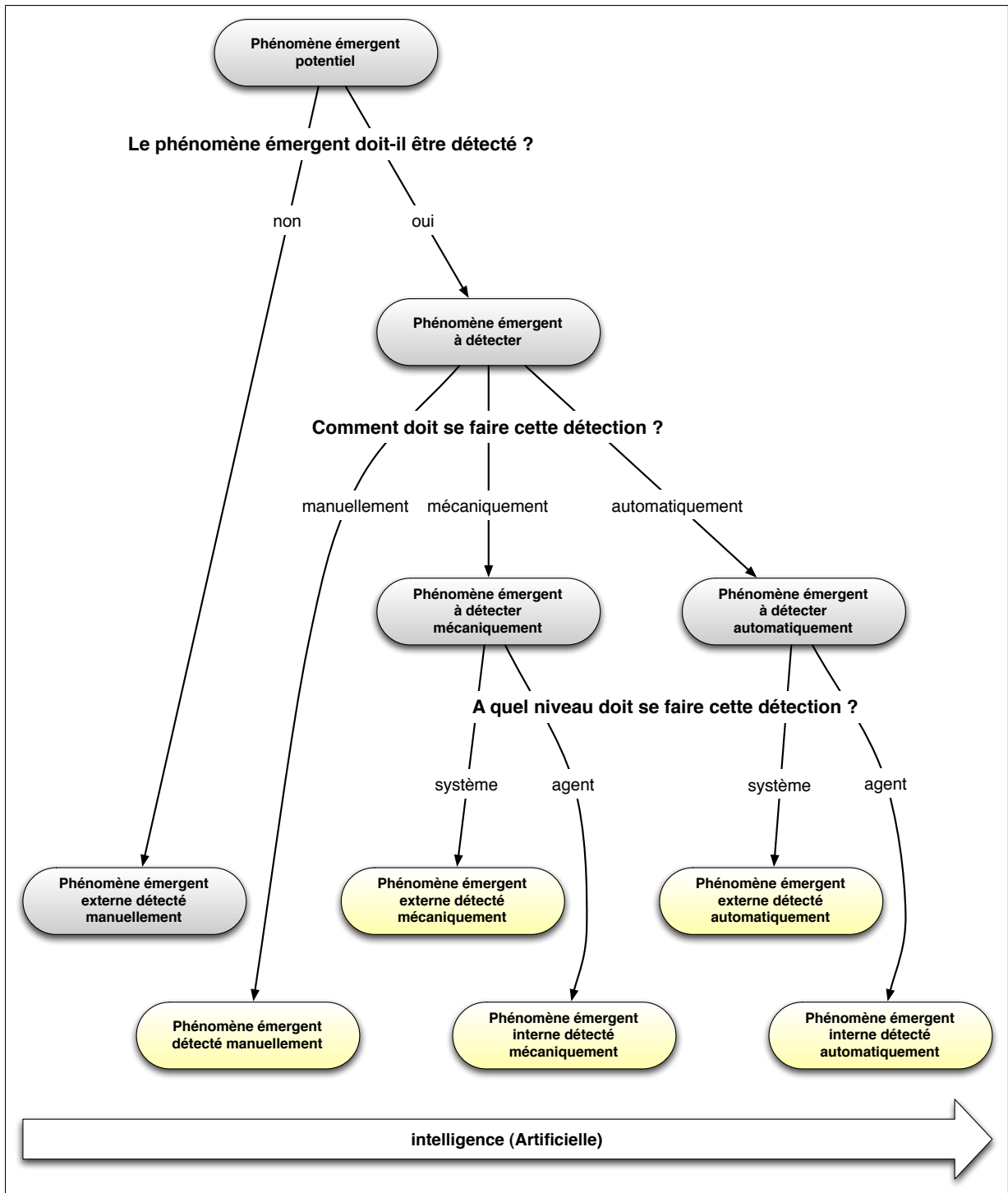


FIG. 4.3 – Les différentes stratégies de détection d'un phénomène émergent

à l'exception éventuelle de celles relevant d'une détection purement automatique de phénomènes émergents s'exprimant de manière très complexes (par exemple comment peut-on imaginer caractériser et donc détecter la conscience humaine?). En effet, et nous y reviendrons au prochain chapitre, nous ne devons pas oublier qu'il est souvent très difficile de décrire les causes et effets d'un phénomène émergent, mais qu'il est souvent possible d'observer un ensemble d'éléments qui suffisent à caractériser son apparition. Ceci étant à la fois valable pour un phénomène émergent détectable de façon logicielle ou humaine, et c'est par exemple ce qui se produit lorsque un scientifique regarde une simulation et d'un coup s'arrête, interloqué, en ayant l'intuition que "tiens, là on dirait qu'il se passe quelque chose d'intéressant".

Enfin, précisons que les techniques de détections mécanique et automatique, essentiellement lorsqu'elles ne sont pas mises en œuvre en tant qu'éléments internes au modèle conceptuel du domaine, peuvent être vues comme étant des aides pour les utilisateurs des outils de simulations développés. En effet, elles peuvent aboutir à la détection par le système (par exemple grâce à des mécanismes standards d'apprentissage ou de détection de patterns) de phénomènes qui seront alors mis en avant. Les thématiciens pourront alors prendre conscience de l'existence de ces phénomènes pour les confronter à leurs connaissances d'experts afin de déterminer s'ils présentent un intérêt notable.

4.4 Phase de matérialisation des phénomènes émergents

Comme pour la phase de détection, les trois questions qui se posent lors de la phase de matérialisation des phénomènes émergents peuvent paraître évidentes, mais elles ont chacune leur importance, et doivent se poser pour chacun des phénomènes émergents qui ont été détectés lors de la phase précédente. En raison de la dualité des étapes permettant de réaliser les phases de détection et de matérialisation, certains passages pourront paraître redondants relativement à la section précédente, mais nous avons préféré offrir ici une description complète du processus de réification.

4.4.1 Le phénomène émergent doit-il être matérialisé ?

La première question qui se pose est de savoir si un phénomène émergent détecté donné doit être matérialisé afin de compléter son processus de réification. La réponse à cette question découle naturellement d'une chose : l'utilité de cette réification.

La réponse à la question dépend évidemment des thématiciens puisque, étant les experts des domaines concernés par le modèle, ils sont ceux qui possèdent les connaissances nécessaires à sa (bonne) conception et à sa (bonne) compréhension, et ils sont par conséquent ceux qui sont les plus habilités à porter un tel jugement. Ce premier questionnement se situe donc au niveau des thématiciens, et ce même si un regard extérieur peut toujours leur permettre de confronter le modèle qu'ils ont co-construit à de nouvelles idées.

Les réponses possibles sont évidentes ("oui" ou "non"), mais le choix ne l'est pas pour autant. En effet, s'il était clair lors de la phase de détection qu'un thématicien avait tout à gagner

à tenter de détecter des phénomènes émergents dans son modèle, il n'a pas forcément intérêt à voir ceux-ci y être réifiés. Le cas le plus simple l'illustrant est celui dans lequel le modèle dans son état actuel a d'ores et déjà permis de répondre aux questions que se posaient les thématiciens au début de la démarche de modélisation. On est ici à la fin du processus de conception de simulations que nous avons présenté au chapitre précédent : les résultats du modèle obtenus grâce aux simulations sont conformes aux attentes, et il n'est pas nécessaire d'entreprendre de nouvelles modifications.

Les phénomènes émergents qui auront été détectés mais qui n'auront pas besoin d'être matérialisés – et par là même réifiés – entrent dans le cadre classique de l'émergence vue comme étant un résultat.

Mais évidemment, même si les résultats sont conformes aux attentes et que les différents protagonistes sont satisfaits du modèle obtenu, il se peut que l'on trouve quand même un intérêt à réifier des phénomènes émergents, puisque les modèles simulés obtenus *in fine*, tout satisfaisants qu'ils puissent être, peuvent jouer pleinement leur rôle de découvreurs de nouvelles questions et de producteurs de nouvelles connaissances, et peuvent donc avoir engendré des phénomènes qui soulèvent de nouveaux questionnements et dont l'étude est donc potentiellement intéressante. De plus, il se peut qu'il existe des phénomènes émergents dont la matérialisation, et donc la réification dans la SOA, n'apporte que peu de choses en terme de connaissances nouvelles sur le système étudié. Mais sur le plan strictement informatique leur réification peut prendre du sens puisque sous certaines conditions elle peut aboutir de façon indirecte à une amélioration significative des temps de calcul (on peut ainsi raisonnablement imaginer qu'il est plus rapide, dans l'absolu, de simuler l'évolution d'un seul agent "banc de poisson" plutôt que celle des centaines de milliers d'agents "poissons" en interaction le constituant).

4.4.2 A quel niveau doit se faire cette matérialisation éventuelle ?

Cette première question fait référence au niveau auquel la matérialisation sera effectuée, et, comme pour la question duale que nous avons traitée dans la phase de détection précédente, on peut penser que cette matérialisation peut être effectuée soit au niveau du système (et sera donc extérieure au modèle conceptuel du domaine), soit au niveau des agents (et donc d'une certaine façon comprise dans le modèle). Nous allons détailler ces deux possibilités que nous avons représentées de façon synthétique sur la figure 4.4.

Niveau système

La technique de matérialisation au niveau système est la plus facile à mettre en œuvre. Elle suppose évidemment une vision globale du modèle, et peut être liée à l'utilisation de mécanismes plateforme tels ceux décrits précédemment.

La matérialisation des éventuels phénomènes émergents se fait ici de façon indépendante

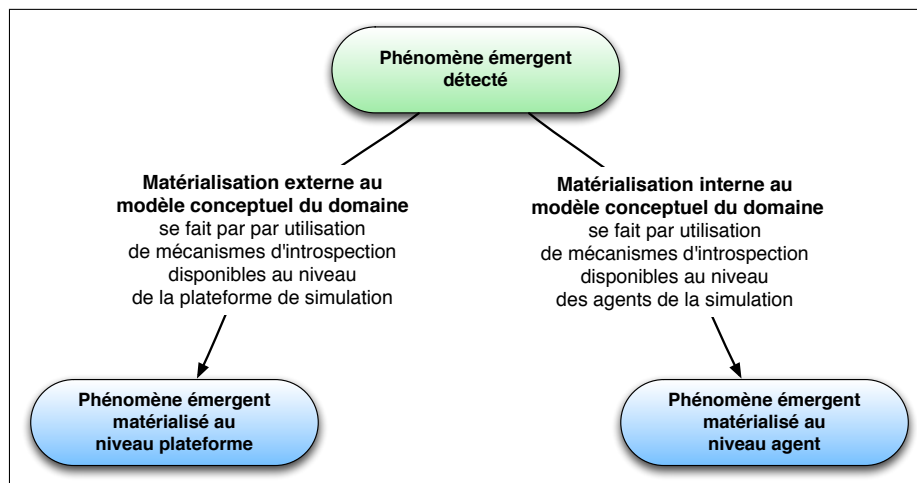


FIG. 4.4 – A quel niveau est déclenchée la matérialisation d'un phénomène émergent

relativement au comportement des agents qui constituent le modèle informatique correspondant au modèle du domaine conçu avec les thématiciens. De nouveaux agents, de nouveaux éléments d'environnement, ou même des nouveaux environnements complets, peuvent ainsi faire leur apparition. En ce sens cette technique de matérialisation des phénomènes émergents est considérée comme étant extérieure au modèle conceptuel du domaine puisqu'aucune des modifications ne l'impacte directement.

Niveau agent

La technique de réification au niveau agent présuppose quant à elle que les agents du modèle initial vont modifier leur comportement (ou vont voir leur comportement être modifié) pour qu'ils puissent intégrer et prendre en compte dans leur comportement le fait qu'ils appartiennent à tel ou tel phénomène qu'ils ont contribué à faire émerger, ou qu'ils ont pris conscience de l'émergence de phénomènes qui leur sont extérieurs.

Cette technique de matérialisation des phénomènes émergents étant directement liée aux modifications des entités constituant le modèle, elle est donc considérée comme étant interne au modèle conceptuel du domaine.

4.4.3 Comment est déclenchée cette matérialisation éventuelle ?

Cette dernière question est celle qui permettra de déterminer la façon concrète dont sera déclenchée la matérialisation d'un phénomène émergent. Comme pour la phase de détection, trois options sont possibles :

- une matérialisation manuelle ;
- une matérialisation mécanique ;
- une matérialisation automatique.

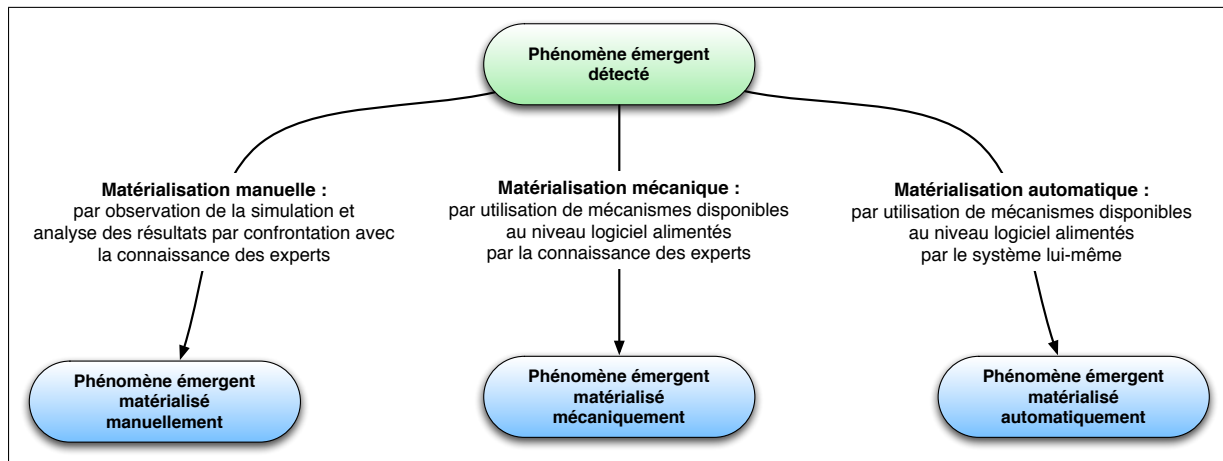


FIG. 4.5 – Comment est déclenchée la matérialisation d'un phénomène émergent

Nous allons détailler chacune de ces possibilités que nous avons représentées de façon synthétique sur la figure 4.5.

Matérialisation manuelle

La technique de réification dite "manuelle" est celle la plus évidente, et, *a priori*, la plus facile à mettre en œuvre. Elle correspond à la matérialisation traditionnelle *via* la modification du code du modèle initial par les informaticiens pour intégrer les phénomènes décrits par les experts thématiques. Cette modification du code peut être effectuée pour modifier les entités du système et leurs comportements, modifiant ainsi le modèle. Mais, dans une démarche expérimentale, elle pourra être effectuée au niveau de mécanismes logiciels externes au modèle, par exemple si au vu des résultats de simulation les informaticiens décident de proposer de nouveaux services dans la plateforme de simulation qu'ils développent. Les matérialisations manuelles qui sont effectuées de façon spécifique relativement à un type de phénomène émergent donné (par exemple les agrégations de gouttes d'eau dans les travaux de Servat et les vortex dans les travaux de Tranouez dont nous avons parlé au chapitre 2), mais dont la mise en œuvre peut entraîner la genèse d'idées qui engendreront la mise en place de mécanismes de matérialisation mécaniques ou automatiques plus génériques.

Matérialisation mécanique

Comme la technique de détection du même nom, la technique de matérialisation dite "mécanique" repose sur l'utilisation, par les thématiques qui souhaitent détecter des phénomènes émergents, de mécanismes disponibles au niveau logiciel, par exemple au niveau de la plateforme de simulation sur laquelle le modèle informatique est implémenté. De tels mécanismes, tels ceux décrits formellement dans [David & Courdier 2008] et dans [David & Courdier 2009] qui sont présentés au chapitre 5, peuvent permettre de matérialiser

directement les phénomènes qui émergent dans une simulation en se basant sur l'existence de révélateurs d'émergence et de lois d'émergence fournies par les thématiciens.

La matérialisation mécanique s'avère toutefois bien plus complexe que la détection mécanique. En effet, comme nous l'avons évoqué plus haut, il est souvent possible d'observer un ensemble d'éléments qui suffisent à caractériser l'apparition d'un phénomène émergent, et qui facilitent donc sa détection, alors que les causes qui ont conduit à l'émergence de ce phénomène, et plus encore les effets résultant de son émergence restent particulièrement difficiles à cerner.

Matérialisation automatique

La technique de matérialisation dite "automatique" est la technique de réification des phénomènes émergents la plus évoluée. Mais elle est aussi la plus difficile à mettre en œuvre, car l'idée est ici d'avoir une réification complètement automatisée, sans que les thématiciens experts du modèle aient à fournir d'informations sur les éventuels phénomènes émergents.

Tout comme pour la détection automatique de phénomènes émergents, les éléments nécessaires à la mise en place d'une telle matérialisation automatique peuvent actuellement être mis en œuvre dans des cas simples, voire simplistes, mais il serait aujourd'hui utopique de penser pouvoir les généraliser. En effet, si l'on reprend l'exemple du Jeu de la vie, on peut aller jusqu'à imaginer qu'un système ou des agents parviennent à compléter par eux-mêmes le processus de réification d'une structure stable ou même d'un des classiques oscillateurs qui y apparaissent, sans pour autant disposer d'une source d'informations exogène. Mais il apparaît impossible d'imaginer une telle réification automatique pour des phénomènes émergents à peine plus complexes.

4.4.4 Bilan et analyse

Les différentes étapes de la phase de matérialisation des phénomènes émergents sont regroupées sur la figure 4.6. Ces étapes, et les questions et réponses correspondantes, permettent de dresser un arbre dont les feuilles constituent l'ensemble des types de réifications possibles pour un phénomène émergent donné. En raison de la dualité des questions qui interviennent lors des phases de détection et de matérialisation, on remarquera que cet arbre présente une structure quasiment isomorphe à celui présenté à la fin de la section précédente. Ceci tend à montrer un caractère générique qui se dégage de la démarche de détection/matérialisation présentée ici. Là aussi, une flèche est représentée sous les différentes feuilles et montre la progression de gauche à droite en terme d'Intelligence Artificielle en place dans le système. Comme pour la phase de détection nous considérons logiquement suivant ce critère que la technique de matérialisation manuelle se situe en-deçà des techniques de matérialisation mécanique et automatique, car la connaissance et l'intelligence sont là dans l'esprit des obser-

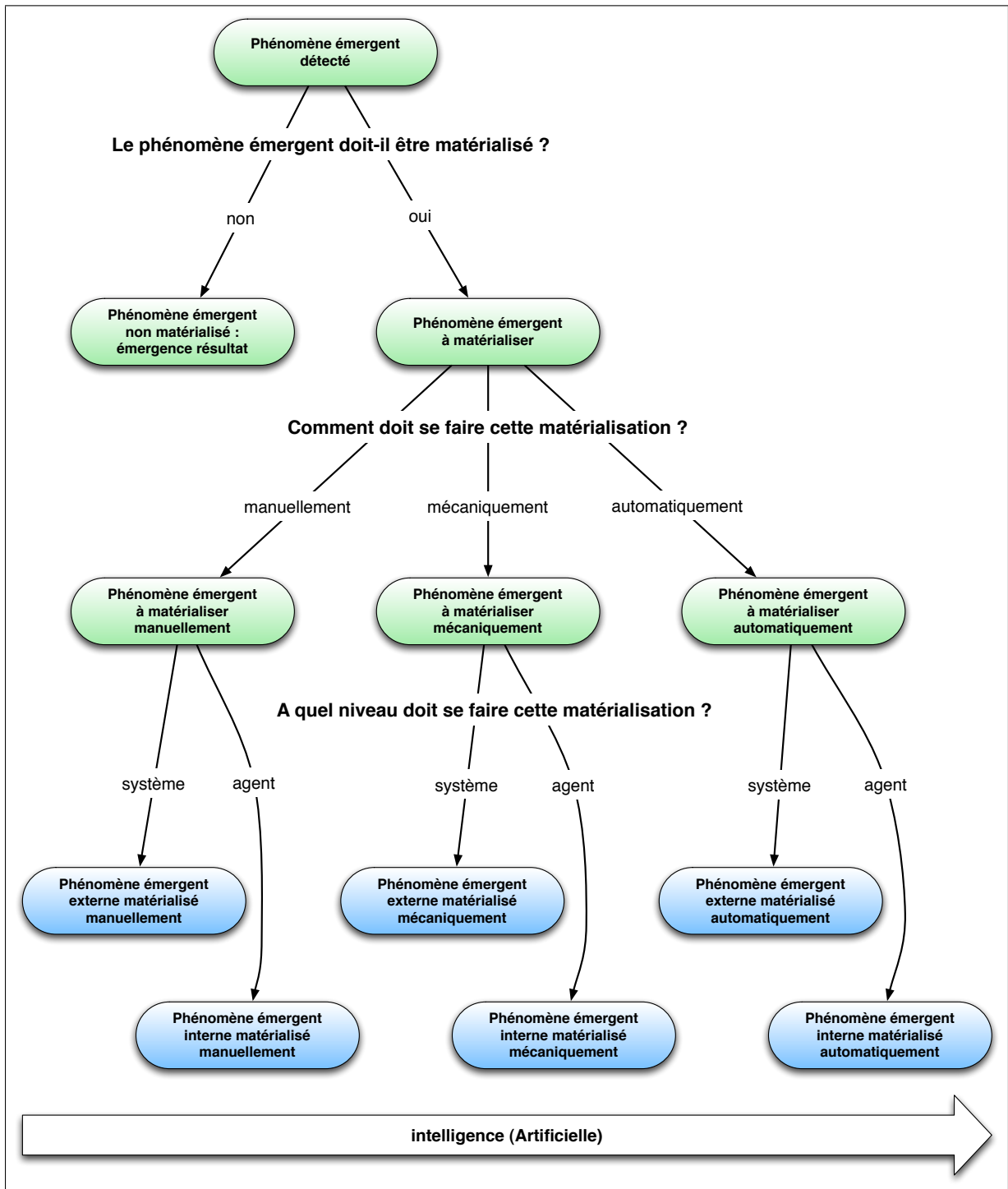


FIG. 4.6 – Les différentes stratégies de réification d’un phénomène émergent

vateurs du système et non pas dans le système lui-même. Et, faisant partie de la communauté SMA, et donc partisans de l'Intelligence Artificielle Distribuée, nous considérons que les techniques de matérialisation internes aux modèles, qui mettent en œuvre des mécanismes au niveau des agents du système, sont – potentiellement – plus performantes que celles qui se situent au niveau système.

Les étapes conduisant à la matérialisation d'un phénomène émergent, et qui permettent donc de compléter le processus de sa réification, sont cependant moins abordables sur le plan de la mise en œuvre effective que ne le sont les étapes décrites lors de la phase de détection. Comme nous l'avons dit, il est concevable de détecter de nombreux phénomènes dont nous ne comprenons pourtant pas les mécanismes internes, ce qui rend leur matérialisation difficile, car nous ne savons pas quelles modifications faire pour donner une place concrète à ces phénomènes dans une SOA sans pour autant la dénaturer. Il faut donc être conscients que si l'outil informatique peut aujourd'hui apporter un appui pertinent à la détection de phénomènes émergents dans des simulations il ne peut en revanche que servir d'outil permettant par exemple de tester différentes hypothèses de matérialisation.

4.5 Conclusion

L'émergence est une notion qui passionne les scientifiques de différents horizons et qui est depuis toujours présentée comme l'un des concepts favorisant les systèmes multi-agents par rapport aux autres techniques de simulation. Comme nous l'avons rappelé en introduction, de nombreux travaux ont permis de définir et de classer les phénomènes émergents observés dans un système, et se sont parfois attaqués à la question de leur réification.

Nous considérons qu'aujourd'hui le problème n'est pas tant de réussir à réifier un phénomène émergent que de parvenir à déterminer les raisons qui font que sa réification devrait être faite ou non, ainsi que les étapes qui permettent de conduire à une telle réification. On peut ainsi légitimement penser qu'il est des cas où une utilité existe tandis qu'il existe d'autres cas où elle demeure plus aléatoire. La question initiale revient donc à se demander quels sont les cas où la réification d'éventuels phénomènes émergents prend un intérêt, et nous sommes convaincus que ceci dépend du contexte de connaissance dans lequel les travaux sont effectués.

Dans ce chapitre, nous avons donc défini la réification d'un phénomène émergent comme étant un processus qui voit se succéder une phase de détection et une phase de matérialisation de ce phénomène. L'étude de ces deux phases nous a permis de décrire différentes stratégies de détection et de matérialisation qui, ensemble, aboutissent à la réification des phénomènes souhaités. Nous avons regroupé l'ensemble des types de réifications possibles d'un phénomène émergent sur la figure 4.7. Cet ensemble des types de réifications possibles est évidemment restreint aux feuilles des deux arbres reflétant les stratégies de détection et de matérialisation qui sont cohérentes entre elles.

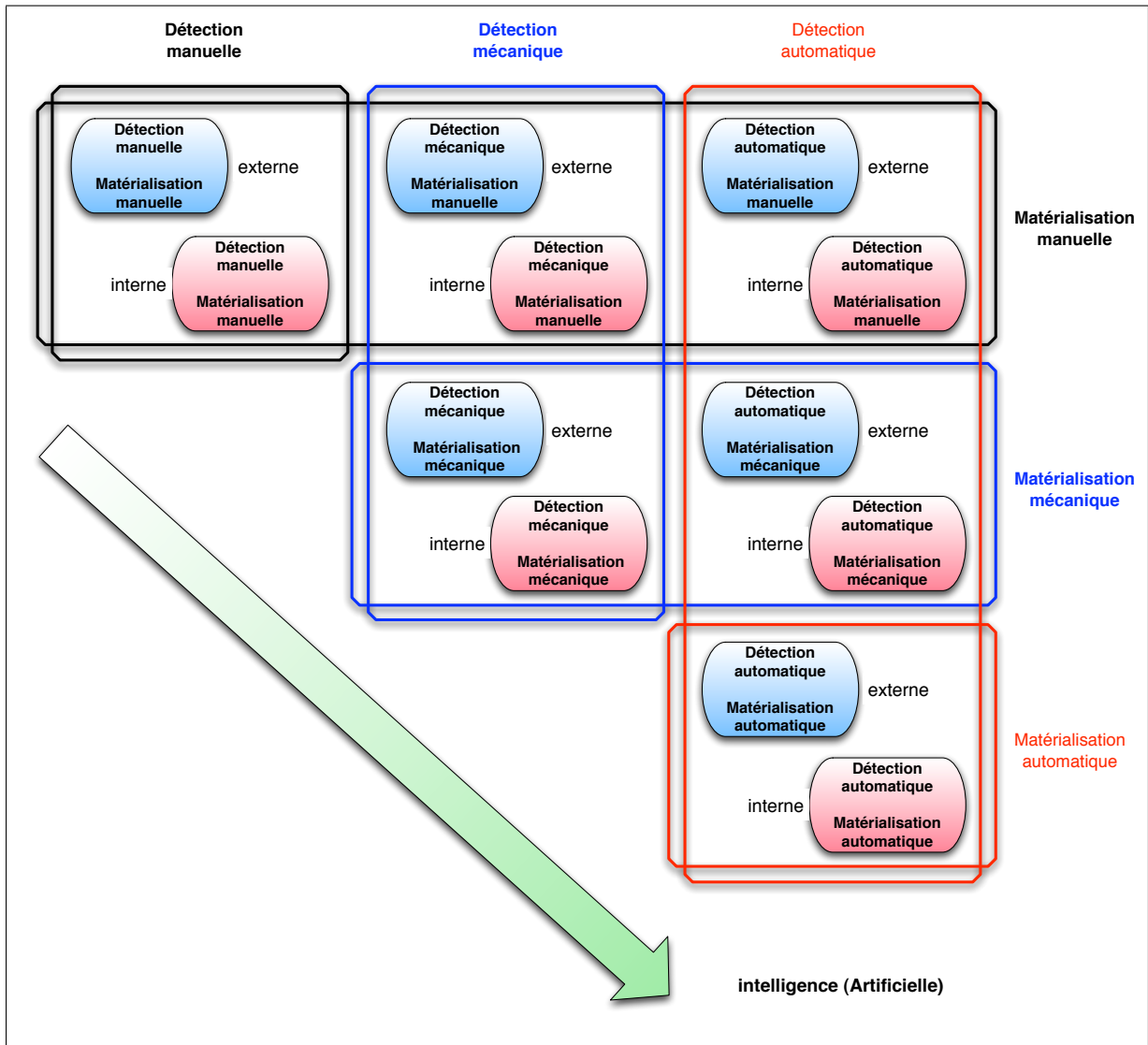


FIG. 4.7 – Classification des types de réifications possibles d’un phénomène émergent dans une SOA

Compte tenu de la "puissance" de ces types de réifications possibles en terme d'Intelligence Artificielle présente dans le système simulé, cette figure fait donc ressortir une classification des types de réifications possibles des phénomènes émergents qui peuvent apparaître dans une SOA. Sur la figure sont représentés de façon croisés les trois critères principaux qui permettent d'établir la classification : la façon dont est effectuée la phase de détection, la façon dont est effectuée la phase de matérialisation, et l'aspect interne ou externe des mécanismes utilisés par rapport au modèle conceptuel du domaine. L'Intelligence Artificielle contenue dans le système est représentée par une flèche dont le dégradé de couleur verte illustre l'acquisition et la production de connaissance associée au système.

En l'état actuel des connaissances, il est donc raisonnable de penser que les techniques permettant de réaliser la phase de détection des phénomènes émergents peuvent être pertinentes, mais que celles permettant de réaliser leur phase de matérialisation peuvent être difficiles à mettre en œuvre de par la difficulté de caractériser ces phénomènes. Mais ces techniques sont sûrement utiles dans un contexte où l'on cherche à simuler différents scénarios, non pas dans le but direct de reproduire la réalité, mais de simuler des hypothèses servant à valider nos connaissances sur des phénomènes particuliers.

En suivant la démarche méthodologique décrite dans le chapitre précédent, et en choisissant d'appliquer les stratégies de réification que nous venons de décrire, il est donc possible de mieux cadrer la façon dont les phénomènes émergents (et à travers eux la notion même d'émergence) sont présents dans les SOA. Afin de rendre cette prise en compte effective, il reste à présent à formaliser certaines notions et à proposer des éléments s'incluant dans l'architecture SMA d'une SOA, ce que nous ferons au prochain chapitre.

Cadre formel et architectural de réification des phénomènes émergents

Sommaire

5.1	Introduction	95
5.2	Vers une formalisation de la réification des phénomènes émergents	97
5.2.1	Cadre formel pour la réification des phénomènes émergents	98
5.2.2	Introspection et intercession	98
5.3	Phase d'introspection	99
5.3.1	Faits et connaissances	99
5.3.2	Révélateurs d'émergence et méta-connaissance	102
5.4	Phase d'intercession	104
5.4.1	Agents d'émergence et éléments d'interposition	105
5.4.2	Lois d'émergence et méta-structures émergentes	105
5.5	Construire une SOA adaptée à l'émergence	108
5.5.1	Le rôle des thématiciens dans la conception et l'utilisation d'une SOA	108
5.5.2	Une plate-forme de SOA adaptée	109
5.5.3	Faisabilité et spécificités liées à la plateforme GEAMAS-NG	110
5.6	Conclusion	113

5.1 Introduction

Dans le chapitre précédent, nous avons décrit un ensemble de stratégies de réification des phénomènes émergents qui peuvent apparaître dans une Simulation Orientée Agent (SOA). Ces stratégies entrent dans le cadre de la méthodologie générale de prise en compte de l'émergence en tant que que méta-connaissance que nous avons décrite au chapitre 4, et elles dé-

pendent directement de choix qui sont à faire lors des deux phases de la réification, la phase de détection et la phase de matérialisation.

Dans ce chapitre, nous allons à présent présenter un cadre conceptuel qui permet de formaliser la réalisation des étapes permettant de parvenir à la réification d'un phénomène, et nous présenterons des éléments architecturaux permettant de procéder à leur matérialisation dans une SOA. Ce chapitre vient donc clore la deuxième partie de ce manuscrit en apportant des éléments qui viennent s'ajouter à ceux vus dans les deux chapitres précédents afin d'améliorer la prise en compte de l'émergence dans les SOA sur les plans méthodologiques, conceptuels, et architecturaux, faisant ainsi avancer chacune des dimensions de la figure 5.1 que l'on retrouve décrites dans [Courdier 2003].

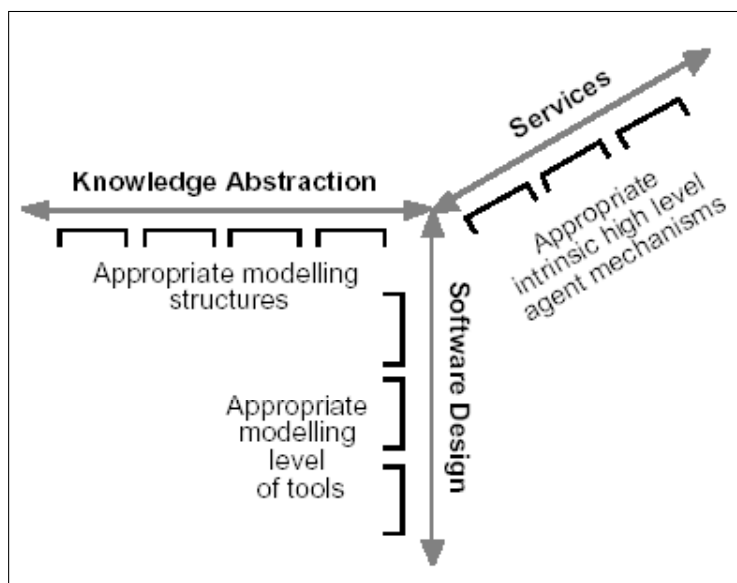


FIG. 5.1 – Les dimensions conceptuelles du développement d'une architecture de plateforme de simulation agent (extrait de [Courdier 2003])

Le cadre formel que nous présentons ici est principalement destiné à faciliter la mise en œuvre des phases de détection et de matérialisation des phénomènes émergents lorsqu'elles sont effectuées grâce à des techniques mécaniques, dans lesquelles les thématiciens sont sollicités uniquement pour décrire et caractériser des phénomènes qui sont alors détectés et matérialisés dans le modèle simulé par l'utilisation de mécanismes disponibles au niveau logiciel. Ceci permet de ne pas avoir recours aux techniques (parfois fastidieuses) d'une détection complètement effectuée par l'observation humaine de la simulation (par les thématiciens experts) et d'une matérialisation uniquement effectuée par des modifications du code (par les informaticiens).

Nous revenons donc dans ce chapitre sur les définitions des phases de détection et de matérialisation des phénomènes émergents vues au chapitre précédent que nous raffinons pour tenir compte de la spécificité de l'utilisation de techniques logicielles, introduisant ainsi les notions de phase d'introspection et de phase d'intercession. Nous présentons les mécanismes

qui permettent de réaliser ces deux phases et qui reposent sur l'analyse des faits qui se passent pendant le déroulement d'une simulation, ces faits étant eux-mêmes traités grâce aux concepts de révélateurs d'émergence et de lois d'émergence.

5.2 Vers une formalisation de la réification des phénomènes émergents

Dans un souci de généralité, la proposition s'inscrit dans le cadre d'un modèle SMA qui se veut le plus général possible. Dans celui-ci, nous concevons évidemment les agents comme étant des entités agissant au travers de mécanismes de perception, d'influence et d'interaction au sein d'un ou de plusieurs environnements.

L'ensemble du processus de réification des phénomènes émergents que nous allons décrire repose sur la volonté d'extraire de la connaissance depuis la SOA ; connaissance qui doit ensuite être analysée et confrontée à des regards experts afin de détecter puis matérialiser les phénomènes émergents éventuels. Ceci est représenté sur la figure 5.2 sur laquelle nous avons fait apparaître les niveaux relevant de la connaissance et de la méta-connaissance.

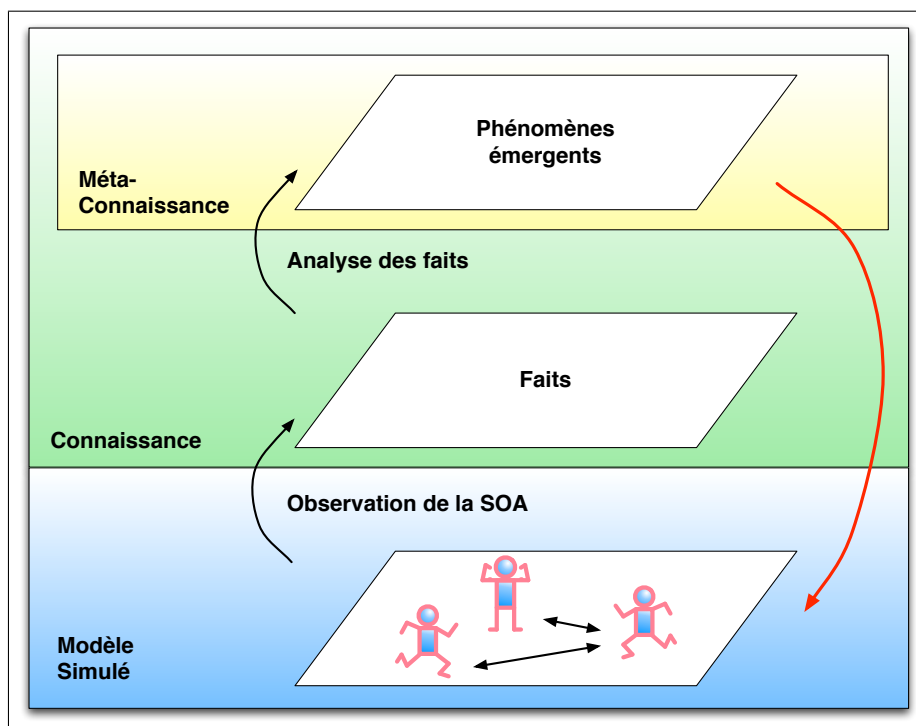


FIG. 5.2 – De la SOA à la méta-connaissance

Nous plaçons notre proposition à un niveau conceptuel qui permet de s'affranchir des concepts spécifiques qui sont intégrés et manipulés dans les différentes plates-formes de SOA existantes et nous monterons à la fin de ce chapitre les spécificités liées à la mise en œuvre de cette proposition conceptuelle générale dans la plateforme de simulation GEAMAS-NG que nous développons dans l'équipe SMART.

5.2.1 Cadre formel pour la réification des phénomènes émergents

Comme nous l'avons énoncé au chapitre précédent, la réification de l'émergence peut être décomposée en deux phases distinctes mais complémentaires :

1. la phase de détection, au cours de laquelle on veut détecter les phénomènes qui émergent dans le système ;
2. et la phase de matérialisation, au cours de laquelle on veut rendre possible leur matérialisation dans le système en tant que tels.

Ces deux phases sont ainsi définies, et ainsi nommées, dans le cadre général de la méthodologie que nous avons proposée pour prendre en compte l'émergence dans un processus de simulation. Dans le cadre de ce chapitre, et par rapport à la classification établie au chapitre précédent, nous allons laisser de côté les aspects (triviaux au niveau de la nouveauté implémentatoire) où l'on souhaite détecter et matérialiser de façon manuelle les phénomènes émergents, et nous allons porter notre attention sur les modifications que l'on souhaite voir apparaître dans un système de façon mécanique ou de façon automatique. Si l'on suit toujours la méthodologie définie au chapitre précédent ceci implique alors également de distinguer les changements (et la façon dont ceux-ci se feront) dans le système qui seront gérés au niveau plateforme ou qui seront gérés au niveau agent.

Dans un cas comme dans l'autre, nous nous intéressons de toute façon à des modifications qui sont directement liées au logiciel, puisque même dans le cas où la réification ne serait "que" mécanique (soit le plus "faible" par rapport à la classification définie précédemment par rapport à une réification qui serait complètement automatique), le thématicien utilisateur ne ferait que guider le système qui se chargerait d'effectuer ensuite pour lui les modifications nécessaires. Les définitions des phases dites de détection et de matérialisation peuvent donc être raffinées pour tenir compte de ce contexte.

5.2.2 Introspection et intercession

Nous nous sommes intéressés ici aux travaux menés dans le champ de la réflexivité et des méta-objets [Kiczales 1991], dans lequel on distingue deux techniques différentes qui sont utilisées par les systèmes réflexifs :

- l'introspection, qui est la capacité d'un programme à examiner son propre état ;
- l'intercession, qui est la capacité d'un programme à modifier son propre état d'exécution ou d'altérer sa propre interprétation ou signification.

Pour aller plus loin, on peut décrire l'introspection comme étant une technique qui est utilisée pour effectuer des mesures de performance, inspecter des modules ou déboguer un programme. Elle est implémentée dans des langages comme Smalltalk ou Java qui fournissent des outils pour connaître la classe d'un objet, ses attributs, ses méthodes, etc. (cette propriété n'existe pas dans des langages comme C ou Pascal). Et on peut décrire l'intercession comme

étant une technique qui permet à un programme d'évoluer automatiquement en fonction des besoins et de l'environnement (cette propriété apparaît dans des langages comme SmallTalk ou Python, mais elle n'existe pas dans des langages comme C).

Considérant ces deux notions, et à partir des définitions des phases de détection et de matérialisation des phénomènes émergents que nous avons établies au chapitre précédent, nous pouvons poser les deux définitions suivantes :

Définition II.6 (Phase d'introspection) *La phase d'introspection correspond à la phase générale de détection des phénomènes émergents lorsque celle-ci est réalisée au niveau système (par des mécanismes de la plateforme de simulation) ou au niveau de l'une ou de plusieurs de ses entités (par les agents eux-mêmes). Elle correspond au fait d'emprunter la voie mécanique ou la voie automatique lors de la phase de détection.*

Définition II.7 (Phase d'intercession) *La phase d'intercession correspond à la phase générale de matérialisation des phénomènes émergents lorsque celle-ci est réalisée au niveau système (par des mécanismes de la plateforme de simulation) ou au niveau de l'une ou de plusieurs de ses entités (par les agents eux-mêmes). Elle correspond au fait d'emprunter la voie mécanique ou la voie automatique lors de la phase de matérialisation.*

Dans la suite de ce chapitre nous allons donc décrire une formalisation que nous proposons pour la phase d'introspection et la phase d'intercession, tout en donnant des éléments d'architecture SMA permettant de mettre en place les mécanismes inhérents pour chacune de ces deux phases.

5.3 Phase d'introspection

En philosophie, l'introspection est définie comme étant une méthode d'observation et d'analyse en vue d'étudier sa propre personne et d'acquérir une connaissance de soi. C'est en quelque sorte en faisant faire au SMA sa propre introspection que nous allons parvenir à lui faire détecter les phénomènes qui émergent en son sein. Cette phase d'introspection repose évidemment sur l'analyse de ce qui se passe au cœur du système.

5.3.1 Faits et connaissances

Pour parvenir à effectuer la détection des phénomènes émergents qui apparaissent nous avons besoin de connaître l'ensemble (ou un ensemble) des faits qui se produisent (ou qui se sont produits, dans le cadre d'une analyse *a posteriori*) au sein du système simulé. Ces faits sont relatifs aux entités qui composent la SOA, et réfèrent donc de façon directe au comportement des agents et à leur environnement. Il peuvent alors être liés à l'espace, au temps, à la communication, et, de façon plus générale, aux différentes entités et aux différentes métriques qui sont présentes dans le système modélisé et simulé.

La figure 5.3 montre ainsi comment un ensemble de faits extraits des différentes couches des environnements d'une SOA peuvent permettre de décrire des phénomènes émergents. Comme nous y reviendrons, les circonstances qui conduisent à l'émergence d'un phénomène dans une simulation sont souvent complexes et difficilement appréhendables, mais les conditions qui permettent de savoir que ce même phénomène a émergé sont souvent plus simples à décrire, facilitant ainsi leur détection. Dans une SOA un phénomène émergent peut donc être détecté en analysant un ensemble de faits qui constituent de la connaissance sur le système simulé, et c'est grâce à la confrontation de cette connaissance à l'expertise des thématiciens concernés que ce phénomène émergent prend son sens en tant que méta-connaissance.

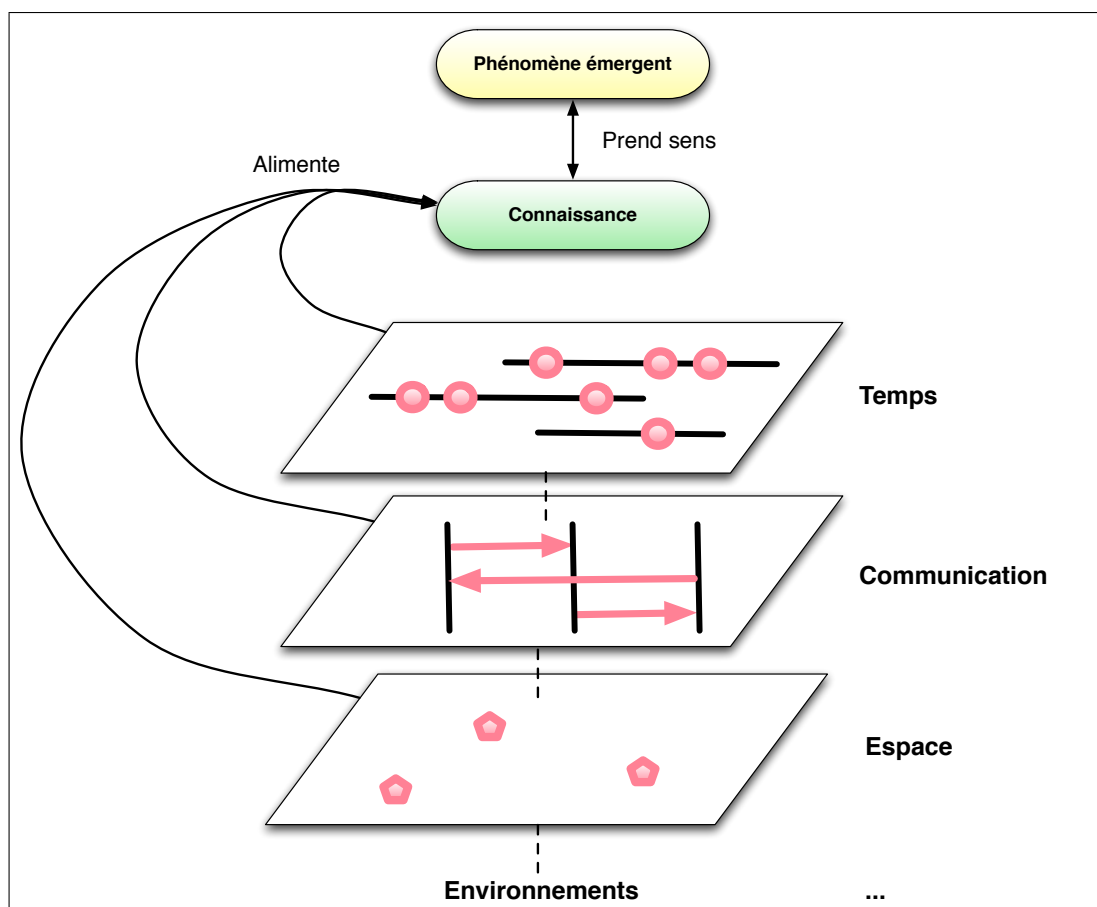


FIG. 5.3 – De l'usage des différents environnements et règles métriques pour caractériser un phénomène émergent

La question qui se pose est alors de savoir comment un système peut parvenir à construire cet ensemble de faits qui sera utilisé dans le cadre de son introspection générale (dans le cadre d'une introspection qui serait externe au modèle conceptuel du domaine), ou comment des agents peuvent arriver à construire le propre ensemble de faits les concernant qui sera utilisé dans le cadre de leur introspection particulière (dans le cadre d'une introspection qui serait interne au modèle conceptuel du domaine).

Cet ensemble de faits peut être construit grâce à une observation rigoureuse de la SOA, et principalement de l'évolution des agents qui la composent, de leurs interactions, et de l'évolution du ou des environnement(s) dans le(s)quel(s) ces agents évoluent. Il est donc important d'utiliser des plateformes de SOA proposant des mécanismes d'observation de la simulation qui soient performants. Ces mécanismes permettent généralement d'ajouter simplement des sondes particulières au niveau des agents et des environnements, sondes qui permettent alors de suivre un ensemble d'indicateurs ciblés. En fonction des types d'entités et d'interactions à observer il est aussi possible d'utiliser des mécanismes d'observation plus évolués. Ainsi, si l'on souhaite par exemple extraire des faits résultant des interactions de certains agents au sein d'un environnement de communication, on peut envisager l'utilisation de cartes et de fils de conversation qui résultent de précédents travaux de notre équipe [Ralambondrainy 2009] et dont un exemple est donné sur la figure 5.4.

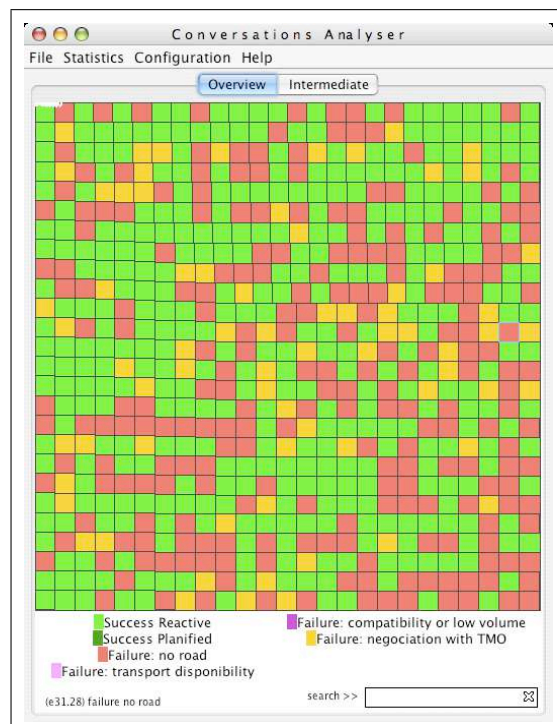


FIG. 5.4 – Un exemple de carte des conversations (extrait de [Ralambondrainy 2009])

L'existence de tels mécanismes d'observation dans les plateformes de SOA est souvent liée au besoin humain d'observer les résultats des simulations pour obtenir de la connaissance sur le système simulé, mais il est possible de les exploiter pour alimenter de la connaissance interne à ce système. Un système en général, et ses agents en particulier, peuvent donc utiliser de tels mécanismes d'observation pour construire les ensembles de faits leur permettant de réaliser leur introspection.

En utilisant des mécanismes d'observation de la simulation, le système ou les entités le

constituant peuvent construire un ensemble \mathcal{K} que nous définissons ainsi :

$$\mathcal{K} = \{faits\}$$

Cet ensemble regroupe l'ensemble des faits qui représentent la connaissance utile (pour l'analyse des phénomènes émergents) que l'on peut avoir sur le système étudié pour que celui-ci puisse faire son introspection afin de détecter d'éventuels phénomènes émergents. L'ensemble \mathcal{K} est là défini de façon générale, mais étant dans le contexte d'une SOA sa définition peut être affinée et cet ensemble de faits peut donc être vu comme étant :

$$\mathcal{K} = \mathcal{K}_A \cup \mathcal{K}_E$$

où \mathcal{K}_A est l'ensemble qui comprend les faits qui se déroulent au niveau des agents de la simulation, et que nous définissons donc ainsi :

$$\mathcal{K}_A = \{influences, perceptions, interactions\}$$

et où \mathcal{K}_E est un ensemble qui comprend les faits qui se déroulent au niveau du ou des environnements de la simulation.

Une fois la connaissance contenue dans l'ensemble \mathcal{K} obtenue, on se propose de bâtir une méta-connaissance sur celle-ci, afin de pouvoir l'utiliser pour détecter concrètement les phénomènes qui émergent dans le système. Ceci revient à interpréter cette connaissance en fonction d'autres connaissances qui sont liées à un contexte donné qui peut être l'expertise du thématique dans le cadre d'une introspection mécanique, ou pourquoi pas une "prédiction" des agents ou du système dans le cadre d'une introspection automatique.

5.3.2 Révélateurs d'émergence et méta-connaissance

Pour parvenir à détecter les phénomènes qui émergent au sein d'un système simulé, nous avons, dans le deuxième chapitre de ce manuscrit, posé l'émergence comme étant une méta-connaissance sur le système étudié, cette méta-connaissance étant elle-même définie comme étant une connaissance sur la connaissance. Or nous venons de définir cette connaissance comme étant l'ensemble \mathcal{K} des faits qui se produisent dans le système.

Selon notre approche, un phénomène émergent se définit donc au travers une étude de cet ensemble de faits \mathcal{K} . Cette analyse, qui consiste à établir des relations entre les faits, permet de mettre en lumière les événements qui se produisent dans la SOA et qui, pour une observation particulière, représentent un phénomène émergent.

Notons toujours que ces faits que nous utilisons pour décrire et caractériser des phénomènes émergents sont bien différents de ceux, parfois très difficiles à exprimer, qui sont à l'origine de l'émergence même de ces phénomènes. En effet, il est souvent difficile d'exprimer les conditions qui permettent à un phénomène d'émerger puisque, par nature même de l'émergence (rappelons nous, le "Tout est plus que la somme des Parties"...), ces conditions ne

suffisent pas forcément pour conduire à l'émergence de phénomènes. En revanche, compte tenu d'un contexte donné, d'un niveau d'expertise donné (que celle-ci soit humaine ou logicielle), et ainsi d'une certaine méta-connaissance sur le système étudié et simulé, certains faits peuvent souvent suffire à caractériser les phénomènes émergents et l'analyse de ces faits peut donc conduire à la détection des phénomènes qui y sont associés. Ainsi, si on se rapporte à l'exemple simple du banc de poissons, on conviendra qu'il est certainement difficile d'exprimer les faits qui font que ce phénomène émerge (par exemple : des poissons ont-ils eu peur d'un prédateur ? ont-ils été portés par le courant ? ont-ils "décidé" de chasser ensemble ? etc. ?), mais qu'il est par contre assez facile de reconnaître des faits permettant de caractériser un banc de poissons (par exemple : une centaine de poissons sont regroupés et nagent dans la même direction).

Afin de détecter ces événements intéressants qui sont signes de l'émergence de phénomènes, et de compléter ainsi la phase d'introspection que nous avons définie, nous proposons d'avoir recours à un ensemble extensible de fonctions \mathcal{R}_E défini comme suit :

$$\mathcal{R}_E = \{f : \mathcal{K}^n \rightarrow \text{boolean}\}, \quad n \in \mathbb{N}$$

Chaque fonction de cet ensemble a pour rôle de détecter si les combinaisons de faits passés en paramètres définissent un phénomène émergent. Nous considérons ces fonctions comme étant des *révélateurs d'émergence*, un concept général que nous définissons ainsi :

Définition II.8 (Révélateur d'émergence) *Un révélateur d'émergence est une connaissance particulière caractérisant la présence d'un phénomène particulier. Ces révélateurs – qui peuvent être très abstraits lorsqu'ils n'existent que dans l'esprit des observateurs de la simulation ou qui peuvent être très concrets en prenant la forme de fonctions utilisées au niveau logiciel – sont utilisés afin de détecter les phénomènes qui émergent dans une simulation.*

A partir de ces fonctions révélateurs d'émergence, nous définissons comme suit l'ensemble \mathcal{P}_E qui représente les phénomènes émergents qui ont effectivement été détectés dans le système :

$$\mathcal{P}_E = \{(f, k) \in \mathcal{R}_E \times \mathcal{K}^n / f(k) = \text{true}\}$$

Un phénomène émergent est donc vu comme étant un couple composé d'une fonction (qui est son révélateur d'émergence associé) et de faits particuliers qui sont ceux pour lesquels la fonction a été activée. Ceci permet d'utiliser les fonctions de \mathcal{R}_E pour décrire des phénomènes émergents en général, et de retrouver dans \mathcal{P}_E les situations particulières dans lesquelles ces phénomènes ont émergé, permettant ainsi de différencier d'éventuels phénomènes émergents de même type.

Parvenir à construire cet ensemble \mathcal{P}_E marque la fin de la phase d'introspection, que ce soit au niveau d'un service système qui intervient dans le cadre d'une technique de détection des phénomènes émergents qui est externe au modèle conceptuel du domaine, ou que ce soit au niveau d'agents de la SOA qui voient ces mécanismes d'introspection inclus dans leur

comportement et qui procèdent donc à la mise en œuvre d'une technique de détection des phénomènes émergents qui est interne au modèle conceptuel du domaine.

La formalisation que nous venons de présenter est en accord avec les concepts que nous avons avancés dans le chapitre 3, et notamment avec la méthodologie de conception de simulations que nous avons proposée. En effet, puisque les thématiques experts d'un domaine particulier ne possèdent pas une connaissance innée sur celui-ci, il est normal que les phénomènes émergents qu'ils appréhendent sont ceux qu'ils ont pu voir apparaître en mettant en œuvre des mécanismes d'observation particuliers. C'est seulement à partir de cette observation qu'ils ont pu apprendre à les reconnaître, les caractériser, les nommer, etc.

Les premiers phénomènes qui émergent et qui auront pu être détectés par des mécanismes d'observation mis en place, résultent essentiellement de l'analyse des descriptions fournies initialement par les experts thématiques. Il est important d'ajouter cette nouvelle connaissance sur ces phénomènes qui ont émergé à la connaissance globale du système. Ceci permettra d'une part de tendre vers une meilleure compréhension du système lui-même, et d'autre part de pouvoir mettre en évidence de nouveaux phénomènes émergents qui n'auraient pas pu être appréhendés à partir des seules connaissances formulées initialement par les experts.

Une fois qu'il a été créé lors de la phase d'introspection, cet ensemble \mathcal{P}_E sera utilisé comme base à partir de laquelle est réalisée la phase d'intercession.

5.4 Phase d'intercession

Cette phase rend possible la matérialisation dans la SOA des phénomènes émergents qui auront été détectés pendant la phase d'introspection.

Comme nous l'avons vu au chapitre 2 lorsque nous avons passé en revue le large panel des définitions et classifications qui se rattachent à la notion d'émergence, la plupart des types de phénomènes considérés comme étant les plus évolués (par exemple ceux relevant de l'émergence bilatérale ou de l'émergence diachronique) se définissent en fonction d'interactions qui ont lieu entre les niveaux microscopiques et macroscopiques.

En effet, les phénomènes émergents qui sont détectés dans le monde réel se manifestent souvent par des comportements qui font que l'existence même de ces phénomènes influe sur les entités du monde réel : certaines de ces entités peuvent participer directement à l'émergence de phénomènes, tandis que d'autres subissent les influences exercées par ces phénomènes, et que d'autres encore voient leur perception modifiée de par la présence de ces phénomènes.

Il en va évidemment de même dans le monde virtuel que nous manipulons dans une SOA puisque notre but est de reproduire des phénomènes qui se produisent dans des systèmes ou processus du monde réel, et il est donc important de proposer des solutions pour représenter ces phénomènes dans l'architecture d'une plateforme de SOA.

5.4.1 Agents d'émergence et éléments d'interposition

Comme nous l'avons dit, la détection des phénomènes émergents comme nous l'avons spécifiée dans la section précédente peut parfois aboutir à définir des entités qui influencent directement le comportement des agents au sein de la SOA.

Dans la SOA il est donc intéressant de mettre en place des structures émergentes qui représenteront les phénomènes émergents à matérialiser. Nous proposons notamment deux formes de structures :

- des agents d'émergence ;
- des éléments d'interposition.

Nous définissons ces deux structures de la façon suivante :

Définition II.9 (Agent d'émergence) *Un agent d'émergence est un agent qui s'exécute au sein d'une plate-forme de SOA. Il évolue ainsi dans le ou les mêmes environnements que tous les autres agents du système et interagit avec eux au travers des mécanismes d'influence et de perception sur lesquels s'appuie la plate-forme hôte. Plusieurs agents d'émergence peuvent être créés pour réifier un même phénomène.*

Définition II.10 (Élément d'interposition) *Un élément d'interposition est une structure permettant la modification pour un ou plusieurs agents de l'un ou de plusieurs des environnements dans lesquels ils évoluent. Une telle structure modifie (selon les cas en les altérant, en les améliorant, en les restreignant, etc.) les mécanismes de perception ou d'influence qui sont utilisés par les agents de la SOA.*

Ces deux types de structures, agent d'émergence et éléments d'interposition, sont représentées sur la figure 5.5. Y sont représentées notamment schématiquement les influences et perceptions modifiées qui sont gérées par les éléments d'interposition.

La matérialisation des phénomènes émergents dans une SOA se fait en utilisant ces deux types de structures dont la mise en place dépend de lois d'émergence.

5.4.2 Lois d'émergence et méta-structures émergentes

Les deux structures d'émergence que nous avons définies, agents d'émergence et éléments d'interposition, sont contrôlées par des méta-structures émergentes, que l'on appellera ms_E , qui sont elles-mêmes régies par des lois d'émergence, un concept général que nous définissons ainsi :

Définition II.11 (Loi d'émergence) *Une loi d'émergence est une connaissance particulière caractérisant le comportement d'un phénomène particulier et les répercussions qui sont liées à l'existence de ce phénomène. Ces lois sont utilisées pour caractériser la façon dont se comportent les entités logicielles qui sont utilisées afin de matérialiser les phénomènes qui émergent dans une simulation.*

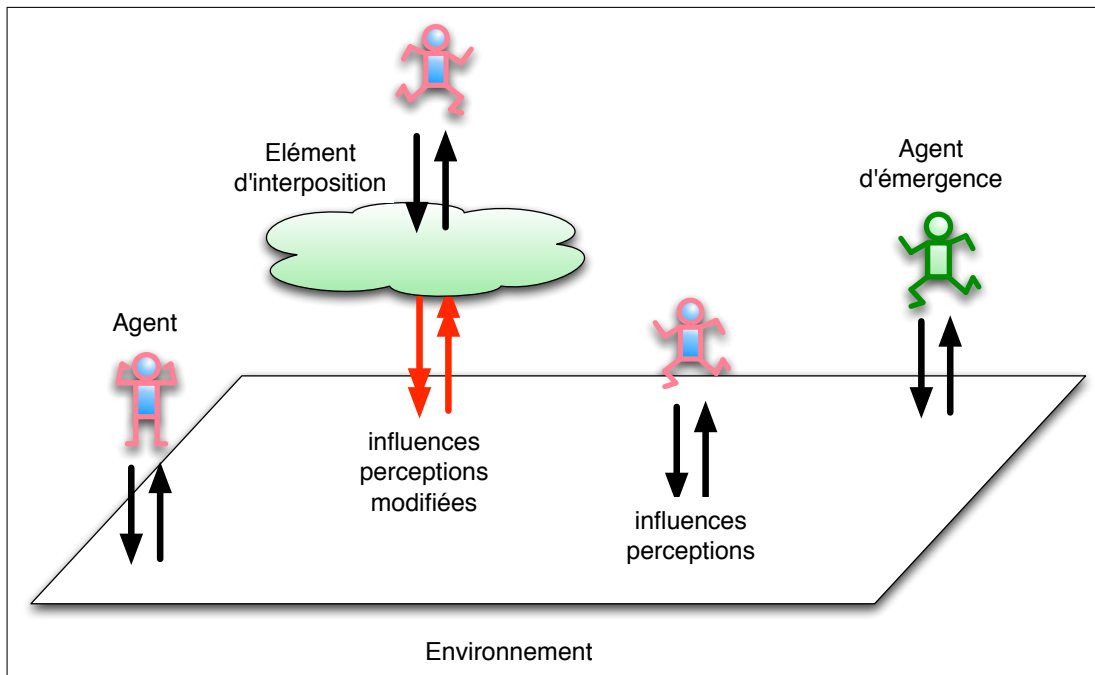


FIG. 5.5 – Les structures d'émergence (agents d'émergence et éléments d'interposition) et les mécanismes d'influences/perceptions modifiées

Ici, nos lois d'émergence sont définies comme étant les éléments de l'ensemble \mathcal{L}_E , dont nous posons la définition suivante :

$$\mathcal{L}_E = \{f : \mathcal{P}_E^n \rightarrow \mathcal{S}_E\}, \quad n \in \mathbb{N}$$

où \mathcal{S}_E désigne l'ensemble des structures émergentes, chaque structure émergente étant définie par un triplet $\langle \text{méta-structure émergente, agents d'émergence, éléments d'interposition} \rangle$.

Les deux types d'éléments que sont les agents d'émergence et les éléments d'interposition, considérés de façon complémentaire ou indépendante, nous permettent de prendre en compte différents types de phénomènes qui se manifestent dans le SMA étudié.

Ainsi, dans un exemple d'émergence intrinsèque (au sens de la classification de [Boschetti & Gray 2007]), comme l'apparition d'un banc de poissons, le banc de poissons sera représenté de façon directe dans le système par l'agent d'émergence. Et les différents poissons qui constituent le banc de poissons pourront continuer à évoluer dans leur environnement mais verront leurs influences et perceptions modifiées par des éléments d'interposition qui seront pilotés par la ms_E représentant le banc de poissons.

Dans le même ordre d'idées, citons un exemple d'émergence faible (au sens de [Stephan 2002]) : les brindilles (objets de l'environnement) qui auront été entassées par des agents termites faisant ainsi émerger un tas de bois. Ici le tas de bois n'a pas de comportement propre, et il n'y a donc pas lieu de faire apparaître un agent d'émergence pour le représenter. En revanche, si certaines entités du système doivent percevoir le tas de bois

en tant que tel, ceci sera rendu possible grâce aux éléments d'interposition gérés par la m_{SE} représentant le tas de bois qui modifieront les influences et perceptions de ces entités.

Le processus complet de réification des phénomènes émergents est représenté sur la figure 5.6 qui reprend les ensembles \mathcal{K} , \mathcal{P}_E , \mathcal{S}_E , et le rôle clé dans chacune des phases d'introspection et d'intercession des ensembles \mathcal{R}_E et \mathcal{L}_E .

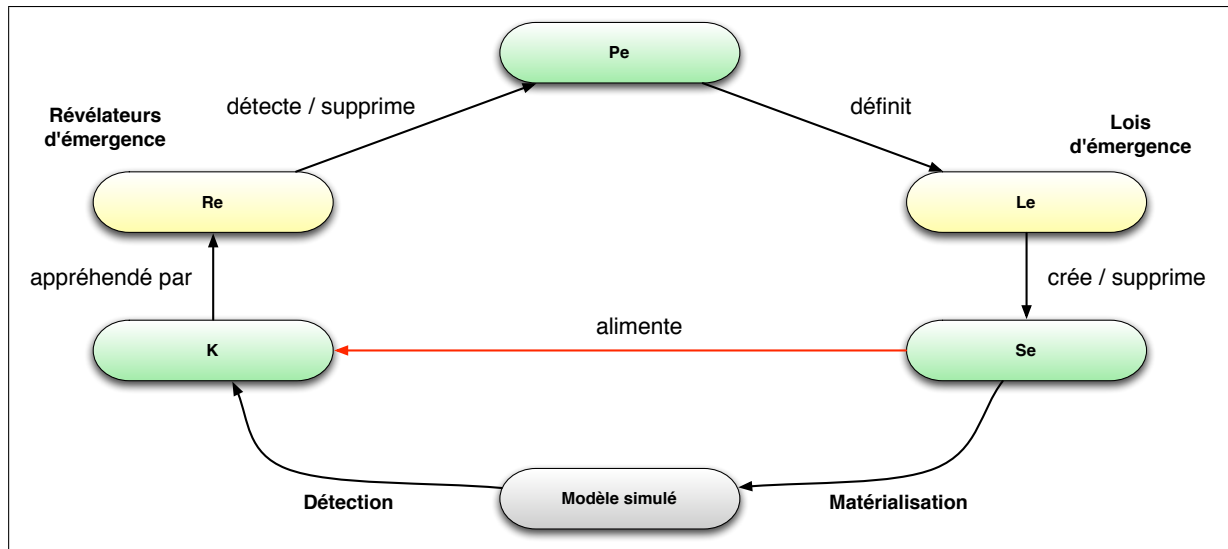


FIG. 5.6 – Le processus formalisé de prise en compte de l'émergence

Sur cette figure, les ensembles \mathcal{K} , \mathcal{P}_E , \mathcal{S}_E sont représentés en vert, puisqu'ils sont tous – à leurs niveaux respectifs – des éléments relevant directement de la connaissance, et les ensembles \mathcal{R}_E et \mathcal{L}_E sont représentés en jaune puisqu'à chaque étape ils sont le reflet de la méta-connaissance nécessaire pour analyser des connaissances potentielles et pour en extraire ainsi de nouvelles connaissances. La fin du cycle de réification voit apparaître en rouge une flèche visant à rappeler que toute nouvelle connaissance matérialisée dans un système (par exemple au travers les structures d'émergence que nous avons définies) se traduira, dans les cycles d'exécutions suivants de la SOA, par la production de nouveaux faits qui eux-mêmes alimenteront alors la connaissance observable sur la simulation, et qui pourraient alors être utilisés pour décrire de nouveaux phénomènes de niveaux supérieurs.

Ceci correspond bien à notre vision de l'émergence comme étant une méta-connaissance, puisqu'on retrouve dans ce processus de réification de l'émergence les différentes déclinaisons de la méta-connaissance que nous avons listées dans ce cadre au début du chapitre 3 : l'aspect description de connaissances, puis découverte de ces connaissances, et enfin manipulation de ces connaissances.

Enfin, nous terminons cette section en évoquant le cas que l'on oublie souvent lorsque l'on considère l'émergence comme étant un "résultat" d'une simulation : celui où un phénomène

émergent donné... n'émerge tout simplement plus. Ce cas est pris en charge de façon naturelle dans le processus de réification que nous avons décrit dans la mesure où celui-ci repose sur l'analyse des faits qui ont lieu dans la simulation. Si les faits qui décrivent un phénomène donné ne sont plus observés, et le révélateur d'émergence qui lui est associé n'est alors plus activé. L'ensemble \mathcal{R}_E contient évidemment toujours la fonction permettant de détecter ce phénomène s'il émergeait à nouveau, mais dans l'immédiat cette fonction ne retourne pas "true" et n'appartiendra donc pas à l'ensemble \mathcal{P}_E .

Au bout du processus de réification le phénomène qui au cycle de simulation précédent était encore matérialisé grâce à d'éventuels éléments d'interpositions et agents d'émergence finit donc par être "auto-dissout" par sa propre ms_E .

5.5 Construire une SOA adaptée à l'émergence

Pour pouvoir être mis en œuvre, le processus de réification que nous venons de présenter nécessite de faire appel à plusieurs mécanismes disponibles dans la plateforme sur laquelle s'exécute la SOA, afin que la plateforme elle-même (dans le cas d'une réification niveau système, et donc externe au modèle conceptuel du domaine) ou que ses entités (dans le cas d'une réification niveau agent, et donc interne au modèle conceptuel du domaine) puissent réaliser leurs phases d'introspection et d'intercession.

5.5.1 Le rôle des thématiciens dans la conception et l'utilisation d'une SOA

Dans le cadre des simulations dédiées à l'étude de systèmes naturels et sociaux – ce qui, au delà de la prospective territoriale, constitue notre axe de recherche privilégié au sens large – l'informatique est vue comme un outil pour les thématiciens divers qui sont, comme nous l'avons dit à plusieurs reprises, les détenteurs de la connaissance sur le système.

Dès le début d'une démarche de conception de simulations, l'idéal est donc que les SOA soient conçues en tenant compte de cette connaissance des experts et le cadre formel que nous venons de décrire entre évidemment dans le cadre d'une telle méthodologie de conception pour parvenir à une réification des phénomènes émergents.

Relativement au cadre formel que nous venons de présenter, les thématiciens interviennent pour définir les révélateurs d'émergence et ainsi créer les ensembles \mathcal{R}_E et \mathcal{L}_E qui sont au cœur du processus de réification d'un phénomène émergent.

Identifier les différents éléments de ces ensembles est en fait un point précis dans la méthodologie de conception du système qui doit être traité par les thématiciens experts des domaines auxquels se rattache le système complexe qu'ils se consacrent, avec les informaticiens, à reproduire de façon virtuelle.

Chacun d'entre eux va enrichir \mathcal{R}_E en définissant des fonctions, combinaisons de faits qu'il sait – relativement à la connaissance particulière qu'il a de son domaine d'expertise – être susceptibles d'apparaître et auxquelles il trouve du sens. Et cet ensemble \mathcal{R}_E pourra alors être

utilisé avec pertinence pour détecter les phénomènes émergents lors de la phase d'introspection.

Et chacun d'entre eux va également enrichir \mathcal{L}_E en définissant la façon dont les phénomènes émergents qui l'intéressent seront matérialisés dans la SOA en utilisant les structures d'émergences mises à leur disposition. Arriver à créer un système le plus performant (et le plus réaliste) possible se fait donc grâce à des interactions continues entre experts thématiques et informaticiens, les uns bénéficiant sans cesse des retours des autres.

Tout ceci est évidemment valable dans le cadre d'un processus de réification mécanique, tel que nous l'avons défini au chapitre précédent, dans lequel la détection et la matérialisation des phénomènes émergents se fait en utilisant des mécanismes disponibles sur les plateformes de SOA, et dans lequel ces mécanismes s'appuient sur une connaissance apportée par les utilisateurs (qui sont le plus souvent les thématiques co-concepteurs des modèles et outils).

Mais le cadre formel et architectural que nous avons décrit est également valable dans le cas d'un processus de réification qui serait automatique. Ce serait alors aux agents eux-mêmes (dans le cas d'un processus de réification interne au modèle conceptuel du domaine) ou à un service système (dans le cas d'un processus de réification externe au modèle conceptuel) de définir les ensembles \mathcal{R}_E et \mathcal{L}_E qu'ils utiliseront pour détecter puis matérialiser les phénomènes qui émergent dans la SOA. Comme nous l'avons dit au chapitre précédent, ceci peut paraître utopique, mais peut tout de même être envisagé en donnant des capacités cognitives poussées aux entités du système, par exemple en fournissant aux agents des capacités de reconnaissance de formes, ou en leur donnant des facultés d'apprentissage.

5.5.2 Une plate-forme de SOA adaptée

La figure 5.7 illustre la vision générale de notre approche qui a été décrite dans la section précédente. Pour chacune des deux phases détaillées, la phase de détection des phénomènes émergents (ici considérée comme phase d'introspection) et leur phase de matérialisation *via* des structures émergentes (ici considérée comme phase d'intercession), nous avons identifié les services attendus par une plate-forme de SOA pour parvenir à nos fins et ainsi permettre la réification de l'émergence.

Ces services sont au nombre de trois :

- un service d'observation, grâce auquel nous pouvons mettre en place les différents mécanismes d'observation qui permettent d'établir l'ensemble des faits définissant la connaissance sur la SOA ;
- un service de manipulation d'agents qui permet la création, la suppression et la modification du cycle de vie des agents ;
- un service de manipulation d'objets de l'environnement qui permet la mise en place des éléments d'interposition.

Ces trois services sont représentés sur le schéma comme étant à l'interface entre le cadre formel dédié à la réification des phénomènes émergents que nous venons de présenter – qui si

situé sur un plan conceptuel – et l’exécution de la simulation – qui se situe au niveau système et logiciel.

Le cadre général dans lequel s’inscrit l’ensemble de notre proposition permet l’expression de comportements associés aux structures émergentes au sein de la SOA. Par conséquent, ces structures par influence / perception rajoutent de nouvelles connaissances (pour peu qu’elles puissent être observées) dans l’ensemble des faits. Ceci raffine les connaissances initiales sur le système simulé et par conséquent, dans notre démarche de production de connaissance, sur le système complexe étudié.

Notons également qu’il nous faudra utiliser une formalisation afin de décrire au mieux les ensembles de *faits* si on veut être capables d’en manipuler facilement les éléments. Ceci pourra être fait, au niveau représentation, en utilisant un formalisme issu des différentes logiques (logiques de descriptions, logique des défauts, logiques temporelles), et, au niveau implémentation, par une technique de programmation déclarative bien adaptée à l’unification de règles (par exemple en utilisant un langage comme Prolog).

Ainsi une plateforme de SOA doit pouvoir fournir des outils facilitant la saisie – et l’interprétation – des éléments de \mathcal{R}_E que doivent fournir les thématiciens utilisateurs qui souhaitent détecter les phénomènes émergents correspondant à ces révélateurs d’émergence. Et dans l’idéal – même si cela s’avère sans nul doute bien plus complexe – elle doit aussi proposer des outils facilitant la description des lois d’émergence contenues dans l’ensemble \mathcal{L}_E qui décrivent la façon dont les phénomènes émergents seront matérialisés en utilisant d’éventuels éléments d’interpositions et agents d’émergence. Ceci pourrait permettre à ces lois de ne pas avoir à être saisies *via* des créations et modifications directes dans le code du modèle informatique implémentant le modèle du domaine.

5.5.3 Faisabilité et spécificités liées à la plateforme GEAMAS-NG

Nous donnons ici quelques éléments de validation pour la mise en œuvre de ce cadre formel pour la réification des phénomènes émergents au sein de SOA qui seraient implémentées sur la plateforme de simulation multi-agents GEAMAS-NG que nous développons au sein de l’équipe SMART de l’IREMIA.

Dans GEAMAS-NG une séparation claire est mise en place entre les agents et les environnements, dans lesquels les agents interagissent au moyen de mécanismes d’influences et de perceptions. La plateforme, qui possède un ensemble de services destinés à l’observation de la simulation, possède toutes les caractéristiques souhaitées pour la réification de l’émergence au sein de modèles simulés.

La génération de l’ensemble des faits décrivant la connaissance sur le SOA se fait alors en intégrant les mécanismes d’observation décrits dans [Payet *et al.* 2006b] ou en s’inspirant de mécanismes d’observations proposés dans [Ralambondrainy 2009]. Pour être rendue pratique, l’utilisation de ces mécanismes d’observation passe par la mise en place de mécanismes

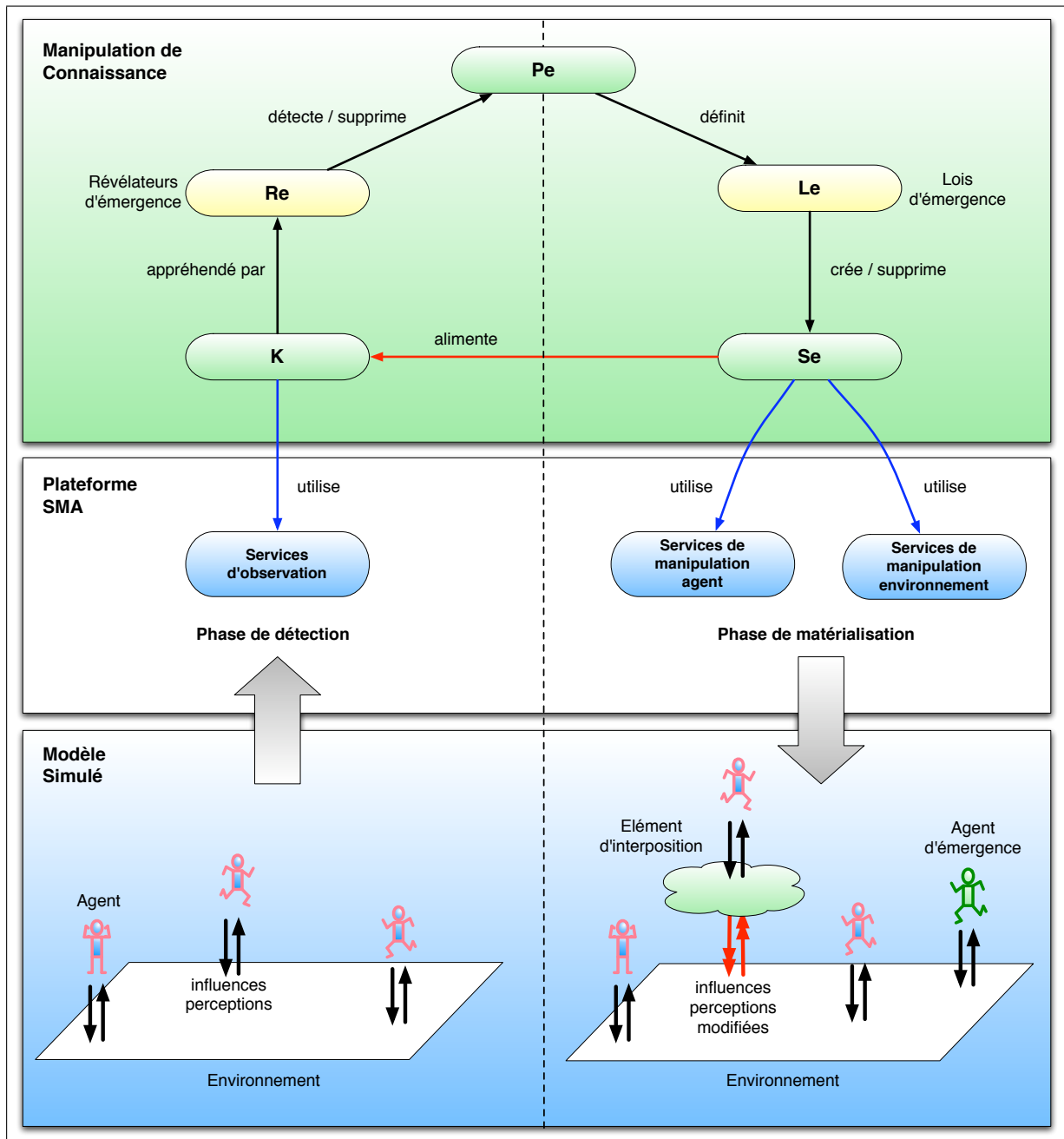


FIG. 5.7 – Architecture générale pour la réification de phénomènes émergents

fournis par *Pilobs*, le Pilote d'Observation de la plateforme. Ce pilote, dont le fonctionnement est détaillé dans [Ralambondrainy 2009], permet de centraliser l'accès aux sondes systèmes et fournit ainsi une interface unique entre l'extérieur du modèle et les sondes systèmes donnant accès au modèle simulé. Dans l'architecture de GEAMAS-NG (représentée sur la figure 5.8), *Pilobs* fait partie du moteur d'exécution, ce qui permet de garantir une acquisition d'éléments observables en phase avec l'exécution de la simulation. Ainsi, dès qu'un attribut est modifié, la sonde système associée notifie le changement à *Pilobs*. Un attribut désignant ici une variable du modèle simulé, sans lien direct avec le concept d'attribut en conception orientée objet.

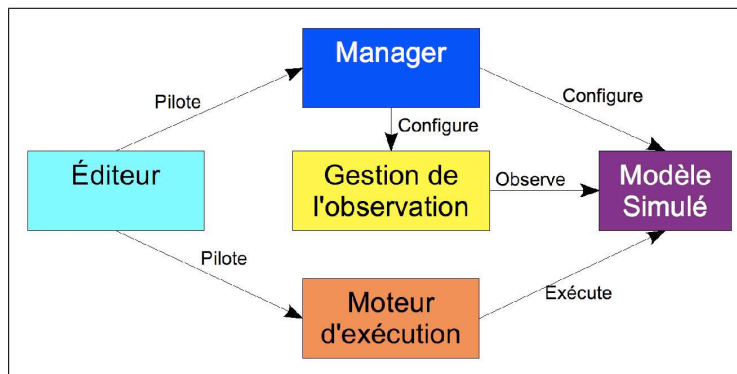


FIG. 5.8 – Architecture générale de GEAMAS-NG (extrait de [Ralambondrainy 2009])

Dans GEAMAS-NG, ces mécanismes permettent le suivi de l'état des variables utilisées par les environnements et les agents. Chaque fait qui se produit dans la simulation peut ainsi être défini par un couple (clé ; valeur) et la valeur de la clé peut elle-même dépendre de certaines variables. Dans le contexte de nos simulations spatialisées, la localisation géographique d'une entité pourrait ainsi être utilisée pour déterminer la valeur de la clé, ce qui peut s'avérer pratique pour observer des faits pour lesquels la relation de voisinage a son importance, comme nous le verrons dans les expérimentations décrites dans le dernier chapitre de ce manuscrit.

De plus, précisons que dans GEAMAS-NG toute la communication entre les agents se fait *via* un environnement de communication, et les interactions qui sont dans l'ensemble \mathcal{K}_A se font donc elles aussi *via* des mécanismes d'influences et de perceptions, et on a alors :

$$\mathcal{K}_A = \{influences, perceptions, interactions\} = \{influences, perceptions\}$$

On peut également s'interroger légitimement sur la présence des états des agents dans cet ensemble, donnant ainsi :

$$\mathcal{K}_A = \{etats, influences, perceptions, interactions\}$$

Ceci est parfaitement envisageable, mais dépend directement du degré d'intrusivité que l'on s'accorde dans les entités faisant partie du modèle de la SOA. Ceci dépend également des techniques de réifications qui seront choisies. Par exemple, dans le cas d'une réification mécanique ou automatique qui soit interne au modèle du domaine, les agents qui feraient leur propre introspection auraient naturellement accès à leur état, mais leur donner accès à celui des autres agents irait à l'encontre des principes élémentaires des SMA.

Par ailleurs, le fait de voir l'ensemble \mathcal{K} comme étant l'union des ensembles \mathcal{K}_A et \mathcal{K}_E est lui aussi dépendant des mécanismes d'influences et perceptions (puis réaction de l'environnement), mais aussi du respect d'autres principes agents qui sont en place dans GEAMAS-NG, comme la séparation entre l'état et les corps éventuels d'un agent. En effet, si ces principes ne sont pas respectés, et qu'ainsi par exemple toute volonté d'un agent se traduise par une modification directe de l'environnement, ou qu'un agent donné puisse avoir un accès direct aux autres agents du système, ou que l'environnement soit conçu comme un tout ayant une vision complète des entités de la simulation, il est probable que, selon les cas, \mathcal{K} puisse se réduire à l'étude de \mathcal{K}_A ou (exclusif) à celle de \mathcal{K}_E .

Enfin, notons que nous utilisons la possibilité offerte par GEAMAS-NG qui permet de définir plusieurs environnements au sein d'une même simulation (comme l'ont montré nos travaux sur la modélisation orientée dynamiques [Payet *et al.* 2006b]) afin d'utiliser un environnement particulier pour modéliser chaque élément d'interposition associé à une structure émergente. La séparation nette qui existe dans GEAMAS-NG entre les agents et leur(s) environnement(s) simplifie alors la mise en place de tels éléments d'interposition.

5.6 Conclusion

Dans ce chapitre, nous avons proposé un cadre formel et architectural pour faciliter la réification des phénomènes émergents dans une SOA. Cette proposition se place dans la continuité directe de la méthodologie de conception de simulations présentée au chapitre 4 et de la méthodologie de réification des phénomènes émergents présentée au chapitre 5. En particulier le cadre formel et architectural que nous venons de présenter ici permet d'apporter une solution de mise en œuvre aux techniques de réification mécaniques – qu'elles soient externes ou internes au modèle conceptuel du domaine – des phénomènes qui émergeraient dans une SOA.

Notre proposition s'appuie sur la définition, par les experts thématiques, de révélateurs d'émergence et de lois d'émergence qui permettent au système ou à ses entités de faire leur introspection afin de détecter les phénomènes qui apparaissent et qui sont ensuite matérialisés au cours d'une phase d'intercession. Ces révélateurs et lois d'émergence fonctionnent en opérant l'un sur l'ensemble \mathcal{K} des faits qui se produisent dans la SOA et l'autre sur l'ensemble \mathcal{P}_E qui se rattache aux phénomènes qui y auront été détectés.

La matérialisation effective des phénomènes est rendue effective grâce à deux types de structures d'émergence que nous avons introduites : des agents d'émergence et des éléments d'interposition. Les premiers permettent de représenter des phénomènes émergents qui nécessitent d'avoir un comportement propre au sein de la SOA, et les seconds permettent de modifier les influences et perceptions des entités déjà existantes dont le comportement se retrouve modifié de par la présence des phénomènes émergents matérialisés.

La description de cette démarche de réification nous a permis de faire ressortir les

exigences que nous devons avoir au niveau des plateformes de SOA que nous utilisons pour implémenter nos modèles afin que la prise en compte des phénomènes émergents s'en retrouve facilitée. Trois exigences principales se dégagent : la nécessité d'offrir des services d'observation de la simulation ; de donner accès à des services de création et de manipulation des agents de la SOA ; et de donner accès à des services de création et de manipulation des environnements de la SOA. Ces exigences raisonnables sous-entendent qu'il doit être possible de mettre en place des mécanismes similaires à ceux que nous avons décrits dans la plupart des plateformes de SOA existantes, même si l'on notera que la gestion du multi-environnements est un plus qui permet de faciliter la mise en place implémentatoire de structures comme les éléments d'interposition que nous avons proposés.

Ce chapitre vient clore la deuxième partie de ce manuscrit, et un exemple de mise en œuvre du cadre formel et architectural que nous avons présenté sera donné dans le chapitre 8 dans lequel nous présenterons essentiellement le fonctionnement interne du modèle DS et les expérimentations qui y sont associées.

Troisième partie

**Applications et outils pour la
prospective territoriale**

Le modèle DS : un outil pour la prospective territoriale à La Réunion

Sommaire

6.1	Introduction	117
6.2	La prospective territoriale : une nécessité à La Réunion	118
6.2.1	La Réunion : un territoire aux multiples contraintes	118
6.2.2	Des espaces naturels et agricoles mités par l'urbanisation	120
6.2.3	L'importance des documents de planification	120
6.3	L'ATP DOMINO du CIRAD	124
6.3.1	L'ATP DOMINO à La Réunion	124
6.3.2	Réalisations issues de l'ATP DOMINO	124
6.4	Le modèle DS	126
6.4.1	DS et les atouts de GEAMAS-NG	126
6.4.2	L'outil DS côté utilisateur	128
6.4.3	Sorties et résultats	133
6.4.4	Quelques résultats exploités régionalement	136
6.5	Conclusion	137

6.1 Introduction

En matière d'aménagement du territoire, l'île de La Réunion est confrontée au défi d'accueillir une population de plus en plus importante tout en valorisant son terroir agricole et en protégeant ses paysages exceptionnels. Dans un tel contexte, faire de la prospective territoriale, et en ce sens parvenir à modéliser et simuler des comportements spatialisés, prend une importance capitale.

Le modèle DS [David *et al.* 2007, Botta *et al.* 2009, Lagabrielle *et al.* 2010] que nous présentons ici est un modèle de simulation d'évolution des espaces fonciers à La Réunion. Implémenté sur la plateforme de simulation multi-agent GEAMAS-NG, il permet de simuler sur l'ensemble de l'île l'évolution de la population et les interactions entre les trois grandes classes d'utilisation du sol (espaces naturels, agricoles et urbains) tout en observant les évolutions induites.

Ce modèle repose sur une spécificité que nous avons élaborée : le couplage de plusieurs dynamiques comportementales liées, à l'échelle microscopique, à l'évolution démographique et, à l'échelle macroscopique, à l'évolution du mode d'occupation des sols. Cette spécificité sera détaillée dans le dernier chapitre de ce manuscrit.

Dans le présent chapitre nous revenons sur les circonstances qui font qu'il est aujourd'hui indispensable de faire de la prospective territoriale à La Réunion. Nous présentons ensuite le modèle de simulation DS comme étant un outil d'aide à la décision dont nous détaillons les principales fonctionnalités. Et nous présentons enfin des exemples de résultats qui ont pu être utilisés dans le cadre concret de la mise en révision du SAR (Schéma d'Aménagement Régional) par le Conseil Régional de La Réunion.

6.2 La prospective territoriale : une nécessité à La Réunion

6.2.1 La Réunion : un territoire aux multiples contraintes

La Réunion, île exiguë aux multiples contraintes, doit gérer aujourd'hui et dans les années à venir de nombreux phénomènes de croissance et leurs conséquences induites. Au cours des prochaines années ce territoire ultra-marin de 2500km² situé en plein océan Indien (cf. figure 6.1) devra en effet faire face à une augmentation de la population qui va passer de 800 000 à 1 million d'habitants dans les vingt années à venir [Actif & Lardoux 2006] ! Cette évolution démographique ouvre à elle seule la porte à de nombreuses questions, et notamment celle du logement : même avec des hypothèses fortes de densification de l'habitat, l'avenir de l'île se traduit par une augmentation inévitable de la demande de foncier urbain.

Dans une perspective de développement durable du territoire, la gestion du foncier réunionnais doit donc se faire selon une politique urbaine claire, car il est évident que la prolongation des tendances constatées ces dernières années en matière de consommation foncière (urbanisation galopante) n'est pas un scénario viable à long terme. Et ce d'autant plus que le foncier potentiellement exploitable ne couvre qu'une partie de l'île dont la géographie complexe (illustrée par les photos de la figure 6.2) combine pitons, cirques, remparts, volcan, rivières et côtes sauvages, soit autant de lieux où il est souvent difficile – et le plus souvent impossible – d'envisager d'habiter dans un pays où les menaces naturelles comme les fortes pluies, la houle, et les cyclones sont toujours présentes.

De plus, sur un territoire aussi restreint, différents enjeux sont à considérer au regard des conséquences de l'évolution démographique. En effet, ce ne sont pas moins de 100 000 hectares d'espaces naturels protégés et maîtrisés qu'il est nécessaire de conserver au mieux afin de

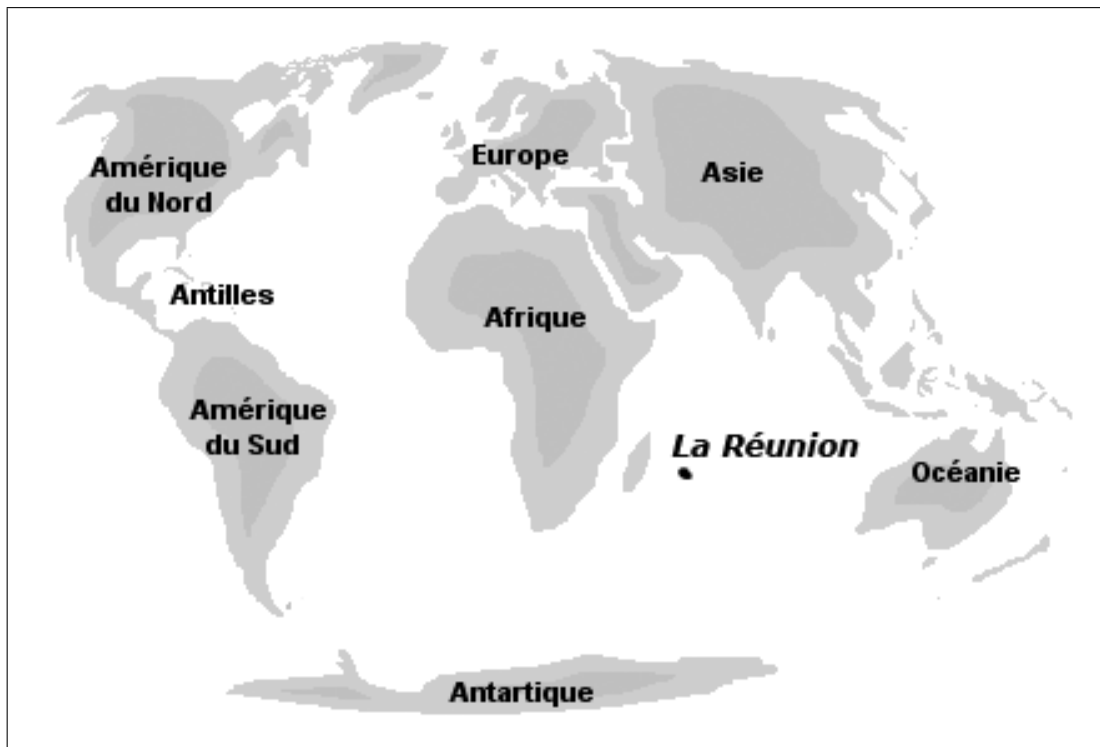


FIG. 6.1 – L'île de La Réunion dans le monde

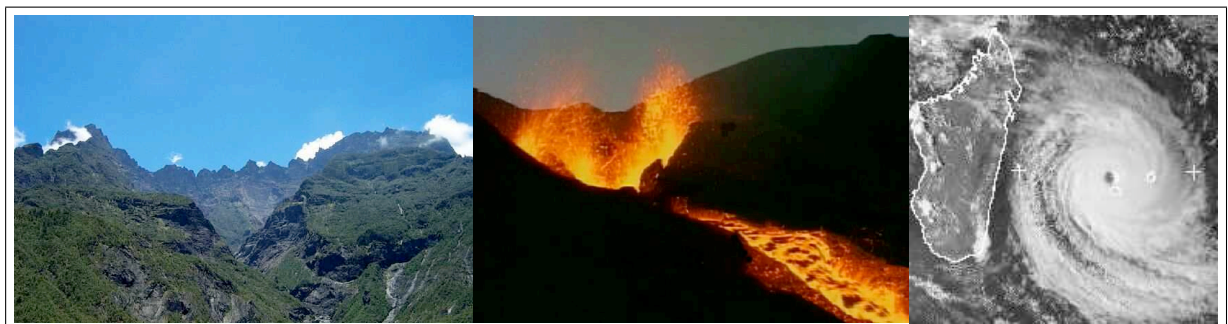


FIG. 6.2 – L'île de La Réunion : un relief tourmenté, un volcan, des cyclones... on ne peut pas s'installer partout! (photographies Observatoire Volcanologique et Météo France)

favoriser le maintien de la biodiversité qui fait la richesse de l'île. Afin d'y admirer encore longtemps une faune et une flore exceptionnelle il convient donc de tenir compte dans les futures politiques d'urbanisation de cet objectif de conservation de la biodiversité.

Par ailleurs, quelques 40 000 hectares du territoire réunionnais sont aujourd'hui occupés par une activité agricole qui constitue un des réels poumons économiques de l'île. Protéger ces espaces agricoles, notamment les plantations de canne à sucre, constitue donc une priorité supplémentaire, et il est important de veiller à freiner leur consommation par l'urbanisation en limitant les extensions urbaines et en empêchant le mitage des meilleures terres agricoles.

Comme rappelé dans [Agorah 2006], en matière d'aménagement du territoire la Réunion se doit donc de relever le défi d'accueillir une population de plus en plus importante tout en valorisant son terroir agricole et en protégeant ses espaces naturels et ses paysages exceptionnels.

Cette problématique, bien définie ici dans le cadre réunionnais, se retrouve dans nombre de pays aux caractéristiques proches. L'ensemble des régions du globe est ainsi soumis aux mêmes contraintes d'aménagement du territoire, et le lien est particulièrement fort que ce soit avec des îles du pacifique, des régions émergentes asiatiques, ou certaines zones africaines.

6.2.2 Des espaces naturels et agricoles mités par l'urbanisation

Conséquence directe de l'évolution importante de la population, les constructions et tout ce qui touche au domaine de l'urbanisation a littéralement explosé au cours des dernières années.

La Réunion est souvent connue pour ses beaux paysages, son volcan, ses plages... et en cette année 2010 elle a été classée au patrimoine mondial de l'UNESCO pour ses pitons, cirques et remparts. Mais il ne faut pas oublier qu'une grande partie des espaces réunionnais, essentiellement ceux des mi-pentes, ressemblent à la photo représentée sur la figure 6.3. Cette photo est particulièrement intéressante car sur une même zone géographique, très restreinte (à peine quelques hectares), on voit que sont présentes les trois grandes classes d'occupation du sol :

- des espaces naturels (en vert) : ici des "fonds de ravines" qui sont bien souvent havres de l'exceptionnelle biodiversité de l'île ;
- des espaces agricoles (en orange) : ici des champs de canne à sucre fraîchement coupés ;
- des espaces urbains (en rouge) : ici un ensemble de maison individuelles, de plein pied.

Ce sont les trois grands types d'espaces dont il est important de prévoir l'évolution dans les années à venir à La Réunion ; évolution que nous souhaitons donc simuler en fonction de différentes scénarios d'aménagement du territoire.

6.2.3 L'importance des documents de planification

Pour faire référence aux *Mutins de la Liberté* – un ouvrage évoquant la genèse d'un monde idéal encore dans ses balbutiements – dont nous avons cité un passage au début de ce manuscrit, il vient un moment où une île comme La Réunion ne peut plus être dirigée comme un grand bateau. De l'accroissement de la population et du besoin de contrôler l'urbanisation



FIG. 6.3 – Un paysage typique réunionnais : zones naturelles (en vert), zones agricoles (en orange), et zones urbaines (en rouge) coexistent sur un même territoire

dans l'île découle la nécessité de produire des documents de planification qui permettent de fixer les grandes orientations de l'aménagement du territoire réunionnais.

A La Réunion, il existe ainsi trois principaux types de documents de planification, dont la valeur et le domaine d'application sont fonction d'une échelle de représentation spatiale et administrative :

- le SAR, Schéma d'Aménagement Régional, qui concerne la région Réunion dans son ensemble ;
- les SCoT, Schémas de Cohérence Territoriale, qui concernent chacun l'ensemble des communes d'un même bassin de vie ;
- les PLU, Plans Locaux d'Urbanisme, qui concernent chacune des communes de l'île.

Le premier de ces documents de planification, le SAR, dont un schéma chorématique de la dernière version en date est représenté sur la figure 6.4, est un document qui précise les orientations fondamentales et les principes d'aménagement d'un territoire régional, ici l'île de La Réunion dans sa globalité, que ce soit en terme de développement, en terme de mise en valeur du territoire, ou en terme de protection de l'environnement. De tels schémas ont été institués dans des régions à statut particulier de par leur éloignement ou leur insularité, ce qui est le cas de La Réunion, comme de l'ensemble des Départements d'Outre-Mer français



FIG. 6.4 – Le Schéma d'Aménagement Régional

Les SCoT, eux, permettent aux communes d'un même bassin de vie de mettre en cohérence des politiques sectorielles comme l'habitat, les déplacements, ou les équipements commerciaux. A La Réunion il y a quatre SCoT différents, un pour chacune des micro-régions de l'île :

le SCoT de la CINOR¹, le SCoT Grand Sud, le SCoT du TCO², et le SCoT de la CIREST³.

Enfin, les PLU sont les principaux documents de planification de l'urbanisation essentiellement à l'échelle communale. Remplaçant des anciens POS, Plans d'Occupation des Sols, ils représentent l'espace local sous forme de cartes qui distinguent les zones naturelles et forestières, les zones agricoles, les zones urbaines, et les zones à urbaniser. A La Réunion chaque commune doit donc posséder son propre PLU, sur lequel elle s'appuiera par exemple pour décider (ou non) d'accorder les futures demandes de permis de construire.

Evidemment, ces différents documents, d'échelles différentes, sont en relation les uns avec les autres afin que les planifications locales soient en accord avec la planification régionale. Comme représenté sur la figure 6.5, le SAR s'impose ainsi aux différents SCoT, qui eux-mêmes s'imposent aux différents PLU, et l'ensemble permet de rendre plus claires et plus cohérentes les politiques d'urbanisation menées dans l'île.

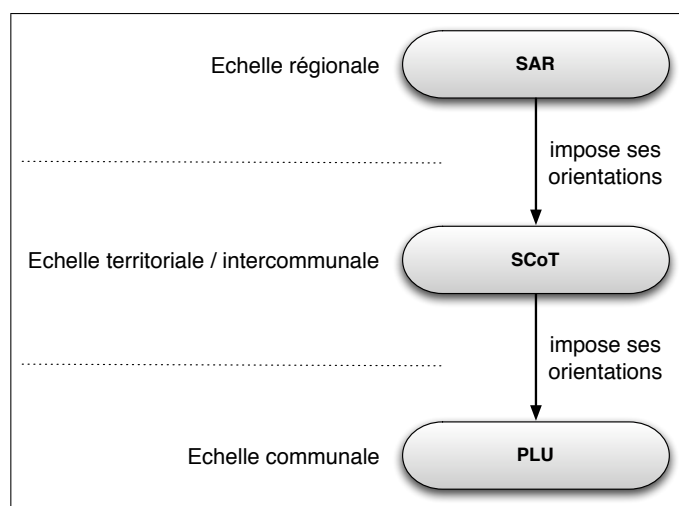


FIG. 6.5 – Articulation entre le SAR, les SCoT, et les PLU

L'idée pour nous en faisant des outils de simulation est de fournir des outils d'aide à la décision à de différents acteurs, qui peuvent être des chercheurs, des agents de terrain, mais aussi des décideurs politiques en charge de l'élaboration des documents de planification que nous venons de présenter. A ce niveau, l'utilisation d'outils de simulation s'avère par ailleurs être un bon élément pour ouvrir le dialogue, puisque les simulations permettent par exemple de mettre en avant des incohérences qui existent entre différents documents de planification d'échelles différentes (par exemple des zones qui seraient considérées comme étant "à urbaniser" d'après le SAR alors qu'elles ne le seraient pas selon les PLU concernés, qui doivent alors être mis en conformité).

¹Communauté Intercommunale du NOrd de la Réunion

²Territoire de la Côte Ouest

³Communauté Intercommunale Réunion EST

6.3 L'ATP DOMINO du CIRAD

Le projet DOMINO (Démarche Objet Multisite pour l'étude des Interactions entre Niveaux d'Organisation) est un projet piloté par le CIRAD de 2005 à 2007 dans le cadre d'une Action Transversale Programmée (ATP). Coordonné par Aurélie Botta, William's Daré et Grégoire Leclerc, il a été mené conjointement à La Réunion et au Sénégal, ces deux sites étant confrontés à des problématiques démographiques similaires dans des contextes géographiquement inversés : La Réunion est entourée d'eau, alors que la région du Sénégal étudiée constituait tout le pourtour d'un lac.

6.3.1 L'ATP DOMINO à La Réunion

A La Réunion, ce projet a été réalisé en collaboration avec l'Université de la Réunion, le Comité de pilotage de la canne, l'Association pour la Promotion en milieu Rural et la Chambre d'Agriculture de la Réunion. Et l'un de ses buts était de parvenir au développement d'outils d'aide à la décision publique sur les choix de règles d'affectation des terres.

En tant qu'informaticiens et modélisateurs, nous sommes intervenus dans ce projet pour apporter notre appui au développement de ces outils et pour formuler les concepts informatiques nécessaires à l'implémentation des modèles, facilitant ainsi le passage du modèle conceptuel du domaine à sa version implémentée.

Une collaboration entre conseillers et techniciens de l'aménagement du territoire et scientifiques a conduit à la production de modèles simulant les interactions entre dynamiques sociale et environnementale. Ces modèles représentant les liens impliqués dans les choix d'affectation des terres à différents niveaux d'organisation doivent permettre de tester la cohérence de différents scénarios prospectifs et ont vocation à être appropriés et exploités par les décideurs comme des outils d'aide à la décision.

C'est dans le cadre de ce projet qu'ont été élaborés, en suivant une démarche de modélisation participative, plusieurs outils que nous allons présenter dans la section suivante.

6.3.2 Réalisations issues de l'ATP DOMINO

Nous présentons ici brièvement quelques outils qui sont les fruits de l'ATP DOMINO. Un seul de ces outils, le modèle DS, résulte des travaux directs menés dans l'équipe SMART, mais les autres méritent d'être présentés rapidement puisqu'ils s'inscrivent dans le contexte global et participatif du projet DOMINO. D'autre part ces différents outils, quand ils ne servent pas à s'alimenter entre eux de façon directe (comme on le verra par la suite avec les cartes de l'Atlas DOMINO qui sont utilisées pour initialiser DS), servent tout de même à alimenter ensemble la réflexion de fond sur l'avenir des territoires réunionnais.

L'Atlas DOMINO

L'Atlas DOMINO est un recueil de couches d'informations cartographiques géographiques. Sous la forme de données SIG, sont ainsi regroupées des données qui serviront (entres autres) pour initialiser nos simulations. Un aperçu de plusieurs cartes issues de cet Atlas sera donné au chapitre suivant.

Metronamica Réunion

Le modèle Metronamica Réunion, dont l'interface est représentée sur la figure 6.6, est un automate cellulaire qui a été créé afin d'explorer la dynamique d'évolution du sol à La Réunion. Cet automate cellulaire a été développé par Gilles Lajoie qui a utilisé la plateforme Geonamica [White & Engelen 1993, White *et al.* 1997] qui est développée par l'équipe néerlandaise du Research Institute for Knowledge Systems.

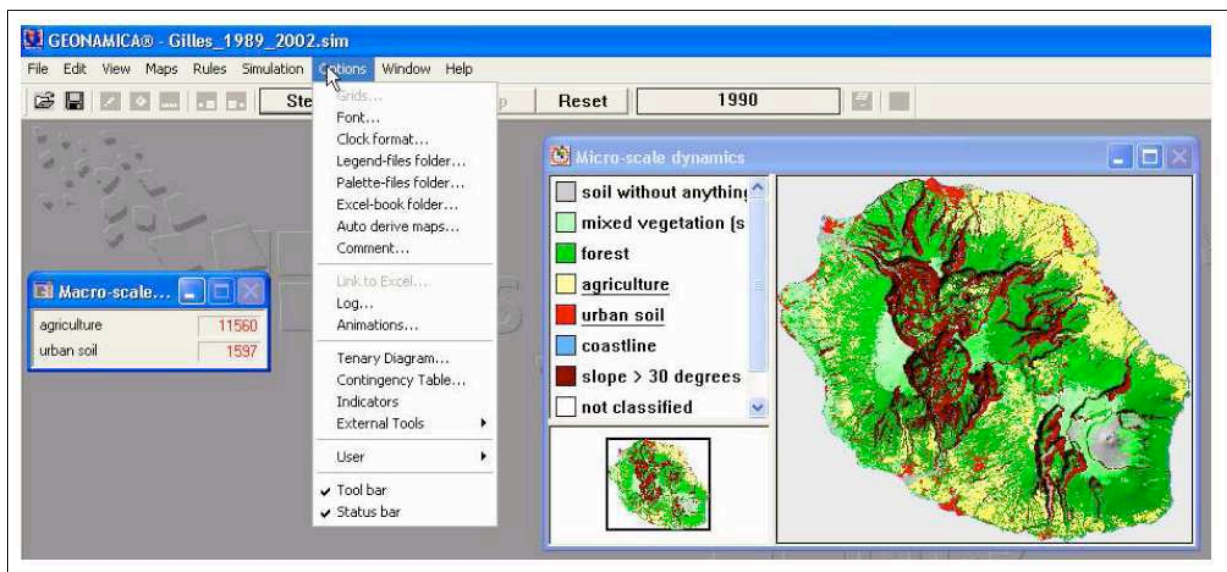


FIG. 6.6 – Le modèle Metronamica Réunion

Ce modèle repose sur des mécanismes de changement d'états des cellules en fonction de leurs potentiels de transitions. Comme souligné dans [Lajoie & Hagen-Zanker 2007], il permet de simuler différents scénarios d'évolution de l'espace en faisant varier le statut des terres ou en fixant les objectifs à atteindre en fin de simulation en terme de surface. Très bon outil pour ouvrir le dialogue, la sensibilité de sa paramétrisation le rend cependant assez complexe à exploiter pour des non-spécialistes qui seront parfois confrontés à l'apparition d'aberrations spatiales dans leurs simulations.

Le Modèle DS

Comme nous l'avons évoqué au début de ce chapitre, le modèle DS permet de réaliser sur l'ensemble de l'île des simulations relatives aux interactions entre les trois grandes classes d'utilisation du sol (les espaces naturels, agricoles et urbains) et d'observer les évolutions qu'elles induisent. Les premières versions du modèle DS ont été réalisées dans le cadre du projet DOMINO, et d'autres versions plus spécifiques ont vu le jour par la suite. Grâce aux travaux de S. Manglou une version DS-SAR a par exemple été réalisée pour coller le plus possible aux demandes de la Région. Et d'autres versions sont utilisées en interne par chacune des équipes de chercheurs ayant contribué à l'élaboration du premier DS pour réaliser leurs expérimentations diverses. Au sein de l'équipe SMART nous l'utilisons donc pour mener des essais implémentatoires liés à nos recherches (sur l'observation, la distribution, ou la prise en compte de l'émergence), mais aussi à des fins pédagogiques avec les étudiants que nous avons en cours ou que nous suivons en stages.

6.4 Le modèle DS

Le modèle DS (qui porte ce nom en référence aux initiales de DOMINO-SMAT, contraction du nom du projet et de celui d'un prototype, SMAT, issu des travaux précédents de Denis Payet et Gilles Lajoie) a été implémenté en langage Java sur la nouvelle version de la plateforme de simulation multi-agents GEAMAS-NG.

Dans le présent chapitre nous présentons essentiellement les aspects liés à création et à l'utilisation de DS ainsi qu'à la valorisation que cet outil nous a apporté. Le fonctionnement interne du modèle sera détaillé au chapitre 8, dans lequel nous ferons le lien entre la façon dont a été conçu DS et la méthodologie de conception de simulations que nous avons présentée au chapitre 3, et dans lequel nous présenterons une expérimentation visant à prendre compte un phénomène émergent lié à l'urbanisation dans l'île, mettant ainsi en œuvre les propositions décrites aux chapitres 4 et 5.

6.4.1 DS et les atouts de GEAMAS-NG

Les possibilités fournies par l'utilisation d'une plateforme de simulation plutôt qu'une autre conditionnent souvent certains choix de modélisation tels l'ambition de pouvoir travailler avec une échelle de simulation suffisamment fine (peut-on découper le territoire en très petites entités ?) sur une zone suffisamment large (peut-on travailler à l'échelle de l'île entière, soit 2500km² ?).

Pour implémenter DS nous avons fait le choix de GEAMAS-NG comme plateforme de simulation, car son utilisation se révélait avantageuse relativement à trois besoins que nous avons identifiés :

- pouvoir être utilisée dans le cadre d'une démarche de modélisation pluridisciplinaire ;
- permettre une importante montée en charge ;

- offrir des procédés d’initialisation souples, pertinents, et visuellement expressifs.

Le choix de cette plateforme était d’autant plus naturel puisqu’étant nous mêmes chargés de son développement cela nous permettait d’avoir un retour direct sur les expérimentations et sur son utilisation, facilitant ainsi tous les ajustements nécessaires. Le choix de GEAMAS-NG a donc permis d’améliorer la conception de DS ; et en retour la conception de DS a contribué à améliorer GEAMAS-NG.

Une modélisation pluridisciplinaire

Le modèle DS ayant été conçu dans le cadre d’une modélisation participative il était important de pouvoir implémenter un modèle réalisé par des thématiciens se préoccupant de dynamiques diverses. De nombreux travaux soulignent l’importance de l’environnement dans les systèmes multi-agents [Odell *et al.* 1997], mais GEAMAS-NG a la particularité d’offrir à ce niveau une séparation claire avec les agents. Ainsi, comme l’ont montré les travaux précédents menés sur la modélisation orientée dynamique [Payet *et al.* 2006b], et dans le cadre d’une démarche méthodologique comme celle présentée dans la deuxième partie de ce manuscrit, notre objectif de modélisation et de couplage de dynamiques est ainsi rendu possible.

Nous reviendrons sur certains de ces aspects dans le dernier chapitre de ce manuscrit dans lequel nous nous focaliserons sur le fonctionnement interne de DS, et dans lequel nous détaillerons donc les techniques que nous avons utilisées pour coupler des dynamiques qui ont pu être pensées et réfléchies de façon relativement indépendante.

Un procédé d’initialisation souple, pertinent, et visuellement expressif

Suivant toujours la logique d’une modélisation qui soit la plus ouverte possible établie avec les thématiciens qui interviennent dans la conception des modèles, il est important de pouvoir exploiter sous leur meilleure forme (qui peut selon les cas être la plus pratique ou la plus parlante) les données qui sont nécessaires pour initialiser le modèle. Les travaux menés en ce sens sur GEAMAS-NG nous font bénéficier de possibilités avantageuses au niveau de l’initialisation des simulations, en permettant par exemple de charger directement des données en utilisant un jeu de cartes colorées sémantiques (exemple : une carte de la répartition initiale de la population sur le territoire) qui nous servent de plans d’information et qui sont utiles dans le cadre d’un environnement spatialisé.

Nous reviendrons de manière exclusive sur ce dernier point dès le prochain chapitre dans lequel nous présenterons les avantages de l’initialisation par cartes couplée au langage XELOC.

Une importante montée en charge

Lorsque l’on s’intéresse à l’évolution du territoire (avec ici le cas concret Réunionnais), il est *a priori* préférable de travailler à l’échelle la plus fine possible, ce qui croît inexorablement le nombre d’agents présents dans le système. Les travaux menés sur le moteur à temporalité [Payet *et al.* 2006a] présent dans GEAMAS-NG nous permettent d’obtenir des performances acceptables en terme de temps de traitement. La Réunion faisant 2500km², nous faisons ainsi

fonctionner sur une machine locale des simulations dans lesquelles évoluent jusqu'à 250 000 agents, puisque notre nombre d'agents est égal au nombre d'hectares de terre de l'île. Ceci justifie donc nos choix d'implémentation pour ce projet par rapport aux autres plateformes de simulation du domaine comme CORMAS, pourtant largement utilisée dans des modélisations tournant autour de problématiques liées à l'aménagement du territoire [Bousquet *et al.* 2007], ou comme la précédente version de GEAMAS encore utilisée dans des projets comme BIOMAS [Courdier *et al.* 2002].

Pour autant, la performance ne doit pas se faire au détriment de la cohérence. En effet, rares sont les simulations que nous avons faites avec effectivement 250 000 agents, puisque certaines des données que nous utilisons, et notamment certaines des cartes présentées au chapitre suivant, étaient cohérentes avec une précision de 2ha, et non pas de 1ha, ce qui rendait donc caduques les simulations réalisées avec une échelle de précision trop fine.

Les aspects techniques liés au côté "large échelle" de nos simulations ne seront pas détaillés dans ce manuscrit, car ils ont fait l'objet de travaux récents dans notre équipe au sein de laquelle les travaux de Tiana Ralambondrainy sur l'observation de simulations large échelle et ceux de Nicolas Sébastien sur la distribution et la parallélisation de simulations ont fait suite aux travaux initiaux de Denis Payet sur le moteur à temporalité.

6.4.2 L'outil DS côté utilisateur

DS : vue d'ensemble

L'interface de l'outil DS est représentée sur la figure 6.7. De façon relativement classique, quatre zones différentes sont identifiables :

- au milieu, une zone de visualisation dans laquelle on reconnaît l'île de La Réunion, et qui permet d'afficher des indicateurs colorés qui donnent une impression visuelle d'ensemble de l'évolution des différents paramètres choisis par l'utilisateur ;
- en haut à gauche, une "minimap" qui permet de choisir en "drag & drop" la zone de l'île qui sera affichée dans la vue centrale précédente ;
- sur la droite, un ensemble d'onglets qui permettent de configurer la simulation, de choisir différents paramètres ou scénarios, et de lancer la simulation ;
- en bas, un ensemble d'informations qui sont disponibles et qui sont relatives aux agents qui seront sélectionnés et dont l'utilisateur souhaite avoir le détail.

Scénarios de simulation et interfaces indispensables

Nous ne détaillerons évidemment pas ici l'intégralité de l'outil et de ses possibilités, mais nous présenterons quelques interfaces utilisateur qui sont utilisées pour définir des scénarios de simulation. La diversité des paramètres qui sont modifiables dans ces interfaces illustre de façon pertinente le côté pluridisciplinaire de la conception du modèle DS.

La figure 6.8 met en avant les deux onglets d'une interface qui permettent de choisir différentes options d'urbanisation en fonction des quatre micro-régions réunionnaises dont les

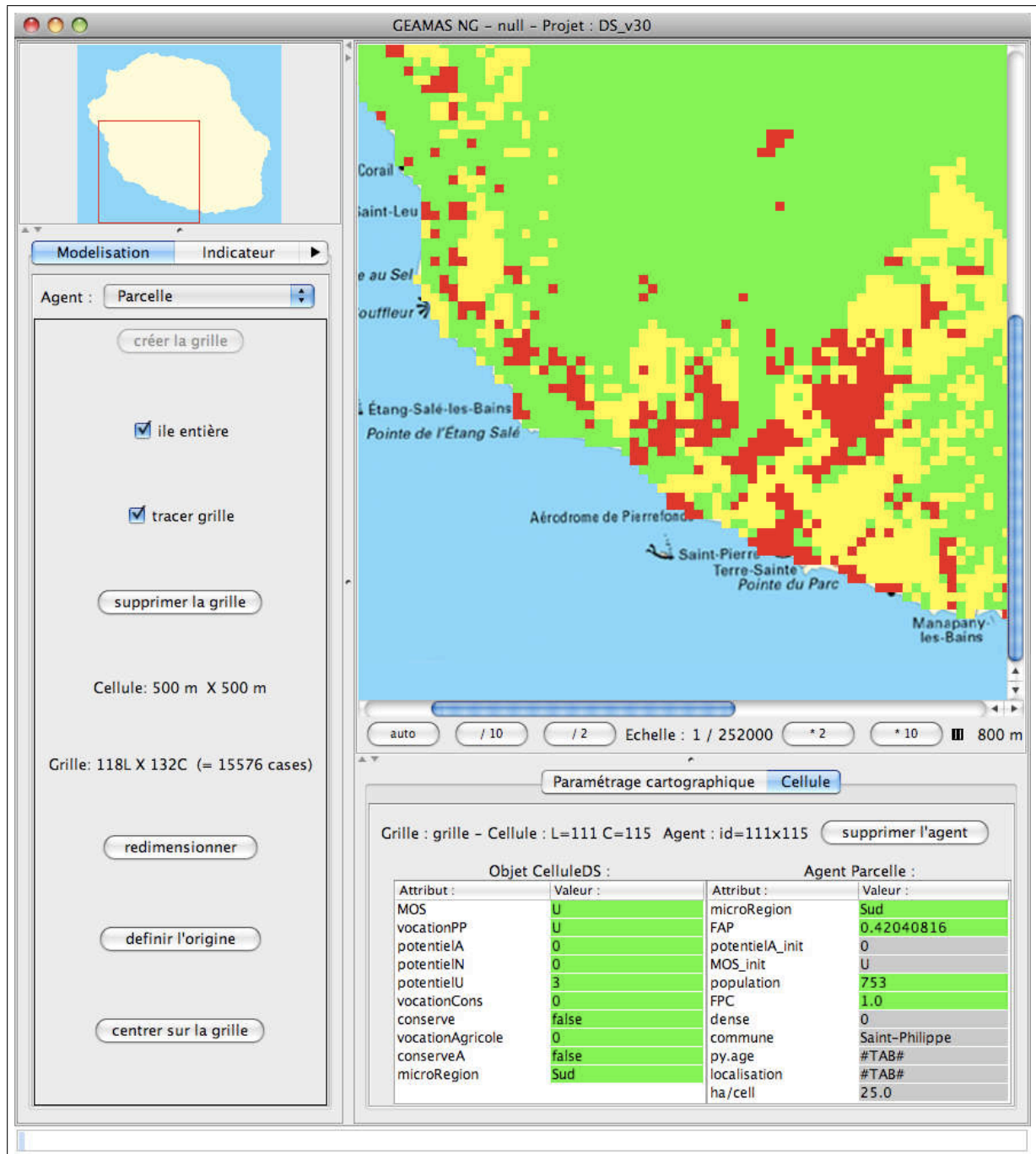


FIG. 6.7 – La principale interface utilisateur de DS

évolutions sont planifiées au sein des quatre SCoT (Schémas de Cohérence Territoriaux) correspondants. Ce paramètre doit absolument être pris en compte lors des simulations. En effet, et par exemple, au cours des dernières années si la côte Ouest, ensoleillée, et proche des plages, a été fortement attractive et a donc été fortement urbanisée, elle se retrouve aujourd’hui bien plus saturée que ne l’est la côte Est de l’île. Ainsi il est donc raisonnable d’imaginer qu’il faut tenir des politiques de densification de l’habitat différentes dans ces deux micro-régions, l’étalement urbain étant encore davantage possible dans l’Est qu’il ne l’est sur le littoral de l’Ouest.

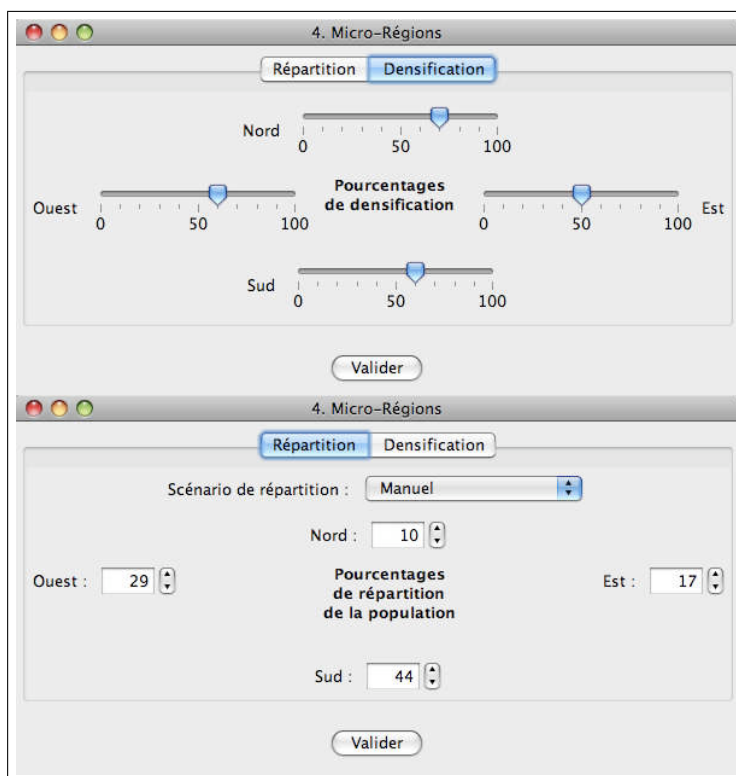
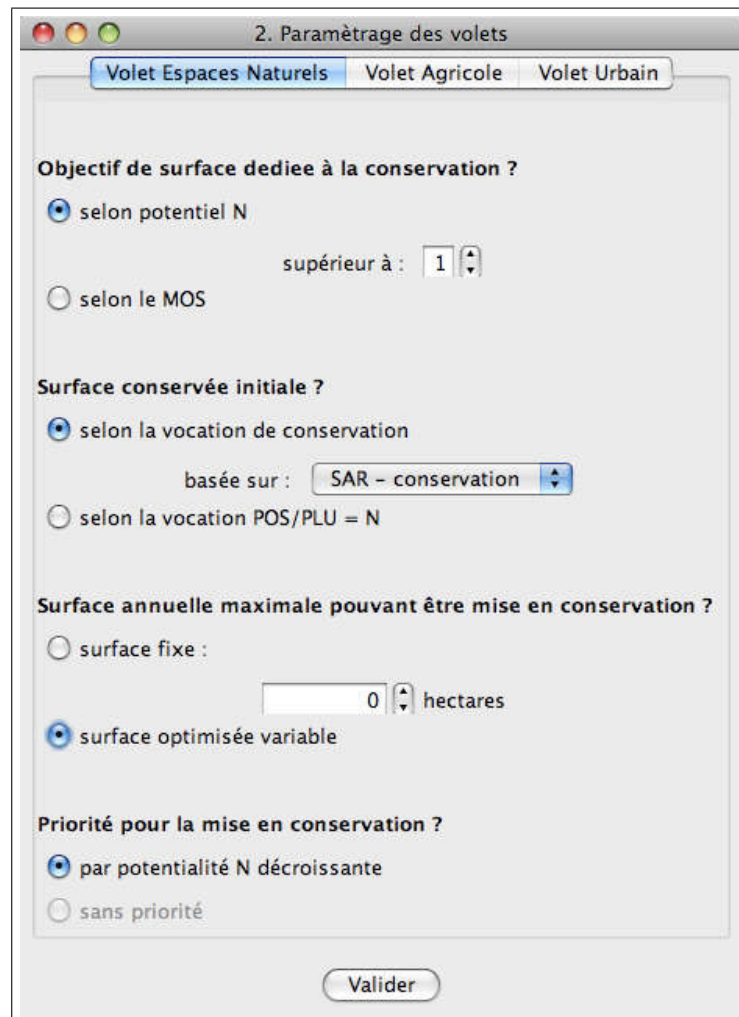


FIG. 6.8 – La répartition de la population et la densification de l’habitat dans les micro-régions

Sur la figure 6.9 on voit trois onglets qui font chacun référence aux paramètres configurables pour les 3 secteurs d’utilisation du sol : les espaces naturels, agricoles ou urbains. Grâce à cette interface les utilisateurs experts d’un secteur particulier peuvent venir expérimenter différents paramètres d’évolution sans toucher aux réglages des autres volets.

Enfin, sur la figure 6.10, l’utilisateur peut choisir les interactions qui sont possibles entre les trois grands types d’utilisation du sol. Sur l’exemple de la figure, il est ainsi permis qu’au cours de la simulation des espaces qui ont a priori une vocation naturelle puissent être mis en culture ou être urbanisés, de même qu’il est permis que des espaces ayant une vocation agricole puissent être urbanisés par la force des choses. Cette interface est essentielle pour permettre de créer des scénarios de simulation dans lesquels on souhaiterait par exemple préserver les espaces naturels actuels. Il est également permis ici de choisir si lors de la simulation les évolutions de chacun des trois grands types d’espaces doivent se faire (ou non) en fonction des orientations qui sont celles indiquées dans les PLU (Plans Locaux



The image shows a software window titled "2. Paramétrage des volets" with three tabs: "Volet Espaces Naturels" (selected), "Volet Agricole", and "Volet Urbain". The window contains several configuration sections:

- Objectif de surface dédiée à la conservation ?**
 - selon potentiel N
 - supérieur à : 1
 - selon le MOS
- Surface conservée initiale ?**
 - selon la vocation de conservation
 - basée sur : SAR - conservation
 - selon la vocation POS/PLU = N
- Surface annuelle maximale pouvant être mise en conservation ?**
 - surface fixe : 0 hectares
 - surface optimisée variable
- Priorité pour la mise en conservation ?**
 - par potentialité N décroissante
 - sans priorité

A "Valider" button is located at the bottom center of the window.

FIG. 6.9 – La configuration du volet naturel

d'Urbanisme). Ceci permet de tester si l'évolution des territoires de l'île se fait bien en accord avec la réglementation définie dans les documents de planification. Mais ceci permet aussi de tester si ces mêmes documents de planification ne sont pas trop restrictifs, ou de mettre en avant des orientations qui seront difficiles à mettre en œuvre dans le monde réel si l'on constate des impossibilités d'évolution dans les résultats obtenus en simulation.

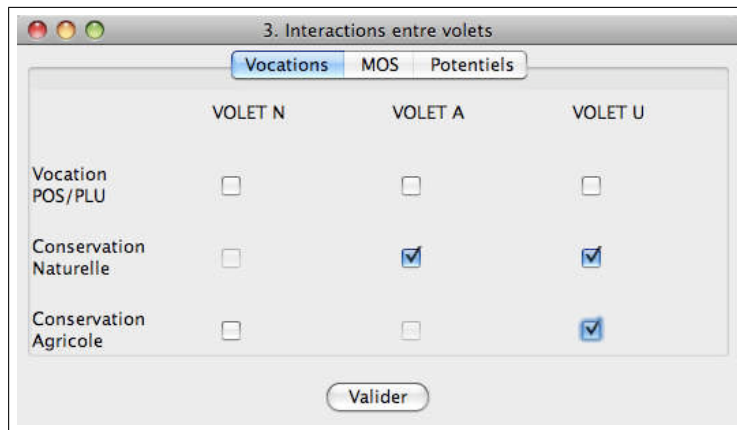


FIG. 6.10 – Les interactions entre les volets naturel, agricole, et urbain

L'ensemble des choix qui peuvent être effectués dans ces interfaces par les utilisateurs de DS permettent de définir des scénarios de simulation. Ces scénarios peuvent être sauvegardés, et les scénarios existants peuvent ainsi être chargés au lancement de l'outil pour limiter le côté fastidieux qui existe si l'utilisateur doit à chaque fois modifier un ensemble de paramètres. Quelques scénarios sont ainsi pré-définis (favorisant le développement des villes, favorisant le développement agricole, etc.), et donnent la possibilité à un utilisateur donné, qui serait par exemple un expert dans le domaine de la conservation de la biodiversité, de charger un scénario favorisant le développement des villes, chargeant ainsi tout un ensemble de paramètres et de règles d'évolutions qui sont hors de son domaine d'expertise (par exemple les critères de densification ou d'étalement de l'habitat). Cet utilisateur peut alors faire l'impasse sur la configuration fastidieuse d'éléments hors de sa portée pour se concentrer sur la configuration d'éléments en lien direct avec son expertise (par exemple la délimitation de tel ou tel espace de conservation dans les Hauts de l'île) afin de lancer ses simulations.

Simulation

Une fois un scénario de simulation défini, l'utilisateur choisit une échelle qui servira à découper le territoire en parcelles de terres (parcelles de 1ha ? de 2ha ? de 10ha ? ...) et il peut ensuite lancer sa simulation. Pour cela il fournit à DS les dates de début et de fin de la simulation par rapport au temps réel. Ceci se fait via une interface, représentée sur la figure 6.11, qui permet évidemment de lancer et/ou d'arrêter une simulation donnée. Sur le plan strictement informatique, les dates qui sont entrées comme étant celles de début et de fin de la simulation

servent simplement à définir un nombre de cycles de simulation qui devront être effectués. La cohérence de ces dates par rapport au temps réel dépend là directement des données qui sont utilisées en entrée pour initialiser le modèle. Par exemple sur la figure 6.11 on considère que la simulation démarre en 2003, car c'est la date pour laquelle sont cohérentes les données et cartes fournies en entrée de cette simulation.

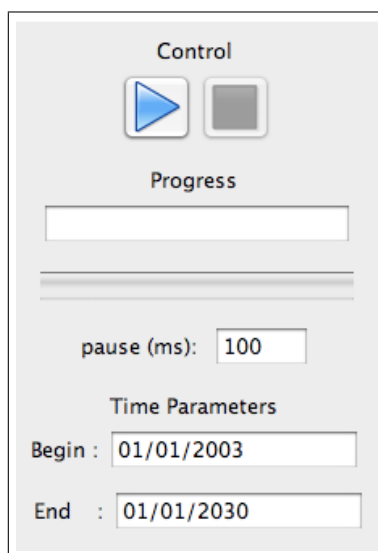


FIG. 6.11 – L'interface de lancement des simulations où se spécifient les plages temporelles

Le fait de faire saisir des dates de début et de fin de simulation, plutôt que de faire saisir une simple durée temporelle (par exemple 30 ans de simulation), permet par ailleurs de rappeler de façon indirecte aux utilisateurs ce critère de cohérence et de validité des données qu'ils ont parfois tendance à oublier. Cela montre aussi que DS n'est pas restrictif par rapport à l'aspect temporel, et qu'il est ainsi possible de lancer des simulations montrant l'évolution des territoires réunionnais des années 1950 à nos jours, en admettant évidemment que les données de l'époque soient disponibles.

Une fois la simulation lancée, la dynamique du modèle se met en place. Le territoire est découpé selon l'échelle choisie par l'utilisateur et celui-ci peut suivre en direct l'évolution aux différents niveaux (de la région en général à chaque parcelle de terre en particulier) d'un ensemble d'indicateurs, les principaux étant évidemment la densité de population et le MOS (Mode d'Occupation des Sols).

6.4.3 Sorties et résultats

Dans l'application DS trois types de sorties sont privilégiées :

- des fichiers de données ;
- des graphiques d'évolution ;
- des représentations cartographiques (fichiers images et exports SIG).

Cela permet de fournir un plus large panel de résultats exploitables en fin de simulation. Chaque type de sortie peut ensuite être exploité préférentiellement par les thématiques géographes, économistes, statisticiens, etc. qui s'intéressent à l'avenir de leurs territoires.

Fichiers de données

Des données chiffrées (regroupées dans des fichiers de logs) permettent de se donner une idée globale des changements qui interviennent au niveau de l'occupation de l'espace, par exemple en donnant la valeur totale en hectares des zones agricoles de l'île de la Réunion, ou en donnant la valeur de chacune des demandes insatisfaites des agents superviseurs.

De façon classique ces données sont ensuite utilisées par les thématiques et/ou par les décideurs pour produire, grâce à différents jeux de simulations, des ensembles de statistiques illustrant les futurs possibles de l'île.

Graphiques d'évolution

Des sorties graphiques permettent de fournir à l'utilisateur un rapide aperçu de l'évolution d'une donnée particulière, par exemple l'évolution de la population totale de l'île au cours de la simulation, ou celle détaillée pour chacune des micro-régions comme le montre la figure 6.12. Sur ce graphique d'évolution on voit cinq courbes différentes : la première (la plus haute)

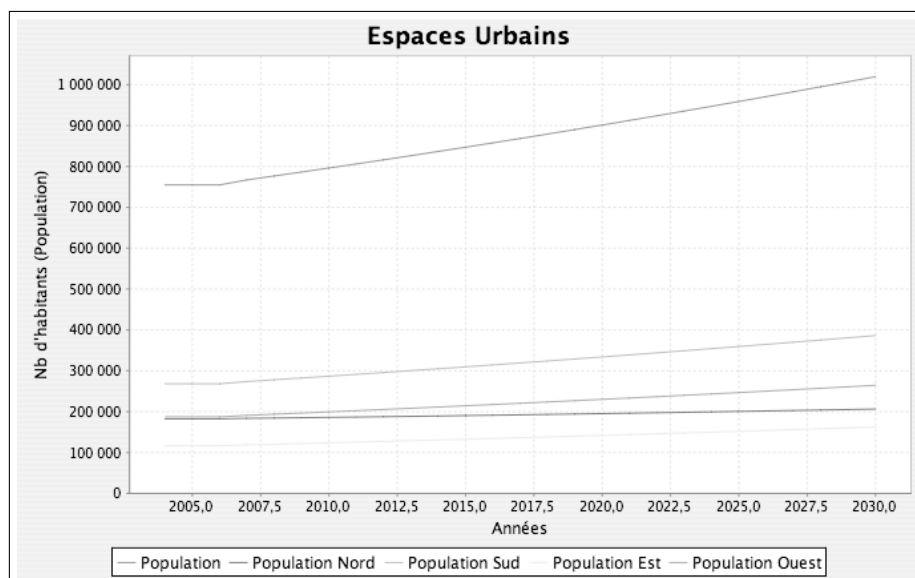


FIG. 6.12 – L'évolution de la population : de 750 000 habitants en 2003 à 1 million en 2030

représente l'évolution totale de la population réunionnaise, et les quatre autres représentent l'évolution de la population dans chacune des quatre micro-régions de l'île.

Ce type de sortie nous est particulièrement utile pour jauger rapidement les tendances d'évolution qui se dégagent dans l'une ou l'autre des dynamiques comportementales.

Les données présentes sur ce graphique, issu d'une simulation favorisant le développement des villes, nous servent également, dans une certaine mesure, d'élément de validation. En effet, on constate dans les résultats issus de nos simulations que la population totale de l'île aura dépassé le million d'habitants dans les années 2030, ce qui est en parfaite cohérence avec les projections de l'INSEE⁴ pour notre région comme on peut le voir sur les données de la figure 6.13.

		EST	NORD	OUEST	SUD	LA RÉUNION (*)
Population	1999	101 700	176 400	179 800	248 300	706 200
	2000	103 400	178 000	182 900	252 500	716 800
	2005	113 000	187 000	200 300	275 500	775 900
	2010	122 300	194 900	217 200	298 000	832 500
	2015	131 700	201 600	234 000	320 600	887 900
	2020	141 100	206 900	250 200	342 500	940 600
	2025	149 900	210 400	265 000	362 900	988 300
	2030	158 100	212 100	278 000	380 900	1 029 100
Augmentation, 1999 à 2030		56 400	35 700	98 200	132 600	322 900
Augmentation 1999 à 2030 (%)		55,5	20,2	54,6	53,4	45,7
Poids des zones en 2030 (%)		15,4	20,6	27,0	37,0	100,0
Répartition de l'augmentation (%)		17,5	11,1	30,4	41,1	100,0

FIG. 6.13 – Les projections de l'INSEE pour l'évolution de la population entre 1999 et 2030 (extrait de [Actif & Lardoux 2006])

Représentations cartographiques

DS permet d'obtenir deux types de sorties cartographiques :

- des cartes sous forme de fichiers images ;
- des exports vers des fichiers SIG.

Cartes avec indicateurs graphiques

Des cartes sauvegardées sous forme de fichiers images (plutôt des fichiers .bmp que des fichiers .jpg pour offrir la meilleure résolution exploitable possible) permettent, en se plaçant sur un indicateur coloré choisi, de se faire une idée précise des zones de l'île de la Réunion qui ont changé de type d'occupation (cf. figure 6.14). Ainsi, dans un scénario où la population de l'île augmente, on verra logiquement croître les étendues colorées correspondant aux zones urbanisées au détriment des zones colorées correspondant aux zones naturelles ou agricoles. Sur la figure présentée ici les cases de la grille qui a permis le découpage du territoire et la création des agents font 1 km de côté, ce qui permet d'illustrer (par exemple en regardant sur la

⁴Institut National de la Statistique et des Etudes Economiques

figure la très faible proportion de zones urbaines dans l'intérieur des terres) les incohérences liées à la validité des données dont nous parlions plus tôt.

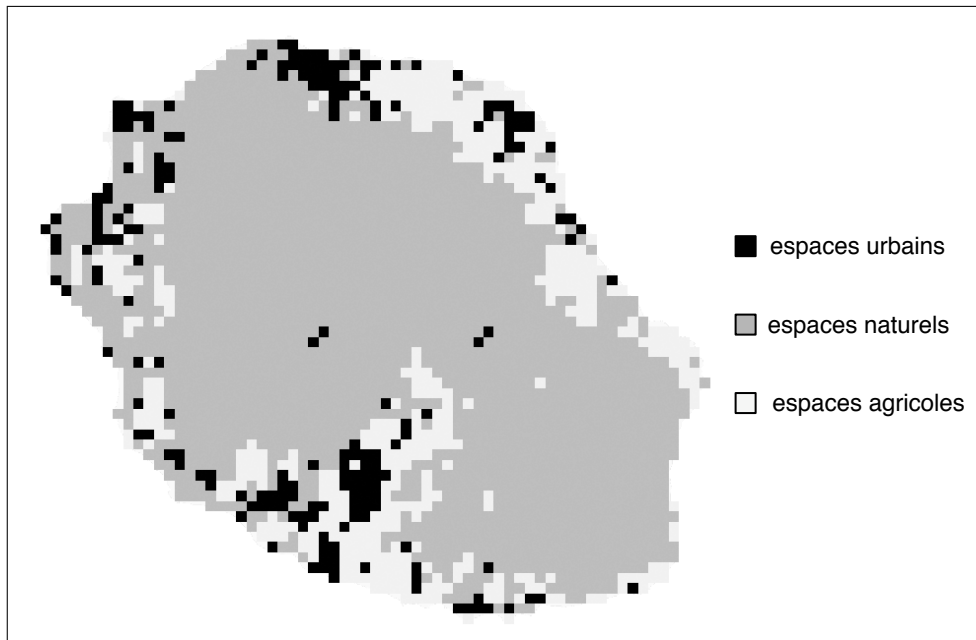


FIG. 6.14 – Un exemple de sortie cartographique (extrait de [David *et al.* 2007])

Lien avec un SIG

L'état courant d'une simulation peut être sauvegardé de sorte à rendre les résultats exploitables par des Systèmes d'Information Géographique. Le système de sauvegarde est basé sur l'utilisation de l'API Geotools [Custer 2005], et le fichier shapefile généré est un ensemble de points correspondant aux cellules de DS. Ceci permet de faciliter encore plus les interactions avec les thématiciens, souvent géographes lorsque l'on travaille dans le domaine de l'aménagement du territoire, qui sont familiers des systèmes SIG et qui peuvent récupérer les données issues de simulations avec ArcGIS ou MapInfo.

6.4.4 Quelques résultats exploités régionalement

Une des réussites du projet DOMINO fut la collaboration qui a pu être tissée en 2007 avec la cellule SAR du Conseil Régional de La Réunion.

Lors de cette collaboration avec la cellule SAR, nous avons donc fait différents aménagements à l'outil DS, et nous l'avons utilisé pour produire des simulations dont les scénarios seraient en accord avec les quatre scénarios d'aménagement globaux envisagés pour l'île à l'époque :

- un scénario tendanciel, au fil de l'eau, dans lequel l'urbanisation suivrait peu de contraintes, à la manière de ce qui s'était produit au cours des dernières années ;
- un scénario visant à organiser la ville d'un million d'habitants ;

- un scénario visant à organiser l'espace en fonction du patrimoine naturel, des risques, et du recours aux ressources ;
- un scénario visant à organiser en profondeur l'aménagement économique de l'espace.

Le modèle DS a donc été utilisé pour produire un ensemble de simulations dont les résultats ont été un élément de dialogue lors des réflexions sur le prochain SAR, et est donc intervenu comme élément de processus sur la prospective d'avenir du territoire réunionnais.

Etant informaticiens, notre but n'est pas ici de nous étendre sur les résultats de cette collaboration, mais nous allons présenter brièvement quelques cartes produites par DS lors des simulations faites avec la cellule SAR. Ceci permettra de se faire une idée visuelle des avenir possibles pour La Réunion.

Les figures 6.15 et 6.16 montrent ainsi le résultat de simulations d'évolution du Mode d'Occupation des Sols. Les espaces urbains sont représentés en rouge, les espaces agricoles sont représentés en jaune, et les espaces naturels sont représentés en vert. La figure 6.15 illustre un scénario de simulation laxiste au niveau des contraintes d'urbanisation, et on voit donc apparaître de nombreuses zones rouges, signe d'une nouvelle urbanisation, sur le pourtour littoral de l'île et autour des centres urbains déjà existants. En revanche, la figure 6.16 illustre un scénario de simulation favorable au développement économique et agricole de l'île. On voit donc les zones jaunes prendre une plus grande importance alors que la population se concentre sur les zones urbaines déjà existantes, ce qui entraîne peu de nouvelles constructions.

La figure 6.17 est, elle, d'un type différent puisqu'elle illustre la densité de population qui augmente au fur et à mesure que le temps se passe dans la simulation. On remarque une teinte rouge bien plus prononcée en fin de simulation dans les zones d'agglomérations principales de l'île, certaines étant même représentées en noir, signe d'une densification importante.

Ces résultats de simulations ont fait partie de la réflexion qui a conduit la cellule SAR et différents cabinets d'études à produire leurs documents pour la révision du SAR datant de 1995. L'utilisation de DS a ainsi permis de traduire en termes d'évolutions chiffrées et d'illustrer en termes d'évolutions spatiales des documents comme celui représenté à la figure 6.18. Ce type de schémas chorématiques étant l'un de ceux utilisés par les bureaux d'études en charge d'établir et d'analyser les scénarios d'aménagement et de développement à La Réunion pour le Conseil Régional.

6.5 Conclusion

Dans ce chapitre, nous avons présenté de façon générale le modèle DS qui permet de simuler les interactions qui se produisent entre les trois grandes classes d'utilisation du sol (espaces naturels, agricoles et urbains) à l'île de La Réunion.

L'originalité de ce modèle tient dans le couplage de deux dynamiques comportementales. Une première dynamique, dite d'évolution du mode d'occupation des sols, est présente à

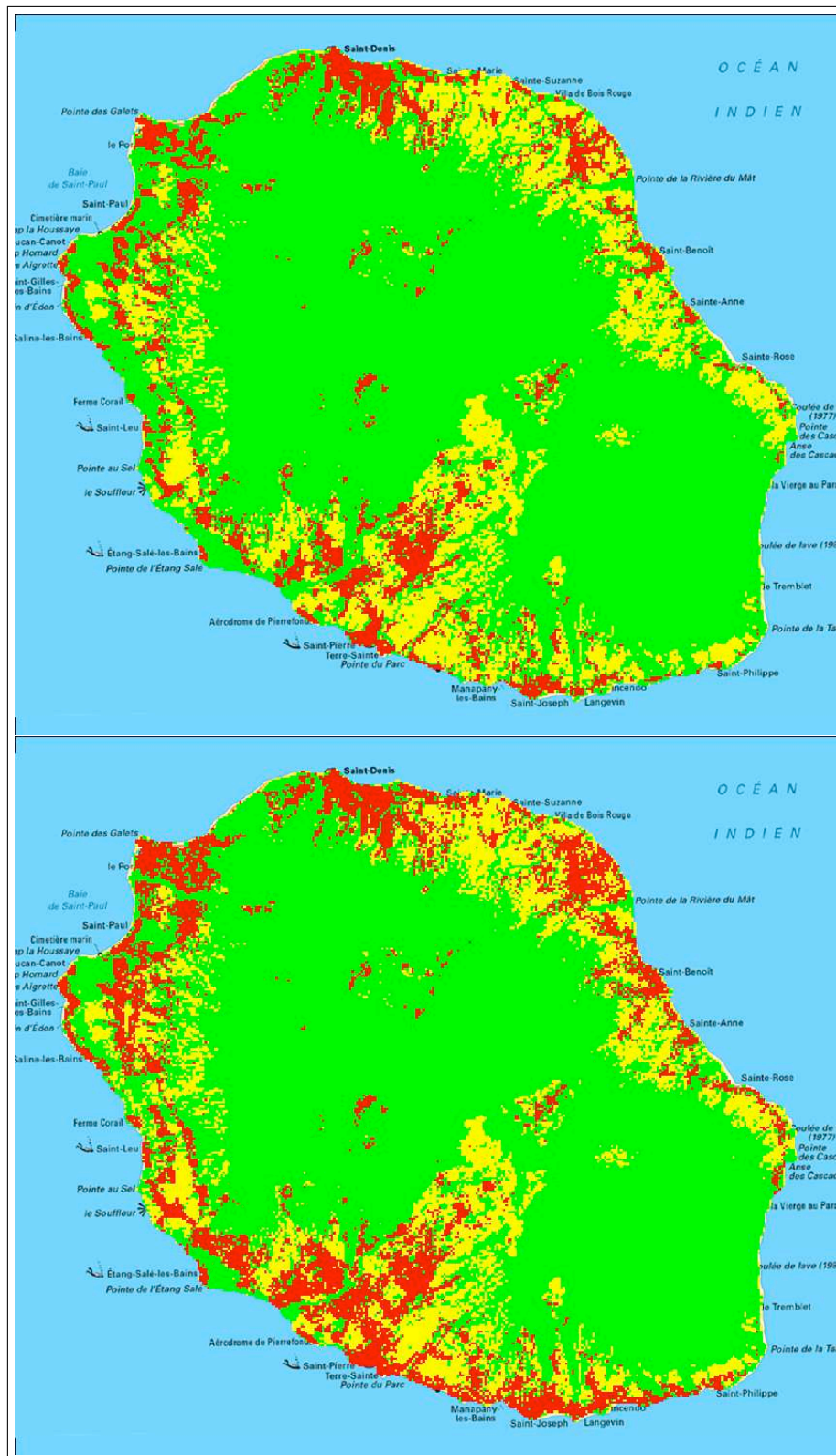


FIG. 6.15 – Un exemple d'évolution du Mode d'Occupation des Sols dans un scénario laxiste au niveau contraintes d'urbanisation (en haut l'état initial en 2003, en bas l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.

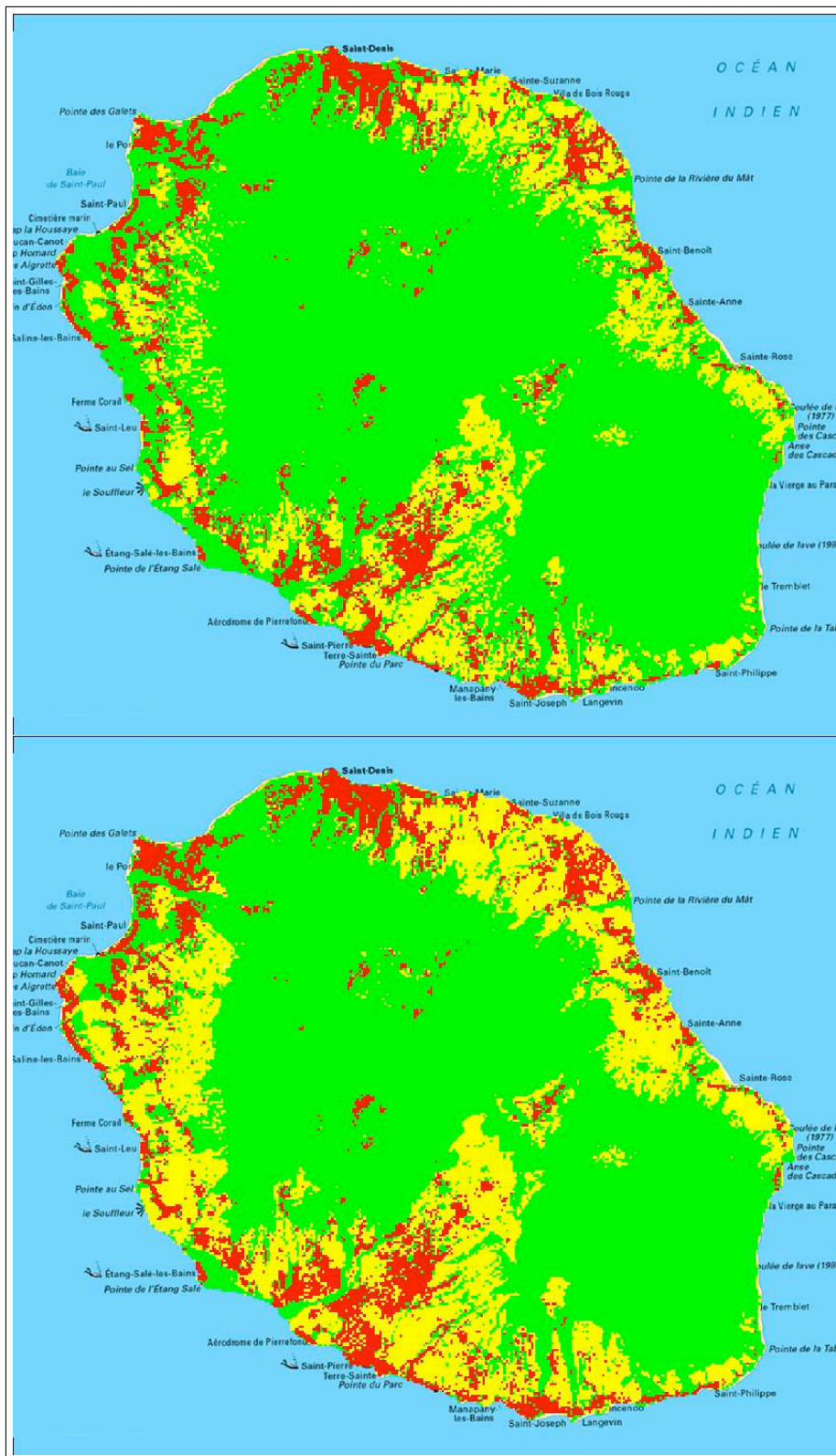


FIG. 6.16 – Un exemple d'évolution du Mode d'Occupation des Sols dans un scénario favorable au développement économique (en haut l'état initial en 2003, en bas l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.

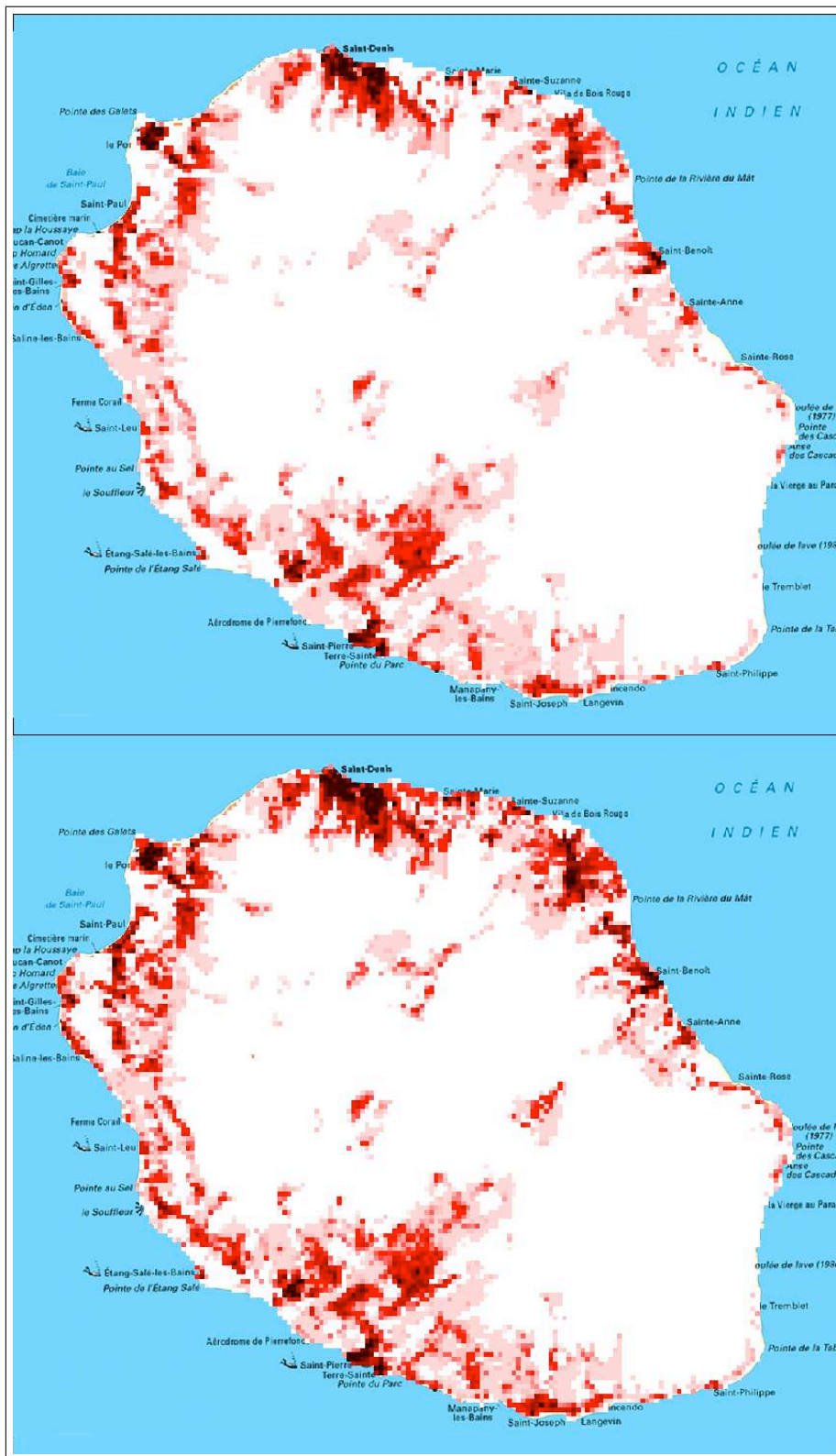


FIG. 6.17 – Un exemple d'évolution de la densité de population croissante selon le dégradé de couleur rouge (en haut l'état initial en 2003, en bas l'état final en 2030)

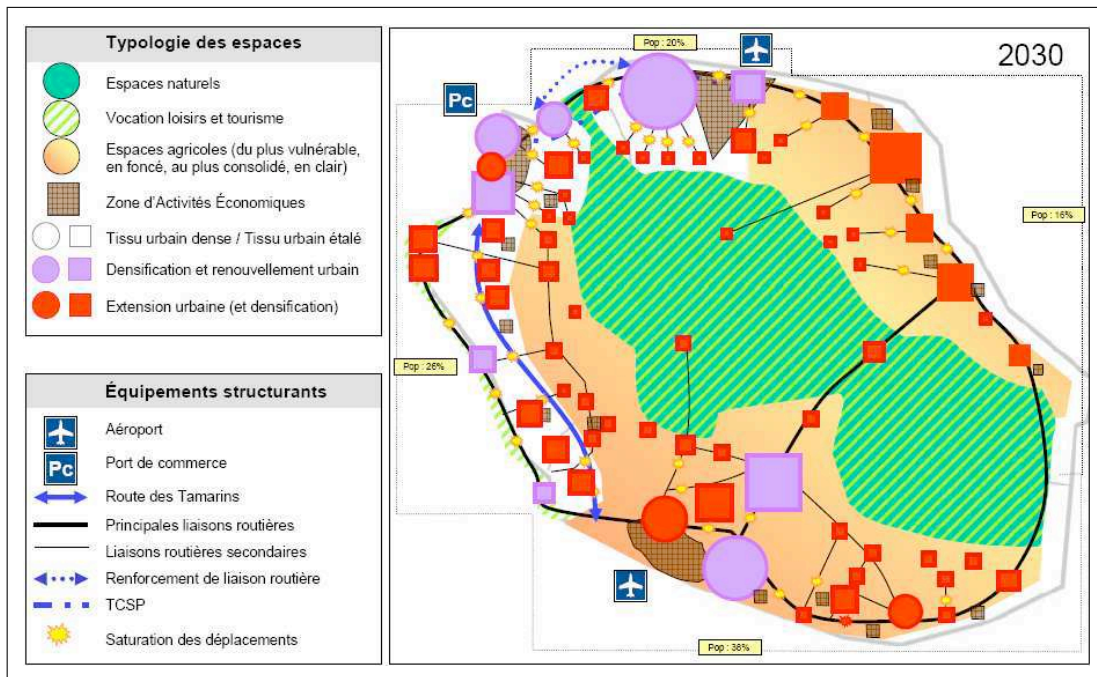


FIG. 6.18 – Schéma chorématique d'un scénario d'évolution tendancielle (extrait de [SETEC & SAFEGE 2007])

l'échelle globale de l'île. Une seconde dynamique, dite d'évolution de la population, repose, en fonction de la granularité choisie au moment de la configuration de la simulation, sur l'évolution de centaines, de milliers (ou mieux, de centaines de milliers !) d'agents qui couvrent, tous ensemble, la totalité du territoire. Le fonctionnement interne du modèle, le comportement des différentes entités, la communication entre les différents agents et leurs interactions avec l'environnement seront présentés en détail dans le chapitre 8. Nous y insisterons notamment sur la façon dont sont couplées les deux dynamiques – le point de couplage se faisant via l'environnement au niveau de l'évolution de la population qui influe sur l'urbanisation – et nous y présenterons un exemple expérimental de prise en compte dans la simulation de l'émergence de nouvelles zones urbaines.

Les simulations produites dans DS permettent de mettre en évidence les choix qui conduiront l'île à une urbanisation plus ou moins importante au détriment des zones agricoles ou naturelles. En testant différents scénarios prospectifs, DS permet ainsi d'observer en quoi l'évolution et la répartition de la population, les politiques d'urbanisation ou les volontés de conservation influenceront sur l'image qu'auront demain les espaces réunionnais.

Le caractère global de la problématique d'aménagement du territoire, et les possibilités offertes au niveau de l'initialisation du modèle confèrent à celui-ci un caractère générique. DS peut ainsi fonctionner avec des cartes et des données relatives à une région autre que l'île de La Réunion, ce qui confère au modèle un aspect réutilisable non négligeable.

Comme pour tout modèle il est cependant nécessaire de relativiser les résultats produits car leur interprétation est étroitement liée aux hypothèses de modélisation. Rappelons en ce sens

que la résolution spatiale et les simplifications du modèle permettent d'observer des grandes tendances sur un territoire donné, mais que vouloir faire une lecture trop fine des sorties obtenues à l'échelle des parcelles et cellules pourrait être trompeur. Le but recherché par DS n'étant bien évidemment pas de prévoir l'avenir, mais simplement d'aider à comprendre et faire comprendre les dynamiques qui sont modélisées.

Pour toutes ces raisons, DS constitue un outil de choix pour faire de la prospective territoriale, et ce quel qu'en soit l'utilisateur : politique, décideur, chercheur ou agent de terrain, tous ont à gagner à utiliser de tels outils pour mieux réfléchir les aménagements futurs. Et même si parfois les résultats ne sont pas satisfaisants par rapport aux espérances de l'un ou de l'autre, l'expérience aura au moins servi à entamer une discussion entre ces différentes parties souvent bloquées au moment d'entamer ensemble des projets d'aménagement ayant une portée à long terme.

Soulignons enfin qu'un outil comme DS trouve même aujourd'hui sa place dans le monde éducatif, puisqu'il permet de donner aux jeunes générations d'étudiants une vision de ce que c'est qu'un outil de simulation, de comment il est conçu, mais aussi (et presque surtout) de voir que cela sert concrètement à quelque chose.

L'initialisation : charger et utiliser des données cartographiques

Sommaire

7.1 Introduction	143
7.2 Des cartes colorées comme plans d'informations	144
7.2.1 L'initialisation : un point clé dans les simulations	144
7.2.2 Les cartes, un vecteur de transmission idéal	145
7.2.3 Quelques cartes plans d'informations	145
7.3 Le langage XELOC	149
7.3.1 Les bases du Langage	150
7.3.2 Configurer agents et environnements	152
7.3.3 Charger les cartes pour le modèle DS	155
7.4 Conclusion	157

7.1 Introduction

Dans le chapitre précédent nous avons décrit le modèle DS, un outil de simulation pour l'aménagement du territoire à l'île de La Réunion. Les données qui sont prises en compte pour faire fonctionner ce modèle, et qui doivent être prises en compte de façon générale dans un système multi-agents, sont souvent nombreuses et de nature complexe. Mais, alors que beaucoup de temps est consacré à la modélisation du système et à son implémentation, les techniques dédiées à l'initialisation sont, elles, souvent non explicitées et les schémas de configuration mis en place sont rarement réutilisables. Dans ce chapitre, nous revenons donc sur ce point clé que constitue l'initialisation d'un système, et nous présentons XELOC, un support pour la configuration et l'initialisation de systèmes multi-agents.

XELOC [David *et al.* 2009] est établi sur les bases du langage XML que nous avons enrichi par une sémantique de langage script. Ainsi, tout en permettant de définir un paramétrage sous la forme d'une description XML classique, XELOC offre la possibilité de synthétiser une configuration complexe à l'aide d'une procédure dynamique. En outre, XELOC intègre une technique d'initialisation riche utilisant des cartes sémantiques fournies par des thématiciens.

En effet, les interactions que nous avons avec les thématiciens divers nous ont montré qu'il était important de trouver des vecteurs de transmission de l'information qui soient le plus pratique possible pour tout le monde, aussi bien les informaticiens que les experts thématiciens. Et utiliser des plans d'informations constitués de cartes colorées s'est avéré un excellent vecteur de ce type. Nous présenterons donc quelques cartes qui ont été utilisées pour transmettre des données utiles à l'initialisation de nos simulations et nous détaillerons la façon dont elles sont chargées dans DS via l'utilisation du langage XELOC.

7.2 Des cartes colorées comme plans d'informations

7.2.1 L'initialisation : un point clé dans les simulations

Comme nous l'avons vu dans les premiers chapitres de ce manuscrit, un modèle informatique est la description formelle d'un système, physique ou abstrait, qui a pour objectif de permettre de reproduire virtuellement la dynamique du système à l'aide d'un ordinateur. Pour atteindre cette reproduction virtuelle il faut d'abord implémenter le modèle, c'est à dire le traduire dans un langage de programmation. Le code obtenu devra alors être assemblé et compilé pour constituer une application. Finalement, c'est l'exécution de cette application par un ordinateur qui va permettre la réalisation des traitements spécifiés par le modèle et ainsi mimer le fonctionnement du système.

Une fois l'application obtenue, se pose alors le problème de sa configuration. Celle-ci peut être faite de différentes façons de sorte que chaque exécution conduise à des résultats qui ne seront pas forcément identiques les uns des autres. En quelque sorte, chacune des configurations représente une expérience (ou un scénario de simulation) dans le sens où elle a pour effet de soumettre l'application (le modèle informatique, résultant de l'implémentation du modèle conceptuel du domaine, et exécuté pour produire le modèle simulé) à un paramétrage bien défini.

Ce problème se pose dans le modèle DS en particulier, mais il se pose aussi de façon générale dans l'ensemble des systèmes multi-agents utilisés pour la modélisation de systèmes complexes où le nombre d'éléments à configurer est très important [Hill & Vickers 2006, Holgate & Partain 2002]. Le volume et la complexité des données d'initialisation sont tels que l'approche de description choisie est bien souvent spécifique et n'est pas réutilisable dans d'autres applications que celle pour laquelle elles ont été formulées.

7.2.2 Les cartes, un vecteur de transmission idéal

En suivant notre logique d'une modélisation la plus ouverte possible établie avec les thématiciens concernés, il est important de pouvoir exploiter sous leur meilleure forme (qui peut, selon les cas, être la plus pratique ou la plus parlante) les données qui sont nécessaires pour initialiser le modèle DS. Un format de données s'est révélé être particulièrement parlant, notamment pour les géographes avec lesquels nous travaillons : les cartes sémantiques qu'ils sont habitués à manipuler, comme celles des figures 7.1 à 7.6 qui sont détaillées dans la suite.

Ce format s'est imposé comme étant, pour eux, le moyen le plus pratique de nous transmettre certaines informations riches et non textualisables dont nous avons besoin pour initialiser nos simulations. Mais parvenir à initialiser ces simulations en se servant de telles cartes sémantiques n'était pas, *a priori*, une chose triviale pour nous, informaticiens. Alors que cela peut s'avérer particulièrement utile, notamment dans le cadre d'environnements spatialisés. Et ce d'autant plus que pouvoir utiliser directement des cartes sémantiques faciliterait d'autant plus l'édition de scénarios de simulation. En effet, pour obtenir une nouvelle configuration d'initialisation, il suffirait de choisir une carte plutôt qu'une autre parmi tout un jeu disponible (par exemple des cartes représentant une même donnée, mais fournies par des chercheurs différents).

7.2.3 Quelques cartes plans d'informations

Dans cette section, nous présentons quelques cartes colorées qui sont utilisées pour initialiser le modèle DS et qui sont présentées dans l'Atlas DOMINO [Botta *et al.* 2007]. DS utilise un grand nombre de cartes pour son initialisation, mais nous avons choisi de présenter spécialement 6 d'entre elles car les données qu'elles représentent sont parmi les plus parlantes relativement à notre problématique d'aménagement du territoire :

- une carte purement administrative (la carte des communes de l'île) ;
- une carte représentant le déséquilibre relatif dans la répartition humaine sur l'île (la carte de densité de population) ;
- une carte sur laquelle ressort la mixité des espaces réunionnais (la carte de Mode d'Occupation des Sols) ;
- trois cartes montrant les enjeux des trois secteurs principaux d'occupation du sol (cartes de potentialités naturelle, agricole, et urbaine).

La carte des communes

Cette carte (figure 7.1) représente les 24 communes de l'île de La Réunion. Elle correspond à la carte des limites qui sont enregistrées au niveau administratif. Dans les simulations réalisées avec DS, elle est utilisée afin de localiser administrativement parlant chaque parcelle de terre créée, et ainsi pouvoir ajuster certains paramètres (par exemple on peut envisager que la population vivant à Saint-Denis, le chef-lieu de l'île, soit plus jeune que celle vivant à l'Entre-Deux, commune traditionnelle des Hauts) et les règles comportementales qui en découlent.

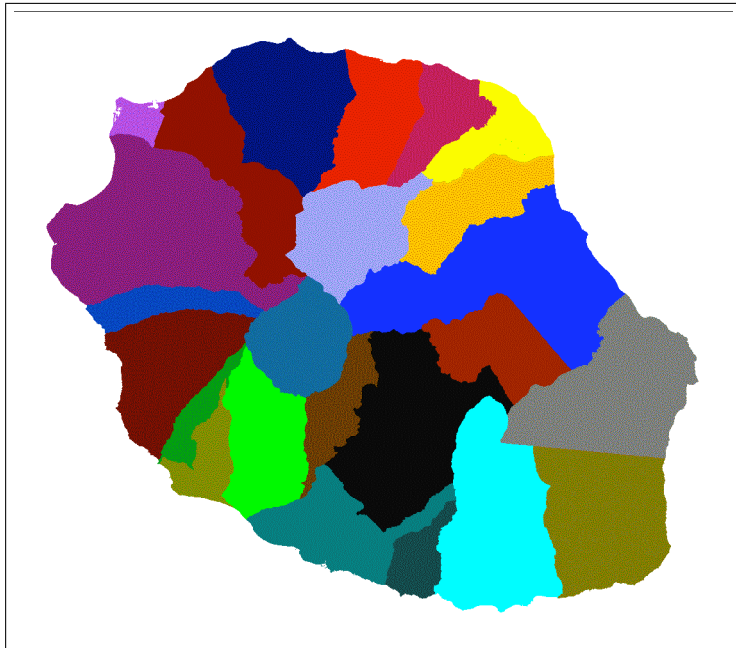


FIG. 7.1 – Les 24 communes de L'île de La Réunion

La carte de densité de population

Cette carte (figure 7.2) représente la densité de population sur l'île de La Réunion en 2003 via un dégradé de couleur rouge. Elle est particulièrement parlante puisqu'elle montre bien une des particularités de la population réunionnaise qui est de vivre en étant concentrée essentiellement sur le pourtour littoral de l'île. Dans DS cette carte est naturellement utilisée pour charger la population de l'île en 2003, date qui nous sert de date d'état initial. Le plan d'information défini par cette carte sera amené à évoluer au fil de la simulation, permettant ainsi de suivre les évolutions de la population.

La carte du MOS

Cette carte (figure 7.3) représente une version du Mode d'Occupation des Sols (MOS) de l'île de La Réunion. Elle est issue des projets DOMINO et APICBIO [Lagabrielle *et al.* 2010]. Fruit d'un travail de synthèse à partir d'une quarantaine de types d'espaces différents, l'ensemble du territoire réunionnais y est finalement représenté selon les trois grandes classes d'utilisation du sol qui sont présentes dans DS : les espaces naturels (en vert), les espaces agricoles (en jaune), et les espaces urbains (en gris). Elle est donc utilisée pour initialiser chaque parcelle de terre créée dans DS en fonction de son MOS. Le plan d'information défini par cette carte sera amené à évoluer au fil de la simulation, permettant ainsi de suivre l'évolution générale du MOS.

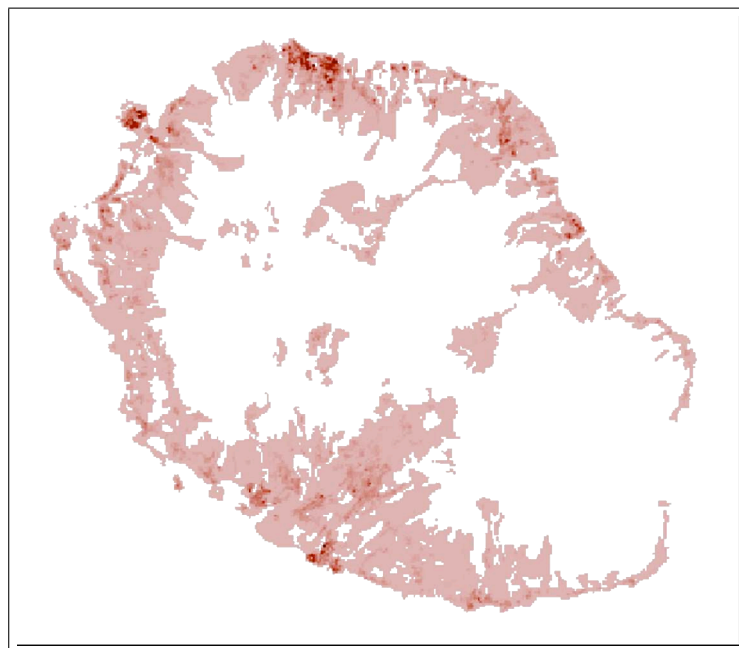


FIG. 7.2 – La répartition de la population à La Réunion

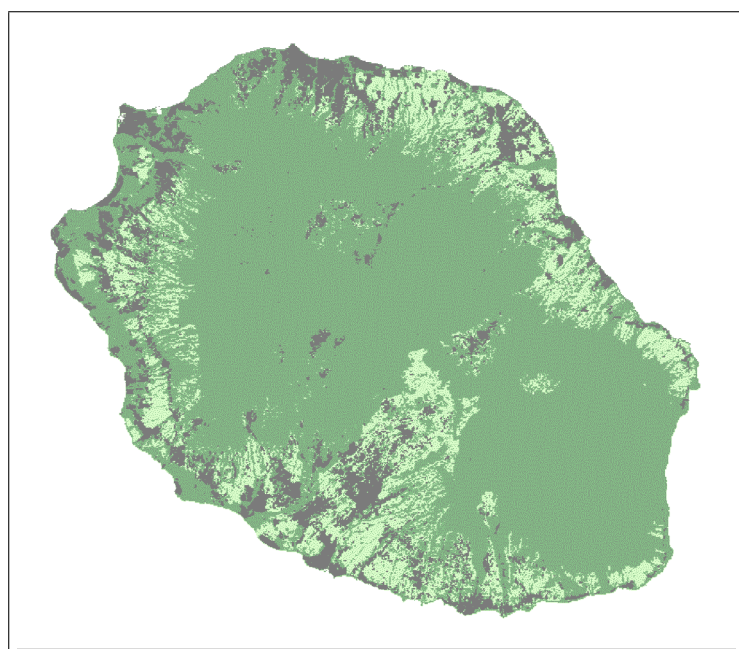


FIG. 7.3 – Le Mode d'Occupation des Sols de L'île de La Réunion en 2003 (extrait de l'Atlas DOMINO)

La carte des potentialités naturelles

Cette carte (figure 7.4) représente les potentialités naturelles de l'île de La Réunion. Cette carte est issue de travaux menés par le CIRAD, l'Agorah, et l'université de La Réunion. Quatre classes de potentiels sont représentées (de la couleur la plus claire à la couleur la plus foncée = du plus faible potentiel au plus fort potentiel) qui représentent respectivement les espaces naturels en bon état (au potentiel le plus fort), les espaces naturels dégradés, les espaces mal connus, et les zones urbaines (au potentiel nul). Selon les scénarios de simulation, les données de cette carte sont prises en compte dans DS lorsque devront être déterminées les zones qu'il est important de préserver essentiellement par rapport à l'urbanisation.

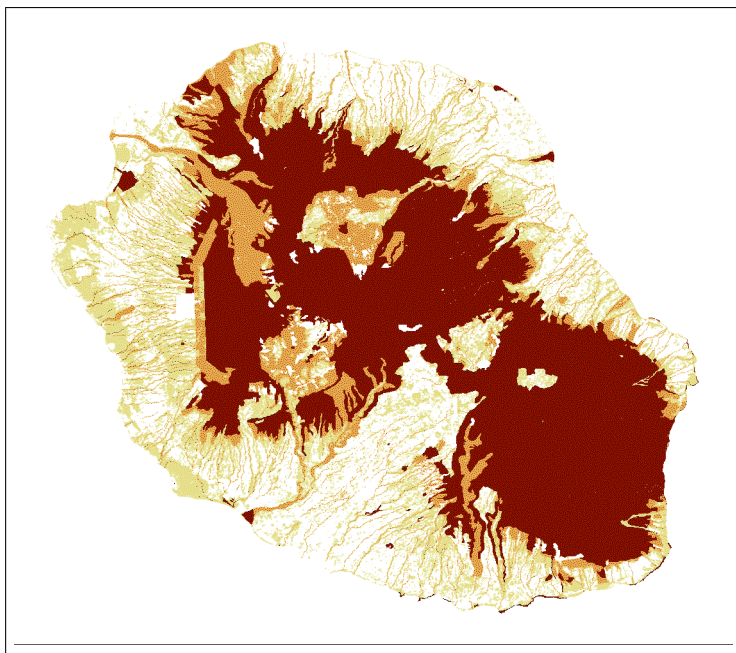


FIG. 7.4 – Les potentiels naturels à l'île de La Réunion (extrait de l'Atlas DOMINO)

La carte des potentialités agricoles

Cette carte (figure 7.5) représente les potentialités agricoles de l'île de La Réunion. Elle a été réalisée en 2006 grâce à des données du Comité de pilotage de la Canne, de la FRCA et de l'UAFP. Quatre classes de potentiels sont représentées (de la couleur la plus claire à la couleur la plus foncée = du plus faible potentiel au plus fort potentiel). Elle se base sur les potentialités d'élevage et sur les potentialités de culture de la canne à sucre, culture traditionnelle qui occupe la plupart des territoires agricoles réunionnais. De par l'importance de la canne à sucre dans l'économie locale, les données contenues sur cette carte sont réellement importantes à prendre en compte dans nos simulations. Ainsi, selon les scénarios de simulation, les données de cette carte sont prises en compte dans DS lorsque devront être déterminées les parcelles de terre dont la mise en culture aura les meilleures répercussions.

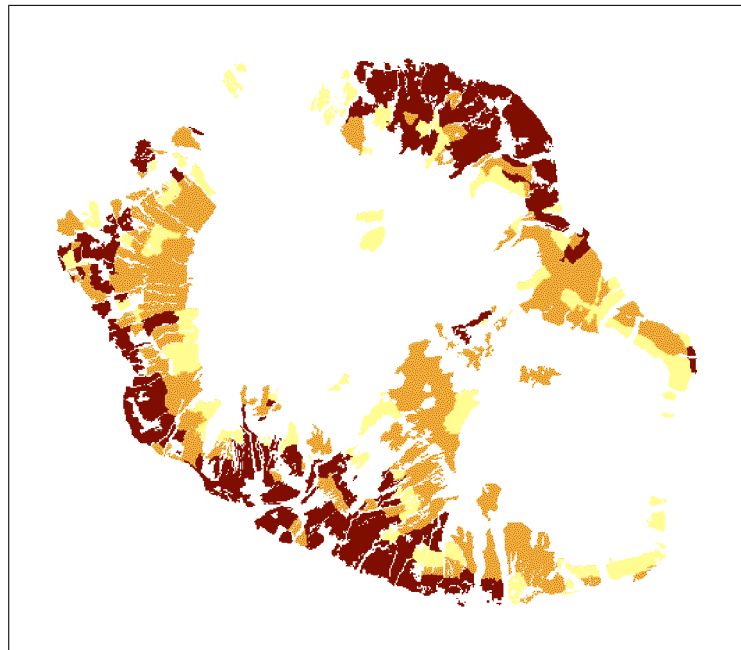


FIG. 7.5 – Les potentiels agricoles à l'île de La Réunion (extrait de l'Atlas DOMINO)

La carte des potentialités urbaines

Cette carte (figure 7.6) représente les potentialités urbaines de l'île de La Réunion. Cette carte est issue des projets SMAT et DOMINO. Quatre classes de potentiels sont représentées (de la couleur la plus claire à la couleur la plus foncée = du plus faible potentiel au plus fort potentiel). Elles représentent le côté attractif sur le plan urbain des différentes zones de l'île, et dépendent directement, pour chacune, de données comme l'altitude, la pente, ou encore la proximité aux grandes agglomérations. Les données de cette carte sont prises en compte dans DS lorsque devront être déterminées les zones déjà urbanisées qui accueilleront de la population, et celles qui seront à urbaniser pour répondre à la demande en nouveaux logements.

7.3 Le langage XELOC

Pour faciliter l'initialisation et la configuration des outils de simulation que nous utilisons, nous avons mis en place le langage XELOC, dont l'acronyme signifie eXtensible Editing Language Of Configuration. Il s'agit d'un langage script à base d'XML dont la finalité est de permettre d'exprimer la configuration d'un système complexe sous la forme d'une suite d'instructions. L'exécution de cette suite d'instructions va indiquer les traitements nécessaires pour produire la configuration précise qui a été décrite.

Un des intérêts du langage XELOC est que celui-ci, de par sa syntaxe XML, autorise une édition par interface graphique via des outils existants pour l'édition XML. Par ailleurs, XELOC permet de synthétiser un paramétrage complexe sous la forme d'une procédure, définie par quelques lignes de code, et dont l'exécution produira le contenu du paramétrage. Ainsi

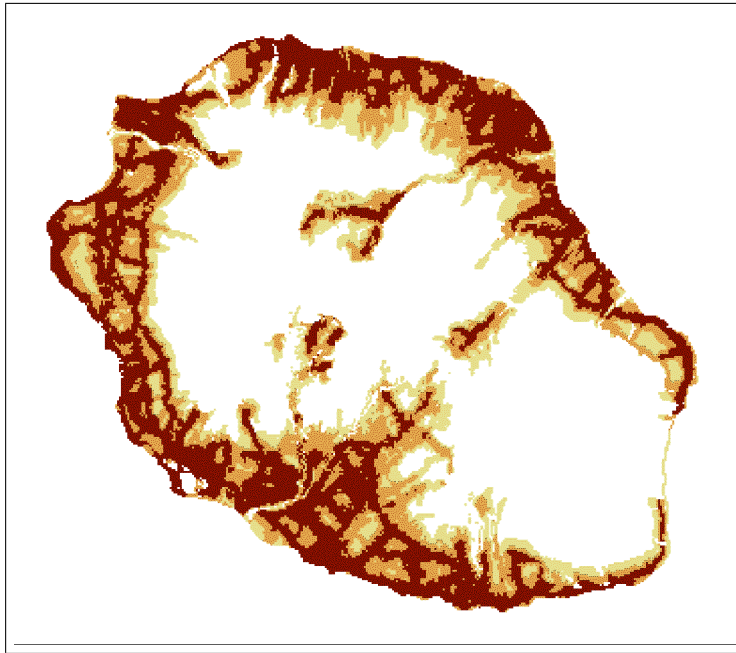


FIG. 7.6 – Les potentiels urbains à l'île de La Réunion (extrait de l'Atlas DOMINO)

une économie importante peut être faite sur le volume du fichier de configuration. De plus, de par le caractère dynamique d'un script [Niemeyer & Poteet 2003], un paramétrage conditionnel ou probabiliste peut être envisagé de sorte que la configuration résultante ne soit pas toujours identique mais couvre un intervalle de possibilités bien calibré.

C'est dans ce langage que nous avons intégré une technique d'initialisation basée sur l'utilisation directe de cartes sémantiques. L'idée étant de se servir d'informations contenues dans des cartes pour initialiser les agents et leurs environnements. *Via* ces cartes, les thématiciens transmettent des données qu'ils comprennent aisément et sont habitués à manipuler, et ceci favorise donc leurs interactions avec les informaticiens.

7.3.1 Les bases du Langage

Un code XELOC est ainsi une succession de balises qui sont évaluées (exécutées) les unes après les autres par un interpréteur. Chaque instruction XELOC a une fonction précise et son exécution retourne une valeur récupérée par la balise parente et exploitée en fonction du traitement réalisé par cette dernière.

Un fichier XELOC prendra la forme suivante :

```
<?xml version="1.0" encoding="utf-8" ?>  
<XELOC legend="Ma simulation" >  
    ...  
</XELOC>
```

Les deux lignes d'en-tête et la dernière ligne correspondent à la structure générale d'un fichier XML [W3C 2004, W3C 2006]. Mais XELOC offre une dimension supplémentaire qui se

traduit au travers l'évaluation de l'ensemble des sous-balises portées par la balise `<XELOC>`. Quand ces sous-balises seront exécutées, la valeur retournée par le programme XELOC sera la valeur retournée par la dernière balise évaluée.

En tant que langage de programmation, XELOC permet la définition de variables. Ceci se fait au moyen des balises `<setVar>` et `<setGlobal>` qui les définissent respectivement au niveau local et global. Ces variables contiennent toutes une valeur unitaire qui est automatiquement transtypée en fonction du contexte.

La balise `<XELOC>` initiale constitue le programme principal, celui par lequel démarre l'exécution quand on traite le fichier source. Mais il est possible de définir des sous-programmes dont l'exécution pourra être lancée à tout moment et autant de fois que l'on souhaite. Naturellement, XELOC permet également de coder des instructions conditionnelles (instructions `<if>` `<then>` `<else>`) et différentes boucles de type for, dont certaines seront présentées dans les sections suivantes.

Une des singularités du langage tient dans ce que nous avons appelé les "attributs balisables" : en dehors des attributs `value`, `setVar` et `setGlobal`, tous les attributs du langage sont "balisables". Ceci signifie qu'ils peuvent s'écrire :

- soit sous forme in-line, comme un attribut classique :

```
<maBalise nomAttribut="{valeur}" >
    ...
</maBalise>
```

- soit sous forme de sous-balise :

```
<maBalise>
    ...
    <nomAttribut>
    ...
</nomAttribut>
</maBalise>
```

Cette considération syntaxique permet d'atténuer la rigueur du typage imposé par XML et d'obtenir un langage de programmation plus souple. Dans cette forme, la valeur de l'attribut est alors celle retournée par l'évaluation des balises comprises entre `<nomAttribut>` et `</nomAttribut>`.

XELOC a donc été conçu pour disposer de toute la force d'un vrai langage de programmation, et ce sous une forme mieux adaptée à notre problématique de la configuration et de l'initialisation de systèmes que les langages déjà existants (Javascript, Perl, etc.) [Loui 2008, Ousterhout 1998, Winder 2006]. Dans le chapitre précédent, nous avons présenté le modèle DS, qui est un modèle de simulation de l'évolution des espaces fonciers à l'île de La Réunion. Implémenté sur la plateforme de simulation multi-agents GEAMAS-NG, il permet de simuler sur l'ensemble du territoire l'évolution de la population réunionnaise et les interactions entre les trois grandes classes d'utilisation du sol (espaces naturels, espaces agricoles, et espaces urbains), tout en observant les évolutions induites. Dans la suite, nous illustrons

certaines des fonctionnalités offertes par XELOC en détaillant comment peut être réalisée la configuration des agents et de l'environnement dans un cas simple inspiré du modèle DS et nous montrons ensuite comment est réalisée l'initialisation de DS *via* l'utilisation des cartes sémantiques colorées présentées plus tôt dans le présent chapitre.

7.3.2 Configurer agents et environnements

Si XELOC facilite la configuration d'un système, il se destine ainsi surtout à rendre cette configuration réutilisable quel que soit le contexte d'implémentation du modèle théorique. Ainsi du code XELOC peut être utilisé pour configurer les agents et environnements qui sont présents dans une simulation de système complexe spatialisé comme c'est le cas de DS.

L'initialisation d'une simulation consiste alors le plus souvent en la définition d'un environnement spatial, généralement de la forme d'une grille, sur lequel se trouvent des agents qui permettront de mettre en œuvre la ou les dynamiques du modèle simulé. Il faut donc en particulier spécifier la taille de la grille et déclencher la création des agents qui occuperont certaines de ses cases en fonction des volontés de l'utilisateur.

Imaginons, à titre d'exemple, que l'on veuille initialiser une aire spatiale, qui pourrait représenter une surface de terre, sur laquelle seraient présents un certain nombre d'agents, qui pourraient représenter des parcelles urbanisées. Le code XELOC permettant de générer une configuration de ce type commence par les lignes d'en-tête que nous avons présentées précédemment, et les premières instructions rencontrées ensuite sont des balises `<setGlobal>` :

```
<!-- Paramétrage global -->
<setGlobal name="nbRow"          value="400" />
<setGlobal name="nbColumn"      value="400" />
<setGlobal name="densityUrbanCell" value="40" />
```

Grâce à cette commande de base, on définit ici trois variables globales : `nbRow` et `nbColumn` pour les dimensions de la grille ; et `densityUrbanAgent` pour la valeur de densité de remplissage des agents parcelles urbaines que nous expliciterons plus loin.

La suite des instructions contenues dans le code XELOC nécessaire à la configuration d'une simulation voit ensuite apparaître l'instruction `<launchParam>`, indispensable à toute simulation :

```
<!-- Paramètre de lancement -->
<launchParam beginTime="0"   endTime="3000"
              step="1"       pause="500" />
```

Il s'agit ici de configurer les paramètres de lancement de la simulation. Sur cet exemple il s'agit simplement de définir la façon dont le temps s'écoulera, en précisant une date de début, une date de fin, un pas, et une pause éventuelle qui sont tous exprimés dans une unité temporelle arbitraire (ici des millisecondes).

Si l'on continue à dérouler le code XELOC de configuration de notre exemple, on trouve ensuite un ensemble d'instructions `<defineComponent>` :

```
<!-- Enregistrement des composants -->
<defineComponent name="LandArea"
    class="myApplication.environment.LandArea"/>
<defineComponent name="UrbanCellAgent"
    class="myApplication.agent.UrbanCellAgent" />
```

Ces instructions permettent d'instancier des composants à partir de leur nom en donnant leur classpath associé ; les classes agent et environnement étant définies à cet endroit.

Une fois les agents et environnements instanciés, on voit apparaître une instruction qui a son importance dans le contexte d'utilisation d'une plateforme de SOA car elle permet d'instancier un noyau d'exécution :

```
<!-- Instanciation du noyau -->
<createKernel name="DefaultKernel"
    timeSpace="ConstantStep"/>
```

A ce noyau d'exécution est associé un nom, mais aussi un environnement temporel. Ceci est particulièrement intéressant car cela nous permet de basculer entre les différentes approches temporelles – pas de temps constant, évènementiel, moteur à temporalité [Payet *et al.* 2006a] – qui sont disponibles dans le contexte de la plateforme de simulation sur laquelle l'on se trouve.

Dans le code de configuration XELOC arrivent ensuite des instructions concernant la configuration directe de l'environnement spatial :

```
<!-- Ajout de l'environnement spatial -->
<addEnvironment name="LandArea"
    setGlobal="worldName" />
<!-- Configuration de cet environnement -->
<configEnvironment environment="@worldName">
    <multiValue>
        <champ name="request" value="size"/>
        <champ name="row" value="@nbRow"/>
        <champ name="column" value="@nbColumn"/>
    </multiValue>
</configEnvironment>
```

La balise `<addEnvironment>` permet ici d'instancier un environnement spatialisé représentant une aire terrestre correspondant au monde dans lequel vont être situés les agents parcelles urbaines. La balise `<configEnvironment>` permet de lancer une requête de configuration de l'environnement spécifié. On voit ici apparaître le symbole @ en première lettre de la valeur d'un attribut. C'est un élément syntaxique du langage XELOC qui signifie que cette valeur correspond à celle de la variable dont le nom est indiqué à droite de ce symbole. Dans cette

suite de commandes, nous faisons ainsi appel aux variables `nbRow` et `nbColumn` définies précédemment et à la variable `worldName` définie lors de l'ajout de l'environnement. Apparaît également la balise `<multiValue>`, qui dénote une structure multi-champs qui permet de manipuler des variables complexes, sous forme d'informations structurées, sans se soucier du type des champs qui la composent.

Enfin, le code XELOC propose les instructions qui permettent de configurer les agents parcelles urbaines.

```
<!-- Création des agents -->
<forGrid row="@nbRow"      column="@nbColumn"
        varRow="curRow"   varColumn="\curColumn\">
  <if>
    <cond>
      <booleanAlea luck="@densityUrbanCell"/>
    </cond>
    <then>
      <!-- On crée une parcelle urbaine -->
      <addAgent name="UrbanCellAgent"
                setVar="curAgent" />
      <addBody environment="@worldName" agent="@curAgent" >
        <multiValue>
          <champ name="type" value="urbanCell" />
          <champ name="row" value="@curRow" />
          <champ name="column" value="@curColumn"/>
        </multiValue>
      </addBody>
    </then>
  </if>
</forGrid>
```

Dans ce bloc de code, la première instruction rencontrée est la balise `<forGrid>`. C'est une commande native qui permet d'exécuter en boucle l'ensemble des commandes situées à l'intérieur de la balise. Cette boucle correspond à deux boucles `for` classiques imbriquées. En effet, la boucle `<forGrid>` décrit ici une grille virtuelle en faisant varier deux variables dont les noms sont donnés par la valeur des attributs `varRow` et `varColumn`. Classiquement, ces variables prennent les valeurs comprises entre 0 et la valeur de l'attribut `row` pour la variable désignée par `varRow`; et entre 0 et la valeur de l'attribut `column` pour la variable indiquée par `varColumn`. Le bloc qui est exécuté à l'intérieur de cette boucle permet de configurer l'ensemble des cellules de l'environnement. Les variables dont les noms sont indiqués par les attributs `varRow` et `varColumn` de la balise `<forGrid>` donnent à tout moment la ligne et la colonne de la cellule dans laquelle on se trouve, notamment pour pouvoir les intégrer à la structure d'état des objets créés.

Ici, le bloc qui est exécuté dans la boucle permet de configurer l'ensemble des agents que l'on veut créer. On remarquera la présence de balises `<if>` ou `<then>` dont le sens parle de

lui même. La première instruction de cette boucle correspond à un test : pour chaque cellule de l'environnement si la condition qui est entre les balises `<cond>` et `</cond>` est vérifiée, alors on exécute la suite des opérations. Ici, la condition est simplement un "lancer de dés" : la commande `booleanAlea`. C'est une commande qui fait partie de la base du langage XELOC et qui permet de générer une valeur booléenne aléatoirement. L'attribut `luck` permet de spécifier le pourcentage de chances que la valeur générée soit la valeur true. Ici ce pourcentage est donné par la variable `densityUrbanCell` que nous avons définie au début du code de cet exemple.

Les instructions suivantes réfèrent directement à la création des agents parcelles urbaines. La balise `<addAgent>` permet d'enregistrer un nouvel agent. Ce n'est pas le cas sur cet exemple, mais il est évidemment possible d'ajouter une balise `<state>` optionnelle qui permet, le cas échéant, de spécifier l'état initial de l'agent.

Dans la boucle étudiée, on voit ensuite apparaître l'instruction `<addBody>` qui, pour chaque agent précédemment enregistré, permet de créer un corps dans un environnement donné, ici l'environnement terrestre `landArea` précédemment spécifié auquel se réfère la variable `worldName`. En définitive, ce code permet donc d'initialiser des agents parcelles urbaines et un environnement spatial de type grille de dimension 400x400. Les cellules de l'environnement étant occupées par agents parcelles urbaines dans environ 40% des cas.

7.3.3 Charger les cartes pour le modèle DS

Nous allons maintenant montrer une possibilité d'initialisation avancée offerte par le langage XELOC : l'initialisation par cartes sémantiques. Nous avons rendu ceci possible, par l'ajout d'instructions XELOC spécifiques travaillant directement sur les cartes qui sont fournies en entrée.

La principale de ces instructions est la balise `<forMap>`, balise dont nous allons illustrer le fonctionnement au travers du code XELOC utilisé pour initialiser le modèle DS avec les données de l'une des cartes présentées à la section précédente, la carte des potentiels d'urbanisation qui présente 4 classes de potentiels, et donc 4 couleurs différentes.

Initialisation des données de potentialité urbaine

Les données de la carte de potentialité urbaine (figure 7.7), qui a été présentée à la section précédente, seront chargées dans DS en utilisant un bloc de code XELOC tel que celui présenté ci-dessous :

```
<forMap      name="UrbanPotential"
            url="./maps/urbanPotential.bmp"
            resolution="16" >
  <legend name="4classes" >
    <label name="Nul"      r="255"v="255"b="255"/>
    <label name="Faible" r="255"v="255"b="128"/>
    <label name="Moyen"  r="242"v="167"b="46" />
    <label name="Fort"   r="107"v="0"  b="0"  />
```



```

</legend>
<cell legend="4classes"
      operator="average"
      mode="closer" >
  <case label="\Nul\">
    <!-- suite d'instructions 1 -->
  </case>
  <case label="\Faible\">
    <!-- suite d'instructions 2 -->
  </case>
  <case label="\Moyen\">
    <!-- suite d'instructions 3 -->
  </case>
  <case label="\Fort\">
    <!-- suite d'instructions 4 -->
  </case>
</cell>
</forMap>

```

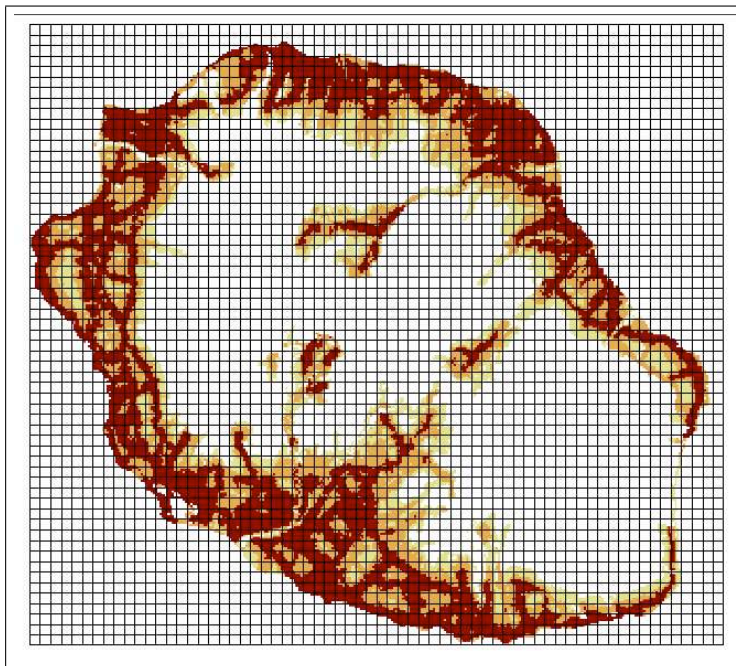


FIG. 7.7 – Une carte sur laquelle est appliquée une grille qui sera parcourue pour lire le code couleur de chaque case lors de l'exécution de la balise `<forMap>`

Dans cette boucle `<forMap>`, on remarque la présence de trois attributs : `name`, qui précise le nom de la carte ; `url`, qui précise son emplacement (qui peut être distant, ce qui se révèle pratique vu la taille importante des cartes sémantiques, et qui permet de manipuler des données

dynamiques localisées sur des serveurs) ; et **resolution**, qui précise la résolution avec laquelle seront lus les blocs sur la carte (ici 16, soit un bloc de 4x4 pixels). Vient ensuite la description d'une légende. Celle-ci porte un nom (ce qui nous permet d'avoir plusieurs légendes de noms différents dans un même code), et permet de faire le lien entre des couleurs et les valeurs associées. On reconnaît ici des couleurs exprimées selon les trois composantes du code RGB, mais il est évidemment possible d'imaginer une légende similaire avec des couleurs exprimées différemment, par exemple en code hexadécimal.

On trouve enfin l'instruction `<cell>`, qui va permettre l'analyse des différents blocs de la carte en fonction d'une légende, d'un opérateur et d'un mode associé. L'opérateur pouvant être : moyenne des couleurs, majorité, etc. Et le mode associé pouvant être : la couleur la plus proche parmi celles spécifiées dans la légende, la couleur exacte, etc. Par la suite, l'opérateur et le mode vont permettre d'associer au bloc lu le label qui lui correspond dans la légende. C'est la valeur de ce label qui va définir le `<case>` qui verra sa suite d'instructions exécutées. Ces suites d'instructions encapsulées dans les `<case>` sont de même type que celles vues au cœur de la boucle `<forGrid>`, le test conditionnel étant ici remplacé par la ventilation dans les différents `<case>`. Ils permettent une initialisation fondée sur les valeurs lues sur la carte.

Notons enfin qu'il est possible de dépasser ce "simple" chargement de cartes pour l'initialisation pour proposer, par exemple, une création/édition de ces cartes via un outil de dessin qui serait présent dans l'application. Nous avons ainsi ajouté une fonctionnalité pour offrir à un utilisateur thématique la possibilité de dessiner directement à l'écran des cartes ou zones de cartes. Ainsi, en fonction de sa connaissance du terrain ou de sa volonté de prospective, il peut affiner des zones précises d'une carte et y rajouter de l'information par rapport à une carte de base qui serait trop générale à son goût. De telles modifications graphiques sont ensuite transcrites en instructions XELOC qui, de là, permettent une initialisation complète du système, ce qui facilite grandement l'édition de scénarios de simulation.

Il va de soi qu'il est possible d'utiliser cette technique d'initialisation utilisant des cartes colorées sémantiques pour améliorer la façon dont est effectuée la configuration des agents et de leurs environnements. En particulier dans DS l'initialisation des agents se fait en appliquant une grille (dont l'échelle est déterminée par l'utilisateur) sur le territoire réunionnais, puis en chargeant une carte représentant un masque servant à délimiter le pourtour de l'île (car La Réunion n'est évidemment pas un territoire rectangulaire), ce qui nous permet de procéder à la création des agents dans chacune des cellules de la grille... sans pour autant créer des agents incohérents qui seraient situés en plein océan.

7.4 Conclusion

Dans ce chapitre, nous nous sommes volontairement éloignés de la prospective territoriale au sens strict pour nous attarder sur une notion qui est souvent négligée dans les simulations : l'importance de tout ce qui est relatif à l'initialisation des modèles. De plus, dans un contexte de projets pluridisciplinaires, où les informaticiens doivent travailler avec des géographes, des

sociologues, des économistes, etc., et où tous les intervenants ont donc des degrés d'expertise et d'interprétation différents, il est important de faire en sorte que l'information qui sera utilisée dans les modèles soit comprise par tous de la même façon.

Nous avons donc présenté une technique d'initialisation qui repose sur l'utilisation de cartes colorées sémantiques qui font office de plan d'informations. Par nature, ces cartes s'avèrent être relativement compréhensibles par tout un chacun, et sont donc un vecteur de transmission de l'information idéal pour les différents thématiciens, entre eux, et vers les modélisateurs et/ou informaticiens qui participent avec eux à la construction d'outils de simulation. Afin de rendre ces cartes parfaitement utilisables pour initialiser nos simulations, nous avons donc mis en place un procédé basé sur l'analyse des codes couleur présents sur chacune d'entre elles, faisant ainsi de ce bon vecteur de transmission de l'information du thématicien à l'informaticien un bon vecteur de transmission de l'information des utilisateurs homme à la machine.

Pour ce faire, nous avons présenté et illustré le langage XELOC qui est un langage de programmation défini sur la base syntaxique du XML auquel nous avons ajouté la sémantique d'un langage script. XELOC a pour objectif de répondre au problème de la réutilisation de configurations de simulations en proposant une approche qui se situe à la frontière entre une description statique et une description dynamique exprimée dans un langage de programmation de haut niveau. Ainsi cette solution conserve les avantages d'une description souple et accessible que confère l'utilisation du XML, avec la puissance et le potentiel synthétique et dynamique d'un langage de programmation.

Le modèle DS : dynamiques et émergence de nouvelles zones urbaines

Sommaire

8.1 Introduction	159
8.2 Une modélisation adaptée à la simulation territoriale	160
8.2.1 La modélisation orientée dynamique	160
8.2.2 Vue d'ensemble du modèle	161
8.2.3 Couplage de dynamiques via l'environnement	164
8.2.4 Couplage et retour méthodologique	168
8.3 L'émergence de nouvelles zones urbaines	171
8.3.1 Un phénomène bien particulier	171
8.3.2 Analyse et mise en place architecturale	173
8.3.3 Détection dans le modèle DS	174
8.3.4 Matérialisation dans le modèle DS	179
8.4 Conclusion	187

8.1 Introduction

Dans les chapitres précédents nous avons présenté le modèle DS dont le fonctionnement se décompose en deux dynamiques : une dynamique démographique (évolution et répartition de la population sur le territoire) et une dynamique d'évolution de l'espace (changements du MOS, le Mode d'Occupation des Sols).

Pour faire fonctionner ces dynamiques, nous avons identifié, avec l'appui des thématiciens concernés, plusieurs facteurs qui entrent en compte dans la dynamique d'évolution du mode d'occupation des sols. Citons, à titre d'exemple, sa probabilité d'urbanisation, sa potentialité agricole, ou encore son degré de conservation naturelle.

Mais le plus important des facteurs devant être pris en compte est sans conteste... l'Homme ! Car c'est bien l'Homme qui vit, meurt, se loge, se nourrit, etc. S'interroger sur l'évolution d'un territoire habité nécessite donc de mener en amont une réflexion sur la façon dont évolue la population concernée. Aussi, pour parvenir à modéliser et simuler l'évolution de l'espace, nous avons été amenés à modéliser une dynamique de la population.

Naturellement, ces deux dynamiques ont besoin l'une de l'autre pour fonctionner. Grâce à l'utilisation de l'approche agent, au choix de GEAMAS-NG comme plateforme de développement, et à l'application d'un certain nombre de règles méthodologiques, nous avons pu, dans un premier temps, les modéliser rigoureusement d'une façon relativement indépendante l'une de l'autre. Puis, dans un second temps, nous avons pu coupler les deux sous-modèles que nous avons obtenus, et constituer ainsi la modélisation complète du système.

Dans ce chapitre, nous présentons donc l'implémentation globale de l'architecture du modèle DS. Nous présentons comment fonctionnent chacune des deux dynamiques qui le composent, et nous nous attardons sur la façon dont elles sont couplées entre elles. Nous montrons également que la façon dont ont été conçues ces deux dynamiques entre dans le cadre de la méthodologie de conception de simulations que nous avons décrite au chapitre 3.

Enfin, nous présentons une expérimentation liée à la prise en compte dans DS d'un phénomène émergent particulier, les nouvelles zones urbaines, en nous appuyant sur le cadre formel et architectural que nous avons décrit dans le chapitre 5. Ceci nous permet de donner un exemple concret de mise en œuvre des propositions énoncées dans toute la deuxième partie de ce manuscrit.

8.2 Une modélisation adaptée à la simulation territoriale

Le couplage des deux dynamiques présentes dans DS est le fruit des travaux entamés par Denis Payet dans [Payet *et al.* 2006b] qui portent sur la Modélisation Orientée Dynamiques.

8.2.1 La modélisation orientée dynamique

La Modélisation Orientée Dynamiques propose de se focaliser sur les dynamiques qui entrent en jeu dans les systèmes que l'on souhaite modéliser (puis simuler) pour découper le modèle "global" en sous-modèles. Chacun de ces sous-modèles, appelé un MDM, Mono Dynamical Model, contient donc exclusivement la connaissance qui lui est nécessaire, ce qui permet d'être pleinement concentré sur cette dynamique là. Ceci étant valable à la fois pour le concepteur et pour les entités du système.

En effet, au niveau des entités du système, les agents de nos simulations, il est possible d'utiliser à bon escient le découpage entre état et comportement, et de se servir de l'environnement comme étant le point de couplage des différentes dynamiques qui sont chacune modélisée dans un MDM particulier, l'ensemble parvenant à rester un tout cohérent. Et même d'autant plus cohérent dans la mesure où avec les MDM chaque thématicien, a priori expert

dans un domaine donné, et donc souvent expert consulté à propos d'une dynamique donnée, peut se concentrer lors de la modélisation et des interactions avec les informaticiens sur "son" MDM sans être perturbé par les éléments à prendre en compte qui proviendraient des thématiques experts dans des champs différents. Les principes de base des MDM sont illustrés sur la figure 8.1.

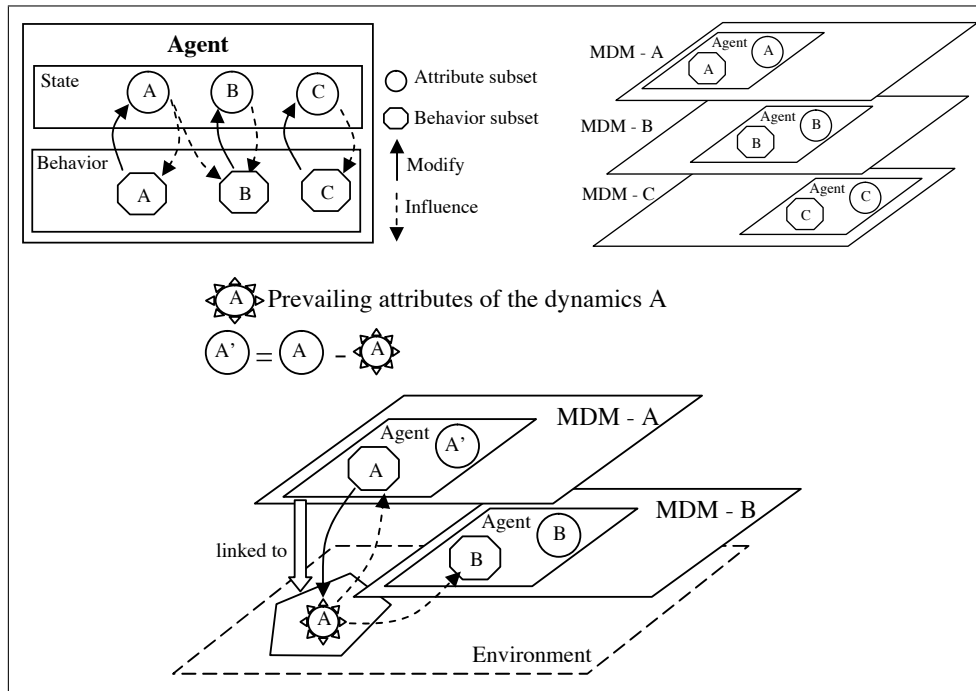


FIG. 8.1 – La modélisation orientée dynamiques (extrait de [Payet *et al.* 2006b])

Cette technique de modélisation est au cœur du modèle DS puisque c'est elle qui nous permet de coupler les dynamiques d'évolution de la population et d'évolution du mode d'occupation des sols. Typiquement, dans ce cas précis, des géographes ayant une grande connaissance dans les questions relatives à l'évolution démographique de la population ont pu, dans les premiers cycles de construction du modèle DS, se concentrer pleinement sur la modélisation de la dynamique d'évolution de la population sans être perturbés par la façon dont allaient réfléchir d'autres acteurs spécialisés dans les domaines d'urbanisation, d'expansion agricole, ou de conservation naturelle qui allaient de leur côté réfléchir à la mise en place de la dynamique d'évolution de l'espace.

8.2.2 Vue d'ensemble du modèle

8.2.2.1 Structure du modèle

Pour mettre en place les comportements liés à la dynamique de population, nous avons découpé le territoire réunionnais en fonction des 4 microrégions (Nord, Sud, Est, Ouest) qui élaborent des SCoT (Schémas de Cohérence Territoriale) qui doivent se conformer au SAR (Schéma d'Aménagement Régional), documents de planification dont nous avons parlé au

chapitre 6. Chacune de ces micro-régions est ensuite découpée en parcelles de terre de forme et de taille identique en appliquant sur l'île le masque d'une grille (de résolution variable selon les choix des utilisateurs et/ou des experts).

En se basant sur les entités obtenues lors de ce découpage, nous créons les agents qui interviendront dans la dynamique de population et qui forment une hiérarchie de classes similaire à la hiérarchie administrative : la *Région* est constituée de *Microrégions* qui sont elles-mêmes constituées de *Parcelles*.

Les deux étapes de ce processus sont schématisées sur la figure 8.2.

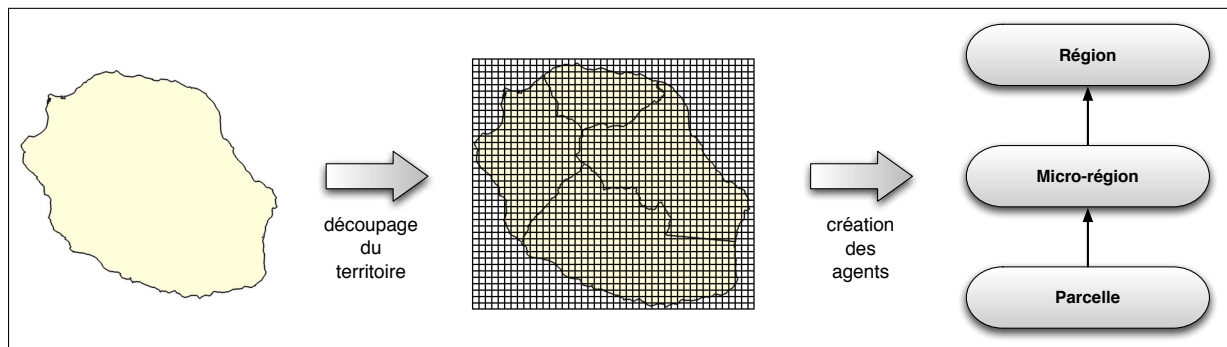


FIG. 8.2 – Lien entre le découpage de l'île et la création des agents

Pour ce qui est de la dynamique d'évolution de l'espace, nous avons mis en place trois agents de niveau supérieur qui correspondent à chacune de ces classes d'utilisation du sol.

Comme illustré sur la figure 8.3, le modèle distingue donc :

- trois niveaux d'agents correspondant aux niveaux d'organisation qui entrent en jeu dans la dynamique démographique du modèle :
 - un agent *Région* ;
 - quatre agents *Microrégions* ;
 - des agents *Parcelles* (nombre variable selon la résolution spatiale choisie).
- un *environnement*, constitué d'objets (que l'on appellera *cellules*) correspondant à la localisation géographique de chacun des agents *Parcelle*.
- trois agents superviseurs qui gèrent, chacun pour leur domaine, l'affectation des terres à l'échelle de l'île en agissant directement sur les *cellules de l'environnement* :
 - un *agent superviseur Agricole* ;
 - un *agent superviseur Naturel* ;
 - un *agent superviseur Urbain*.

Le fait que les deux dynamiques présentes dans le modèle ne se situent pas sur un même plan donne à leur couplage encore plus d'importance. En effet, la dynamique de la population, qui repose sur l'évolution des agents *Parcelles*, est située à l'échelle micro, tandis que la dynamique d'évolution du mode d'occupation des sols, basée sur les évolutions de l'environnement

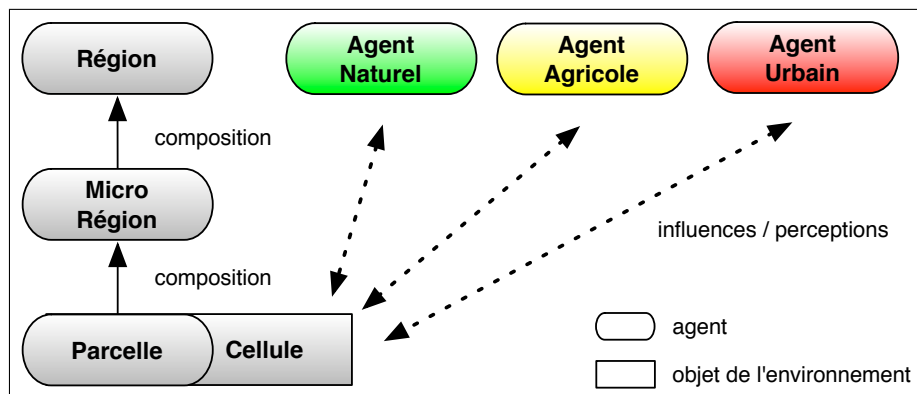


FIG. 8.3 – Structure conceptuelle du modèle DS

provoquées par les *agents superviseurs*, se situe à l'échelle macro.

Cela rend le couplage entre les deux dynamiques d'autant plus important. Or, s'il n'est pas rare d'observer les interactions micro/macro dans un système, que celui-ci soit modélisé via un SMA [Sanders *et al.* 1996b] ou de façon plus classique via un automate cellulaire [Couclelis 1985], celles-ci se limitent souvent à la mise en place d'une dynamique au niveau micro et à l'observation des phénomènes qui se produisent au niveau macro. Ici les deux dynamiques comportementales coexistent bel et bien, non causalement l'une envers l'autre, mais en inter-influence l'une de l'autre.

8.2.2.2 Comportement général du modèle

Le pas de temps du modèle est annuel et les trois *agents superviseurs Agricole, Urbain et Naturel* impactent directement sur les cellules constituant *l'environnement*. La temporalité du modèle est calée de sorte que la cohérence soit maintenue : comme le montre la figure 8.4, les trois *agents superviseurs* se déclenchent successivement et *l'environnement* agit en se calant entre chacune de ces temporalités.

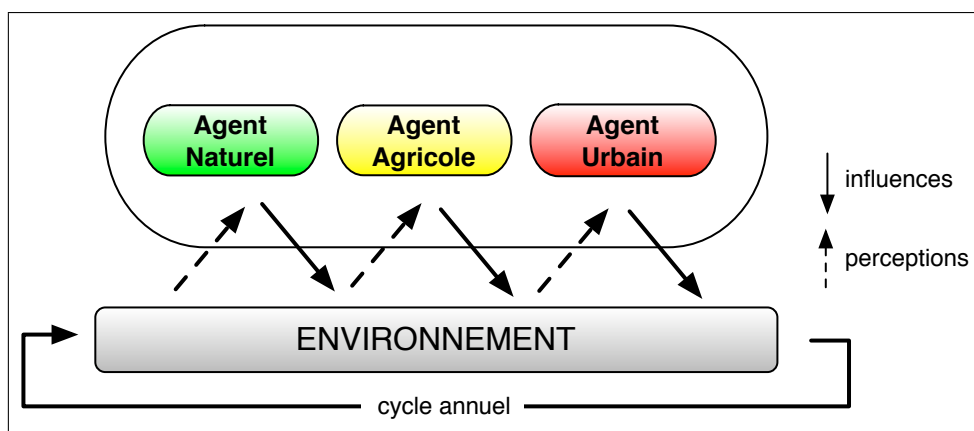


FIG. 8.4 – Les actions des agents superviseurs sur l'environnement

Au cours d'un cycle de simulation, plusieurs attributs d'une *cellule* peuvent être modifiés

par les *agents superviseurs*. On se focalisera ici sur l'évolution du MOS (Mode d'Occupation des Sols : espace naturel, agricole ou urbain) de chaque *cellule* puisque c'est le critère de base dans notre dynamique d'évolution de l'espace.

8.2.3 Couplage de dynamiques via l'environnement

8.2.3.1 Une évolution démographique : la dynamique de la population

La dynamique de la population repose sur l'évolution des agents *Parcelles*, *Microrégions*, et *Région*. Les *Parcelles* sont la base de son fonctionnement et la dynamique comportementale qu'elles produisent est à l'échelle micro.

Les Parcelles

Principe général

Les *Parcelles* sont les agents de base qui œuvrent dans la dynamique de population. Chaque agent *Parcelle* possède une population initiale et, tous les ans, chaque agent calcule à partir de sa population précédente une nouvelle population en fonction de l'évolution de la natalité et de la mortalité, et un pourcentage de cette nouvelle population qui cherchera à quitter cette *Parcelle*, appelée "population migrante".

La nouvelle population totale et la population migrante de chaque *Parcelle* sont communiquées successivement aux agents *Microrégions* et à l'agent *Région*. Ensuite la population migrante est redistribuée sur chaque *Parcelle* en tenant compte de critères divers comme le flux d'immigration de l'île, le facteur d'attraction de la *Microrégion*, le facteur d'attraction de chaque *Parcelle*, etc.

Dynamique

Au cours d'un cycle annuel de simulation, chacun des agents *Parcelle* effectue les opérations suivantes :

- calcul des nombres de départs, naissances, et morts de l'année ;
- mise à jour de la pyramide des âges ;
- mise à jour de la population totale.

Pyramide des âges, natalité, mortalité, départs

La pyramide des âges est propre à chaque agent *Parcelle*. Dans la version actuelle de notre modèle, toutes les *Parcelles* ont initialement la même pyramide des âges. Mais cette dernière évolue différemment au cours du temps en fonction des fluctuations de population subies par chaque *Parcelle*.

Les Microrégions

Principe général

Les agents *Microrégions* interviennent dans la dynamique de population comme intermédiaires entre l'agent *Région* et les agents *Parcelles*. C'est à leur niveau que s'opère la répartition de la population entre les zones d'habitat dense et les zones d'habitat étalé ou dispersé. Ce sont aussi ces agents *Microrégions* qui vont interagir, via l'environnement, avec l'agent *superviseur Urbain*, et faire le lien entre dynamique démographique et dynamique d'urbanisation. Ce lien s'établit sur la base du différentiel entre population à l'instant t et population à l'instant $t+1$. Ce différentiel sera utilisé par l'agent Urbain et sera traduit en population participant à l'étalement pour chaque microrégion.

Dynamique

Au cours d'un cycle annuel de simulation, chacun des agents *Microrégion* effectue les opérations suivantes :

- calcul de la population à attribuer à chacune de ses *Parcelles* ;
- mise à jour de sa population totale ;
- calcul du différentiel de population entre t et $t+1$;
- calcul de la part de la population étalée (qui servira à l'agent *superviseur Urbain*) ;
- mise à jour des facteurs liés à la dynamique démographique.

La Région

Principe général

L'agent *Région* permet l'intégration de la population migrante de toutes les *Parcelles* avant redistribution par *Microrégion* en fonction de leur facteur d'attractivité. A ce niveau est aussi pris en compte le flux externe de population qui immigre dans l'île.

Dynamique

Au cours d'un cycle annuel de simulation, l'agent *Région* effectue les opérations suivantes :

- calcul de la population à attribuer à chaque *Microrégion* ;
- mise à jour de la population totale ;
- mise à jour des facteurs liés à la dynamique démographique.

8.2.3.2 Une évolution spatiale : la dynamique d'évolution de l'espace

La dynamique comportementale décrite ici est relative aux évolutions du MOS pour la totalité du territoire concerné. Cette dynamique a pour moteur les *agents superviseurs*, qui ont une vision globale de l'*environnement*, et est à l'échelle macro.

Les agents superviseurs et l'environnement

Pour pouvoir effectuer des actions sur l'environnement de façon optimale, chacun des trois *agents superviseurs* est doté de la connaissance des *cellules* sur lesquelles il pourra d'agir. C'est ce que nous appelons pour chacun sa "liste de cellules cibles".

Les filtres

Les "filtres" sont des éléments de *l'environnement* qui ont une importance capitale dans ce modèle : pour chaque *agent superviseur* un filtre permet de constituer la liste de cellules cibles qui lui est associée en "filtrant" l'ensemble des *cellules* pour ne retenir que celles répondant à un ensemble de critères, attributs des cellules, qui peuvent être :

- Mode d'Occupation des Sols Naturel, Agricole ou Urbain ;
- conservation naturelle ;
- conservation agricole ;
- ...

L'utilisateur doit donc, via une interface de configuration, définir les critères composant les trois filtres (l'agricole, le naturel et l'urbain). Ici apparaît un premier degré de paramétrage du modèle : jouer sur les valeurs des filtres permet de créer différents scénarios de simulation.

La construction des listes de cellules cibles

A partir des valeurs des trois filtres, chacun des *agents superviseurs* construit sa liste de cellules cibles lors de l'initialisation du modèle. Les cellules qui ne possèdent pas les propriétés établies dans un filtre donné ne seront simplement pas ajoutées à la liste de cellules cibles de l'*agent superviseur* correspondant.

Le tri des cellules

Chaque *cellule de l'environnement* possède des attributs relatifs à sa potentialité urbaine, agricole et naturelle.

Afin d'optimiser le système, les *cellules* appartenant à la liste de cellules cibles d'un agent superviseur donné sont ordonnées en fonction de la valeur de leur potentiel relatif à cet agent (c'est à dire : le potentiel d'urbanisation pour l'*agent superviseur Urbain*, etc.). Les potentiels étant regroupés par "classes" de valeurs (exemple : 4 classes, du potentiel 0 au potentiel 3), les opérations suivantes sont effectuées :

- trier la liste des cellules par potentiel concerné décroissant ;
- classer aléatoirement les groupes de cellules de même potentiel.

La liste de cellules cibles d'un agent superviseur donné contient donc dans l'ordre les "meilleures" *cellules* qu'il vise à conquérir. Lorsqu'un agent superviseur veut transformer des cellules de l'environnement, il n'a donc qu'à consommer sa liste de cellules cibles depuis le début. Ceci évite de fastidieux calculs pour déterminer à chaque cycle quelles sont les meilleures cellules du domaine. Notons que les cellules présentant un potentiel Agricole, Urbain ou Naturel nul ne sont pas considérées comme étant des cellules cibles par le secteur concerné.

La mise à jour des listes de cellules cibles

Pour garder à jour les listes de cellules cibles de chacun des *agents superviseurs*, deux cas de figure se présentent : l'ajout et le retrait.

Les opérations de retrait sont gérées lorsque l'état d'une *cellule* appartenant à une des listes de cellules cibles est modifié. Au vu des valeurs choisies pour le filtre concerné :

- soit la modification n'a pas d'influence, auquel cas aucune modification de la liste de cellules cibles concernée n'est nécessaire ;
- soit la modification est incompatible avec le filtre établi, auquel cas la *cellule* dont il est question est retirée de la liste des cellules cibles à laquelle elle appartenait.

A chaque modification de l'un des attributs d'une *cellule* existante, si celle-ci possède à présent les caractéristiques pour intégrer l'une des listes de cellules cibles, son cas sera réexaminé par l'*agent superviseur* concerné.

Agents superviseurs

Agents superviseurs Naturel et Agricole

Le comportement des *agents superviseurs Naturel* et *Agricole* peut être décrit selon le même schéma :

- L'objectif de l'*agent Naturel* est d'obtenir une surface d'espaces naturels à conserver sur l'île en mettant des *cellules* en conservation (prioritairement les cellules de plus fort potentiel naturel).
- L'objectif de l'*agent Agricole* est d'atteindre une surface de terre agricole donnée sur l'île dans son ensemble en faisant passer des *cellules* (prioritairement les cellules de plus fort potentiel agricole) en MOS Agricole.

Pour cela, à chaque cycle annuel de simulation, ces agents superviseurs effectuent les opérations suivantes :

- détermination de l'objectif (qui peut selon les cas être calculé à partir des données initiales, être une donnée fixée par l'utilisateur, etc.) ;
- calcul de l'espace existant concerné (espace naturel conservé pour l'*agent Naturel* ; ensemble des cellules dont le MOS est Agricole pour l'*agent Agricole*) ;
- de ces deux variables, l'agent extrait une demande de surface (à mettre en conservation pour l'*agent Naturel*, à mettre en culture pour l'*agent Agricole*) ;
- pour essayer de pourvoir à cette demande, et en tenant compte des contraintes définies par l'utilisateur, l'agent transforme le plus de cellules possibles en les prélevant dans sa liste de cellules cibles et en modifiant leur état (mise en conservation pour l'*agent Naturel*, passage en MOS agricole pour l'*agent Agricole*) ;
- selon les paramètres choisis pour la simulation (valeurs seuils, etc.) l'ensemble de la demande peut ne pas être affectée, le modèle enregistre alors une demande insatisfaite.

Ces étapes du comportement des agents superviseurs Naturel et Agricole peuvent être exprimées synthétiquement ainsi :

$$\text{Objectif} - \text{Espace} = \text{Demande}$$

Les Cellules concernées de l'environnement sont transformées (mise en culture ou mise en conservation). Comme toutes les cellules ne peuvent être transformées, il y a apparition d'une demande insatisfaite.

$$Demande - nombre\ de\ cellules\ transformees = Demande\ Insatisfaite$$

Agent superviseur Urbain

L'objectif de cet agent superviseur est d'assurer l'étalement urbain nécessaire à l'installation de la nouvelle population.

Ceci contribue à l'objectif recherché en couplant les dynamiques d'évolution de la population et d'évolution du MOS : loger la population dans les espaces urbains en faisant passer, si nécessaire, des cellules en MOS Urbain (prioritairement les cellules de plus fort potentiel urbain). Aussi, c'est au niveau de l'*agent superviseur Urbain* que se situe le point de couplage des deux dynamiques présentes dans DS.

Pour cela, à chaque cycle annuel de simulation, l'*agent superviseur Urbain* effectue les opérations suivantes :

- l'agent récupère pour chaque *Microrégion* la population migrante participant au phénomène d'étalement via notre dynamique de population ;
- en multipliant cette dernière par un ratio d'étalement (exprimé en "nombre de mètres carrés par habitant migrant participant à l'étalement", fixé par l'utilisateur), qui peut être nul (on est alors dans un scénario à tache urbaine fixe), l'agent obtient une demande en terme surfacique.
- en fonction des contraintes définies par l'utilisateur, l'agent fait passer le plus de *cellules* possible en MOS Urbain en les prélevant dans sa liste de cellules cibles et en modifiant leur état.

Ces étapes du comportement de l'agent superviseur Urbain peuvent être exprimées synthétiquement ainsi :

$$Différentiel\ de\ population * Ration\ d'etalement\ urbain = Demande$$

Les Cellules concernées de l'environnement sont transformées (mise en culture ou mise en conservation). Comparativement aux agents superviseurs Naturel et Agricole, il est admis que l'agent superviseur Urbain ne sera pas affublé d'une demande insatisfaite puisqu'il faudra de toute façon loger la population produite par le système.

8.2.4 Couplage et retour méthodologique

Comme nous l'avons décrit précédemment, les deux dynamiques présentes dans DS ont chacune leur fonctionnement propre. Les thématiciens concernés par chacune d'elles ont pu les formaliser indépendamment et c'est au niveau de la modélisation et de l'implémentation que nous les avons couplées.

Ce couplage est rendu possible grâce à l'utilisation de l'environnement. La figure 8.5 montre bien la séparation qui existe entre les deux "sous-modèles" de DS, celui de la dynamique de population et celui de la dynamique d'évolution du mode d'occupation des sols.

Les *agents superviseurs* de la dynamique d'évolution de l'espace effectuent des modifications sur les cellules de l'environnement. Les *cellules* qui ont été urbanisées (l'*agent superviseur Urbain* leur a affecté un MOS urbain) influencent alors non seulement les agents superviseurs (qui doivent prendre en compte les changements des cellules de l'environnement pour être à jour, notamment au niveau de leur liste de cellules cibles), mais également les agents qui interviennent dans la dynamique de population.

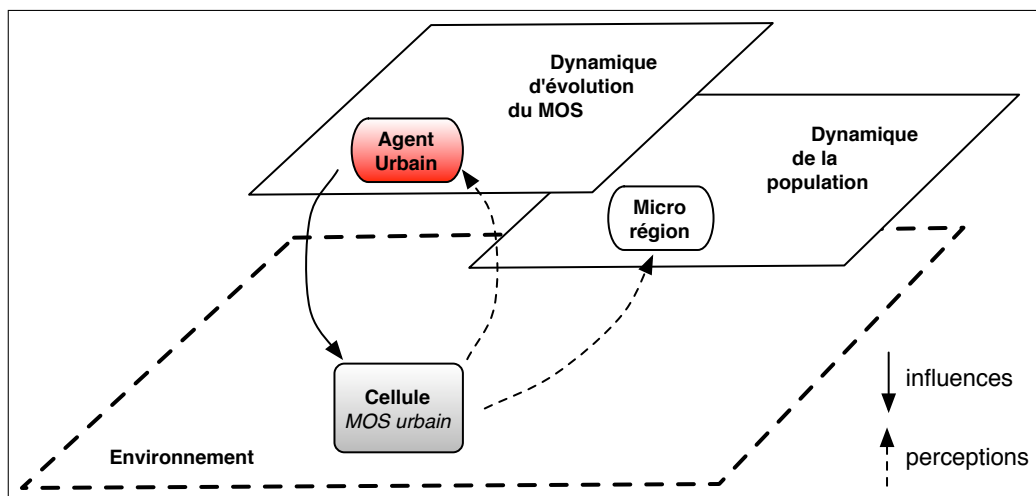


FIG. 8.5 – Le couplage des dynamiques via l'environnement

En effet, lors de la redistribution de la population sur le territoire, les agents *Microrégions* redistribuent leur nouvelle population sur les agents *Parcelles* qui correspondent géographiquement à des cellules de l'environnement dont le mode d'occupation des sols est urbain. L'*environnement* influe donc en ce sens la façon dont la population sera répartie sur le territoire et se pose comme un réel moyen de couplage des dynamiques.

Au travers l'utilisation de l'*environnement*, le comportement général résultant des deux dynamiques présentes dans DS peut alors être décomposé, pour chaque pas de temps, en neuf étapes clés représentées sur la figure 8.6 :

1. Calcul des nouvelles populations totale et migrante de chaque *Parcelle*.
2. Dans chacune des quatre *Microrégions*, somme de la population totale et de la population migrante de toutes leurs *Parcelles*.
3. Somme de la population totale des quatre *Microrégions*.
4. Ajout du flux migrant arrivant sur l'île.
5. Calcul du différentiel de population des quatre *Microrégions*.

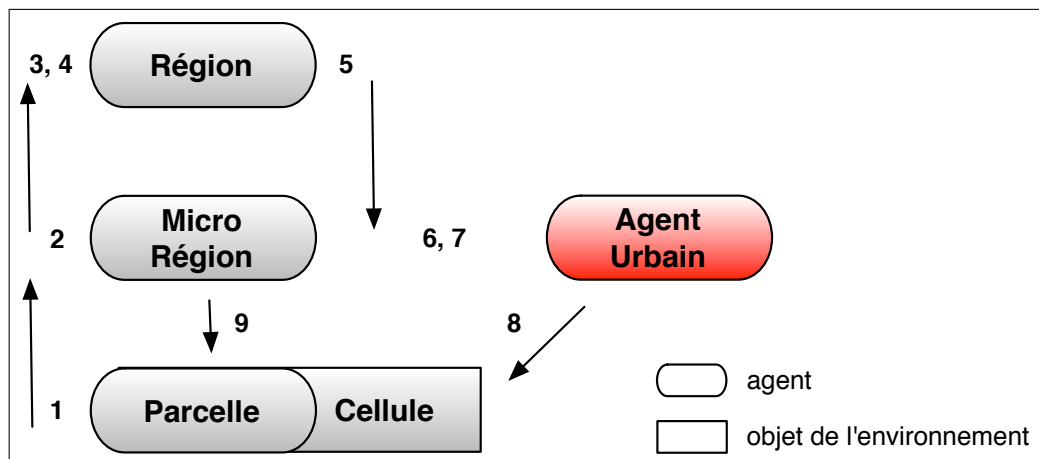


FIG. 8.6 – La dynamique générale du modèle

6. Dans chaque *Microrégion*, calcul à partir du différentiel de population de la part qui va participer à l'étalement urbain, le reste participant au processus de densification.
7. Pour chaque *Microrégion*, calcul du besoin de nouvelles *cellules* à urbaniser en fonction de la nouvelle population participant à l'étalement urbain.
8. Changement de MOS des *cellules* nouvellement urbanisées.
9. Dans chaque *Microrégion*, redistribution de la population migrante participant à l'étalement sur les *Parcelles* correspondant à de nouvelles *cellules* urbanisées, et de la population migrante participant à la densification sur les *Parcelles* correspondant à des *cellules* anciennement urbanisées.

Comme nous l'avons dit, l'environnement joue le rôle d'un réel point de couplage entre les deux dynamiques du modèle. La modélisation de chacune de ces deux dynamiques entre dans le cadre de la méthodologie de conception de simulations que nous avons présentée au chapitre 3. La conception des modèles relevant de chaque dynamique peut être faite de façon indépendante, en associant les informaticiens, les modélisateurs et les thématiques experts de chaque champ. Ainsi, si l'on suit la méthodologie que nous avons décrite sur la figure 3.6, les étapes de modélisation de la dynamique d'évolution de la population et celles de la modélisation de la dynamique d'évolution du MOS peuvent – dans une certaine mesure, évidemment, puisque tout le monde a à gagner à être au courant de ce qui se fait... sans pour autant en être perturbé ou freiné – être menées en parallèle, les deux modèles ne "fusionnant" qu'en fin de modélisation informatique pour faire émerger le modèle général, que l'on peut espérer comme étant producteur – direct ou indirect – de connaissance.

Les deux dynamiques que nous avons présentées – et la façon dont elles sont couplées entre elles – décrivent le fonctionnement interne du modèle DS, et explicitent donc les mécanismes sous-jacents de l'outil tel qu'il a été présenté au chapitre 6. Nous nous sommes donc conformés

à décrire le modèle DS tel qu'il est utilisé et validé par ses différents concepteurs et utilisateurs (thématiciens, informaticiens, décideurs, agents de terrain, etc.).

Dans la suite du présent chapitre, nous allons maintenant décrire la façon dont nous avons enrichi DS afin qu'il soit possible, à ce stade à titre expérimental, de prendre en compte l'émergence de certains phénomènes qui apparaissent au cours de nos simulations. Nous nous appuyerons sur l'étude d'un phénomène que nous avons observé : les nouvelles zones urbaines.

8.3 L'émergence de nouvelles zones urbaines

Nous nous intéressons ici à la façon dont sont pris en compte certains des phénomènes émergents qui peuvent apparaître dans DS. Nous montrons notamment comment le cadre formel et architectural que nous avons présenté au chapitre 5 peut être utilisé pour permettre de détecter un phénomène émergent particulier et de le matérialiser en testant ainsi différentes hypothèses de réification.

Précisons que cette section se situe au sein de ce dernier chapitre puisque la volonté de prendre en compte les phénomènes émergents dans DS relève d'expérimentations que nous avons souhaité mener. Au chapitre 6 nous avons présenté les éléments relatifs à l'utilisation de DS et à la valorisation que cet outil nous avait apporté ; nos expérimentations sur la prise en compte de l'émergence n'entrant pas dans le cadre direct de cette utilisation et de la valorisation associée nous les présentons donc ici.

Ceci se justifie d'autant plus que les deux sections que nous venons de voir, qui illustrent la conception des deux dynamiques internes du modèle DS, ont permis de générer les idées qui ont abouti à la méthodologie de conception de simulations décrite au chapitre 3, puis d'en montrer la validité *a posteriori*. La méthodologie de conception de simulations que nous avons décrite se pose donc comme un complément méthodologique permettant de concevoir des outils de simulations qui mettent en œuvre, au niveau implémentatoire, la technique des MDM qui a été utilisée pour coupler les deux dynamiques présentes dans DS. Nos expérimentations sur l'émergence se plaçant dans la lignée des propositions conceptuelles énoncées aux chapitres 4 et 5, le présent chapitre permet donc de présenter des éléments qui rendent concrètes les propositions de toute la seconde partie de ce manuscrit.

8.3.1 Un phénomène bien particulier

Dès que l'on utilise DS pour effectuer quelques simulations, on remarque le côté très important pris par l'urbanisation dans l'île, et ce quels que soient les scénarios de simulation. En effet, la population réunionnaise est dans un tel accroissement que même avec des hypothèses fortes de densification de l'habitat le besoin en logements et en constructions diverses liées à cet accroissement (zones d'activités, centres commerciaux, etc.) augmente inévitablement.

Lorsque l'on étudie plus attentivement les résultats des simulations, on se rend compte que des zones urbaines particulières apparaissent au fil du temps : des zones qui se sont urbanisées rapidement, en à peine quelques cycles de simulation, et qui correspondent à

des territoires très concentrés auparavant considérés comme étant des zones naturelles ou agricoles.

La figure 8.7 donne un exemple visuel illustrant une telle urbanisation. Sur la portion de territoire réunionnais qui y est représentée, on constate l'évolution de la zone de Cambaie, une zone autrefois préservée en raisons de réglementations, et dont le potentiel d'utilisation est tel qu'elle sera sûrement entièrement occupée d'ici quelques années par des constructions.

Dans la simulation avec laquelle cette figure a été produite, les *cellules* de l'environnement qui correspondent aux *agents parcelles* font chacune 4 hectares. En haut se situe la zone du Port, déjà fortement urbanisée, et en bas se situe la zone de Saint-Paul. On remarque bien qu'au fil de la simulation ces deux zones continuent à s'urbaniser puisque des *cellules* de couleur rouge sont venues s'ajouter à celles existantes. Mais la plupart du temps ces *cellules* nouvellement urbanisées ne font que combler des espaces (restreints) qui n'étaient pas encore urbanisés alors qu'ils étaient déjà dans un milieu urbanisé.

En revanche le cas de l'urbanisation de la zone centrale, celle de Cambaie, est différent puisque les cellules urbanisées y étaient minoritaires au début des simulations, et que l'on constate qu'elles s'urbanisent, ensemble, dans un espace temporel restreint.

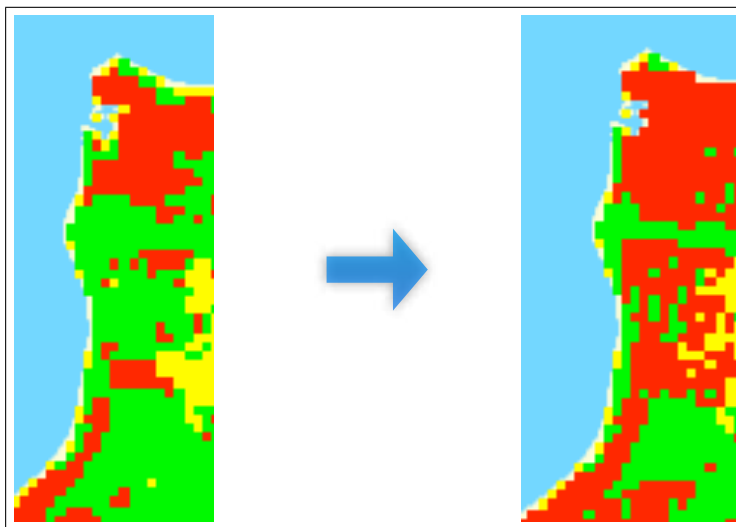


FIG. 8.7 – Une simulation sur la zone de Cambaie (à gauche l'état initial en 2003, à droite l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.

Ce genre de phénomène est bien connu localement puisque de nombreuses zones de ce type peuvent être repérées sur le territoire réunionnais. Elles correspondent généralement à des zones pour lesquelles la réglementation en vigueur a été modifiée dans les documents de planification (les PLU, les SCoT, le SAR). C'est notamment ce qui se passe lorsque de grandes étendues de terres agricoles sont déclassées, devenant ainsi constructibles, et destinées à une urbanisation rapide, que ce soit par des habitations destinées à combler le besoin en nouveaux

logements ou par des locaux, entrepôts, halls, qui apparaîtront dans le cadre d'entreprises s'implantant dans de nouvelles zones d'activités.

Dans la suite, nous allons ainsi nous intéresser aux évolutions virtuelles d'une zone de la commune de Saint-Pierre qui était encore vierge il y a quelques années, qui est aujourd'hui occupée par de nombreux bâtiments commerciaux, et qui, d'après nos simulations, semble vouée à connaître une urbanisation encore plus importante. Un exemple de résultat d'une simulation (de 2003 à 2030) obtenu sur une étendue terrestre contenant cette zone est donné sur la figure 8.8.

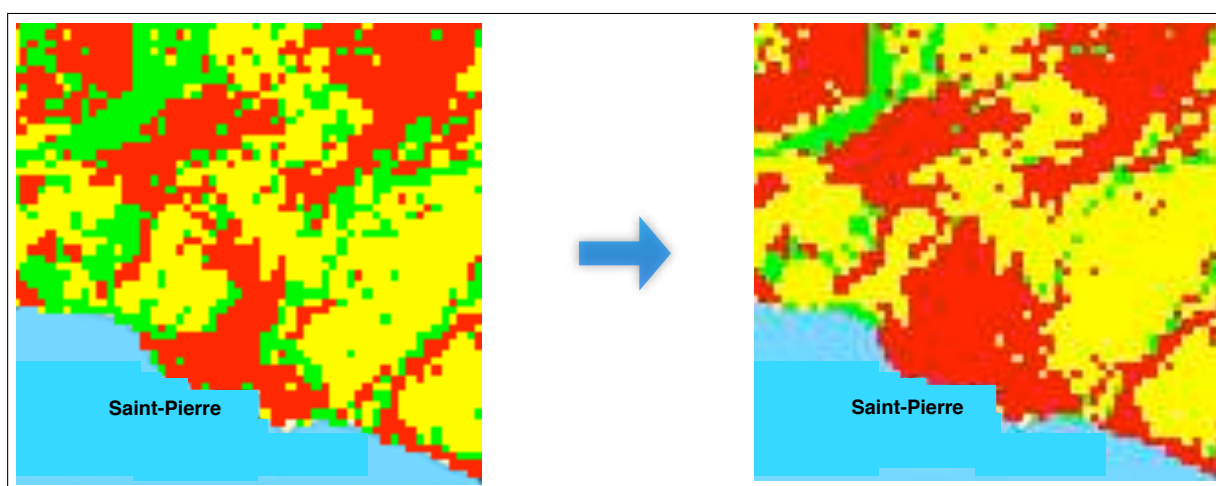


FIG. 8.8 – Une simulation sur la zone de Saint-Pierre (à gauche l'état initial en 2003, à droite l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.

Sur cette figure, on constate une évolution importante de l'urbanisation, mais on constate également par rapport à l'exemple de Cambaie qu'il est ici plus difficile de distinguer et délimiter visuellement le phénomène émergent correspondant à une nouvelle zone urbaine (ou même tout simplement de dire si nous sommes en présence d'un tel phénomène), ce qui rend d'autant plus pertinent l'utilisation des mécanismes que nous proposons.

8.3.2 Analyse et mise en place architecturale

Le phénomène que nous venons de décrire correspond, en terme d'urbanisation, à un réel phénomène émergent dont l'étude mérite d'être approfondie. En particulier, il serait intéressant de rendre possible la détection de tels phénomènes dans DS, afin que ceux-ci ne soient pas remarqués qu'après une analyse (minutieuse et fastidieuse pour les thématiciens experts) des résultats et cartes produits par les simulations. Et il serait tout aussi intéressant de pouvoir prendre en compte l'émergence de ces nouvelles zones urbaines dans le système afin de tester les différentes hypothèses d'urbanisation qui leur sont liées et les conséquences qu'elles induisent.

Evidemment, et bien que les *cellules* qui la composent soient à présent toutes des *cellules* ayant un MOS urbain, toutes les nouvelles zones urbaines ne sont pas à considérer de la même manière, et on imagine facilement que les éventuels comportements qui leurs seraient associés puissent différer selon que l'urbanisation récente concerne, par exemple, des zones de logements ou des zones d'activités. Nous pouvons donc utiliser le cadre formel et architectural que nous avons décrit au chapitre 5 de ce manuscrit pour, dans un premier temps, parvenir à détecter les nouvelles zones urbaines qui émergent dans DS, puis, dans un deuxième temps, pour parvenir à les matérialiser afin d'expérimenter différentes hypothèses.

Afin de réaliser le processus de réification de ces nouvelles zones urbaines, nous avons donc mis en œuvre une technique de réification mécanique externe, telle que nous l'avons décrite au chapitre 4. Nous avons fait le choix de cette technique mécanique car elle se situe au centre de la classification des types de réification possibles que nous avons résumés sur la figure 4.7, et sa mise en place n'est donc ni trop triviale, ni trop dépendante de l'introduction d'algorithmes cognitifs poussés. Par ailleurs, ce choix nous offre un cas concret d'utilisation pour les structures d'émergence que nous avons définies au chapitre 5. Enfin, étant dans une pure démarche expérimentale, nous avons choisi d'effectuer cette réification de façon externe au modèle conceptuel du domaine, afin de ne pas perturber le fonctionnement stable du modèle DS qui a acquis ainsi une certaine validité.

Dans la suite nous allons donc décrire la phase de détection et la phase de matérialisation dont la réalisation aboutit à la réification du phénomène qui nous intéresse. Le processus de réification étant réalisé en suivant une technique mécanique externe, il se fait donc au moyen de services plateforme que nous avons mis en place dans GEAMAS-NG au niveau système. Conformément au cadre décrit dans le chapitre 5, la phase de détection est vue comme une introspection, la phase de matérialisation est vue comme une phase d'intercession, et la mise en place des éléments relatifs à ces deux phases doit se faire en tenant compte des spécificités de DS.

8.3.3 Détection dans le modèle DS

La première étape du processus de réification des nouvelles zones urbaines dans DS est donc de parvenir à détecter la formation de ces zones au cours des simulations. Ceci se fait au moyen de services plateforme que nous avons mis en place dans GEAMAS-NG auxquels – en tant qu'utilisateurs du système – nous devons passer les éléments permettant de décrire le phénomène émergent que constitue une nouvelle zone urbaine.

Suivant le cadre formel décrit au chapitre 5, il faut donc fournir une fonction f qui fera office de révélateur d'émergence associé au phénomène des nouvelles zones urbaines. Cette fonction f appartient donc à l'ensemble \mathcal{R}_E des éléments qui sont utilisés en tant que révélateurs d'émergence des phénomènes qui apparaissent dans les simulations effectuées avec DS. Et elle dépend ainsi directement des faits qui se passent dans ces simulations.

Peu importe l'éventuelle complexité d'analyse et de compréhension des processus internes au modèle DS qui ont conduit à l'émergence de nouvelles zones urbaines, les faits qui ca-

ractérisent l'apparition de ces zones appartiennent, eux, à l'ensemble \mathcal{K} des faits observables pendant une simulation. Pour parvenir à détecter notre phénomène émergent, il nous faut donc :

- définir le révélateur d'émergence qui y est associé ;
- construire et analyser un ensemble de faits qui se produisent dans DS.

Le premier point se fait au niveau des utilisateurs de DS (thématiciens, concepteurs, décideurs, etc.), et le second se fait au niveau des services plateforme présents dans GEAMAS-NG.

Définir les révélateurs d'émergence

La première des choses à faire pour détecter les nouvelles zones urbaines est donc de définir le révélateur d'émergence associé à ce phénomène. Ceci peut paraître surprenant puisqu'on devrait logiquement commencer par construire l'ensemble \mathcal{K} des faits qui se passent dans DS avant de pouvoir soumettre cet ensemble au filtre que constitue l'ensemble \mathcal{R}_E et qui aboutira à l'ensemble \mathcal{P}_E dans lequel ne seront alors présentes que les fonctions liées aux faits qui décrivent les phénomènes qui nous intéressent. Mais au niveau de la logique implémentatoire on peut légitimement s'interroger sur l'intérêt qu'il y aurait à construire un ensemble \mathcal{K} comprenant l'ensemble des faits qui se passent dans DS si l'on sait déjà que l'analyse d'une partie de ces faits ne présente pas d'intérêt dans la suite de la démarche. L'ensemble \mathcal{K} sera donc construit en n'incluant que les faits dont le type est présent dans l'ensemble de départ des différentes fonctions appartenant à \mathcal{R}_E .

On commence donc par définir le révélateur d'émergence associé au phénomène des nouvelles zones urbaines, en sachant que ce phénomène se traduit généralement par l'urbanisation rapide et conjointe d'un ensemble de cellules situées dans une même zone géographique. Nous devons donc traduire cela en terme de faits pouvant être observés dans le système ; faits que nous devons exprimer sous une forme logique afin d'illustrer le cadre formel du chapitre 5.

Définir les faits révélant l'émergence du phénomène

Soit la fonction $MOS : \mathbb{N}^3 \rightarrow \{U, A, N\}$ qui prend en entrée un numéro de ligne i , un numéro de colonne j et une année a , et qui renvoie naturellement le Mode d'Occupation des Sols (Urbain, Agricole, ou Naturel) de la cellule située à la ligne i et à la colonne j à un instant t .

L'utilisation de cette fonction MOS nous permet de définir les faits k_1 et k_2 suivants :

$$k_1(i, j) : MOS(i, j, t) = U$$

$$k_2(i, j) : MOS(i, j, t - p) = \neg U$$

Relativement à la notation (clé ; valeur) proposée au chapitre 5, ceci nous donne :

$$k_1(i, j) : (MOS(i, j, t); U)$$

$$k_2(i, j) : (MOS(i, j, t - p); \neg U)$$

Ces faits permettent :

- pour k_1 , de définir si une *cellule* donnée, située à la ligne i et à la colonne j , est une *cellule* urbanisée ($MOS = U$) à l’instant t ;
- pour k_2 , de définir si cette même *cellule* était auparavant libre de toute construction ($MOS = -U$) dans un délai précédent l’instant t défini par p .

Ces deux faits peuvent nous servir afin de définir les *cellules* qui ont été urbanisées au cours de la simulation. Par exemple la fonction $f \in \mathcal{R}_E$ telle que $f(k_1, k_2) = k_1 \wedge k_2$ est un révélateur d’émergence qui serait activé dès l’observation des faits k_1 et k_2 et qui pourrait être utilisé si l’on considérait que la mise en évidence des toutes les *cellules* nouvellement urbanisées – peut importe leur localisation – constituait un phénomène émergent dans les simulations de DS. Ceci est illustré sur la figure 8.9 qui représente un résultat de simulation dans laquelle nous avons détecté l’ensemble des *cellules* (représentées en gris) qui n’étaient pas urbanisées à l’état initial et qui ont été urbanisées en cours de simulation. On remarque mieux sur cette figure le caractère spécifique de la zone centrale dans laquelle les *cellules* nouvellement urbanisées sont toutes regroupées et relèvent du phénomène des nouvelles zones urbaines qui nous intéresse.

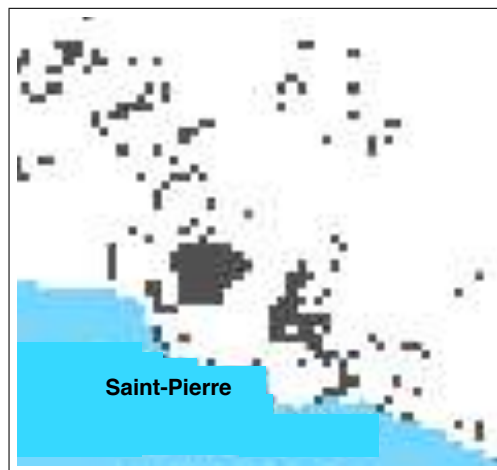


FIG. 8.9 – Toutes les cellules urbanisées au cours d’une simulation (en gris)

Mais pour créer le révélateur d’émergence associé au phénomène émergent des nouvelles zones urbaines, nous devons prendre en compte l’effet de groupe qui se traduit par l’urbanisation de plusieurs cellules avoisinantes. Nous devons donc appliquer k_1 et k_2 à des *cellules* proches les unes de autres, et si nous avons introduit la variable p c’est pour pouvoir fixer la plage temporelle (par exemple de 2, 5, ou 10 ans) dans laquelle on considère que des cellules sont en quelque sorte les victimes du même phénomène d’urbanisation.

Nous introduisons donc les faits $k_{x,y}$ qui vont nous permettre de tester si le fait $k_1(x, y) \wedge k_2(x, y)$ est observé pour chacune des *cellules* situées à la ligne x et à la colonne y en considérant $(x, y) \in \mathbb{N}^2, i - 1 \leq x \leq i + 1$, et $j - 1 \leq y \leq j + 1$, ce qui fait donc référence aux sept cellules voisines de celle située à la ligne i et à la colonne j .

Nous introduisons enfin le fait $k_n : nombre(k_{x,y}) \geq n$. Ce fait k_n utilise la fonction *nombre*

qui est à disposition des mécanismes d'observation en place dans GEAMAS-NG et qui permet de renvoyer le nombre de faits qui sont effectivement observés parmi ceux qui interviennent dans la clé fournie. Ce fait nous indique ainsi si les faits k_1 et k_2 sont observés pour suffisamment de *cellules* voisines à la *cellule* située à la ligne i et à la colonne j , signifiant ainsi qu'elles ont toutes été urbanisées au cours de la même plage temporelle. La valeur de n permet de décider du nombre minimal de *cellules* vérifiant les critères pour que les *cellules* de la zone soient considérées comme constituant le phénomène émergent ; en ce sens cette valeur fait le lien avec la densité d'urbanisation de la zone.

Notons que nous redéfinissons ici k_2 comme étant $k_2(i, j) : MOS(i, j, abs(t - p)) = -U$. Nous avons ainsi remplacé le paramètre $t-p$ par sa valeur absolue, afin que les *cellules* qui vérifient k_1 et k_2 ne soient pas seulement celles pour lesquelles les voisines vérifiaient déjà les conditions, mais aussi celles pour lesquelles les voisines seraient amenées à vérifier les conditions dans un futur proche.

Précisons que la formalisation que nous avons détaillée en exemple, choisie pour sa relative simplicité d'expression dans notre cadre expérimental, n'est évidemment pas unique puisqu'il existe de multiples façons différentes – parfois plus pratiques, notamment en ce qui concerne l'implémentation – de représenter sous une forme logique notre phénomène des nouvelles zones urbaines.

Détecter le phénomène

Tous les faits d'urbanisation que nous venons de lister relèvent de l'état du MOS des *cellules*. Ces *cellules* étant des objets de l'environnement de DS, nous utilisons donc un des services que nous avons mis en place dans GEAMAS-NG afin de détecter leur apparition éventuelle à chaque mise à jour de l'environnement. Ce service prend la forme d'un service plateforme spécifique, qui n'intervient nullement dans les entités de la simulation, et n'a donc aucune interaction avec les agents constituant le modèle de base de DS. Ce service possède en revanche une vision globale de l'environnement.

Il nous permet ainsi simplement de passer en revue les *cellules* de notre environnement spatial, individuellement ou collectivement, afin de vérifier si elles correspondent aux critères activant le révélateur d'émergence que nous avons défini. C'est donc ce service qui permet de prendre en charge la détection des faits qui sont présents dans les révélateurs d'émergence qui décrivent les phénomènes émergents. Par exemple, relativement au phénomène des nouvelles zones urbaines, à un instant "t" c'est donc lui qui va tester si, pour chaque objet de type "cellule", la valeur de son attribut "MOS" est égale à "U", vérifiant ainsi l'existence du fait k_1 . Et il en va de même pour vérifier l'existence des autres faits. Précisons que nous restons ici assez génériques, mais que dans GEAMAS-NG l'utilisation des mécanismes fournis par Pilobs – le Pilote d'observation de la plateforme dont nous avons parlé au chapitre 5 – permet à notre service de détection de ne pas avoir à examiner constamment l'évolution de la simulation. Il lui suffit d'utiliser Pilobs pour suivre l'état des variables qui apparaissent dans les faits qui

sont utilisés pour définir les révéléteurs d'émergence.

Globalement, par rapport à notre cadre formel, la détection de ce phénomène signifie que la fonction $f \in \mathcal{R}_E$ associée à ce phénomène renvoie "true", et qu'ainsi elle appartient également à l'ensemble \mathcal{P}_E , qui regroupe les révéléteurs d'émergence qui ont été activés par des faits observés dans la simulation, et qui fait donc ressortir les phénomènes ayant réellement émergé.

Dans notre exemple le révéléteur d'émergence est la fonction f qui renvoie "true" si tous les faits que nous avons définis (k_1 , k_2 , tous les $k_{x,y}$ et le k_n associé) sont observés. L'ensemble \mathcal{P}_E contiendra alors les couples constitués, d'une part, de cette fonction et, d'autre part, des faits ayant activé le révéléteur d'émergence. Chacun de ces couples pointe ainsi vers une cellule donnée.

La figure 8.10 illustre ainsi la nouvelle zone urbaine qui est constituée d'une trentaine de d'objets de l'environnement de type *cellule* qui ont tous été détectés en utilisant ce révéléteur d'émergence détectant les *cellules* qui sont au moins au nombre de 5 (la valeur n), dans un même voisinage, à avoir vu leur urbanisation effectuée dans une même période temporelle de 5 ans (la valeur p).

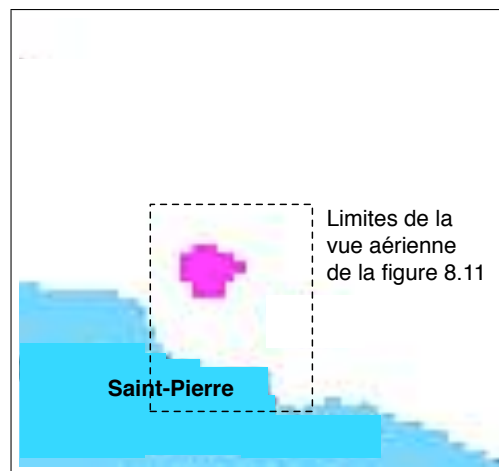


FIG. 8.10 – Le phénomène émergent "nouvelle zone urbaine" qui a été détecté (en rose)

On voit bien ressortir sur cette figure l'ensemble des cellules regroupées qui font partie, ensemble, de la nouvelle zone urbaine détectée. Cette zone est d'ailleurs bien connue localement puisqu'elle correspond à une récente Zone d'Aménagement Concerté de la commune de Saint-Pierre : la ZAC Canabady dont une vue aérienne est donnée sur la figure 8.11.

Sur cette figure, on distingue bien l'agglomération de Saint-Pierre, déjà fortement urbanisée, et la zone encore peu urbanisée dans laquelle notre phénomène émergent se produira dans nos simulations (la zone encadrée) mais dans laquelle on distingue déjà dans la partie de droite les premiers bâtiments de la ZAC Canabady. Sur cette vue aérienne on remarque également les nombreuses parcelles de terres agricoles cultivées qui, selon les choix effectués dans les différents scénarios de simulations, pourraient donc être vouées à une urbanisation dans les années à venir.

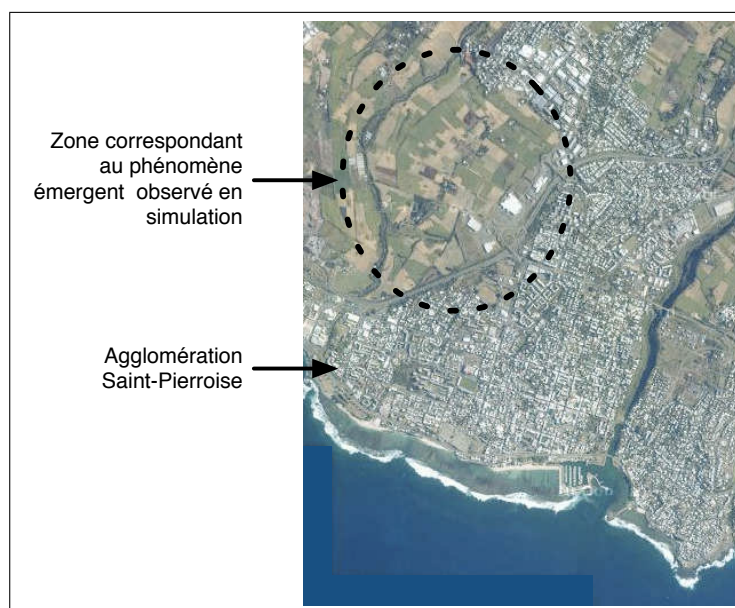


FIG. 8.11 – Vue aérienne actuelle de la zone correspondant à notre phénomène émergent (photographie provenant de Google Maps)

La détection ainsi faite des nouvelles zones urbaines qui sont susceptibles d'apparaître dans les simulations effectuées avec DS nous permet de fournir un appui aux thématiciens spécialistes chargés d'analyser les résultats des simulations. En ce sens, cette expérimentation constitue donc une proposition d'avancée importante par rapport aux possibilités offertes par les versions stables et utilisées du modèle DS. Pour aller plus loin que la "simple" détection assistée, nous allons à présent montrer comment les nouvelles zones urbaines qui ont émergé peuvent être matérialisées dans les simulations.

8.3.4 Matérialisation dans le modèle DS

Nous venons de voir que nous pouvons détecter l'émergence de nouvelles zones urbaines dans les simulations effectuées avec DS. Relativement au cadre formel et architectural que nous avons décrit au chapitre 5, nous pouvons donc compléter le processus de réification de ces nouvelles zones urbaines en accomplissant la phase de matérialisation qui leur est associée. Ceci permet essentiellement de mieux prendre en compte l'existence de ces phénomènes dans les simulations, et nous allons l'illustrer en poursuivant l'étude de l'exemple donné précédemment, à savoir l'émergence – selon certains scénarios de simulation – d'une nouvelle zone urbaine dont l'actuelle ZAC Canabady de la commune de Saint-Pierre fait aujourd'hui partie.

Par rapport aux concepts que nous avons mis en avant dans le chapitre 5, la phase de matérialisation de nos nouvelles zones urbaines passe par deux étapes :

- la définition des lois d'émergence de l'ensemble \mathcal{L}_E ;

- la création des structures d'émergence donnant corps aux nouvelles zones urbaines détectées.

Comme pour la phase de détection, le premier point se fait au niveau des utilisateurs de DS (thématiciens, concepteurs, décideurs, etc.), et le second se fait au niveau des services plateforme présents dans GEAMAS-NG. Mais le lien entre ces deux points est extrêmement étroit.

En effet, comme nous l'avons détaillé dans le chapitre 5, il est parfois très difficile de caractériser les conditions et processus qui conduisent à l'émergence d'un phénomène, alors qu'il est plus facile d'exprimer des faits qui ne font que révéler l'existence de ce phénomène, et c'est d'ailleurs ce que nous exploitons dans la phase de détection. De la même façon, il est difficile de parvenir à caractériser les répercussions de l'existence d'un phénomène particulier sur les entités de la SOA et sur le système dans sa globalité. Les mécanismes que nous proposons, qui utilisent des lois d'émergence pour déclencher la création de structures d'émergence et en gouverner le fonctionnement, impliquent donc des descriptions de comportements et des réflexions qui entraînent souvent de se plonger à nouveau dans le code de la SOA, même si les entités du modèle simulé ne sont pas modifiées directement.

Définir les lois d'émergence et les structures d'émergence associées

La première des choses à faire pour matérialiser les nouvelles zones urbaines qui émergent dans une simulation est donc de définir les lois d'émergence associées à ce type de phénomène tout en s'interrogeant sur la façon dont ce phénomène sera intégré à la SOA en cours au moyen des structures d'émergence.

On peut notamment s'interroger, selon la place que l'on veut donner au phénomène, si un phénomène émergent doit être matérialisé en utilisant uniquement des éléments d'interposition, s'il est nécessaire d'avoir recours de façon unique à un agent d'émergence, ou si on doit s'orienter vers une utilisation conjointe des deux types de structures. Rappelons que si les éléments d'interposition permettent simplement de modifier les influences et perceptions (et donc au bout du compte le comportement) des agents de la SOA qui seraient concernés par l'émergence de phénomènes, les agents d'émergence permettent, eux, de donner un comportement propre au phénomène lui-même.

Dans notre exemple des nouvelles zones urbaines, nous avons donc commencé par analyser la façon dont les entités du système sont concernées par ce phénomène. Ainsi, comme l'illustre la figure 8.12, on remarque qu'il y a naturellement des *cellules* de l'environnement et des *agents parcelles* qui sont dans sa zone géographique et qui sont directement concernées par l'émergence d'une nouvelle zone urbaine, tandis que les *agents Parcelles* et les *cellules* qui en sont éloignées ne sont pas directement concernées par son émergence.

Si nous voulons que la matérialisation des nouvelles zones urbaines dans la SOA serve à quelque chose il va donc de soi que nous devons, au moins, mettre en place des éléments d'interposition avec les *agents Parcelles* qui sont concernés par l'émergence du phénomène. Ceci nous permettra d'ailleurs de tester différentes hypothèses influant sur l'évolution de ces

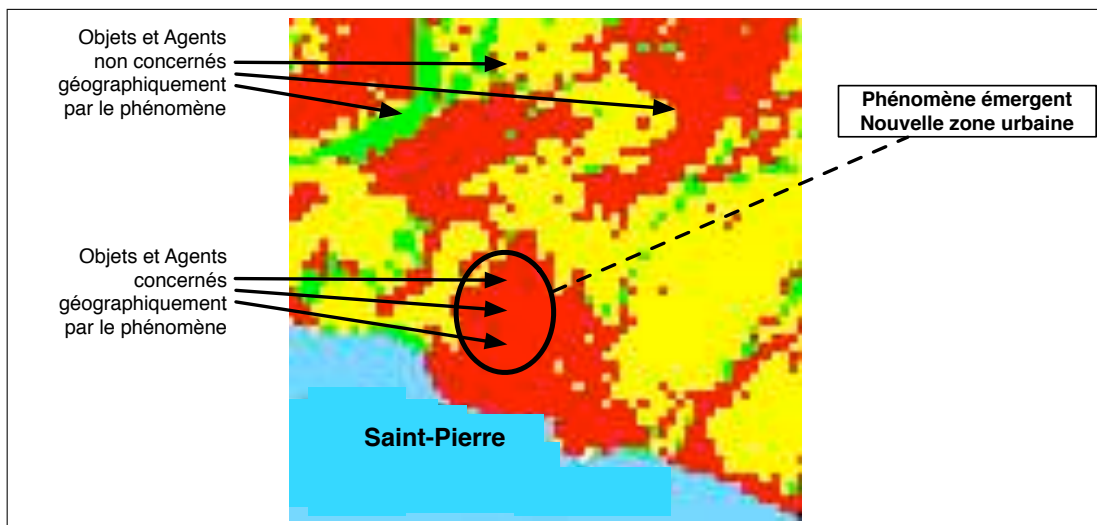


FIG. 8.12 – Le phénomène et les entités (*agents* et *cellules* d'environnement de DS) concernées géographiquement par son émergence

agents qui étaient internes au modèle de DS sans pour autant les modifier directement.

Mais n'oublions pas que les entités les plus petites ne sont pas les seules à être concernées par l'émergence d'une nouvelle zone urbaine. En effet, comme nous l'avons vu au début de ce chapitre, DS est composé de différents niveaux d'agents et chaque nouvelle zone urbaine émerge donc au sein d'une micro-région particulière, et au sein de la région Réunion elle-même. Il nous faut donc prendre ceci en compte au niveau des agents correspondants, et mettre en place des éléments d'interposition pour l'*agent Région* et pour chaque *agent Microrégion* concerné par l'émergence du phénomène.

Enfin, nous pouvons donner corps à nos nouvelles zones urbaines en mettant en place, pour chaque nouvelle zone urbaine, un agent d'émergence qui permettra d'attribuer un comportement spécifique au phénomène et qui pourra interagir avec l'environnement de DS *via* ses propres influences et perceptions. Ceci nous permettra là aussi de tester différentes hypothèses d'évolution.

Mise en place architecturale

Dans notre expérimentation, nous avons choisi de réifier chaque nouvelle zone urbaine qui émergerait dans une simulation et y serait détectée en utilisant :

- un agent d'émergence ;
- un élément d'interposition utilisé pour les *agents Microrégions* (dans notre exemple uniquement pour l'*agent Microrégion* Sud mais il est possible qu'une nouvelle zone urbaine émerge à cheval sur les territoires de plusieurs *agents Microrégions*) ;
- un élément d'interposition utilisé pour les *agents superviseurs* (dans notre exemple uniquement pour l'*agent superviseur Urbain*).

Ces éléments d'interposition nous suffisent puisque nous considérons dans nos expérimentations qu'il est raisonnable de penser que tous les agents d'un même type qui sont concernés par un phénomène émergent seront affectés de la même manière. En revanche il paraît évident que des agents de types différents (et d'échelles différentes) seront affectés de façon différente. Dans notre exemple ces choix s'expliquent car pour la nouvelle zone urbaine détectée à Saint-Pierre l'agent *Microrégion* Sud et l'agent *superviseur Urbain* sont ceux qui sont au cœur du fonctionnement du processus d'urbanisation. Ainsi la figure 8.5 qui présentait le couplage des deux dynamiques de DS via l'environnement peut être reprise, de façon spécifique pour la zone terrestre concernée par notre nouvelle zone urbaine, pour donner la figure 8.13.

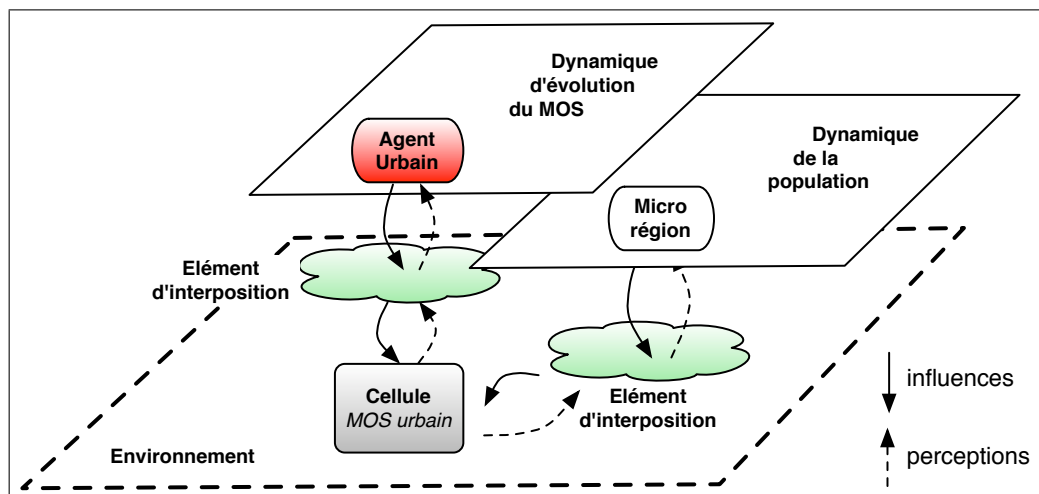


FIG. 8.13 – Le couplage des dynamiques dans la zone géographique de la nouvelle zone urbaine

Pour ce qui est de l'agent d'émergence représentant le phénomène, il va nous permettre, au travers du comportement qui lui sera donné, de tester différentes hypothèses qui permettront d'affiner le comportement général de DS relativement à l'émergence de ce phénomène en particulier. Nous l'appellerons ici *agent gestionnaire d'urbanisation*.

La mise en place architecturale de ces différentes structures d'émergence est synthétisée sur la figure 8.14 qui reprend le style de la figure 5.5 du chapitre 5 qui présentait cette mise en place architecturale dans un cadre général.

Toutes ces structures d'émergence relatives à un même phénomène émergent sont dirigées par une structure d'émergence particulière : sa méta-structure m_{sE} dont nous parlions au chapitre 5. Dans notre architecture les différentes m_{sE} sont toutes regroupées par un service plateforme qui est en charge de matérialiser les phénomènes émergents. Ce service prend simplement en charge la création et la mise en place des différentes structures d'émergence, et évidemment leur suppression lorsque les phénomènes émergents qui y sont associés n'existent plus. Il est ainsi le garant de la cohérence de la matérialisation. Ceci est représenté sur la figure 8.15.

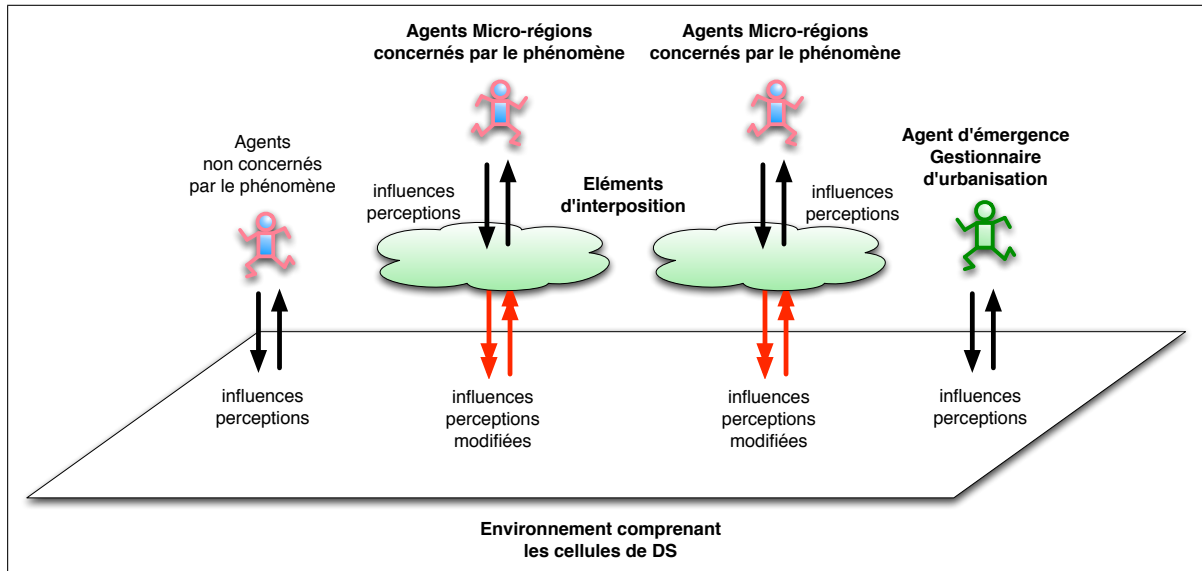


FIG. 8.14 – La mise en place du phénomène émergent des nouvelles zones urbaines dans DS sur le plan architectural

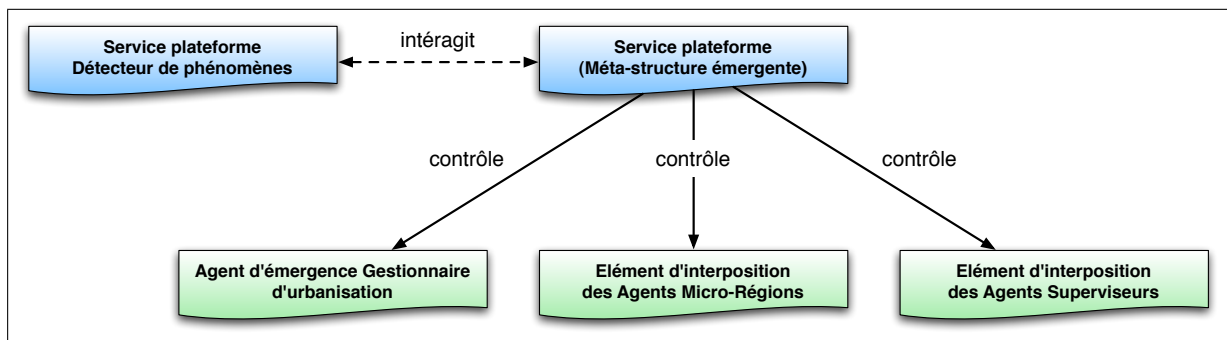


FIG. 8.15 – Les services plateforme en charge de la détection et de la matérialisation

Ce service plateforme est en relation avec celui dont nous avons parlé plus tôt lors de la phase de détection et ils sont donc les deux garants du processus complet de réification de notre phénomène.

Dynamique de détection et de matérialisation

Le fonctionnement des deux services plateforme qui nous permettent de réaliser le processus complet de réification de nos nouvelles zones urbaines, en suivant une technique mécanique externe au modèle du domaine de DS, donne donc une vision synthétique de ce processus.

Dans une simulation, à chaque mise à jour de l'environnement, le service plateforme en charge de la détection des phénomènes émergent effectue les actions suivantes :

- analyse des faits qui doivent être pris en compte dans le révélateur d'émergence des nouvelles zones urbaines (les faits k_1 , k_2 , k_3 relatifs à nos nouvelles zones urbaines concernent les *cellules* de DS et impliquent par exemple d'avoir accès à leur MOS, mais pas à leur potentiel agricole) ;
- construction d'un ensemble de faits \mathcal{K} (restreint aux types des faits précédents) en parcourant l'ensemble des *cellules* de DS ;
- analyse de cet ensemble \mathcal{K} pour déterminer pour chaque *cellule* de DS si les faits k_1 , k_2 , k_3 sont observés et donc si la fonction qui sert de révélateur d'émergence à nos nouvelles zones urbaines renvoie "true" ;
- construction de l'ensemble \mathcal{P}_E dans lequel seront regroupées les fonctions décrivant les nouvelles zones urbaines ayant émergé (ce qui permet par exemple ensuite de faire ressortir ces zones visuellement pour un observateur de la simulation) ;
- transmission de \mathcal{P}_E à l'agent plateforme en charge de la matérialisation des nouvelles zones urbaines.

Et dans la même simulation, à chaque mise à jour de l'environnement, le service plateforme qui gère l'ensemble des structures émergentes effectue les actions suivantes :

- réception de l'ensemble \mathcal{P}_E (qui fait donc référence aux nouvelles zones urbaines détectées) transmis par l'agent plateforme en charge de la détection des nouvelles zones urbaines ;
- analyse de cet ensemble \mathcal{P}_E relativement aux lois d'émergence de l'ensemble \mathcal{L}_E transmises par les thématiciens ;
- pour chaque nouvelle zone urbaine détectée, création de son *agent d'émergence gestionnaire d'urbanisation* et création des différents éléments d'interposition associés ;
- pour chaque nouvelle zone urbaine qui n'est plus détectée, destruction éventuelle de son *agent d'émergence gestionnaire d'urbanisation* et destruction éventuelle des différents éléments d'interposition associés.

Ces deux services sont donc en relation étroite. Et il va de soi que la phase de détection devra être réalisée plusieurs fois de façon séparée de la phase de matérialisation afin que les experts puissent analyser les expérimentations qu'ils ont envie de mettre en place dans le système une fois que des phénomènes émergents y ont été découverts. En effet, comme nous l'avons déjà dit plusieurs fois, si la phase de détection ne demande qu'une description des faits qui permettent de détecter un phénomène émergent, la phase de matérialisation demande elle une réflexion plus profonde pour savoir quelles nouvelles entités (et munies de quels comportements) on souhaite injecter dans le système. Et c'est là ce que nous proposons de faire avec nos structures d'émergence, qui certes doivent être programmées, mais qui permettent de mettre en place des nouveaux éléments dans un système sans pour autant modifier le code des entités constituant le modèle initial.

De l'intérêt de tester différentes hypothèses de matérialisation

Le principal intérêt qui se dégage de la démarche de réification des nouvelles zones urbaines est de permettre à un utilisateur du modèle DS de tester différentes hypothèses qui permettent d'affiner les comportements du modèle pour tenir compte des spécificités liées aux phénomènes qui ont été mis en évidence.

En effet, comme nous l'avons vu dans les premières sections de ce chapitre, le modèle DS permet dans sa version d'origine de prendre en compte des comportements à l'échelle de la parcelle, de la région et des micro-régions. Il est clair que les centaines de milliers d'*agents Parcelles*, de petite taille, et qui ont tous des spécificités qui sont liées à leur localisation, ont donc une finesse suffisante pour supposer que les traitements qu'ils effectuent sont adéquats relativement à ces spécificités. Mais pour des agents d'échelle supérieure, comme les *Microrégions*, qui sont au cœur de la hiérarchie d'agents *Parcelles/Microrégions/Région*, les comportements initiaux prévus dans DS sont parfois trop généraux pour tenir compte de spécificités comme les nouvelles zones d'urbanisation.

Avec les structures d'émergence que nous avons mises en place nous pouvons donc, d'une façon indirecte, modifier l'évolution des entités du modèle DS en modélisant puis programmant de nouvelles entités qui viennent s'insérer dans le modèle initial.

Sur notre exemple, les expériences que nous avons menées nous permettent de tester différentes hypothèses affinant le comportement de l'*agent superviseur Urbain* et de l'*agent Microrégion Sud* dans la zone correspondant à notre nouvelle zone urbaine. De façon concrète, si l'on souhaite faire de cette zone une zone commerciale plutôt qu'une zone de logements, l'élément d'interposition utilisé avec l'*agent Microrégion Sud* permet de faire en sorte qu'il ne perçoive plus les corps des *agents Parcelles* concernés par la nouvelle zone urbaine et qu'il ne puisse donc plus les prendre en compte lorsqu'il cherche à répartir sur le territoire la nouvelle population calculée (ce qui se fait à l'étape 9 de la figure 8.6). Et l'*agent gestionnaire d'urbanisation* de la nouvelle zone urbaine permet de modifier les *cellules* d'environnement qui sont sur le territoire de la nouvelle zone urbaine afin d'augmenter par exemple le paramètre d'attractivité

du sol qui leur est associé, ce qui permettrait alors de simuler une urbanisation de la zone par l'*agent superviseur Urbain* (ce qui se fait à l'étape 8 de la figure 8.6) qui serait encore plus rapide.

La figure 8.16 illustre ainsi les résultats de deux simulations dans lesquelles l'urbanisation rapide a été observée et dans lesquelles nous avons donc détecté la nouvelle zone urbaine présente sur la commune de Saint-Pierre, mais dans lesquelles nous l'avons matérialisée de deux façons différentes. Les deux images représentent à travers un dégradé de couleur rouge la densité de population observée à la fin de deux simulations :

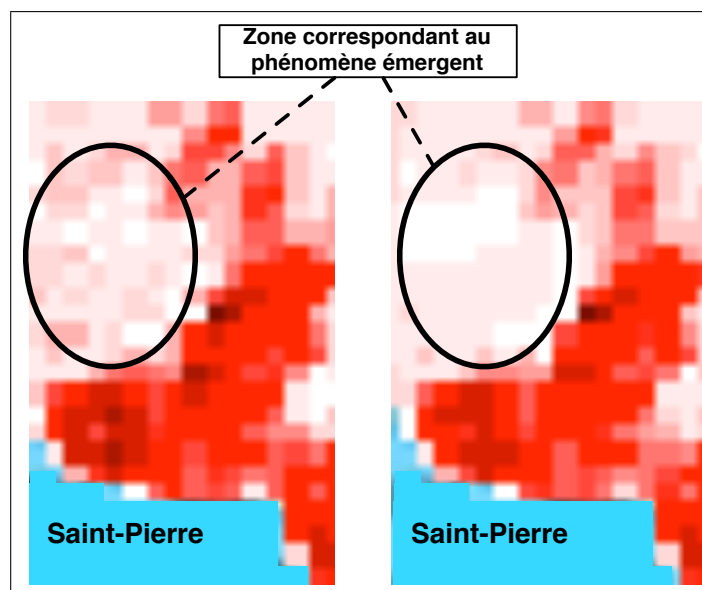


FIG. 8.16 – La densité de population observée dans les résultats de deux simulations (à gauche la nouvelle zone urbaine est matérialisée en considérant qu'elle est destinée à des logements, à droite elle est matérialisée en considérant qu'elle est destinée à des locaux d'entreprises)

- dans le premier cas, représenté sur l'image de gauche, la nouvelle zone urbaine est matérialisée en considérant qu'elle serait majoritairement dédiée à être occupée par des logements ;
- dans le second cas, représenté sur l'image de droite, la nouvelle zone urbaine est matérialisée en considérant qu'elle serait majoritairement dédiée à des bâtiments commerciaux et des locaux d'entreprises.

On remarque la différence de teintes rouges dans la zone encadrée, ce qui montre logiquement que la population a commencé à être répartie sur la zone dès son urbanisation dans le premier cas mais pas dans le second pour lequel, perturbé par un élément d'interposition, l'*agent Microrégion Sud* n'a pas attribué de nouvelle population aux *agents Parcelles* contenues dans la zone.

8.4 Conclusion

Dans ce chapitre, nous avons commencé par présenter le fonctionnement interne de DS en détaillant la façon dont sont articulées les deux dynamiques du modèle : la dynamique de population et la dynamique d'évolution du MOS. Nous avons présenté les différentes entités qui interviennent dans ces dynamiques et nous avons montré le rôle clé joué par l'environnement qui sert de point de couplage entre elles.

Ce couplage de dynamique est rendu possible par l'utilisation des MDM qui facilitent la réalisation de modèles généraux constitués de sous-modèles qui sont chacun consacrés à une dynamique particulière. Et la méthodologie de conception de simulations que nous avons présentée au chapitre 3, qui place la connaissance et le niveau de méta-connaissance des thématiciens experts de chaque dynamique au cœur du processus de modélisation, offre ainsi un cadre méthodologique qui facilite la réalisation de tels modèles.

Nous nous sommes ensuite focalisés sur l'étude des nouvelles zones urbaines, un phénomène particulier qui émerge dans les simulations effectuées avec DS. A titre expérimental nous avons donc détaillé comment il était possible de réaliser le processus de réification de ces nouvelles zones urbaines en nous appuyant sur la formalisation et sur le cadre architectural que nous avons décrits au chapitre 5. Cette réification qui se fait au moyen de mécanismes dédiés à la détection et à la matérialisation des nouvelles zones urbaines nous a ainsi permis de mettre en œuvre sur un exemple réel les propositions énoncées dans la deuxième partie de ce manuscrit.

Ces propositions n'ont pas pour ambition de fournir LA solution pour prendre en compte les phénomènes émergents dans les SOA puisque pour parvenir à réifier un phénomène, il faut, comme nous l'avons vu au travers de l'expérimentation des nouvelles zones urbaines et notamment pendant leur phase de matérialisation, passer par des étapes – qui sont parfois fastidieuses – d'analyse, de modélisation, puis de programmation des structures d'émergence que constituent les éléments d'interposition et les agents d'émergence.

Mais le déroulement de cette expérimentation montre que notre approche permet d'intégrer la prise en compte des phénomènes émergents dans des modèles de simulations qui ne l'avait pas prévue. Et nous pouvons ainsi étendre les fonctionnalités de modèles comme DS – dont la structure complexe fait qu'il est parfois difficile de modifier les comportements de certaines entités sans provoquer un déséquilibre global – afin d'affiner le comportement général pour tenir compte de spécificités nouvelles, ce qui nous place donc au cœur du processus d'injection de connaissances dans une simulation.

Conclusion générale

Prospective Territoriale par Simulation Orientée Agent

Synthèse des travaux

L'île de La Réunion, territoire ultra-marin de 2500km^2 en plein océan Indien, est réputée pour la beauté de ses paysages et pour sa biodiversité exceptionnelle qui lui valent d'être inscrite depuis 2010 au patrimoine mondial de l'UNESCO. Mais, au cours des prochaines années, l'île va devoir faire face à une importante augmentation de sa population ; augmentation dont les conséquences directes ou indirectes menacent la préservation des espaces agricoles et naturels actuels. Les documents de planification réglementant l'évolution de l'urbanisation dans l'île doivent donc tenir compte de ces critères afin de gérer au mieux l'évolution et la structuration du foncier local.

La mise en place d'outils d'aide à la décision s'avère ici être une nécessité afin de tenir compte du maximum d'alternatives possibles pour les avenir de l'île, et c'est là ce qui a entraîné la réalisation des travaux que nous avons présentés tout au long de ce manuscrit.

Dans un contexte de recherche pluridisciplinaire, l'un de nos objectifs principaux était donc de proposer des méthodes et outils facilitant les interactions entre les informaticiens et les thématiciens qui apportent leur expertise à la construction de modèles de simulation dédiés à la prospective territoriale. L'autre objectif étant bien évidemment de faire en sorte que de tels modèles puissent voir le jour, et qu'ils puissent éventuellement fournir des pistes de réflexions aux décideurs confrontés aux choix de demain afin d'aider à mettre en place une organisation des territoires qui soit la plus cohérente possible.

Nous avons ainsi participé à la construction de DS, un modèle qui permet de simuler conjointement l'évolution de la population et celle du Mode d'Occupation des Sols dans l'île. Les premières versions de ce modèle, réalisées pendant le projet DOMINO du CIRAD, ont été exploitées dans le cadre de la révision du Schéma d'Aménagement Régional par le Conseil Régional de La Réunion, permettant ainsi à ce projet d'obtenir une certaine valorisation sur le plan régional.

En tant qu'informaticiens spécialistes en modélisation et simulation de systèmes complexes, la réalisation de DS nous a permis de nous poser plusieurs questions relatives à la conception de modèles de simulations, et notamment sur la place qui était accordée dans ces modèles à la notion d'émergence. Nous avons donc consacré la majeure partie de nos travaux à l'étude de cette notion et, considérant qu'un phénomène émergent n'en est un qu'en fonction des connaissances de l'entité (réelle ou virtuelle) qui l'observe, nous avons défini l'émergence comme étant une méta-connaissance.

Cette vision des choses nous a permis de replacer la notion même de connaissance au cœur du processus de modélisation. Nous avons ainsi proposé une méthodologie de conception de simulations dans laquelle la méta-connaissance en général, et donc l'émergence en particulier, sont représentées explicitement. Dans cette méthodologie, nous avons mis en avant l'importance de différentes phases et nous avons souligné la nécessité, pour les concepteurs des modèles, de bien distinguer, d'une part, l'émergence de phénomènes et, d'autre part, la capacité à détecter, caractériser et manipuler ces mêmes phénomènes afin de décider ensuite de la façon dont leur émergence sera prise en compte.

Nous avons ensuite étendu cette méthodologie afin de cadrer de façon spécifique les choix à opérer qui permettent de distinguer les différentes façons de prendre en compte les phénomènes qui peuvent émerger dans une Simulation Orientée Agent. Pour cela nous nous sommes attardés sur le concept de réification des phénomènes émergents, un processus que nous avons défini comme étant la succession d'une phase de détection et d'une phase de matérialisation. De façon transversale, cette méthodologie nous a permis de proposer une classification des différents types de réifications possibles (technique manuelle, mécanique ou automatique ; et technique interne ou externe au modèle) des phénomènes émergents dans une SOA.

Afin de faciliter la mise en œuvre de telles réifications, nous avons proposé un cadre conceptuel (qui permet de formaliser la réalisation des étapes permettant de parvenir à la réification de phénomènes ayant émergé) et des éléments architecturaux (qui permettent de procéder à leur matérialisation dans une SOA). Cette proposition s'appuie sur la définition de révéléateurs d'émergence et de lois d'émergence qui permettent au système ou à ses entités de faire leur introspection afin de détecter les phénomènes qui apparaissent et qui sont ensuite matérialisés au cours d'une phase d'intercession. Ces révéléateurs et lois d'émergence fonctionnent en opérant, pour les premiers, sur l'ensemble \mathcal{K} des faits qui se produisent dans la SOA et, pour les seconds, sur l'ensemble \mathcal{P}_E qui se rattache aux phénomènes qui y auront été détectés. Cette formalisation est utilisée par les structures émergentes que sont les agents d'émergence et les éléments d'interpositions, deux types d'éléments architecturaux qui nous permettent de donner corps à des phénomènes émergents grâce à l'utilisation de mécanismes de détection et de matérialisation que nous avons intégrés dans GEAMAS-NG.

Enfin, nous avons montré la faisabilité et l'intérêt de nos propositions en expérimentant la réification de nouvelles zones urbaines, un phénomène particulier dont nous avons observé

l'émergence dans les simulations effectuées avec DS. Ceci nous a permis de montrer qu'il était possible d'étendre les fonctionnalités d'un modèle comme DS – dont la structure complexe fait qu'il est parfois difficile de modifier des mécanismes internes sans provoquer un déséquilibre global – pour affiner son comportement général afin de tenir compte de spécificités nouvelles.

D'une façon générale, nous espérons que les travaux que nous avons menés dans le cadre de cette thèse – que ce soit au niveau de nos avancées sur la prise en compte de l'émergence dans les SOA, au niveau du modèle DS (participation à son élaboration, mise en place de techniques d'initialisation, expérimentations liées à la réification de phénomènes émergents), ou au niveau des démarches méthodologiques visant à améliorer la conception de simulations dans des projets pluridisciplinaires – seront une source d'inspiration pour les chercheurs, décideurs et acteurs de terrain qui, à La Réunion ou ailleurs, se sentent concernés par l'avenir de leurs territoires et se placent dans une démarche de prospective territoriale similaire à la notre.

Contributions

Au regard des objectifs que nous nous étions fixés au début de ce travail de thèse, six contributions principales peuvent être mises en avant :

- le fait de poser l'émergence comme étant une méta-connaissance (chapitre 3, définition II.1) ;
- un cadre formel pour tendre vers une réification des phénomènes émergents (chapitre 5, ensembles \mathcal{K} , \mathcal{R}_E , \mathcal{L}_E , \mathcal{P}_E) ;
- une base architecturale permettant de représenter les phénomènes émergents dans une SOA (chapitre 5, agents d'émergence et éléments d'interposition) ;
- une classification des types de réifications possibles des phénomènes émergents (chapitre 4, figure 4.7) ;
- une méthodologie de conception de simulations (chapitre 3, figure 3.6) ;
- un panel d'outils pour la simulation à des fins de prospective territoriale (chapitres 6, 7 et 8, co-conception et expérimentations sur le modèle DS, le langage XELOC).

Les trois premières de ces contributions sont les résultats d'objectifs que nous nous étions fixés dans le but d'améliorer la prise en compte de l'émergence dans les SOA. Elles ont abouti à la mise en place d'éléments permettant à une plateforme de SOA ou à des entités de la SOA elles-mêmes de prendre conscience de l'émergence de phénomènes et de les matérialiser en ayant recours à la connaissance de thématiciens qui peuvent à présent mener des expériences manipulant les phénomènes qui émergent dans DS. Comme nous l'évoquerons dans les perspectives, cette solution est cependant perfectible puisque les pistes menant à une réification automatique des phénomènes émergents restent à approfondir.

La quatrième contribution, la classification des différents types de réifications possibles des phénomènes émergents, ne répond pas à un objectif fixé au début de ce travail, mais elle est

particulièrement intéressante puisqu'elle a elle-même émergé des réflexions sur la façon dont les phénomènes émergents devaient être pris en compte dans une SOA. Elle se rattache ainsi à notre problématique de fond.

Les deux dernières contributions permettent de répondre à notre objectif visant à faciliter la conception de modèles devant intégrer les connaissances de nombreux experts. La méthodologie de conception de simulations que nous avons proposée permet ainsi de placer la méta-connaissance au cœur du processus, rappelant ainsi que les connaissances introduites dans un modèle – de mêmes que celles qui y sont produites – n'ont du sens que pour les thématiciens possédant eux-mêmes un certain niveau de connaissance.

Les travaux menés sur DS nous ont quant à eux permis d'expérimenter la réification de phénomènes émergents, mais aussi de prouver qu'il était possible d'utiliser la technique des MDM afin de coupler des dynamiques différentes, conçues chacune de façon – relativement – indépendante avec les thématiciens concernés. Enfin, en permettant le chargement de cartes colorées sémantiques pour initialiser nos simulations, l'utilisation du langage XELOC facilite là aussi les interactions diverses.

Cet ensemble de contributions permet à nos travaux d'avoir une valeur sur le plan scientifique. Mais il est important de rappeler qu'au-delà de leur portée purement scientifique certains des éléments décrits dans cette thèse – et notamment tout ce qui touche au modèle DS – nous ont permis d'obtenir une valorisation sur le plan régional au travers notre participation à des projets comme la révision du Schéma d'Aménagement Régional de La Réunion. Ceci étant finalement un juste retour des choses, puisque le Conseil Régional de La Réunion a financé l'essentiel de ce travail de thèse.

Un ensemble de publications scientifiques ont été réalisées durant cette thèse et y sont directement ou indirectement rattachées :

- 3 articles publiés et présentés dans des conférences internationales ;
- 1 chapitre d'ouvrage ;
- 1 article publié dans une revue internationale ;
- 4 articles publiés et présentés dans des conférences nationales.

La majorité de ces articles touchent directement au sujet de cette thèse, et leur contenu est donc parfois repris, de façon diffuse, dans les différents chapitres de ce manuscrit. En revanche, certains d'entre eux concernent des travaux annexes qui ont été menés en parallèle à cette thèse et, s'ils touchent tous à la simulation, ils s'écartent cependant de nos problématiques directes. Ils ne sont donc pas (ou peu) repris dans ce document. C'est par exemple le cas des travaux menés afin de simuler l'évolution des tortues vertes dans la zone océan Indien. Et c'est également le cas de travaux basés sur une approche génie logicielle de type "génération procédurale" qui permet l'auto-génération d'environnements consistants, cohérents, non-cycliques et non-bornés qui peuvent être utilisés avec de telles propriétés dans les SOA.

Par ailleurs, un poster scientifique a été réalisé dans le cadre direct de cette thèse en vue d'être présenté aux Doctoriales de l'université de La Réunion ; ce poster a été repris pour servir de base illustrative à la page de garde de ce manuscrit.

Perspectives proches...

Les travaux que nous avons menés au cours de cette thèse ont apporté avec eux leur lot de nouvelles connaissances ; connaissances qui ont elles-mêmes apporté des solutions à quelques uns des problèmes que nous nous posions au départ. Mais ils ont aussi apporté avec eux leur lot de nouvelles questions, de nouveaux problèmes et de nouvelles idées dont l'approfondissement demandera plus ou moins de temps de cerveau disponible. Plusieurs pistes peuvent ainsi être avancées afin d'améliorer, à court ou moyen terme, certaines des propositions que nous avons énoncées tout au long de ce manuscrit.

Tout d'abord, concernant nos travaux sur l'émergence, il est évident que nos propositions visant à réifier les phénomènes émergents qui apparaissent dans une simulation peuvent être améliorées, notamment en ce qui concerne les techniques de réification mécaniques que nous avons identifiées. En effet, pour permettre à tout un chacun d'effectuer les expérimentations qu'il souhaite mener, il est indispensable d'améliorer les services plateformes mis en place dans GEAMAS-NG. Ceci permettrait de faciliter les interactions, d'une part, au niveau de la saisie des révéléteurs d'émergence de l'ensemble \mathcal{R}_E et des lois d'émergence de l'ensemble \mathcal{L}_E , et, d'autre part, au niveau de la description des structures d'émergence que sont les agents d'émergence et les éléments d'interposition. Ceci peut s'envisager, en termes de représentation, en utilisant un formalisme issu des différentes logiques (logiques de descriptions, logique des défauts, logiques temporelles) et, en termes d'implémentation, en utilisant une technique de programmation déclarative bien adaptée à l'unification de règles (par exemple en utilisant un langage comme Prolog)... ou un langage comme XELOC.

L'ensemble pourrait alors se faire au travers d'interfaces facilitant la saisie de règles comportementales pour un thématicien expert de son domaine mais n'ayant pas la connaissance informatique suffisante pour se plonger directement dans l'intégralité du code source d'un modèle implémenté sur une plateforme comme GEAMAS-NG. Dans leurs expérimentations les différents experts pourraient alors se focaliser d'autant plus sur les aspects qui les concernent ou les intéressent.

Ensuite, concernant nos travaux sur l'initialisation de simulations, le langage XELOC pourrait voir ses fonctionnalités étendues afin d'être utilisé comme standard dédié à la configuration de SOA. Particulièrement efficace pour décrire des scénarios de simulations, il gagnerait à être diffusé pour une utilisation qui soit extérieure à l'équipe SMART et qui ne soit pas restreinte aux seules applications développées avec GEAMAS-NG.

Ceci passe par le développement d'un interpréteur qui ne soit pas spécifique à notre pla-

teforme de simulation. Celui-ci pourrait être inclus au sein de plug-ins qui seraient alors être déclinés en fonction des principales plateformes de SOA existantes. L'utilisation d'un même langage pour initialiser des simulations effectuées avec différentes plateformes permettrait en outre de comparer les performances de chacune d'entre elles sur des cas d'école en étant sûrs de l'identité des configurations initiales.

Enfin, concernant nos travaux dédiés à l'aménagement du territoire, nous pouvons continuer à enrichir la base solide que constitue le modèle DS. Nos propositions méthodologiques ayant montré qu'il était possible – dans une certaine mesure, évidemment – d'étendre les comportements déjà en place dans DS, nous pouvons donc ajouter de nouveaux éléments (comme les phénomènes émergents que nous réifions) qui viennent affiner les dynamiques existantes, ou même ajouter de nouvelles dynamiques complètes en utilisant la technique des MDM.

C'est ce que nous avons déjà commencé à faire en ajoutant dans DS une nouvelle dynamique liée à la gestion de l'énergie sur le territoire réunionnais. Cette problématique est bien entendu liée à l'évolution de la population et à l'urbanisation, mais elle est aussi liée aux ressources naturelles disponibles (soleil, vent, houle) et elle nécessite donc de mener des réflexions avec des thématiciens experts dans des domaines différents.

EDMMAS (pour Energy Demand Management by MultiAgent Simulation) – un premier prototype issu de l'ajout de cette nouvelle dynamique dans DS – a ainsi d'ores et déjà vu le jour et a été présenté dans [Gangat *et al.* 2009a] et dans [Gangat *et al.* 2009b]. A terme, l'objectif est de réaliser un outil d'aide à la décision pour l'aménagement énergétique d'un territoire. Ceci permettrait, grâce aux différents scénarios de simulations, de tester les meilleurs emplacements pour l'implantation des nouvelles installations ou encore de mieux dimensionner l'infrastructure pour parer aux cas de dysfonctionnements et de maintenances qui peuvent survenir sur les différents équipements.

...et perspectives plus lointaines

Les exemples que nous venons de citer font partie des pistes qui pourraient nous permettre d'améliorer nos propositions et outils sur du court et du moyen terme. Mais certains des nouveaux problèmes, nouvelles questions et nouvelles idées qui ont émergé (sur nos écrans ou dans nos esprits) au cours de nos travaux sont à considérer sur une échelle temporelle plus importante, dépassant ainsi le cadre des suites directes de cette thèse.

C'est essentiellement le cas de nos travaux sur la notion d'émergence qui pourraient être approfondis. Le potentiel de nos propositions pourrait par exemple être accru en proposant des techniques permettant de réaliser une réification automatique des phénomènes qui auraient émergé dans une SOA. Ceci peut s'envisager en mettant en place des mécanismes cognitifs dans le système simulé, soit au niveau de services plateforme (qui permettent une réification externe au modèle initial), soit au niveau des entités de la SOA (qui permettent une réification

interne au modèle).

Cette possibilité est sans doute celle la plus difficile à mettre en œuvre, mais c'est aussi celle qui ouvre le plus de perspectives. Elle présuppose qu'au sein d'une SOA les agents eux-mêmes seraient capables, de façon autonome ou de façon coopérative, de faire leur introspection et, s'il en ressentent le besoin, de modifier le système en cours de simulation pour y faire apparaître de nouveaux éléments, structures ou entités grâce aux connaissances qu'ils auraient acquises. A ce titre elle constitue donc une piste privilégiée pour des recherches futures.

Enfin, et pour revenir à la prospective territoriale de façon plus concrète, nous avons bien conscience qu'un modèle comme DS ne pourra pas être indéfiniment augmenté (en y ajoutant de nouvelles dynamiques, comme ce que nous avons fait dans nos travaux sur EDMMAS en y introduisant une dynamique de gestion de l'énergie) ou affiné (en y ajoutant de nouvelles structures, comme nous le faisons en réifiant des phénomènes émergents sans pour autant déstabiliser le fonctionnement global du modèle). Il est donc important de mener à bien de nouveaux projets, alliant recherche fondamentale et développement d'applications, qui permettront à des outils comme DS de voir le jour.

Ceci est une nécessité puisque dans les années à venir la population, qu'elle soit réunionnaise ou mondiale, va continuer à augmenter; la conservation des milieux naturels, la préservation de la biodiversité, la valorisation des terres agricoles et l'accroissement de l'urbanisation constituent donc autant de thèmes soumis à de forts enjeux actuels et futurs. En tant qu'informaticiens, le fait de développer des outils de simulation, alimentant les réflexions liées à ces domaines, nous permettrait d'avancer dans le développement de notre plateforme GEAMAS-NG avec l'espoir d'aboutir à l'émergence de nouveaux concepts. Et, en tant que réunionnais, il est de notre responsabilité d'utiliser nos humbles connaissances pour penser aux futurs possibles de notre île.

La Réunion a la chance d'avoir une population et un territoire qui font d'elle un formidable terrain de jeu pour nous, experts en modélisation et simulation de systèmes complexes naturels et sociaux. Faisons en sorte qu'elle le demeure.

Bibliographie

- [Actif & Lardoux 2006] Nelly Actif et Jean-Marc Lardoux. *Projections démographiques 2030 : les dynamiques territoriales*. Économie de La Réunion, vol. 125, pages 9–24, 2006. 118, 135, 211
- [Agorah 2006] Agorah. *Une politique foncière, une des clefs pour aménager le territoire*. 2006. 4, 120
- [Ali & Zimmer 1997] Syed Mustafa Ali et Robert Zimmer. *The question concerning emergence*. CASYS'97, Abstract Book, First International Conference on Computing Anticipatory Systems, CHAOS, 1997. 48
- [Allen & Sanglier 1979] Peter Allen et Michèle Sanglier. *A Dynamic Model of Growth in a Central Place System*. Geographical Analysis, pages 256–272, 1979. 32
- [Aristote] Aristote. *Metaphysics*. In W. D. Ross, editeur, *The Works of Aristotle translated into English*, Volume VIII. Oxford University Press, Oxford, 1908. 39
- [Atlan 2000] Henri Atlan. *La finalité*. Hors série Science et Avenir, 2000. 47
- [Aurnague & Plenat 1996] Michel Aurnague et Marc Plenat. *La prefixation en é- et la relation de partie a tout*. Representation et Outils pour les bases lexicales. Morphologie Robustes, 1996. 49
- [Banks 2000] Jerry Banks. *Simulation in the future*. Proceedings of the 32nd conference on Winter simulation, pages 7–13, 2000. 15, 40
- [Benchimol 2005] Guy Benchimol. *What about metaknowledge?* www.knowledgeboard.com, 2005. 67, 68, 210
- [Benenson & Torrens 2004] Itzhak Benenson et Paul M. Torrens. *Geosimulation : Automata-based modeling of urban phenomena*. Chichester, 2004. 19
- [Berger 1959] Gaston Berger. *L'attitude prospective*. 1959. 27
- [Berlekamp *et al.* 2002] Elwin R. Berlekamp, John H. Conway et Richard K. Guy. *Winning ways for your mathematical plays*, volume 2nd edition. Academic Press, 2002. 44, 82
- [Bonabeau & Dessalles 1997] Eric Bonabeau et Jean-Louis Dessalles. *Detection and emergence : Emergence et explication*. Intellectica(Orsay. 1985), 1997. 41

- [Bonabeau *et al.* 1997] Eric Bonabeau, Guy Thelauraz, Jean-Louis De Neubourg, Serge Aron et Scott Camazine. *Self-organization in social insects*. Trends in Ecology and Evolution, vol. 12, pages 188–193, 1997. 43
- [Bone & Dragicevic 2010] Christopher Bone et Suzana Dragicevic. *Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management*. Computers, Environment and Urban Systems, vol. 34, no. 2, pages 162 – 174, 2010. 32
- [Bonneaud *et al.* 2007] Stéphane Bonneaud, Pascal Redou, Damien Thébault et Pierre Chevaillier. *Agents-Modèles pour la co-construction de simulations multi-modèles - Application aux écosystèmes exploités*. In Philippe Mathieu et Valerie Camps, éditeurs, Journées Francophones sur les Systèmes Multi-Agents (JFSMA'07), pages 119–128, Carcassonne, France, 2007. 32
- [Bonnefoy *et al.* 2001] Jean-Luc Bonnefoy, François Bousquet et Juliette Rouchier. *Modélisation d'une interaction individus, espace et société par les systèmes multi-agents : pâture en forêt virtuelle*. L'espace géographique, no. 1, pages 13–25, 2001. 32
- [Boschetti & Gray 2007] Fabio Boschetti et Randall Gray. *A Turing test for Emergence*. Advances in Applied Self-organizing Systems, pages 349–364, 2007. 40, 52, 53, 106, 210
- [Botta *et al.* 2007] Aurélie Botta, Williams Daré, Grégoire Leclerc, Sigrid Aubert, Rémy Courdier, Ibrahima Diop Gaye, Gilles Lajoie et Jean-Pierre Müller. *DOMINO : Démarche Objet Multi-site pour l'analyse des Interactions entre Niveaux d'Organisation*. Rapport Scientifique, 2007. 145
- [Botta *et al.* 2009] Aurélie Botta, Williams Daré, Martine Antona et Grégoire Leclerc. *Integration of multi-scale stakes in governance by applying companion modelling to land-use foresight*. Eighteenth World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, pages 2377–2383, 2009. 118
- [Bousquet & Le Page 2004] François Bousquet et Christophe Le Page. *Multi-agent simulations and ecosystem management : a review*. Ecological Modelling, vol. 176, no. 3-4, pages 313–332, 2004. 32
- [Bousquet *et al.* 1998] François Bousquet, Innocent Bakam, Hubert Proton et Christophe Le Page. *Cormas : Common-Pool Resources and Multi-agent Systems*. In IEA/AIE '98 : Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pages 826–837, London, UK, 1998. Springer-Verlag. 26, 32
- [Bousquet *et al.* 2007] François Bousquet, Christophe Castella, Guy Trébuil, Cécile Barnaud, Stanislas Boissau et Suan Pheng Kam. *Using multi-agent systems in a companion modelling approach for agroecosystem management in South-east Asia*. Outlook on Agriculture, vol. 36, pages 57–62, 2007. 128
- [Bretagnolle *et al.* 2007] Anne Bretagnolle, Benoit Glisse, Thomas Louail, Denise Pumain et Céline Vacchiani-Marcuzzo. *Deux types de systèmes de villes identifiés par la modélisation multi-agents (Europe, Etats-Unis)*. 2007. 33

- [Caldéroni 2002] Stéphane Caldéroni. *Ethologie Artificielle et Contrôle Auto-Adaptatif dans les Systèmes d'Agents Réactifs*. Thèse de Doctorat, Université de La Réunion, 2002. 5
- [Castelfranchi 1998] Cristiano Castelfranchi. *Modelling social action for AI agents*. *Artificial Intelligence*, vol. 103, no. 1-2, pages 157–182, 1998. 40
- [Confucius] Confucius. Lun yu, entretiens, -479. 41
- [Couclelis 1985] Helen Couclelis. *Cellular Worlds, a framework for modelling micro-macro dynamics*. *Environment and Planning B*, vol. 17, pages 585–596, 1985. 163
- [Courdier et al. 2002] Rémy Courdier, François Guerrin, Fenintsoa Andriamasinoro et Jean-Marie Paillat. *Agent-based simulation of complex systems : Application to collective management of animal wastes*. *Journal of Artificial Societies and Social Simulation (JASSS)*, vol. 5, 2002. 5, 30, 128
- [Courdier 2003] Rémy Courdier. *Simulation Multi-Agents : Architecture, Ingénierie et Expérimentations*. Habilitation à Diriger des Recherches, 2003. 20, 96, 210
- [Custer 2005] Adrian Custer. *Geotools User's Manual*. The Geotools Management Committee, 2005. 136
- [Daniels 1999] Marcus Daniels. *Integrating simulation technologies with SWARM*. Workshop on Agent Simulation : Application, Models, and Tools, University of Chicago, 1999. 26, 42
- [David & Courdier 2008] Daniel David et Rémy Courdier. *Emergence as Metaknowledge : Refining Simulation Models through Emergence Reification*. In Cyrille Bertelle et Aladdin Ayesh, éditeurs, European Simulation and Modelling Conference (ESM'08), pages 25–27, Le Havre, France, 2008. 69, 81, 88
- [David & Courdier 2009] Daniel David et Rémy Courdier. *See Emergence as a Metaknowledge, a Way to Reify Emergent Phenomena in Multiagent Simulations ?* In International Conference on Agents and Artificial Intelligence (ICAART'09), pages 564–569, Porto, Portugal, 2009. 69, 81, 88
- [David et al. 2007] Daniel David, Denis Payet, Aurélie Botta, Gilles Lajoie, Sébastien Manglou et Rémy Courdier. *Un couplage de dynamiques comportementales : le modèle DS pour l'aménagement du territoire*. In Philippe Mathieu et Valerie Camps, éditeurs, Journées Francophones sur les Systèmes Multi-Agents (JFSMA'07), pages 129–138, Carcassonne, France, 2007. 5, 118, 136, 211
- [David et al. 2009] Daniel David, Denis Payet, Rémy Courdier et Yassine Gangat. *XELOC : un support générique pour la configuration et l'initialisation de systèmes multi-agents*. In Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09), pages 251–256, Lyon, France, 2009. 144
- [Deguet et al. 2007] Joris Deguet, Yves Demazeau et Laurent Magnin. *Emergence and Software development Based on a Survey of Emergence Definition*. vol. 56, pages 13–21, 2007. 40
- [Deguet 2008] Joris Deguet. *Emergence and Software development Based on a Survey of Emergence Definition*. Thèse de Doctorat, Université Joseph Fourier, 2008. 53

- [Dessalles & Phan 2006] Jean-Louis Dessalles et Denis Phan. *Emergence in multi-agent systems : cognitive hierarchy, detection, and complexity reduction part I : methodological issues*. *Artificial Economics*, pages 147–159, 2006. 64, 65, 210
- [Dessalles et al. 2007] Jean-Louis Dessalles, Jean-Pierre Müller et Denis Phan. *Emergence in multi-agent systems : conceptual and methodological issues*. *Agent-based modelling and simulation in the social and human sciences*, pages 327–355, 2007. 40, 52, 210
- [Dessalles et al. 2008] Jean-Louis Dessalles, Jacques Ferber et Denis Phan. *Emergence in Agent based Computational Social Science : conceptual, formal and diagrammatic analysis*. *Intelligent Complex Adaptive Systems*, 2008. 40
- [Destatte & Van Doren 2003] Philippe Destatte et Pascale Van Doren. *La prospective territoriale comme outil de gouvernance*. Institut Destrée, 2003. 28
- [Di Marzo Serugendo et al. 2006] Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes et Anthony Karageorgos. *Self-Organisation and Emergence in Multi-Agent Systems : An Overview*. *Informatica*, vol. 30, no. 1, pages 45–54, 2006. 39
- [Dimou 2008] Michel Dimou. *Urbanisation, agglomeration effects and regional disparities : an introduction*. *Région et Développement*, vol. 27, pages 7–12, 2008. 29
- [Drogoul et al. 2003] Alexis Drogoul, Diane Vanbergue et Thomas Meurisse. *Simulation Orientée Agent : où sont les agents ?* Actes des Journées de Rochebrune, Rencontres interdisciplinaires sur les systèmes complexes et artificiels, 2003. 22, 23, 70, 209
- [Drogoul 1993] Alexis Drogoul. *De la simulation multi-agents à la résolution collective de problèmes*. Thèse de doctorat, Université de Paris VI, 1993. 47, 79
- [Drogoul 2005] Alexis Drogoul. *Cours de Systèmes Multi-Agents*. Master 2 Intelligence Artificielle et Décision, Université de Paris VI, 2005. 25
- [Dépigny 2008] Sylvain Dépigny. *Le modèle PAYSAGRI : expérimentation de la sensibilité au paysage des agriculteurs comme facteur des évolutions du paysage rural*. Thèse de Doctorat, AgroParisTech, 2008. 34, 35, 209
- [Epstein & Axtell 1996] Joshua M. Epstein et Robert L. Axtell. *Growing artificial societies ; Social science from the bottom up*. Washington : Brookings Institution / MIT Press, 1996. 42
- [Epstein 2008] Joshua M. Epstein. *Why model ?* *Journal of Artificial Societies and Social Simulation*, vol. 11, no. 4, page 12, 2008. 14, 16, 18
- [Ferber 1995] Jacques Ferber. *Les systèmes multi-agents - Vers une intelligence collective*. Masson, 1995. 19, 24
- [Fishwick 1997] Paul A. Fishwick. *Computer simulation : growth through extension*. *Transactions of the Society for Computer Simulation*, vol. 14, no. 1, pages 13–24, 1997. 19, 20, 209
- [Fourny & Denizot 2007] Marie-Christine Fourny et Damien Denizot. *La prospective territoriale, révélateur et outil d'une action publique territorialisée*. *Territoires en action et dans l'action*, Presses universitaires de Rennes, pages 29–44, 2007. 27

- [Franc & Sanders 1997] Alain Franc et Lena Sanders. *Modèles et systèmes multi-agents en écologie et en géographie : état de l'art et comparaison avec les approches classiques*. Colloque SMAGET : Modèles et Systèmes Multi-Agents pour la Gestion de l'Environnement et des Territoires, pages 17–34, 1997. 32
- [Fromm 2005] Jochen Fromm. *Ten Questions about Emergence*. arXiv :nlin/0509049v1 [nlin.AO], 2005. 39
- [Gangat *et al.* 2009a] Yassine Gangat, Daniel David, Denis Payet et Rémy Courdier. *Modélisation et simulation de l'aménagement énergétique d'un territoire*. Colloque sur la Convergence des Réseaux, de l'Informatique et du Multimédia pour les E-Services (CRIMES'09), 2009. 196
- [Gangat *et al.* 2009b] Yassine Gangat, Denis Payet et Rémy Courdier. *Démonstration : Aménagement énergétique d'un territoire - une approche par simulation multi-agents*. In Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09), pages 237–240, Lyon, France, 2009. 196
- [Gangat *et al.* 2010] Yassine Gangat, Mayeul Dalleau, Daniel David, Nicolas Sébastien et Denis Payet. *Turtle are the Turtles*. European Simulation and Modelling Conference (ESM'2010), 2010. 5, 31
- [Georgé 2004] Jean-Pierre Georgé. *Résolution de problèmes par émergence*. Thèse de Doctorat, Université Toulouse III, 2004. 45, 48
- [Gilbert & Troitzsch 1999] Nigel Gilbert et Klaus G. Troitzsch. *Simulation for the social scientist*. Open Univ Pr, 1999. 22, 209
- [Grumbach 1997] Antoine Grumbach. *A propos d'émergence. Emergence...ou...explication*. Intellectica Emergence and explanation, vol. 25, 1997. 48
- [Gutknecht & Ferber 2000] Olivier Gutknecht et Jacques Ferber. *Madkit : a generic multi-agent platform*. In Proceedings of the fourth international conference on Autonomous agents, page 79. ACM, 2000. 26
- [Guérin-Pace *et al.* 1996] France Guérin-Pace, Hélène Mathian, Denise Pumain, Lena Sanders et Stéphane Bura. *Les systèmes multi-agents pour modéliser l'émergence des réseaux urbains. Analyse Spatiale de données démographiques*, pages 281–309, 1996. 32
- [Hagget 1965] Peter Hagget. *Locational analysis in modern geography*. 1965. 14
- [Hill & Vickers 2006] Gillian Hill et Steven Vickers. *A language for configuring multi-level specifications*. Theoretical Computer Science Journal, vol. 351(2), pages 146–166, 2006. 144
- [Hill *et al.* 2000] David R.C. Hill, S. Mechoud, A. Campos, P. Coquillard, J. Gueugnot, D. Orth, Y. Michelin, C. Poix, G. L'Homme, P. Carrere *et al.* *Modélisation de l'entretien du paysage par des herbivores en moyenne montagne : une approche multi-agents= Modelling of landscape management by means of herbivores in medium level mountains : a multi-agent approach*. Ingénieries, no. 21, pages 63–75, 2000. 32

- [Hofstadter 1985] Douglas Hofstadter. *Gödel, Escher, Bach : les brins d'une guirlande éternelle*. Dunod, 1985. 79
- [Holgate & Partain 2002] Matt Holgate et Will Partain. *ARK : an object(ive) view of extensible minimal languages for system configuration*. *Information and Software Technology*, vol. 44(4), pages 259–268, 2002. 144
- [Johnson & Lancaster 2002] Paul Johnson et Alex Lancaster. *Swarm User Guide*. Swarm Development Group, 2002. 26, 33
- [Kalfoglou et al. 2000] Yannis Kalfoglou, Tim Menzies, Klauss-Dieter Althoff et Enrico Motta. *Meta-knowledge in systems design : panacea...or undelivered promise?* *The Knowledge Engineering Review*, vol. 15, no. 4, pages 381–404, 2000. 66, 210
- [Keet 2006] C. Maria Keet. *Part-whole relations in Object-Role Models*. *Object-Role Modelling (ORM 2006)*, pages 1118–1127, 2006. 49
- [Kiczales 1991] Gregor Kiczales. *The Art of the Metaobject Protocol*. The MIT Press, 1991. 98
- [Lagabriele et al. 2010] Erwann Lagabriele, Aurélie Botta, Williams Daré, Daniel David, Sigrig Aubert et Christo Fabricius. *Modelling with stakeholders to integrate biodiversity into land-use planning - Lessons learned in Réunion Island (Western Indian Ocean)*. *Environmental Modelling and Software*, vol. 25, no. 11, pages 1413–1427, 2010. 118, 146
- [Lajoie & Hagen-Zanker 2007] Gilles Lajoie et Alex Hagen-Zanker. *La simulation de l'étalement urbain à La Réunion : apport de l'automate cellulaire Metronamica pour la prospective territoriale*. *Cybergeog : Revue européenne de géographie*, 2007. 125
- [Lajoie 2005] Gilles Lajoie. *Modélisation et prospective territoriale*. *Modélisations en Géographie*, pages 107–134, 2005. 29
- [Lajoie 2007] Gilles Lajoie. *Recherches en modélisation urbaine*. *Rapport d'Habilitation à Diriger des Recherches*, 2007. 29
- [Langton 1990] Christopher G. Langton. *Computation at the edge of chaos, Phase transition and emergent computation*. *Physica D*, vol. 4, 1990. 47
- [Le Page et al. 2000] 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. 32
- [Lewes 1875] George Henry Lewes. *Problems of life and mind*. London : Kegan Paul, trench, Turbner, and Co, 1875. 45, 48
- [Li & Liu 2008] Xia Li et Xiaoping Liu. *Embedding sustainable development strategies in agent-based models for use as a planning tool*. *International Journal of Geographical Information Science*, vol. 22, no. 1, pages 21–45, 2008. 32
- [Loibl & Toetzer 2003] Wolfgang Loibl et Tanja Toetzer. *Modeling growth and densification processes in suburban regions—simulation of landscape transition with spatial agents*. *Environ-*

- mental Modelling Software, vol. 18, no. 6, pages 553 – 563, 2003. Applying Computer Research to Environmental Problems. [33](#), [34](#), [209](#)
- [Loibl & Walz 2010] Wolfgang Loibl et Ariane Walz. *Generic regional development strategies from local stakeholders' scenarios – the Montafon experience*. Ecology and Society, vol. 15, no. 3, 2010. [33](#)
- [Loibl et al. 2007] Wolfgang Loibl, Tanja Tötzer, Mario Köstl et Klaus Steinnocher. *Simulation Of Polycentric Urban Growth Dynamics Through Agents*. Modelling Land-Use Change, vol. 90, pages 219–235, 2007. [33](#)
- [Loinger 2005] Guy Loinger. *La prospective territoriale*. Guide permanent du développeur économique, 2005. [27](#), [28](#)
- [Loui 2008] Ronald P. Loui. *In Praise of Scripting : Real Pro-gramming Pragmatism*. IEEE Computer, vol. 41, no. 7, pages 22–26, 2008. [151](#)
- [Luzeaux 1997] Dominique Luzeaux. *Learning knowledge-based systems and control of complex systems*. 15th IMACS World Congress, 1997. [65](#)
- [Marcenac & Giroux 1998] Pierre Marcenac et Sylvain Giroux. *A generic architecture for agent-oriented simulations of complex processes*. Lecture Notes in Computer Science, vol. 8(3), pages 247–267, 1998. [5](#)
- [Marcenac et al. 1998] Pierre Marcenac, Rémy Courdier, Stéphane Caldéroni et Jean-Christophe Soulié. *Towards an Emergence Machine for Complex Systems Simulation*. Applied Intelligence, vol. 1416, pages 785–794, 1998. [54](#), [55](#), [210](#)
- [May 1976] Robert M. May. *Simple mathematical models with very complicated dynamics*. Nature, vol. 261, pages 459–67, 1976. [18](#)
- [Mena et al. 2010] Carlos F. Mena, Stephen J. Walsh, Brian G. Frizzelle, Yao Xiaozheng et George P. Malanson. *Land use change on household farms in the Ecuadorian Amazon : Design and implementation of an agent-based model*. Applied Geography, vol. In Press, Corrected Proof, 2010. [35](#)
- [Minsky 1965] Marvin L. Minsky. *Master, Mind and Models*. International Federation of Information Processing Congress, pages 45–49, 1965. [14](#)
- [Moncion 2008] Thomas Moncion. *Modélisation de la complexité et de la dynamique des simulations multi-agents. Application pour l'analyse de phénomènes émergents*. Thèse de Doctorat, Université d'Evry val d'Essonne, 2008. [40](#)
- [Morvan et al. 2008] Gildas Morvan, Daniel Jolly, Alexandre Veremme, Daniel Dupont et Damien Charabidze. *Vers une méthode de modélisation multi-niveaux*. 7ème conférence francophone de MOdélisation et SIMulation (MOSIM'08), 2008. [56](#), [210](#)
- [Morvan 2009] Gildas Morvan. *Approche multi-agents d'un système d'aide à la décision en environnement dynamique et incertain - Application à l'entomologie médico-légale*. Thèse de Doctorat, Université d'Artois, 2009. [56](#)

- [M.R.Jean 1997] M.R.Jean. *Emergence et SMA*. In Quinqueton, Thomas et Trousse, editeurs, 5emes JFIADSMA, pages 323–342. Hermes, 1997. 40
- [Müller 2004] Jean-Pierre Müller. *The MIMOSA Generic Modelling and Simulation Platform : the Case of Multi-Agent Systems*. In Proceedings of the 5th Workshop on Agent-based Simulation, Lisbon, 2004. 26
- [Niemeyer & Poteet 2003] Glenn Niemeyer et Jeremy Poteet. *Extree programming with Ants*. Sams, 2003. 150
- [Odell et al. 1997] James Odell, H. Van Dyke Parunak, Mitchell Fleischer et Sven Brueckner. *Modeling Agents and their Environment, Agent-Oriented Software Engineering III*. Lecture Notes in Computer Science, vol. 2585, 1997. 127
- [Ousterhout 1998] John K. Ousterhout. *Scripting : Higher Level Programming for the 21st Century*. IEEE Computer, vol. 31(3), pages 23–30, 1998. 151
- [Paquette 1998] Gilbert Paquette. *Metaknowledge representation : application to learning systems engineering*. TLNCE Technical Reports, 1998. 64
- [Payet et al. 2006a] Denis Payet, Rémy Courdier, Tiana Ralambondrainy et Nicolas Sébastien. *Le modèle à Temporalité : pour un équilibre entre adéquation et optimisation du temps dans les simulations agents*. Journées Francophones sur les Systèmes Multi-Agents (JFSMA), 2006. 4, 127, 153
- [Payet et al. 2006b] Denis Payet, Rémy Courdier, Nicolas Sébastien et Tiana Ralambondrainy. *Environment as support for simplification, reuse and integration of proecesses in spatial MAS*. IRI'06, Hawaï, USA, pages 127–131, 2006. 5, 110, 113, 127, 160, 161, 212
- [Payet et al. 2009] Denis Payet, Daniel David et Nicolas Sébastien. *Auto-génération d'Environnement : l'exemple d'Infinite Forest*. In Journées Francophones sur les Systèmes Multi-Agents (JFSMA'09), pages 165–174, Lyon, France, 2009. 5
- [Peterson 1996] Ivars Peterson. *The gods of sugarscape ; digital sex, migration, trade, and war on the social science frontier - computer model of human social and economic behavior*. Science News, 1996. 42
- [Pitrat 1990] Jacques Pitrat. *Méta-connaissances, futur de l'intelligence artificielle*. Hermes, 1990. 65
- [Ralambondrainy 2009] Tiana Ralambondrainy. *Observation de simulations multi-agents à grande échelle*. Thèse de Doctorat, Université de la Réunion, 2009. 5, 15, 20, 21, 30, 70, 101, 110, 112, 209, 211
- [Ratzé et al. 2007] Cédric Ratzé, François Gillet, Jean-Pierre Müller et Kilian Stoffel. *Simulation modelling of ecological hierarchies in constructive dynamical systems*. Ecological complexity, vol. 4, no. 1-2, pages 13–25, 2007. 32, 54, 55, 210
- [Reynolds 1986] Craig W. Reynolds. *Boids (Flocks, Herds, and Schools : a Distributed Behavioral Model)*. Reynolds, 1986. 44

- [Sanders *et al.* 1996a] Léna Sanders, Jean-Marc Favaro, H el ene Mathian, Denise Pumain et Beno ıt Glisse. *Intelligence artificielle et agents collectifs : le mod ele EUROSIM*. *Cybergeo : European Journal of Geography*, vol. 24, pages 187–305, 1996. 33, 209
- [Sanders *et al.* 1996b] L ena Sanders, Denise Pumain, H el ene Mathian, France Gu er in-Pace et St ephane Bura. *SIMPOP : a multi-agents system for urbanism*. *Environment and Planning B*, vol. 24, pages 187–305, 1996. 32, 163
- [Servat *et al.* 1998] David Servat, Edith Perrier, Jean-Pierre Treuil et Alexis Drogoul. *When agents emerge from agents : Introducing multi-scale viewpoints in multi-agent simulations, Multi-Agent Systems and Agent-Based Simulation*. *Lecture Notes in Artificial Intelligence*, vol. 1534, pages 183–198, 1998. 57, 210
- [SETEC & SAFEGE 2007] SETEC et SAFEGE. *R evision du SAR de La R eunion, Sc enarii d'am enagement et de d evveloppement de La R eunion en 2030*. Rapport commandit e par la R egion R eunion, 2007. 141, 211
- [Souli e *et al.* 1998] Jean-Christophe Souli e, Pierre Marcenac, St ephane Cald eroni et R emy Courdier. *GEAMAS V2.0 : an Object Oriented Platform for Complex Systems Simulations*. *Proceedings of the 26th International Conference on Technology of Oriented-Object Languages and Systems (TOOLS USA'98)*, pages 230–242, 1998. 4, 26
- [Souli e 2001] Jean-Christophe Souli e. *Vers une approche multi-environnement pour les agents*. Th ese de Doctorat, Universit e de La R eunion, 2001. 31
- [Stephan 2002] Achim Stephan. *Emergence*. *Encyclopedia of Cognitive Science*, vol. 1, pages 1108–1115, 2002. 40, 52, 66, 106, 210
- [S ebastien *et al.* 2009] Nicolas S ebastien, R emy Courdier, Didier Hoareau, Marc-Philippe Huet et Denis Payet. *Ordonnancement parall ele de simulations orient ees agent : Une approche bas ee sur l'analyse des d ependances temporelles des influences et perceptions*. *Revue d'Intelligence Artificielle*, vol. 23, no. 5-6, pages 675–697, 2009. 5
- [S ebastien 2009] Nicolas S ebastien. *Distribution et parall elisation de simulations orient ee agent*. Th ese de Doctorat, Universit e de la R eunion, 2009. 5, 15, 20, 21, 209
- [Tranouez *et al.* 2006] Pierrick Tranouez, Cyrille Bertelle et Damien Olivier. *Changing levels of description in a uid ow simulation Chapitre de Emergent Properties in Natural and Articial Dynamical Systems*. *Understanding Complex Systems series*, Springer ed, pages 1108–1115, 2006. 58
- [Tranouez 2005] Pierrick Tranouez. *Contribution   la mod elisation et   la prise en compte informatique de niveaux de descriptions multiples*. Th ese de Doctorat, Universit e du Havre, 2005. 58, 79, 210
- [Turban & Aronson 1997] Efraim Turban et Jay Aronson. *Decision support systems and intelligent systems*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1997. 67, 68, 210
- [Turing 1937] Alan M. Turing. *On computable numbers, with an application to the Entscheidungsproblem*. *Proceedings of the London Mathematical Society*, vol. 2, no. 1, page 230, 1937. 16

- [Vally & Courdier 1999] Jean-Dany Vally et Rémy Courdier. *A Conceptual Role-Centered Model for Design of Multi-Agent Systems*. Lecture notes in computer science, pages 33–46, 1999. 5
- [Van de Vijver 1997] Gertrudis Van de Vijver. *Emergence et explication*. Intellectica, 1997. 47
- [W3C 2004] World Wide Web Consortium W3C. *XML Schema (Second Edition), Parts 0, 1, 2*. W3C Recommendation, 2004. 150
- [W3C 2006] World Wide Web Consortium W3C. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. W3C Recommendation, 2006. 150
- [Walliser 2009] Bernard Walliser. *Emergent Phenomena*. ENPC-EHESS, Ouvrage Collectif, 2009. 51
- [Walsh *et al.* 2008] Stephen J. Walsh, Joseph P. Messina, Carlos F. Mena, George P. Malanson et Philip H. Page. *Complexity theory, spatial simulation models, and land use dynamics in the Northern Ecuadorian Amazon*. *Geoforum*, vol. 39, no. 2, pages 867 – 878, 2008. 35, 36, 209
- [White & Engelen 1993] Roger White et Guy Engelen. *Cellular automata as the basis of integrated dynamic regional modelling*. *Environment and planning A*, vol. 25, pages 1175–1199, 1993. 125
- [White *et al.* 1997] Roger White, Guy Engelen et Inge Uljee. *The use of constrained cellular automata for high-resolution modelling of urban land-use dynamics*. *Environment and planning B : Planning and design*, vol. (24)3, pages 323–343, 1997. 125
- [Wilenski 1999] Uri Wilenski. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University Evanston, IL., 1999. 26, 41
- [Winder 2006] Russel Winder. *Hello Groovy*. *ACCU Journal CVU*, vol. 18, no. 3, pages 3–7, 2006. 151
- [Wooldridge & Jennings 1995] Michael Wooldridge et Nicholas Robert Jennings. *Intelligent agents : Theory and practice*. *The knowledge engineering review*, vol. 10, no. 02, pages 115–152, 1995. 19, 24
- [Yamins 2005a] Daniel Yamins. *Towards a theory of "local to global" in distributed multi-agent systems (I)*. In *AAMAS '05 : Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 183–190, New York, NY, USA, 2005. ACM. 48
- [Yamins 2005b] Daniel Yamins. *Towards a theory of "local to global" in distributed multi-agent systems (II)*. In *AAMAS '05 : Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 191–198, New York, NY, USA, 2005. ACM. 48
- [Zeigler *et al.* 2000] Bernard P. Zeigler, Herbert Praehofer et Tag Gon Kim. *Theory of modeling and simulation*. Academic press New York, 2000. 18

Table des figures

1	Organisation des parties et chapitres de la thèse	8
1.1	Méthodologie de conception de simulations selon Fishwick (adapté d'après [Fishwick 1997])	20
1.2	Méthodologie de conception de simulations selon Ralambondrainy et Sébastien (adapté d'après [Ralambondrainy 2009, Sébastien 2009])	21
1.3	Méthodologie de conception de simulations selon Gilbert et Troitzsch [Gilbert & Troitzsch 1999] (adapté d'après [Drogoul <i>et al.</i> 2003])	22
1.4	Méthodologie de conception de simulations selon Drogoul, Vanbergue et Meurisse (adapté d'après [Drogoul <i>et al.</i> 2003])	23
1.5	Les mécanismes d'influence et perception dans une SOA	26
1.6	Vue sur l'évolution des bassins versants d'une zone agricole avec le modèle BIOMAS	30
1.7	Un prototype de simulation des populations de tortues vertes dans la zone Sud-Ouest de l'océan Indien	31
1.8	L'évolution d'un système de villes avec le Modèle Simpop2/EUROSIM (extrait de [Sanders <i>et al.</i> 1996a])	33
1.9	Un outil de simulation de l'évolution urbaine dans la région de Vienne (extrait de [Loibl & Toetzer 2003])	34
1.10	L'évolution de surfaces agricoles avec PAYSAGRI (extrait de [Dépigny 2008])	35
1.11	L'évolution de surfaces pastorales (en jaune) en Equateur. A gauche l'état initial, à droite l'état final (extrait de [Walsh <i>et al.</i> 2008])	36
2.1	La répartition des agents sur les montagnes de sucre de SugarScape	42
2.2	Les HeatBugs et la chaleur	43
2.3	Le plus court chemin de la colonie de fourmis	43
2.4	Le phénomène de flocking observé chez les oiseaux migrateurs	44
2.5	Les oscillateurs dans le jeu de la vie	45
2.6	Proto-émergentisme et Néo-émergentisme : de la boîte noire à la compréhension des processus d'émergence	46

2.7	L'émergence unilatérale : le micro explique directement le macro	50
2.8	L'émergence bilatérale : influences ascendantes et descendantes entre micro et macro	50
2.9	L'émergence synchronique et l'émergence diachronique	51
2.10	Une classification selon Stephan (adapté de [Stephan 2002] et [Dessalles <i>et al.</i> 2007])	52
2.11	Une classification selon Boschetti [Boschetti & Gray 2007]	53
2.12	Modélisation multi-niveaux en SOA selon l'équipe SMART (extrait de [Marcenac <i>et al.</i> 1998])	55
2.13	Les holons utilisés pour gérer l'articulation des organisations entre niveaux (extrait de [Ratzé <i>et al.</i> 2007])	55
2.14	Modélisation multi-niveaux selon Morvan (extrait de [Morvan <i>et al.</i> 2008])	56
2.15	Modélisation multi-niveaux selon Servat (extrait de [Servat <i>et al.</i> 1998])	57
2.16	Détection et réification de Vortex dans l'écoulement de fluides (extrait de [Tranouez 2005])	58
3.1	Les grandes questions de la modélisation multi-niveau (extrait de [Dessalles & Phan 2006])	65
3.2	Quelques types de méta-connaissance listés dans la littérature (extrait de [Kalfoglou <i>et al.</i> 2000])	66
3.3	Processus d'ingénierie des connaissances (adapté d'après [Turban & Aronson 1997])	68
3.4	Metaknowledge thinking (adapté d'après [Benchimol 2005])	68
3.5	La trame commune des méthodologies de conception de SOA	71
3.6	Connaissance et méta-connaissance dans une méthodologie de conception de SOA	74
4.1	Comment est déclenchée la détection d'un phénomène émergent	81
4.2	A quel niveau est déclenchée la détection d'un phénomène émergent	82
4.3	Les différentes stratégies de détection d'un phénomène émergent	84
4.4	A quel niveau est déclenchée la matérialisation d'un phénomène émergent	87
4.5	Comment est déclenchée la matérialisation d'un phénomène émergent	88
4.6	Les différentes stratégies de réification d'un phénomène émergent	90
4.7	Classification des types de réifications possibles d'un phénomène émergent dans une SOA	92
5.1	Les dimensions conceptuelles du développement d'une architecture de plateforme de simulation agent (extrait de [Courdier 2003])	96
5.2	De la SOA à la méta-connaissance	97
5.3	De l'usage des différents environnements et règles métriques pour caractériser un phénomène émergent	100

5.4	Un exemple de carte des conversations (extrait de [Ralambondrainy 2009])	101
5.5	Les structures d'émergence (agents d'émergence et éléments d'interposition) et les mécanismes d'influences/perceptions modifiées	106
5.6	Le processus formalisé de prise en compte de l'émergence	107
5.7	Architecture générale pour la réification de phénomènes émergents	111
5.8	Architecture générale de GEAMAS-NG (extrait de [Ralambondrainy 2009])	112
6.1	L'île de La Réunion dans le monde	119
6.2	L'île de La Réunion : un relief tourmenté, un volcan, des cyclones... on ne peut pas s'installer partout ! (photographies Observatoire Volcanologique et Météo France)	119
6.3	Un paysage typique réunionnais : zones naturelles (en vert), zones agricoles (en orange), et zones urbaines (en rouge) coexistent sur un même territoire	121
6.4	Le Schéma d'Aménagement Régional	122
6.5	Articulation entre le SAR, les SCoT, et les PLU	123
6.6	Le modèle Metronamica Réunion	125
6.7	La principale interface utilisateur de DS	129
6.8	La répartition de la population et la densification de l'habitat dans les micro-régions	130
6.9	La configuration du volet naturel	131
6.10	Les interactions entre les volets naturel, agricole, et urbain	132
6.11	L'interface de lancement des simulations où se spécifient les plages temporelles	133
6.12	L'évolution de la population : de 750000 habitants en 2003 à 1 million en 2030	134
6.13	Les projections de l'INSEE pour l'évolution de la population entre 1999 et 2030 (extrait de [Actif & Lardoux 2006])	135
6.14	Un exemple de sortie cartographique (extrait de [David <i>et al.</i> 2007])	136
6.15	Un exemple d'évolution du Mode d'Occupation des Sols dans un scénario laxiste au niveau contraintes d'urbanisation (en haut l'état initial en 2003, en bas l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.	138
6.16	Un exemple d'évolution du Mode d'Occupation des Sols dans un scénario favorable au développement économique (en haut l'état initial en 2003, en bas l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.	139
6.17	Un exemple d'évolution de la densité de population croissante selon le dégradé de couleur rouge (en haut l'état initial en 2003, en bas l'état final en 2030)	140
6.18	Schéma chorématique d'un scénario d'évolution tendancielle (extrait de [SETEC & SAFEGE 2007])	141
7.1	Les 24 communes de L'île de La Réunion	146
7.2	La répartition de la population à La Réunion	147

7.3	Le Mode d'Occupation des Sols de L'île de La Réunion en 2003 (extrait de l'Atlas DOMINO)	147
7.4	Les potentiels naturels à l'île de La Réunion (extrait de l'Atlas DOMINO)	148
7.5	Les potentiels agricoles à l'île de La Réunion (extrait de l'Atlas DOMINO)	149
7.6	Les potentiels urbains à l'île de La Réunion (extrait de l'Atlas DOMINO)	150
7.7	Une carte sur laquelle est appliquée une grille qui sera parcourue pour lire le code couleur de chaque case lors de l'exécution de la balise <forMap>	156
8.1	La modélisation orientée dynamiques (extrait de [Payet <i>et al.</i> 2006b])	161
8.2	Lien entre le découpage de l'île et la création des agents	162
8.3	Structure conceptuelle du modèle DS	163
8.4	Les actions des agents superviseurs sur l'environnement	163
8.5	Le couplage des dynamiques via l'environnement	169
8.6	La dynamique générale du modèle	170
8.7	Une simulation sur la zone de Cambaie (à gauche l'état initial en 2003, à droite l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.	172
8.8	Une simulation sur la zone de Saint-Pierre (à gauche l'état initial en 2003, à droite l'état final en 2030). Les zones urbaines sont en rouge, les zones agricoles en jaune, et les zones naturelles en vert.	173
8.9	Toutes les cellules urbanisées au cours d'une simulation (en gris)	176
8.10	Le phénomène émergent "nouvelle zone urbaine" qui a été détecté (en rose)	178
8.11	Vue aérienne actuelle de la zone correspondant à notre phénomène émergent (photographie provenant de Google Maps)	179
8.12	Le phénomène et les entités (<i>agents</i> et <i>cellules</i> d'environnement de DS) concernées géographiquement par son émergence	181
8.13	Le couplage des dynamiques dans la zone géographique de la nouvelle zone urbaine	182
8.14	La mise en place du phénomène émergent des nouvelles zones urbaines dans DS sur le plan architectural	183
8.15	Les services plateforme en charge de la détection et de la matérialisation	183
8.16	La densité de population observée dans les résultats de deux simulations (à gauche la nouvelle zone urbaine est matérialisée en considérant qu'elle est destinée à des logements, à droite elle est matérialisée en considérant qu'elle est destinée à des locaux d'entreprises)	186

Table des définitions

I.1	Modèle	14
I.2	Modèle du domaine	15
I.3	Modèle informatique	15
I.4	Modèle simulé	15
I.5	Simulation informatique	15
I.6	Simulation	15
I.7	Système Complexe	15
I.8	Méthodologie de conception de simulation	19
I.9	Système Multi-Agents	24
I.10	Prospective territoriale	28
II.1	Méta-connaissance	65
II.2	Emergence	69
II.3	Phase de détection	79
II.4	Phase de matérialisation	79
II.5	Réification	79
II.6	Phase d'introspection	99
II.7	Phase d'intercession	99
II.8	Révéléateur d'émergence	103
II.9	Agent d'émergence	105
II.10	Elément d'interposition	105
II.11	Loi d'émergence	105

PROSPECTIVE TERRITORIALE PAR SIMULATION ORIENTÉE AGENT

Résumé :

L'île de La Réunion, inscrite au patrimoine mondial de l'UNESCO pour la beauté de ses paysages et pour sa biodiversité exceptionnelle, va accueillir dans les prochaines années une population de plus en plus importante. Les documents de planification réglementant l'évolution de l'urbanisation dans l'île doivent donc en tenir compte afin de gérer au mieux l'évolution et la structuration du foncier local. Dans un contexte de recherche pluridisciplinaire, l'enjeu de nos travaux est donc de proposer des méthodes et outils facilitant les interactions entre les informaticiens et les thématiciens qui apportent leur expertise à la construction de modèles de simulation dédiés à la prospective territoriale ; et de faire en sorte que ces modèles puissent fournir des pistes de réflexions aux décideurs confrontés aux choix de demain afin d'aider à mettre en place une organisation des territoires qui soit la plus cohérente possible.

Après avoir participé à la réalisation de DS, un modèle qui permet de simuler conjointement l'évolution de la population et celle du Mode d'Occupation des Sols à La Réunion, nous nous sommes focalisés sur l'étude de l'émergence, une notion qui apparaît dans nos simulations en environnement spatialisé, mais pour laquelle il n'existe pas d'outils génériques permettant de la manipuler. Nous l'avons définie comme étant une méta-connaissance, ce qui nous a permis de proposer une méthodologie de conception de simulations et un cadre formel qui ont aboutit à la mise en place de structures émergentes dans la plateforme de simulation GEAMAS-NG. Celle-ci, ou les entités de la SOA elles-mêmes, peuvent ainsi prendre conscience de l'émergence de phénomènes et les matérialiser en ayant recours à la connaissance des thématiciens. Nous avons alors montré l'intérêt de ces propositions en expérimentant la réification de phénomènes émergents observés dans DS.

Mots clés :

Modélisation de Systèmes Complexes, Simulation Orientée Agent, Prospective Territoriale, Emergence, Méta-connaissance.

TERRITORIAL PROSPECTIVE BY AGENT-BASED SIMULATION

Abstract :

Reunion Island will be hosting in the coming years an increasingly important population. Documents dedicated to land-use planning should take this into account to better manage the evolutions of the island. In the context of multidisciplinary research, the challenge of our work is to propose methods and tools to facilitate interactions between computer scientists and thematians who bring their expertise to the construction of simulation models dedicated to territorial prospective.

We co-constructed DS, an agent-based model that simulates the evolutions of the population and of the land-use on the island, and then we focused on the study of emergence, a concept that appears in our simulations in spatial environment. We defined it as a meta-knowledge ; and that enabled us to propose a modelling methodology and a formal framework that lead to the introduction of emerging structures in the Agent-Based Simulation platform GEAMAS-NG. The systems, or the entities of the models themselves, can now be aware of the emergence of phenomena and can materialize these phenomena by using the knowledge of thematians. We finally demonstrated the usefulness of these proposals by experimenting the reification of an emergent phenomenon observed in DS.

Keywords :

Complex Systems Modelling, Agent-Based Simulation, Territorial Prospective, Emergence, Metaknowledge.
