



HAL
open science

Conception d'une approche spatialisée à base d'agent pour coupler les modèles mathématiques et informatiques : application à la modélisation du processus écosystémique du sol

Jean Bienvenue Dinaharison

► **To cite this version:**

Jean Bienvenue Dinaharison. Conception d'une approche spatialisée à base d'agent pour coupler les modèles mathématiques et informatiques : application à la modélisation du processus écosystémique du sol. Système multi-agents [cs.MA]. Sorbonne Université; Université de Fianarantsoa (Madagascar), 2023. Français. NNT : 2023SORUS184 . tel-04199518

HAL Id: tel-04199518

<https://theses.hal.science/tel-04199518>

Submitted on 7 Sep 2023

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.



SORBONNE UNIVERSITÉ
&
UNIVERSITÉ DE FIANARANTSOA



Conception d'une approche spatialisée à base d'agent pour
coupler les modèles mathématiques et informatiques :
Application à la modélisation du processus écosystémique
du sol

Thèse pour l'obtention du diplôme de doctorat en Informatique
Jean Bienvenue Dinaharison
2023

Rapporteurs:

Mamadou Kaba Traoré (Professeur, Université de Bordeaux) (Président du jury)
René Mandiau (Professeur, Université Polytechnique Hauts de France)

Directeurs de thèse :

Nicolas Marilleau (Chargé de recherche, UMMISCO, HDR)
Jean-Pierre Müller (Cadre Scientifique, CIRAD, HDR)

Examineurs :

Christophe Cambier (Maître de Conférences, UMMISCO, HDR)
Hasina Lalaina Rakotonirainy (Maître de Conférences, Université de Fianarantsoa)
Nathalie Corson (Maîtresse de Conférences, Université du Havre)
Laetitia Bernard (Chargé de recherche, Eco&Sol)

Dédicaces

À ma mère

À ma défunte grand-mère

À mon défunt beau-père, celui qui m'a élevé

À mes directeurs de thèses et encadrants

À ma famille et mes amis

Remerciements

Je tiens tout d'abord à remercier du fond du cœur toutes les personnes qui m'ont aidé durant cette thèse, pour votre soutien constant. Vos encouragements ont été inestimables pour moi, et je suis profondément reconnaissant pour tout ce que vous avez fait.

Je tiens particulièrement à remercier:

Mes directeurs de thèses, Docteur Nicolas Marilleau et Professeur Jean-Pierre Müller et mes encadrants Docteur Hasina Rakotonirainy et Docteur Nathalie Corson, pour leurs conseils et soutiens, les encouragements lors des rejets d'articles. Vous m'avez toujours soutenu malgré les difficultés rencontrées.

Ma mère et ma famille qui n'ont cessé de me soutenir pendant tout ce temps.

Les rapporteurs, Professeur René Mandiau et Professeur Mamadou Kaba Traoré d'avoir accepté d'évaluer ce travail.

Les présidents des deux universités auxquelles je suis inscrit, le Professeur Nathalie Drach-Temam de Sorbonne Université et le Professeur Hajalalaina Richard de l'Université de Fianarantsoa, de m'avoir permis de faire ce parcours.

Les directeurs des écoles doctorales, Professeur Habib Mehrez de l'EDITE de Paris et Professeur Raniriharinosy Karyl de l'EDMI Fianarantsoa, de m'avoir accepté dans leurs écoles respectives.

Les directeurs des laboratoires d'accueil, Docteur Jean-Daniel Zucker à l'UMMISCO Bondy ainsi que Professeur Hajalalaina et Docteur Hasina Rakotonirainy à l'IGMA, de m'avoir accueilli dans leurs équipes.

Les initiateurs du programme PDI-MSc, sans qui je n'aurais pas eu l'occasion de faire cette thèse, tout spécialement le Docteur Christophe Cambier et les formateurs du programme.

L'instigatrice du projet Cammisole, Docteur Laetitia Bernard, sans qui cette thèse n'aurait pas été possible. Je remercie également toutes les personnes avec lesquelles j'ai eu la chance de connaître pendant la durée du projet, notamment les ingénieurs de l'AgriSud et les chercheurs au Laboratoire Radio-Isotope. Les personnels administratifs de l'IRD à Bondy et à la Régie de Madagascar pour les différentes démarches administratives relatives aux missions pendant la thèse.

Tous ceux, de près ou de loin, ont contribué à l'achèvement de cette thèse.

Abstract

Soil is a highly heterogeneous environment in which many processes interact to provide ecosystem services. Model coupling approaches propose to define such a system by using a modular architecture in which various processes, represented by models, communicate to reproduce different aspects of a phenomenon such as soil functioning. In this thesis project, we develop such an approach for the purpose of modelling soil functioning.

The challenges of such scheme lie in solving representation problems of soil processes. These representation problems originate from the fact that models from various disciplines are reused to describe the processes. By representations problems, we mean model description (which can be individual or equation based), the temporal execution settings and data resolution.

These coupling constraints are addressed by a number of approaches in the literature. All of them propose satisfactory solutions to these constraints in their respective application fields, but are hard to reuse to properly describe soil functioning. In our approach, we use the agent paradigm to encapsulate the various soil processes. Then processes will communicate through the space by using resources inside it. The behaviour of the processes then depends on the availability of resources. A coordination problem can arise from this type of coupling, as processes may consume the resource simultaneously while the resource may not support this demand. To overcome that matter, we use an action-theoretic technique called Influence-Reaction to define strategies to manage this type of situation. We used algorithms suggested by the abundant literature to manage any processes temporality issues.

This coupling approach is applied to a model of organic matter decomposition in which several processes (earthworms, microbes, and roots) compete for soil resources. Results suggest that the use of our approach is suitable for modelling soil functioning, but also gives more accurate indications of resource availability.

Résumé

Le sol est un environnement fortement hétérogène dans lequel de nombreux processus interagissent pour fournir des services écosystémiques. L'approche de couplage de modèle propose de définir ce type de système en utilisant une architecture modulaire dans laquelle les divers processus représentés par des modèles communiquent pour reproduire les différents aspects du mécanisme d'un phénomène tel que le fonctionnement du sol. Dans cette thèse, nous développons une telle approche dans le but de modéliser le fonctionnement du sol.

Les défis d'un tel projet résident dans la résolution des problèmes de représentation des processus du sol. Ces problèmes de représentation prennent leur origine du fait que l'on réutilise des modèles venant de disciplines variées pour décrire les processus. Cette hétérogénéité de représentation peut se résumer par l'écriture des modèles, le cadre d'exécution temporelle et la résolution des données décrites par les modèles.

Ces contraintes de couplage sont traitées par bon nombre d'approches dans la littérature. Toutes proposent des solutions satisfaisantes à ces contraintes dans leurs domaines d'application respectifs. Cependant, la plupart de ces approches sont trop liées à leurs domaines d'applications pour être réutilisées dans la modélisation du fonctionnement du sol. Dans notre approche, nous utilisons le paradigme agent pour encapsuler les divers processus du sol. Ces processus communiquent à travers l'espace en utilisant les ressources qui y sont présentes. Le comportement des processus dépend alors de la disponibilité des ressources. Un problème de coordination peut jaillir de ce type de couplage, les processus peuvent consommer de la ressource de manière simultanée alors que cette dernière peut ne pas supporter cette demande. Pour remédier à cela, nous utilisons une technique en théorie d'action appelée Influence-Reaction afin de définir des stratégies pour gérer ce type de situation. Nous nous sommes inspirés des approches présentes dans la littérature afin de gérer les différentes temporalités des processus.

Cette approche de couplage a été appliquée à un modèle de décomposition de la matière organique où plusieurs processus (vers de terre, microbes et racine) se disputent de la ressource dans un sol. Les résultats suggèrent que l'utilisation de notre approche sied à la modélisation du fonctionnement du sol, mais donnent aussi des indications plus précises quant à la disponibilité de la ressource.

Liste des abreviations

AA4MM : Agents and Artifacts for Multi-Modeling
ABM : Agent Based Model ou Modèle à base d'agents
AGRE : Agent Group Role Environment
APSF : Arborescent Pore Solid Fractal
BDI : Belief Desire Intention
BOID : Belief Obligation Intention Desire
CAMMiSoIE : Effet du Changement global en Afrique de l'ouest et à Madagascar sur la diversité des Microorganismes du Sol et ses conséquences sur les services Ecosystémiques
DEVS : Discrete EVent System Specification
DOM : Dynamic Oriented Modeling
DS-DEVS : Dynamic-Structure DEVS
EBM : Equation Based Model ou Modèle à base d'équations
FMI : Functional Mock-up Inteface
FMU : Functional Mock-up Unit
GAMA : GIS and Agent-Based Modeling Architecture
GAML : GAMA Modeling Language
HLA : High Level Architecture
IRM : Influence-Reaction Model
IRM4MLS : Influence-Reaction Model For Multi-Level Agent Based Simulation
IRM4S : Influence-Reaction Model For Agent Based Simulation
KGP : Knowledge Goal Plan
LWR : Lighthill-Whitham-Richards
MCT : Model Coupling Toolkit
ML-ABM : Multi-Level Agent Based Model
MM : Matière Minérale
MO (OM) : Matière Organique
Modèle DS : Modèle Domino-Stat
MOMOS : Modeling Organic Transformations by Microorganism of Soils
MPI : Message Passing Interface
OMG : Object Management Group
OMT : Object Model Template
OSIRIS : Outil de Simulation et d'Intégration de processus physiques pour l'étude

d'une Région Irriguée du Sénégal

PADAWAN : Pattern for Accurate Desing of Agent Worlds in Agent Nests

PSF : Pore Solid Fractal

ROT-MAT : Rotation Matrix

RTI : Run Time Interface

SIG : Système d'Information Géographique (GIS en anglais)

SIR : Susceptible-Infectious-Recovered

SMA : Système Multi-Agent

UML : Unified Modeling Language

VLE : Virtual Laboratory Environment

Table des matières

Liste des abreviations	7
Liste des figures	13
Introduction	17
I État de l'art	19
1 Problématiques en modélisation du sol	21
1.1 Description du sol	22
1.2 Modèles compartimentaux	23
1.3 Modèles à base d'agents	24
1.4 Synthèse	27
2 Hétérogénéité du sol	29
2.1 Notion de niveau et échelle	30
2.2 Hétérogénéité de représentation des processus	30
2.2.1 Modèles à base d'équations	30
2.2.2 Modèles à base d'agents	31
2.2.3 Combiner les deux représentations	31
2.3 Hétérogénéité spatiale	31
2.4 Hétérogénéité temporelle	32
2.4.1 Modèle à pas de temps	32
2.4.2 Représentation temporelle à base d'évènements	33
2.4.3 Représentation hybride du temps	33
2.5 Synthèse	34
3 Couplage de modèles	35
3.1 Concept et définitions	37
3.1.1 Concept	37
3.1.2 Définition	37
3.1.3 Types de couplages	40
3.2 Couplage de processus	42

		10
3.2.1	Discrete Event Systems Specifications (DEVS)	42
3.2.2	Functional Mock-up Interface (FMI) et High Level Architecture	43
3.2.3	OSIRIS	44
3.2.4	AA4MM	44
3.2.5	Approches à base d'agents de description des modèles multi-niveaux	45
3.3	Couplage spatialisé	46
3.3.1	Dynamic Oriented Modeling - Model DS	46
3.3.2	Coupling4Modeling	46
3.3.3	Couplage ad-hoc	47
3.4	Plateformes de couplage	48
3.4.1	Gama-Comodel	49
3.4.2	OpenMI	49
3.4.3	VLE-RECORDS	49
3.4.4	NetLogo-Levelscape	50
3.4.5	Model Coupling Toolkit (MCT)	50
3.5	Synthèse	50
4	Coordination de modèles	53
4.1	Coordination par organisation	54
4.1.1	Coordination réactive	54
4.1.2	Coordination hiérarchique	54
4.2	Coordination par réglementation	55
4.3	Coordination par argumentation et négociation	55
4.4	Synthèse : application des approches de coordination en couplage de modèles	56
II	Contribution	57
5	Positionnement	59
5.1	Le paradigme agent pour encapsuler les modèles	60
5.2	L'espace comme élément central du couplage	60
5.3	Un orchestrateur pour gérer les différentes temporalités	61
5.4	Coordination des sous-modèles	62
5.4.1	Influence Reaction: Principe	63
5.4.2	Coordination multi-niveaux	64
6	Architecture et implémentation de l'approche	67
6.1	Architecture de l'approche	68
6.1.1	Macro-Model	68
6.1.2	Micro-Model	68

		11
6.1.3	Space	69
6.1.4	Modification List	69
6.1.5	Scheduler	69
6.1.6	Coordination Unit	69
6.1.7	Strategy	69
6.1.8	Scale Transfert	70
6.2	Implémentation	70
6.2.1	La plateforme Gama	70
6.2.2	Coupler les sous-modèles à travers l'espace	71
6.2.3	Gérer la temporalité des sous-modèles	71
6.2.4	Coordination des sous-modèles	73
6.2.5	Coordination dans un espace multi-niveaux et multi-échelle	76
7	Application sur la modélisation du fonctionnement du sol	79
7.1	Le projet CAMMiSolE	80
7.1.1	Collecte de données	80
7.1.2	Modèle de prédiction: ROT-MAT	82
7.1.3	Modèle Participatif	83
7.2	Sous-modèles	83
7.2.1	Modèle de Sol: APSF	84
7.2.2	Modèle de Vers de terre: SWORM	85
7.2.3	Modèles de Microbe	85
7.2.4	Racines	86
7.3	Simulation	86
7.3.1	Modèle de simulation	86
7.3.2	Résultats et Discussion	87
	Conclusion	95
	Bibliographie	99

Liste des figures

1.1	Modèle conceptuel de sol (tiré du site cammisole.fr)	22
1.2	La dynamique du modèle MOMOS [Pan+10]	23
1.3	La dynamique du modèle CANTIS [Gar+03]	24
1.4	Représentation spatiale de la matière organique et des micro-organismes décomposeurs dans MIOR[Mas+07]	25
1.5	Représentation d’un canevas sur APSF [MLG18], type 1 : matière organique, type 2 : matière minérale	26
1.6	Représentation d’un sol en multi-niveaux avec plusieurs canevas [Mar+08]. La configuration spatiale d’un fractal est déterminée par un canevas.	27
2.1	Évolution de l’état d’un modèle à pas de temps	32
2.2	Représentation temporelle à base d’évènements	33
2.3	Représentation temporelle hybride, en considérant que “modèle 1” utilise une temporalité à pas de temps et “modèle 2” est à base d’évènements	34
3.1	Illustration imagée du couplage de modèles avec un Puzzle	37
3.2	Couplage fort entre deux modèles A et B	41
3.3	Couplage faible entre deux modèles A et B en	41
3.4	Construction d’un “macro-modèle” à partir de trois sous-modèles A, B et C par couplage intégratif	41
3.5	Représentation d’un atomic–devs dans [SHW18]	42
3.6	Représentation d’un modèle DEVS couplé ou les actomic–devs (A1,A2 et A3) sont reliés par leurs entrées et sorties	42
3.7	Architecture simplifiée de HLA	43
3.8	Modèle OSIRIS dans [Fia01]	44
3.9	Représentation d’un couplage entre deux <i>model-agents</i> dans AA4MM [Cam+12]	45
3.10	Représentation d’une interaction entre MDM dans l’approche DOM	46
3.11	Mise à l’échelle des données entre l’espace et les modèles dans cou- pling4modeling [Has15]	47
5.1	Les ressources de l’espace sont utilisées par les sous-modèles	61

5.2	Modèle de temporalité hybride où l'état du système est calculé lorsque les sous-modèles ou le modèle-couplé s'exécutent	62
5.3	Illustration de Influence Réaction sur deux sous-modèles produisant des influences sur l'espace ou ils évoluent.	63
5.4	Passage d'un niveau à un autre	65
5.5	Dynamique du système dans IRM4S [Mic07]: les agents et l'environnement produisent des influences (respectivement $\gamma'_n(t)$ et $\gamma'_w(t)$) au niveau micro, puis la réaction est déterminée au niveau macro pour calculer l'état du système	65
6.1	Diagramme de classe UML de l'approche	68
6.2	Méta-modèle de la plateforme Gama [Tai+19a]	70
6.3	Algorithme de synchronisation des exécutions [Din+22b]	72
6.4	Algorithme de résolution de conflit [Din+22b]: les influences envoyées sont évaluées et traitées par des stratégies lorsqu'un conflit est détecté. Dans le cas contraire, les influences sont appliquées aux sous-espaces.	75
6.5	Coordination dans une structure multi-niveaux	77
7.1	Session de jeu sérieux	81
7.2	Résultat à la fin de session d'un jeu sérieux: Tanim-bary, Tanety, Baiboho et Tendrimbohitra représentent les niveaux où sont situées les parcelles cultivées	82
7.3	Image en 3D de l'APSF sur la plateforme MaDKit[GF00] (image tirée de [Mar+08])	84
7.4	Caneva APSF: les canevas sont représentés par C0,C1,C2,C3 et sont alignés verticalement de sorte que ceux situés plus bas révèlent les patterns utilisés pour décrire les agrégats à des échelles plus petites (figure a). Dans la figure b, un sol en 2D construit en déployant différents canevas.	84
7.5	Représentation simplifiée du sol dans la plateforme gama	85
7.6	Le modèle-couplé à implémenter dans gama	87
7.7	sol de taille 4x4 (25% de matières minérales et de pores et 50% de matières organiques)	88
7.8	sol de taille 6x6 (25% de pores, 33% de matières minérales et 42% de matières organiques)	88
7.9	sol de taille 10x10 (27% de pores, 32% de matières minérales et 41% de matières organiques)	88
7.10	variation des ressources sans stratégie de coordination sur un sol de taille 4x4	89
7.11	variation des ressources avec coordination sur un sol de taille 4x4	89
7.12	variation des ressources sans stratégie de coordination sur un sol de taille 6x6	90
7.13	variation des ressources avec coordination sur un sol de taille 6x6	90

7.14	variation des ressources sans stratégie de coordination sur un sol de taille 10x10	90
7.15	variation des ressources avec coordination sur un sol de taille 10x10 . .	90
7.16	comparaison de la variation de la ressource raw_OM avec et sans coordination	91
7.17	comparaison de la variation de la ressource decomp_OM avec et sans coordination	91
7.18	comparaison de la variation de la ressource soluble_OM avec et sans coordination	91
7.19	comparaison de la variation de la ressource soluble_MM avec et sans coordination	91
7.20	comparaison des conflits détectés avec et sans coordination	92

Introduction

L'espace est l'un des éléments les plus importants lorsqu'on veut décrire des dynamiques comme les processus dans le sol [Bla+09]; [MLG18]; [Din+22b] ou l'occupation de territoire [Pay+06a]; [Has15]. Comprendre et étudier ces dynamiques requiert, dans la plupart des cas, des modèles de simulation afin d'observer leur évolution. Avec l'apparition de nouvelles problématiques comme le changement global, la demande en modélisation pour ce type de phénomène ne cesse de croître et requiert par conséquent l'amélioration ou la création de nouvelles approches de modélisation.

Dans cette thèse, nous nous intéressons à la modélisation du fonctionnement du sol. Le sol est un environnement fortement hétérogène en termes de structure que de population [Bla+09]. Cet environnement méconnu du grand public est cependant très complexe à comprendre en raison du nombre de dynamiques à prendre en compte et la grande variété de la faune fournissant les services écosystémique.

Cette étude du fonctionnement du sol se déroule à Madagascar, une région du monde où un bon nombre d'habitants subsiste grâce à l'agriculture. Cette activité si chère à ce pays présente cependant une forte dépendance à l'accès à l'eau (agriculture pluviale) et les paysans n'ont pas les moyens nécessaires pour se procurer des fertilisants. L'agriculture pluviale est une pratique à risque, car le monde est maintenant sujet au changement climatique. Les périodes de sécheresse nuisent au rendement des paysans et des agriculteurs locaux. De nouvelles pratiques agricoles doivent alors être mises en place pour s'adapter à ces changements. Pour cela, il faut étudier le fonctionnement du sol pour proposer les pratiques agricoles adéquates. Ce travail de thèse contribue à l'objectif du projet Cammisole (Effet du Changement global en Afrique de l'Ouest et à Madagascar sur la diversité des Microorganismes du Sol et ses conséquences sur les services Ecosystémiques), qui est de comprendre, par la modélisation, le fonctionnement du sol par rapport au changement global.

Cette thèse se situe dans la construction d'une approche de modélisation appelée "couplage spatialisé de modèle" pour décrire les différents aspects du fonctionnement écosystémique du sol. Comme le développement de modèles "from

scratch” est une pratique déconseillée [Mar16], le développement des nouveaux modèles se fait par la combinaison d’anciens modèles déjà validés pour décrire les différents aspects des phénomènes comme le fonctionnement du sol. Cette réutilisation de modèles existants pose des problèmes liés à l’hétérogénéité des représentations, due à la discipline d’origine des modèles.

La littérature offre une lecture abondante sur le couplage de modèle. Cependant, les approches existantes ne sont pas adaptées pour la modélisation spatialisée des dynamiques du sol. Les raisons à cela se situent dans la nature spatialisée et hétérogène du sol et les approches de couplages spatialisés existantes ne prennent pas en compte les possibles compétitions de ressources entre les modèles.

Nous nous inspirerons toutefois des démarches utilisées par ces approches existantes afin de construire une approche de couplage spatialisé de modèles, adaptée à la modélisation des processus du sol. Il faut alors répondre aux problématiques générales en couplage:

- comment faire communiquer les modèles écrits dans des formalismes différents (modèles à base d’équation et modèles à base d’agent) ?
- comment gérer l’hétérogénéité des représentations spatiales et temporelles des modèles ?
- comment garantir la cohérence des échanges entre les modèles ?
- comment garantir la cohérence des résultats de simulations ?

Pour répondre à ces problématiques, nous divisons ce mémoire en deux parties. Dans la première partie, nous développons les difficultés de modélisation en science du sol et les approches de couplages de modèles. Nous abordons dans le chapitre 1 les différentes méthodes de description des processus du sol. Dans le chapitre 2 nous évaluons les éléments de conception utilisés pour la description des processus du sol. Afin de rassembler ces éléments de conception, nous développons dans le chapitre 3, la notion de couplage de modèles tout en évaluant les différentes approches de couplage de modèles afin de tirer les avantages et les inconvénients de chaque approche et leur applicabilité en modélisation spatialisée du fonctionnement du sol. Dans le chapitre 4, nous abordons la coordination de modèle afin d’améliorer les approches de couplage de modèles existantes.

Dans la seconde partie, nous détaillons notre contribution en nous basant sur la littérature dans le chapitre 5, puis nous développons, sur la base de ce positionnement, une architecture de couplage spatialisé de modèles dans le chapitre 6. Enfin, dans le chapitre 7, nous évaluons l’efficacité de cette solution dans un modèle de décomposition de la matière organique.

Partie I

État de l'art

Chapitre 1

Problématiques en modélisation du sol

Contenu du chapitre

1.1	Description du sol	22
1.2	Modèles compartimentaux	23
	Modèle MOMOS	23
	Modèle CANTIS	24
	Modèle Min3P	24
1.3	Modèles à base d’agents	24
	Modèle MIOR	25
	Approche APSF	25
	Modèle SWORM	26
1.4	Synthèse	27

Note : Comme le terme “modèle” apparaît souvent et afin d’éliminer toute ambiguïté sur ce terme, nous utiliserons le terme “sous-modèle” (sub-model en anglais) pour nous référer aux modèles à coupler et “modèle-couplé” (coupled-model en anglais) pour indiquer le modèle résultant du couplage. Le terme “modèle” sera utilisé pour se référer aux modèles de manière générale.

Dans le sol, il existe plusieurs éléments, vivants et non vivant, dont les interactions définissent et régulent les propriétés physico-chimiques du sol. Afin de décrire ces éléments, les scientifiques modélisateurs s’aident d’outils de description tels que les mathématiques pour décrire la dynamique d’une population microbienne ou le paradigme agent pour décrire le comportement des vers de terre. Dans ce chapitre, nous détaillons les différentes représentations des éléments biotiques

(les êtres vivants) et les éléments abiotiques (non-vivants) du sol tout en expliquant les difficultés de modélisation liées à la nature hétérogène de la structure du sol et de ses processus [Ver+16].

1.1 Description du sol

Le sol est un espace fortement hétérogène où vivent plusieurs types d'êtres vivants. Pour le modéliser, il faut d'abord considérer sa partie non-vivante (ou abiotique) composée de matière organique, minérale et de pores [PBR00]; [Mar+08]; [Bla+09], et sa partie vivante (ou biotique) présente dans les pores, composée de plusieurs populations, de vers, de bactéries ou de champignons. Pour vivre et se reproduire, certains êtres vivants comme les vers de terre se nourrissent de la matière organique fraîche et rejettent un autre type de matière organique (dans les turricules) décomposée à son tour par les enzymes envoyées par microbes (décomposeurs). Cette décomposition enzymatique donne au soluté présent dans les pores, des nutriments assimilables par les microbes, qui sont à leur tour prédatés par une autre espèce appelé "nématodes" contribuant ainsi à la libération de nutriments pour les plantes.

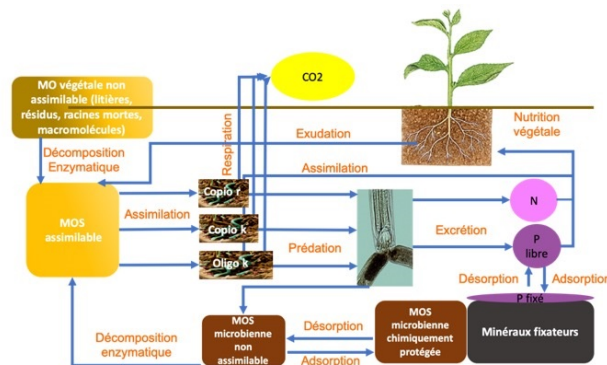


Figure 1.1: Modèle conceptuel de sol (tiré du site cammisole.fr)

Cette dynamique du sol est souvent décrite dans la littérature grâce à deux principaux formalismes, les modèles compartimentaux ou modèle à base d'équations et les modèles à base d'agents.

Les modèles compartimentaux ou spatiaux consistent à utiliser des systèmes d'équations mathématiques appelés "compartiments" pour décrire les différentes dynamiques du sol (évolution de la population microbienne, prédation, etc). Ce type de modèle est construit de façon modulaire, de sorte qu'un compartiment encapsule, grâce à une équation mathématique, une dynamique particulière du système à modéliser. Ces compartiments sont ensuite reliés les uns aux autres pour décrire les échanges entre différents processus du sol (Matières organiques, Microbes, etc).

Les modèles à base d’agents utilisent le paradigme agent, où des entités virtuelles appelées “agent” évoluent dans un environnement. Le sol est alors défini comme étant l’environnement ou évoluent des agents représentant les différents processus du sol. L’évolution de l’environnement qui résulte des interactions des agents (avec d’autre agent ou avec l’environnement) est appelé “phénomène d’émergence”.

1.2 Modèles compartimentaux

Les modèles compartimentaux ou aspatiaux, sont largement utilisés en science du sol et d’autres domaines (comme l’écologie des paysages) pour modéliser de manière modulaire des phénomènes complexes. Non seulement, ils permettent de représenter différentes dynamiques à plusieurs niveaux, mais ils sont surtout plus rapides à construire et à valider grâce à une riche littérature. Dans la suite, nous présentons différents modèles compartimentaux tiré de la littérature.

Modèle MOMOS

“The Modeling Organic Transformations by Microorganisms of Soils” ou MOMOS [Pan+10] consiste à simuler les phénomènes d’humification¹ et de minéralisation de la matière organique par les micro-organismes. Le modèle est constitué de cinq compartiments : les micro-organismes (MB), la nécromasse végétale labile et stable (VL et VS) ainsi que l’humus labile et stable (HL et HS)². La spatialisation des processus se fait par le biais de fonctions reliant les propriétés du sol et l’activité du compartiment contenant la biomasse microbienne.

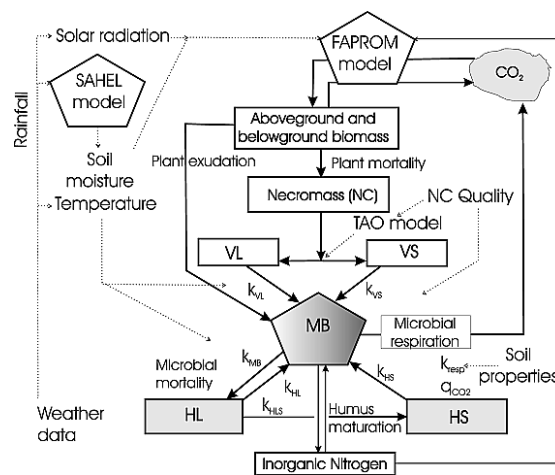


Figure 1.2: La dynamique du modèle MOMOS [Pan+10]

¹processus de transformation de la matière organique fraîche en humus

²Labile désigne les éléments facilement décomposables par les micro-organismes, à l’opposé de Stable

Modèle CANTIS

Le modèle CANTIS [Gar+03] vise à comprendre la transformation du carbone et de l'azote dans le sol. Tout comme MOMOS, CANTIS utilise des compartiments pour représenter des éléments du sol tel que l'humus (HOM) et la matière organique soluble (SOL) ainsi les micro-organismes (ZYB et AUB) responsable de la transformation du carbone et de l'azote.

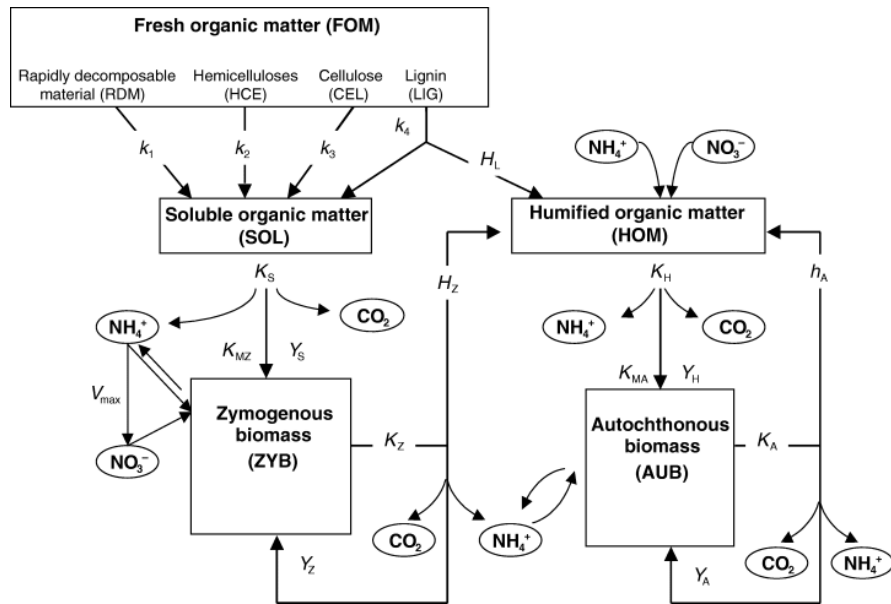


Figure 1.3: La dynamique du modèle CANTIS [Gar+03]

Modèle Min3P

Min3P est un outil de simulation pour décrire des modèles de transport réactive (transport dans les processus géochimiques) dans le domaine de l'hydrologie [May+12]. Son implémentation inclut plusieurs compartiments décrits pour modéliser le mouvement des liquides dans les milieux saturés. Cet outil de simulation a permis l'implémentation de plusieurs extensions dont min3P-Soil [Gér+08], conçu pour comprendre le cycle biogéochimique du silicium dans le sol.

1.3 Modèles à base d'agents

Les modèles à base d'agents permettent de construire une société virtuelle dans laquelle des entités appelées "agents" interagissent dans un environnement [Fer95]; [WC01]. Ce type de modélisation favorise une approche "bottom-up" [Mar16] en créant un modèle centré sur l'individu pour observer les résultats de leurs

interactions dans un environnement (phénomène d'émergence). Nous présentons dans la suite des modèles à base d'agents pour décrire les processus du sol.

Modèle MIOR

MIOR [Mas+07] est un modèle à base d'agents ayant pour objectif de tester des hypothèses sur la distribution spatiale de la matière organique ainsi que des décomposeurs sur les nutriments du sol (Carbone et Azote).

L'environnement est constitué de deux types de sphères réparties aléatoirement: la première représente les décomposeurs (MetaMior) et la seconde la matière organique (OMSphere). L'espace est décrit dans ce modèle, par l'ensemble de ces deux sphères. L'accessibilité des ressources (OMSphere) par les décomposeurs (MétaMior) est calculée en fonction de la distance entre les deux sphères.

MIOR ne représente pas la totalité de la structure du sol. Le modèle propose une représentation spatiale en 3D pour décrire l'accessibilité de la matière organique (OMSphere) par les micro-organismes (MetaMior).

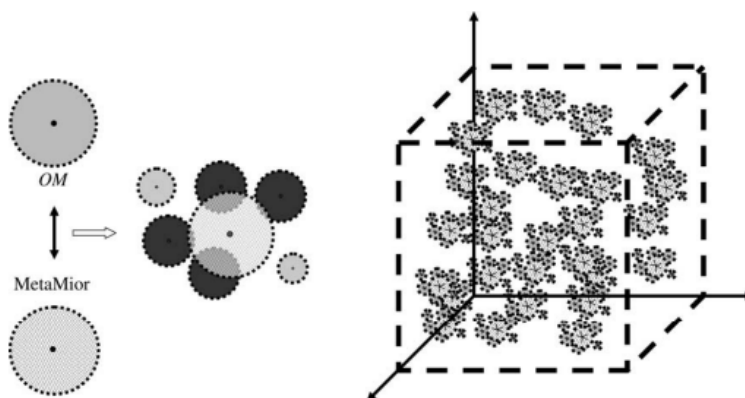


Figure 1.4: Représentation spatiale de la matière organique et des micro-organismes décomposeurs dans MIOR[Mas+07]

Approche APSF

Pore-Solid-Fractal est une approche proposée par [PBR00], permettant de reproduire la structure d'un sol grâce à la représentation fractale. Cette représentation permet de décrire l'hétérogénéité du sol en organisant un arrangement de différentes particules (qui peuvent être des pores, solides ou fractales) de façon répétitive et imbriquée.

Le sol est représenté comme un ensemble de particules. Une particule peut être un Pore (les cavités du sol), un Solide (les matières minérales et organiques) ou un Fractal (les micro-agrégats du sol, qui sont des ensembles de particules

(Pore et Solides) de petite taille).

L'approche PSF permet de représenter les différents types d'éléments composant le sol à différents échelle. Dans [Mar+08], les auteurs proposent une variante du PSF appelé APSF (Agent Pore Solid Fractal), permettant de modéliser grâce au paradigme Agent [Fer95] un sol virtuel en 3D. Une implémentation a été faite sur la plateforme GAMA [Tai+19a]. Grâce au paradigme agent, il est possible de modifier les attributs ou les comportements des particules qui composent l'APSF. On peut alors attacher des processus (qui sont souvent des agents) à ces particules pour modifier leurs états.

Plusieurs modèles ont été construits autour des deux approches (APSF et PSF):

- Le modèle Swarm [Bla+09] est l'un des premiers à faire une intégration de processus dans l'APSF. L'idée est d'intégrer dans le sol virtuel, des agents vers de terre afin d'observer la restructuration du sol par ces derniers. Les vers de terre décomposent la matière organique du sol tout en créant des cavités dans le sol.
- L'étude de la perméabilité des structures poreuses en utilisant la représentation PSF [PBR10].

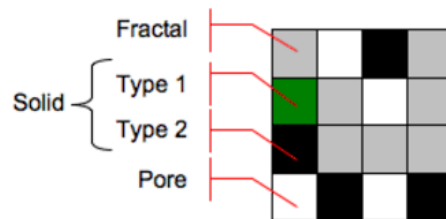


Figure 1.5: Représentation d'un canevas sur APSF [MLG18], type 1 : matière organique, type 2 : matière minérale

Modèle SWORM

Le modèle SWORM [Bla+09] est un modèle à base d'agents, décrivant le comportement des vers de terre dans le sol. Le modèle reprend la représentation APSF de [Mar+08] pour décrire le sol et incluent deux catégories d'agents vers de terre (*Millsonia Anomala* et *Eudrilidae*). L'aspect modulaire des systèmes multi-agent permet à ce modèle de s'améliorer en introduisant d'autres processus comme l'activité microbienne, l'infiltration d'eau, etc.

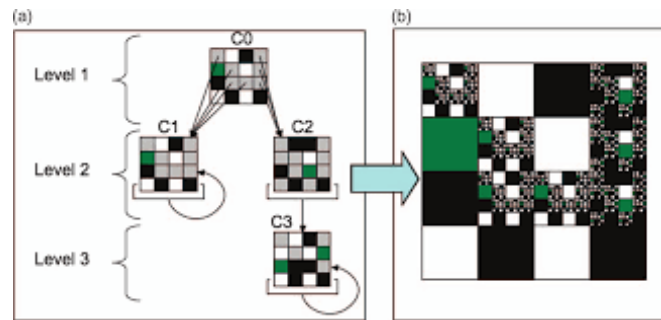


Figure 1.6: Représentation d'un sol en multi-niveaux avec plusieurs canevas [Mar+08]. La configuration spatiale d'un fractal est déterminée par un canevas.

1.4 Synthèse

À partir de ces descriptions du sol et de ces processus, nous pouvons déduire que les modèles de simulation des processus du sol sont décrits par deux types d'écriture, les modèles à base d'agents et les modèles à base d'équations. Les modèles à base d'équations proposent une représentation généralisée des processus du sol. L'aspect spatial du sol est présenté par des compartiments dans lesquels des équations décrivent une dynamique particulière du sol. Les modèles à base d'agents offrent par contre un point de vue individualisé des processus dans un environnement représentant l'espace. Cette représentation propose une description plus détaillée de l'hétérogénéité du sol.

Chapitre 2

Hétérogénéité du sol

Contenu du chapitre

2.1	Notion de niveau et échelle	30
2.2	Hétérogénéité de représentation des processus	30
2.2.1	Modèles à base d'équations	30
2.2.2	Modèles à base d'agents	31
2.2.3	Combiner les deux représentations	31
2.3	Hétérogénéité spatiale	31
2.4	Hétérogénéité temporelle	32
2.4.1	Modèle à pas de temps	32
2.4.2	Représentation temporelle à base d'évènements	33
2.4.3	Représentation hybride du temps	33
2.5	Synthèse	34

Dans ce chapitre, nous abordons l'hétérogénéité du sol sous un angle conceptuel en divisant les problèmes de modélisation. Tout d'abord, dans la section 2.1 où nous développons le concept de niveau et d'échelle pour différencier la manière dont les différents modèles sont organisés, puis dans la section 2.2, nous détaillons les deux types de formalismes (modèle à base d'agents et modèle à base d'équations) principalement utilisés pour la modélisation du sol, leurs avantages, leurs inconvénients et comment choisir entre les deux types d'écritures. Dans les sections 2.3 et 2.4 nous parlerons des méthodes de représentation des échelles spatiales et temporelles.

2.1 Notion de niveau et échelle

Les concepts de niveau et d'échelle sont souvent confondus en modélisation des systèmes complexes. [Mor13] définit un niveau comme étant un point de vue d'un système intégré dans un modèle. On fait souvent référence au niveau lorsqu'on parle de niveau micro, meso ou macro. Dans les modèles de dynamique de population [DRP03] et [MLG18] par exemple, le niveau macro est souvent exprimé par des équations mathématiques et le niveau micro par un modèle à base d'agents décrivant le comportement des individus d'une population. Dans [GLH12], le concept de "niveau" est utilisé lorsqu'on fait référence à un point de vue d'un système, tout en prenant en compte son organisation, sa dimension et ses interactions. [GLH12] associe le terme "échelle" aux dimensions spatiales, temporelles et quantitatives d'un modèle. Dans le sol par exemple, il existe des particules de différente taille (échelle spatiale), des modèles avançant à des temporalités différentes (échelle temporelle) ou utilisant différentes grandeurs physiques sur leurs entrées et sorties (g/cm^2 , kg/cm^3 etc.).

On utilise alors le terme "niveau" lorsqu'on fait référence à un point de vue d'un système et au terme "échelle" pour indiquer une étendue ou une résolution, qu'elle soit spatiale, temporelle ou pour indiquer une quantité.

2.2 Hétérogénéité de représentation des processus

Les processus du sol sont représentés par différentes manières dans la littérature: grâce à des modèles populationnels ou à base d'équations ou des modèles centrés sur l'individu ou à base d'agents.

2.2.1 Modèles à base d'équations

Les représentations populationnelles ou modèles à base d'équations considèrent un aspect homogène d'un phénomène. Dans [VSR98], un modèle à base d'équations est considéré comme étant un ensemble d'équations donnant l'évolution de certaines quantités à travers le temps et l'espace (avec des équations aux dérivées partielles). Les équations représentent chacune la dynamique globale d'un phénomène dans lequel on fait abstraction des comportements individuels des entités qui composent le modèle. Par exemple, les microbes sont souvent représentés par des modèles à base d'équations (dans le sol) car on s'intéresse à la dynamique d'un ensemble de microbes au lieu d'un seul. Dès lors que l'on fait abstraction de cet aspect du comportement individuel, cette représentation est généralement utilisée. Les modèles à base d'équations sont naturellement simplificateurs dans la mesure où l'on veut décrire un phénomène à un niveau global. Cependant, l'utilisation de ces modèles est limitée lorsqu'on doit considérer différents niveaux et échelles

spatiales ou des évènements au niveau individuel des dynamiques composant un phénomène [MLG18].

2.2.2 Modèles à base d’agents

Un modèle à base d’agents consiste à capturer un ensemble de comportements et attributs identifiant un individu. L’exécution des comportements des individus constituant le système crée un phénomène d’émergence, qui décrit l’évolution du phénomène étudié [VSR98]. Grâce au niveau de détails donné par les modèles à base d’agents, ce type de modèle permet de reproduire une dynamique qui est plus proche de la réalité observée [HMK18]. Il est possible de définir des dynamiques multi-niveaux [Mor13] grâce à ce type de représentation, donnant ainsi la possibilité de décrire un phénomène à une échelle plus fine que les modèles à base d’équations.

2.2.3 Combiner les deux représentations

Les modèles à base d’équations et les modèles à base d’agents sont souvent comparés [VSR98] [HMK18]. Bien que les deux approches permettent la description des phénomènes du sol, la principale différence réside dans le niveau d’observation: les modèles à base d’équations décrivent un phénomène de manière globale (niveau macro) alors que les modèles à base d’agents sont souvent utilisés pour une description au niveau de l’individu (niveau micro). Combiner les deux approches est possible afin de donner une représentation plus complète [VSR98]; [DRP03]; [Ban+16]; [MLG18]. On peut alors prendre avantage des deux représentations, c’est à dire, la possibilité de décrire un phénomène à plusieurs niveaux, la prise en compte des différentes échelles des dynamiques composant le phénomène. Cependant, cette combinaison entraîne des contraintes de modélisation que nous détaillerons dans une autre partie (section 3).

2.3 Hétérogénéité spatiale

L’espace peut être décrit de différentes manières dans un modèle. Grâce à des compartiments [Gar+03]; [Pan+10] ou des équations aux dérivées partielles [VSR98] sur les modèles à base d’équations et grâce à différentes représentations de l’environnement (grilles, cartes SIG, images) sur les modèles à base d’agents [Mar+08], [Dro+20], [Din+22b]. Ce problème d’hétérogénéité spatiale est surtout présent lorsque l’on veut combiner divers modèles. Les modèles à base d’équations sont moins concernés par ce problème car les représentations spatiales sont souvent considérées comme homogènes. Cependant, dans les approches à base d’agents comme [Mar+08] et [Bla+09], l’espace est représenté de manière hétérogène, c’est à dire que l’environnement (l’espace) est subdivisé en sous-espaces de différentes

tailles. On peut observer cela dans les modèles en urbanisation [Pay+06b] ou encore [Has15] dans lesquels l’environnement est divisé en plusieurs niveaux (ville, région). Comme le sol est un environnement fortement hétérogène, il est nécessaire d’élaborer des approches permettant de gérer comment les données sont agrégées afin d’avoir une évolution cohérente des dynamiques présentes.

2.4 Hétérogénéité temporelle

Pour représenter un phénomène d’une manière plus complète, il faut prendre en compte l’exécution des processus qui le composent. L’exécution de ces processus dans le temps peut aussi varier d’un processus à un autre. Leurs dynamiques dans le temps sont souvent représentés grâce à un modèle à pas de temps, à un modèle à base d’évènements ou la combinaison des deux types de représentations (On parle alors de représentations hybrides) [Pay+06b]; [Ral+19].

2.4.1 Modèle à pas de temps

Le modèle à pas de temps prend une référence temporelle δ_t appelée “pas de temps”, définie par le modélisateur, pour exécuter un modèle. Un système utilisant ce type de représentation évolue en exécutant les modèles à chaque incrémentation du pas de temps δ_t . En d’autres termes, l’état du système change à une période unique δ_t . Dans les modèles à base d’équations (2.1) par exemple, la temporalité est exprimée en pas de temps.

$$P(t + 1) = r.P(t)\left(1 - \frac{P(t)}{K}\right) \quad (2.1)$$

Équation (2.1) logistique décrivant l’évolution d’une population P , un taux de croissance r et une capacité maximale K . Nous illustrons dans la figure 2.1 l’évolution de ce type de modèle.

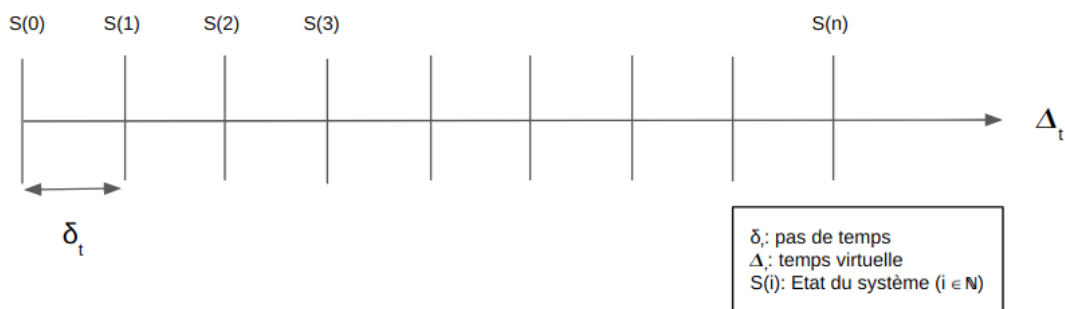


Figure 2.1: Évolution de l’état d’un modèle à pas de temps

Nous pouvons voir ce type de représentation dans les modèles à base d'équations. Ce modèle temporel est utilisé dans les équations logistiques comme le calcul de l'accroissement de la population microbienne [Pan95] ou pour observer l'évolution des modèles numériques telle que [Gar+03]; [Pan+10].

2.4.2 Représentation temporelle à base d'évènements

Dans un modèle évoluant avec une temporalité à base d'évènements, l'état du modèle change lorsqu'un évènement se produit. Contrairement au modèle à pas de temps, l'évolution d'un système à base d'évènements se fait à un intervalle irrégulier, c'est à dire que le temps de passage d'un état à un autre ne correspond pas à la même durée [$\delta(S(1) \rightarrow S(2))$] est différent de [$\delta(S(2) \rightarrow S(3))$]. Cela est dû au fait que les évènements amenant à un changement d'état sont dictés par des conditions d'exécution. Ainsi, la mise à jour de l'état d'un système se fait à différentes périodes, en fonction du déclenchement d'un évènement.

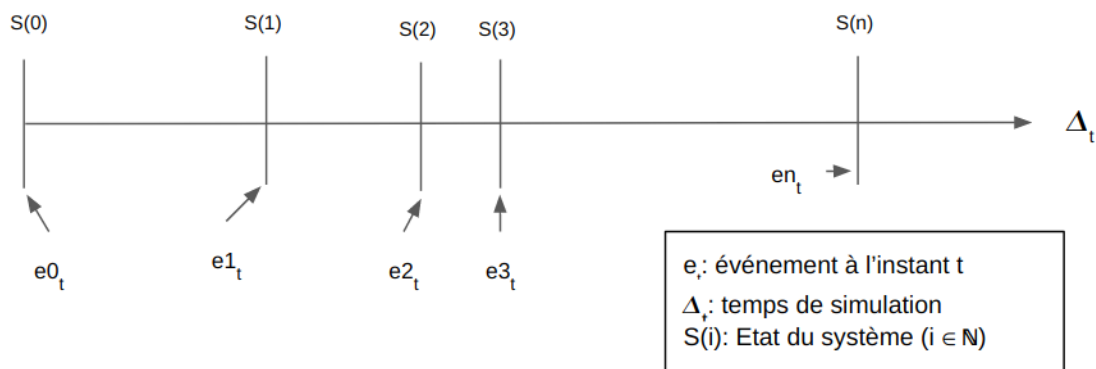


Figure 2.2: Représentation temporelle à base d'évènements

L'apparition des approches telles que DEVS [ZV93] ont permis de reconstruire ce type de représentation temporelle en reliant plusieurs processus. Individuellement, les processus ont leurs propres pas de temps, considérés comme des évènements internes. En reliant les processus entre eux, un processus peut envoyer des informations à ses paires et peut déclencher des évènements externes sur d'autres processus pour qu'ils actualisent leurs états. Ce modèle de temporalité est aussi observé dans les modèles à base d'agents comme dans [Fia01] ou encore [Bla+09].

2.4.3 Représentation hybride du temps

On a recours à ce type de représentation hybride lorsqu'un système possède des sous-systèmes (ou sous-modèles) ayant des modèles de temporalités différentes (à base d'évènements et de pas de temps). Le système globale passe d'un état à

un autre lorsque l'un des sous-systèmes change d'état. L'ensemble des états du système (S) est donc l'union des ensembles des états de ses sous-systèmes (s_i) ($S = \bigcup_{i=0}^n s_i$).

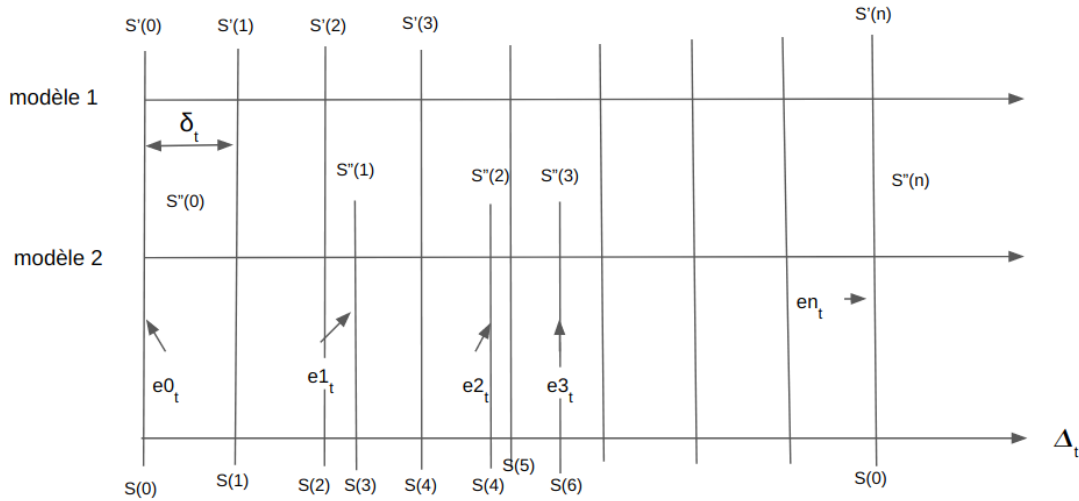


Figure 2.3: Représentation temporelle hybride, en considérant que “modèle 1” utilise une temporalité à pas de temps et “modèle 2” est à base d’évènements

On retrouve souvent ce type de représentation du temps dans les approches de multi-modélisation comme le couplage de modèles [ZV93]; [Fia01]; [Ber+09]; [Has15], dans lesquelles un modèle est considéré comme un ensemble de sous-modèle.

2.5 Synthèse

Comme les processus du sol sont représentés de différentes manières (à base d’équations ou agents) dans la littérature, il est nécessaire de concilier ces différentes écritures afin de prendre avantage de chaque type de représentation [MLG18] et aboutir à une description plus complète du sol [Bla+09]. Pour combiner ces processus, il faut tenir compte de leurs représentations temporelles et spatiales en plus d’autres contraintes de cohérence. Pour ce faire, nous détaillerons dans le chapitre suivant une approche appelée “couplage de modèles”, qui prend en compte toutes ces contraintes.

Chapitre 3

Couplage de modèles

Contenu du chapitre

3.1	Concept et définitions	37
3.1.1	Concept	37
3.1.2	Définition	37
	Pourquoi coupler des modèles?	38
	Contraintes de couplage	39
3.1.3	Types de couplages	40
	Couplage fort	41
	Couplage faible	41
	Couplage intégratif	41
3.2	Couplage de processus	42
3.2.1	Discrete Event Systems Specifications (DEVS)	42
3.2.2	Functional Mock-up Interface (FMI) et High Level Architecture	43
3.2.3	OSIRIS	44
3.2.4	AA4MM	44
3.2.5	Approches à base d'agents de description des modèles multi-niveaux	45
3.3	Couplage spatialisé	46
3.3.1	Dynamic Oriented Modeling - Model DS	46
3.3.2	Coupling4Modeling	46
3.3.3	Couplage ad-hoc	47
3.4	Plateformes de couplage	48
3.4.1	Gama-Comodel	49

3.4.2	OpenMI	49
3.4.3	VLE-RECORDS	49
3.4.4	NetLogo-Levelscape	50
3.4.5	Model Coupling Toolkit (MCT)	50
3.5	Synthèse	50

3.1 Concept et définitions

3.1.1 Concept

Un modèle répond généralement à une problématique existante en faisant abstraction du monde réel. Par exemple, le modèle *sworm* [Bla+09] permet de répondre aux questionnements sur la restructuration du sol par les vers de terre. Cependant, ces problématiques peuvent imbriquer ou engendrer d'autres questionnements comme : comment les bactéries réagissent-elles par rapport à cette restructuration? L'humidité est-elle un facteur important dans l'activité microbienne? Ou encore le changement global a-t-il un effet sur les activités du biote du sol? Pour répondre à ces questionnements, il faut prendre en compte plusieurs dynamiques/processus qui interagissent entre eux. Le progrès en termes de modélisation des phénomènes environnementaux a créé un nombre considérable de modèles répondant à la plupart de ces questionnements [BK00a]. Comme le développement et la validation de modèles est un processus très coûteux en moyen ou/et en temps, pour répondre à ces questionnements complexes, il est suggéré de réutiliser et de combiner les modèles existant [Mar16]. Cependant, ces modèles (appelés "legacy models") viennent souvent de disciplines variées, et constituent un énorme challenge en modélisation à cause de l'hétérogénéité des représentations (souvent propres à la discipline d'origine) à prendre en compte.

On peut alors imaginer le couplage de modèles comme étant un puzzle dont les pièces représentent les modèles à combiner dans le but d'avoir une image. Bien évidemment, pour avoir le résultat souhaité, les pièces doivent répondre à des contraintes de positionnement.

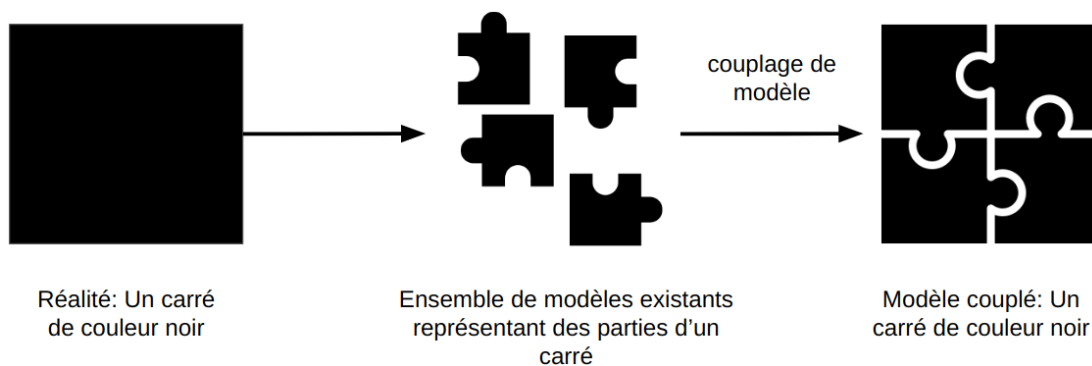


Figure 3.1: Illustration imagée du couplage de modèles avec un Puzzle

3.1.2 Définition

Couplage de modèles En génie logiciel, le couplage se définit comme étant une marque de dépendance fonctionnelle entre plusieurs entités logicielles pour

augmenter le niveau de description d’une application [Fre+19]. En modélisation de systèmes complexes, le couplage de modèles partage ce même désir d’accroître le niveau de détails dans un modèle.

Le couplage de modèles est défini dans la littérature comme étant une démarche combinatoire où l’on fait interagir plusieurs modèles indépendants, dans le but de décrire un système de façon plus complète [Has15]; [Mar16].

Dès que l’on met en interaction deux modèles indépendants¹, on applique une approche de couplage de modèles. Ce fut le cas lorsque les systèmes complexes étaient décrits grâce à des modèles à compartiments [KM27]; [Eri71]; [Bra08], où chaque compartiment représente une dynamique particulière du système à étudier. Le couplage de modèles a été rendu plus sophistiqué grâce à l’apparition des approches telles que DEVS [ZV93] (qui est l’une des approches la plus populaire à ce jour), l’apparition des paradigmes Objet [Boo82] et Agent [Sho93]; [Fer95]; [WC01] sur lesquels plusieurs approches [Fia01]; [Has15] et plateformes de couplage [TW04]; [PM11]; [Tai+19a] ont été construites, ainsi que les approches de couplage de simulateurs [Blo12]; [Awa+13]; [Cam+12] où plusieurs simulateurs intégrant chacun leurs propres dynamiques ou modèles, échangent des informations à travers une entité logicielle.

Couplage spatialisé de modèle Par référence à la définition du couplage de modèles, le couplage spatialisé de modèle est une approche de couplage où les sous-modèles sont combinés à travers l’espace.

Par exemple, dans les modèles de dynamiques urbaines [Dav+07]; [Has+12] on retrouve ce type particulier de couplage, où différents sous-modèles comme le mouvement migratoire des populations et les transports sont spatialisés pour rendre leurs données disponibles à d’autres sous-modèles. Il en est de même pour la modélisation des phénomènes du sol [Gar+03]; [Pan+10] où les différentes parties du sol sont réparties dans des compartiments représentant l’espace. Il est aussi pratique courante de coupler des systèmes d’informations géographiques (SIG) avec des modèles mathématiques pour étudier des systèmes en hydrologie [CEH96]; [GGW07], en épidémiologie [Kha+21]; [Dro+20] ou en écologie du paysage [MLG18].

Pourquoi coupler des modèles?

Le couplage de modèles se révèle être un outil permettant d’économiser le temps et les moyens alloués à la production d’un nouveau modèle, mais se caractérise également par un challenge dû à la discipline d’origine des sous-modèles. Les avantages du couplage de modèles sont étroitement liés à la réutilisation de modèles existants.

¹qui fonctionne indépendamment d’un autre modèle

- Économiser sur les coûts et le temps de développement: il existe des similitudes entre le couplage de modèles et la réutilisation de composants en génie logiciel. En génie logiciel comme en couplage de modèles, un système est un assemblage de plusieurs parties interconnectées plutôt qu’une entité monolithique [Lau06]. Développer un modèle à partir de zéro est un processus laborieux, coûteux en temps et moyens qui nécessite un processus de validation sur les données observables sur le terrain [BK00a]. Il est alors plus judicieux de réutiliser des modèles existants afin de minimiser les coûts matériels.
- Utiliser une démarche participative: l’aspect pluridisciplinaire du couplage de modèles permet d’impliquer les différentes parties prenantes dans le processus de modélisation et ainsi avoir plusieurs points de vue et discussion pendant le processus de modélisation [Tai+19b]. Il est alors plus facile de raffiner ou de modifier les sous-modèles grâce à la participation des experts des domaines.
- Avoir une représentation plus complète du phénomène à modéliser: en reprenant la définition de [Lau06], un système est un ensemble de sous-systèmes interconnectés plutôt qu’une entité monolithique. Le couplage de modèles suit la même optique en regroupant les dynamiques qui composent un phénomène à modéliser. Plutôt que d’avoir un point de vue global sur le phénomène à modéliser, il sera alors possible d’observer les différentes dynamiques qui la composent, mais aussi les influences (uni- ou multidirectionnel) entre les dynamiques en interaction.

Contraintes de couplage

En raison du caractère pluridisciplinaire ainsi que de la réutilisation des modèles, il est possible que les “représentations” des sous-modèles soient différentes les unes des autres. Par représentation, nous nous référons au formalisme utilisé pour écrire le modèle (schématique, mathématique, informatique...), ainsi que les aspects temporels et spatiaux utilisés.

La question la plus évidente lorsqu’on parle de couplage est la manière de mettre les sous-modèles en interactions, puis viennent les questions adjacentes sur l’harmonisation des échelles et les différentes écritures des modèles. Dans sa thèse, [Has15] résume ces questionnements en trois points : Hétérogénéité des formalismes, Interaction et Multi-échelle.

- L’hétérogénéité des formalismes: en communication, deux personnes parlant chacune différentes langues ne peuvent communiquer qu’à travers un langage commun ou un outil de traduction. Il en va de même en couplage de modèles. Les sous-modèles sont écrits ou implémentés dans des langages de programmation ou des simulateurs différents. Pour pouvoir coupler ces modèles, il

faut recourir à des techniques de réécriture (encapsulation) (comme DEVS [ZV93] et les SMA [Fer95]).

- Les contraintes d’interaction regroupent les questionnements relatifs à la manière dont les modèles sont connectés. Il s’agit donc de trouver un “facteur de couplage” permettant aux modèles de communiquer. Ce facteur de couplage peut être une entité logicielle [Blo12]; [Awa+13]; [Cam+12] ou tout simplement les entrées et sorties des sous-modèles [ZV93]; [GGW07]; [HHM19]. En levant ces contraintes, on définit alors la stratégie ou le type de couplage utilisé. S’agit-il de couplage faible? fort ou intégratif? (nous définissons ces types de couplage dans la section 3.1.3).

On peut aussi parler de contrainte d’interaction lorsque les données utilisées par les sous-modèles ne sont pas exprimées dans le même ordre de grandeur (l’un en g/m^2 et l’autre en Kg/m^2 par exemple), on parle alors d’intéropérabilité de données [BK00b]; [Cam+12]; [Has15].

- L’échelle temporelle et spatiale: il est aussi possible que les sous-modèles évoluent à des échelles de temps différentes. Il est alors nécessaire de synchroniser les différentes temporalités pour avoir une représentation cohérente.

Par exemple, dans un sol, un vers de terre consomme de la matière organique toutes les heures alors que les microbes décomposent de la matière organique toutes les quatre heures [Din+22b].

La représentation spatiale des sous-modèles peut aussi être différente. Dans les modèles d’occupation du sol par exemple, l’espace est représenté par des grilles ou des cartes (modèles Système d’Information Géographique (SIG)) [GGW07]. En écologie, l’espace est considéré comme un compartiment [Pan+10] [Gar+03] ou des agents comme dans [Mas+07]; [Mar+08]; [Bla+09]. Il se peut alors que l’on ait plusieurs représentations de l’espace (structure multi-niveaux), qui nécessitent des fonctions de mise à l’échelle [Has15]; [Mar16].

3.1.3 Types de couplages

Trois types de couplages sont communément cités dans la littérature, le couplage faible, fort et intégratif. Le type de couplage détermine la stratégie par laquelle les sous-modèles communiquent.

Naturellement, cette stratégie de couplage se fait de deux manières possibles: directement dans le cas où les sous-modèles utilisent leurs entrées et sorties respectives pour communiquer ou indirectement lorsque les dynamiques décrites par sous-modèles ne partagent aucune entité du système nécessitant alors un intermédiaire pour faciliter leur communication [Mar16]. Une entité logicielle (un bus, les entrées et sorties, un environnement, etc.) appelée “facteur de couplage” est alors déterminée afin de répondre à ce problème de communication.

La stratégie de couplage introduit alors la notion d'interopérabilité des modèles, qui se traduit comme étant la manière dont les sous-modèles s'échangent l'information [BK00b].

Couplage fort

Le couplage fort consiste à relier les sous-modèles de sorte que les entrées et sorties deviennent les points de jointures entre les sous-modèles. C'est la plus simple des techniques de couplage. Les inconvénients résident dans la gestion des échelles temporelles et spatiales des sous-modèles, lorsque ces derniers nécessitent des fonctions de mise à l'échelle pour être interopérables [Mar16].

Couplage faible

Le couplage faible résout les limitations du couplage fort [Has15]. Ce type de couplage consiste à faire communiquer les sous-modèles grâce à une tierce entité, qui peut être une base de données, l'environnement, un bus, etc. de sorte que les données soient mises à l'échelle de façon automatique pour les rendre disponibles aux autres sous-modèles.

Couplage intégratif

Le couplage intégratif ou couplage intégral vise à construire de façon modulaire un "macro-modèle" en combinant différents sous-modèles pour en construire un seul modèle [Mar16]. Dans ce type de couplage, la mise en échelle des données et l'interopérabilité des modèles sont déterminés au préalable.

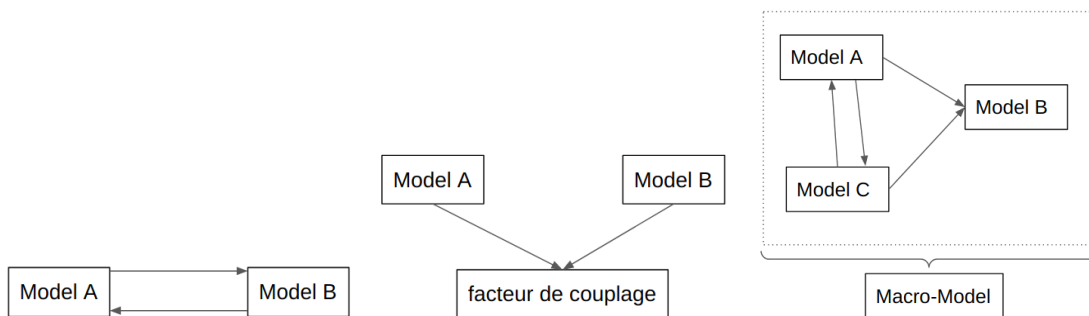


Figure 3.2: Couplage fort entre deux modèles A et B

Figure 3.3: Couplage faible entre deux modèles A et B en

Figure 3.4: Construction d'un "macro-modèle" à partir de trois sous-modèles A, B et C par couplage intégratif

3.2 Couplage de processus

3.2.1 Discrete Event Systems Specifications (DEVS)

Discrete Event System Specification ou DEVS [ZV93] est un formalisme permettant de décrire les systèmes de manière hiérarchique. Les composants du système (les sous-modèles) sont décrits par des “atomicdevs” (noté “A” et décrite par l’équation (3.1)). Ces atomic–devs peuvent se connecter entre eux par leurs entrées (X) et sorties (Y) pour avoir un modèle couplé appelé “coupled-devs”. L’évolution du système est prise en charge par la fonction de transition à chaque occurrence d’événement interne ou externe. Un atomic–devs peut s’écrire comme suit:

$$A = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle \quad (3.1)$$

X : l’ensemble des ports et des valeurs en entrée

Y : l’ensemble des ports et des valeurs en sortie

S : l’ensemble des états système

δ_{ext} : la fonction de transition externe

δ_{int} : la fonction de transition interne

λ : la fonction de sortie

ta : la fonction d’avancement du temps

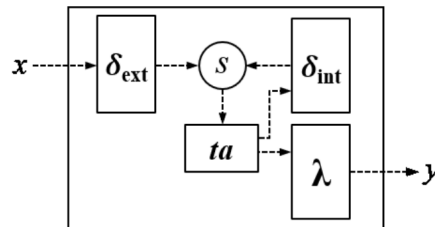


Figure 3.5: Représentation d’un atomic–devs dans [SHW18]

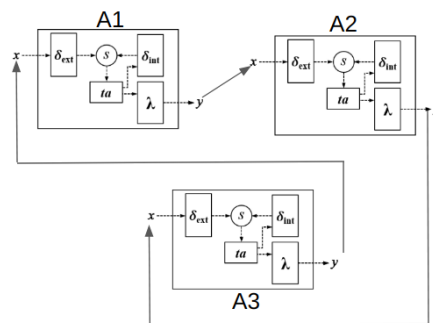


Figure 3.6: Représentation d’un modèle DEVS couplé ou les atomic–devs (A1,A2 et A3) sont reliés par leurs entrées et sorties

DEVS est largement utilisé dans le monde de la modélisation, donnant naissance à des extensions comme Parallel-DEVS [CZ94], Cell-DEVS [WG01] ou des plateformes de modélisation comme VLE [QDR09] et RECORDS [Ber+09].

3.2.2 Functional Mock-up Interface (FMI) et High Level Architecture

FMI [Blo12] est un standard créé pour rendre interopérable et co-simulable des modèles utilisés dans les industries de manufactures. Les sous-modèles doivent implémenter le FMU ou Functionnal Mock-up Unit dans un fichier d’extension “.fmu” dans lequel trois composants sont définies :

- Les données (entrées et sorties des modèles) sont définies dans un fichier XML.
- Un set de fonctions écrit en langage C contenant l’implémentation modèles pour la co-simulation.
- Les données sur la documentation ou d’autres ressources comme les icônes.

HLA [DKW98]; [Awa+13] dans la même philosophie que FMI, fournit un outil pour l’interopérabilité des modèles dans le cadre d’une simulation distribuée (appelé “Fédération” dans le jargon de HLA, tandis que les simulateurs composant ce système distribué est appelé “Fédéré”). L’architecture de HLA inclus un ensemble de règle appelé “HLA rules” que doivent suivre les simulateurs (les Fédérés et la Fédération), une interface appelé “HLA Interface Specification” qui gère tout ce qui est communication entre les fédérés, puis l’OMT (Object Model Template) qui spécifie comment les objets doivent être impémentés. Un bus appelé RTI (Run Time Interface) créé par [Awa+13] sert de médiateur de communication entre les fédérés et gère tout ce qui est synchronisation ou exécutions des fédérés et son implémentation suit les standards imposés par HLA Interface Specification. le bus RTI [Awa+13] a été conçu dans l’objectif d’apporter plus de contrôle sur les simulateurs composant une simulation distribuée. [Awa+13] ont rendu possible l’utilisation des composants FMU en tant que simulateur fédéré à travers le RTI.

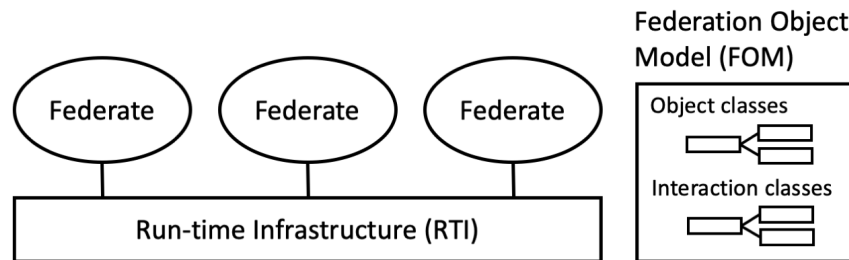


Figure 3.7: Architecture simplifiée de HLA

3.2.3 OSIRIS

OSIRIS [Fia01] (Outil de Simulation et d’Intégration de processus physiques pour l’étude d’une Région Irriguée du Sénégal) est une approche de couplage de processus conçu autour d’un système pivot. Le système pivot est constitué d’agents contrôleurs qui permettent l’exécution des sous-modèles. Un agent contrôleur est rattaché à chaque processus et communique les données aux autres agents contrôleurs, ce qui permet aux processus de communiquer. Un agent contrôleur de temps gère la temporalité des processus en permettant à l’utilisateur de changer de pas de temps en cours de simulation. L’exécution des sous-modèles se fait de façon “pessimiste”, c’est à dire que le système mis en place n’exécute que les évènements sûrs, où aucune erreur de causalité ne survient. Une interface reliée à chaque processus permet de visualiser l’évolution de celle-ci pendant la simulation.

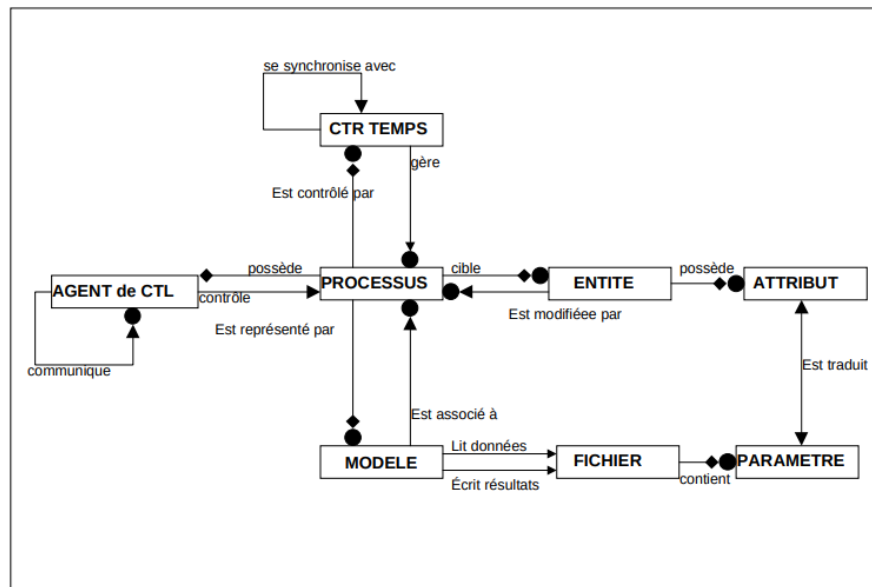


Figure 3.8: Modèle OSIRIS dans [Fia01]

3.2.4 AA4MM

AA4MM (Artefact and Agent for Multi-Modeling) [Cam+12] est un métamodèle destiné au couplage de simulateur. Les artefacts sont considérés comme étant des objets utilisables par les agents et sont présents dans l’environnement d’un système multi-agent [RVO07]. Dans AA4MM, des agents appelés “model-agent” ont pour tâche d’exécuter les sous-modèles, les artefacts quant à eux sont utilisés pour décrire le modèle à plusieurs niveaux : syntaxique (qui décrit les composants du sous-modèle), sémantique (qui décrit comment le sous-modèle est implémenté), dynamique (décrit l’exécution d’un sous-modèle) et technique (pour définir le

simulateur dans lequel le sous-modèle est exécuté). On distingue deux types d'artéfacts: *Model artefact* qui décrit comment les sous-modèles à ces quatre niveaux et *Coupling Artefact* qui décrit comment un *model-agent* communique avec d'autres *model-agents*.

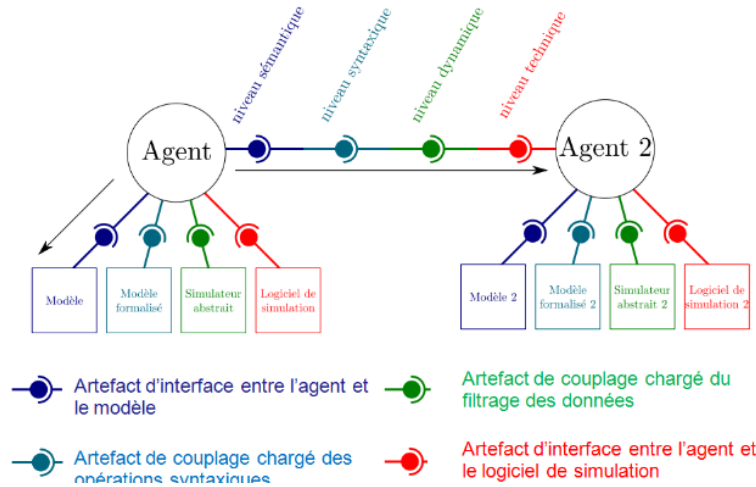


Figure 3.9: Représentation d'un couplage entre deux *model-agents* dans AA4MM [Cam+12]

3.2.5 Approches à base d'agents de description des modèles multi-niveaux

Un modèle multi-niveaux a pour objectif de décrire les différents points de vue d'un système [Mor13]. Toujours selon [Mor13] ces approches peuvent servir pour coupler des modèles hétérogènes (modèles ayant des dynamiques et représentations différentes). On peut citer quelques approches telles que:

Padawan (Pattern for Accurate Design of Agent Worlds in Agent Nests) [PMK10] propose une approche générique pour décrire un SMA à environnement multiple. Elle se base sur deux principes: les agents peuvent encapsuler des environnements et les agents peuvent être dans plusieurs environnements en même temps. Cette approche prend en compte les différentes contraintes de couplage (encapsulation et hétérogénéité d'échelles temporelles et spatiales).

IRM4ML[MVD10] (Influence Reaction Model For Multi-level Agent-Based Models) est un framework basé sur IRM4S (Influence Reaction Model For Agent-Based Simulation) [Mic07]. L'idée est de définir un cadre de simulation en multi-niveaux en appliquant le principe d'influence réaction [FM96] qui sépare l'état mental des agents, c'est dire ce qu'ils veulent faire, et ce qui se passe dans l'environnement. IRM4ML propose des solutions conceptuelles à la problématique d'évolution des agents dans un environnement multi-niveaux. Dans [MK17] les au-

teurs détaillent l'exécution dans le temps, des agents dans une organisation multi-niveaux afin d'établir les bases d'une plateforme de modélisation des modèles multi-niveaux à base d'agents appelé SIMILAR.

3.3 Couplage spatialisé

Dans cette section, nous développons quelques approches de couplage spatialisé, c'est à dire que l'espace est utilisé comme élément central du couplage.

3.3.1 Dynamic Oriented Modeling - Model DS

Dynamic Oriented Modeling (DOM) [Pay+06a] est une approche de modélisation permettant de coupler différentes dynamiques appelées "Mono-Dynamic Model" (MDM). La dynamique d'un MDM est encapsulée dans un agent, ce dernier aura alors la possibilité d'influencer l'environnement ou il se situe en modifiant les caractéristiques du MDM (attributs du sous-modèle) reliées à la portion de l'environnement. Ces attributs reliés à l'environnement peuvent être utilisés par d'autres MDM ce qui rend possible la communication entre les différents MDM à travers l'environnement (Figure 3.10).

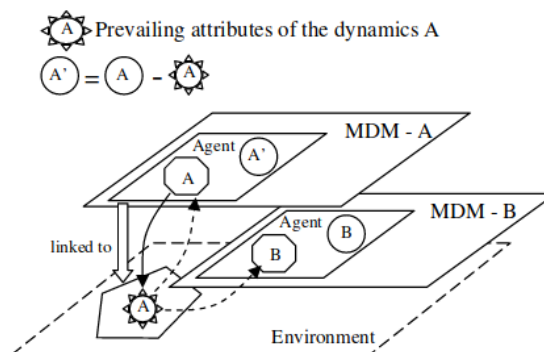


Figure 3.10: Représentation d'une interaction entre MDM dans l'approche DOM

Le modèle DS (Domino-Smat)[Dav+07] est un modèle en urbanisation qui applique les concepts du DOM en couplant une dynamique démographique et une dynamique d'occupation des sols. L'implémentation du modèle a été faite sur GEAMAS [MG98] qui est une plateforme pour la modélisation des modèles spatiaux à base d'agents.

3.3.2 Coupling4Modeling

Coupling4modeling [Has15] est une approche à base d'agents créée pour la modélisation de l'aménagement de territoire de la ville de Metouia en Tunisie. Cette ap-

proche propose un méta-modèle de couplage basée sur l'architecture Agent Group Role Environnement (AGRE) [FMB05], où l'espace est l'élément central de couplage. L'approche propose d'encapsuler les différents modèles par des agents. La gestion des échelles de temps se fait par l'intermédiaire d'un ordonnanceur (ou un orchestrateur, scheduler en anglais) qui implémente un système de calendrier de dates d'exécution sur une base de pas de temps et d'évènements. L'échelle spatiale des modèles est gérée par trois algorithmes : l'algorithme d'exécution des sous-modèles, et deux algorithmes de conversion de données pour convertir les entrées des sous-modèles à leurs échelles et la spatialisation des sorties vers l'espace ou les sous-modèles interviennent (figure 3.11).

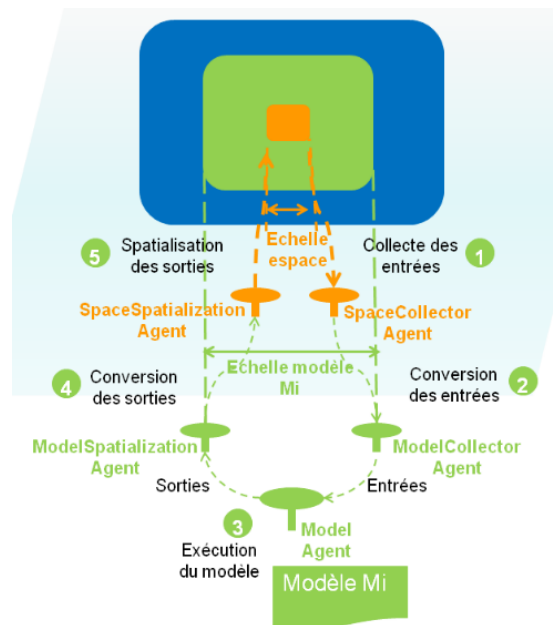


Figure 3.11: Mise à l'échelle des données entre l'espace et les modèles dans couplage4modeling [Has15]

3.3.3 Couplage ad-hoc

Le couplage de modèles est largement utilisé pour la modélisation des systèmes complexes. Les approches “ad hoc” de couplage mentionnées ici, sont faites spécialement pour traiter une certaine problématique dans leurs domaines. Contrairement à ceux que l'on a détaillé précédemment, qui sont davantage dans un cadre générique de modélisation malgré leur appartenance à un certain domaine d'application.

Urbanisation Dans [BB09], les auteurs s'intéressent à l'effet du changement démographique sur les forêts et la disponibilité de l'eau, en couplant trois modèles:

un modèle à base d’agents d’occupation de sol, un modèle individu-centré pour évaluer la dynamique forestière et enfin un modèle spatialisé représentant les cours d’eaux.

[Ban+17] ont développé un modèle de circulation de véhicules en couplant des modèles à base d’agents et des modèles mathématiques représentant la circulation des véhicules à des niveaux différents. Au niveau macro, un modèle mathématique appelé “LWR” qui représente un flux de véhicule, au niveau méso un modèle à base d’agents représentant une concentration de véhicule dans un segment appelé “Underwood” et au niveau micro un modèle représentant individuellement les véhicules “NaSch”.

Écologie du paysage Dans ce domaine particulier, le couplage de modèles est fortement utilisé. L’espace est principalement décrit avec des cartes (SIG) et des données spatialisées. On y étudie l’occupation du sol, l’écoulement des fleuves, les dynamiques forestières, etc. Parmi ces modèles, on peut citer [CEH96] qui est un modèle décrivant l’impact du captage d’eau souterraine sur les cultures reliant des processus avec des données spatialisées. Dans [MLG18], les auteurs décrivent la dispersion d’une population de rongeur sur une carte, en utilisant une équation mathématique pour suivre l’évolution des rongeurs et un modèle à base d’agents pour décrire leur mouvement migratoire au niveau individuel.

Écologie marine L’article [DRP03] propose une technique de transfert d’échelle en couplant un modèle à base d’équations (les phytoplanctons) pour représenter le niveau de la population et un modèle à base d’agents (Copepod, un crustacé) pour représenter le niveau des individus. Les auteurs suggèrent que dans ce couplage hétérogène, l’effet d’émergence du modèle à base d’agents est utilisé pour calibrer le modèle à base d’équations [Mor13].

Épidémiologie De nombreux modèles ont également été développés en épidémiologie pour modéliser la propagation des virus, notamment en couplant le modèle à base d’équations SIR et un (ou des) modèle(s) à base d’agents. C’est le cas du framework SWITCH [Huy+17] ou du modèle MicMac [Ban+16]. Durant la pandémie de la Covid-19, ces modèles ont été grandement sollicités afin de prévoir la propagation du virus ou l’effet de l’application des gestes barrières [Dro+20].

3.4 Plateformes de couplage

Dans cette section, nous abordons les outils permettant de faire du couplage de modèles.

3.4.1 Gama-Comodel

GAMA (GIS and Agent-Based Modeling Architecture) [Tai+19a] est une plateforme de modélisation et de simulation de modèles à base d’agents. La particularité de GAMA est la capacité à manipuler les données spatiales grâce à une large bibliothèque d’opérateurs². GAMA utilise un langage dédié appelé gaml (gama modeling language), facilitant la construction des modèles pour les non initiés à la programmation.

Gama utilise une méthode appelée comodel [Huy17] pour relier les modèles conçus sur la plateforme. Un modèle gaml peut être appelé de la même façon que l’on importe une librairie dans les langages orientés objets.

Comme les modèles à base d’agents sont une approche distribuée, il est aussi possible d’encapsuler des sous-modèles individuellement par des agents et de les faire communiquer sans l’utilisation de comodel.

3.4.2 OpenMI

OpenMI [GGW07] est une plateforme créée pour le couplage de modèles dans le domaine de l’hydrologie. La plateforme s’appuie sur un concept simple de “request and reply”³.

OpenMI propose une interface utilisateur permettant de relier les modèles grâce à des “flèches”. Les modèles sont considérés comme des entités fournissant des données pour d’autres modèles en s’exécutant. Les requêtes sont émises par les modèles en appelant une fonction “get()”, qui récupère les informations reconnaissables par les modèles fournissant l’information requise.

La date d’exécution des modèles se fait au niveau de chaque modèle. Les dates d’exécution peuvent être définies par des pas de temps.

L’espace est défini par une grille et une interface est utilisée afin de définir une résolution standard pour les différentes représentations spatiales des modèles.

3.4.3 VLE-RECORDS

Virtual Laboratory Environment (VLE)[QDR09] est un framework développé dans l’optique de combiner des modèles existants issus de différentes origines, grâce à l’implémentation d’un simulateur basé sur DEVS. RECORD [Ber+09] est une plateforme développée grâce au framework VLE pour la modélisation des systèmes de cultures.

Les interactions et les envois d’évènement aux modèles se font grâce au formalisme DEVS (grâce aux entrées et sorties). Les auteurs proposent des extensions sur lesquels il est possible de:

²fonction prédéfinie dans gama

³request and reply : un modèle demande et attend des données provenant d’un autre modèle, de la même manière que les applications émettant des requêtes à une base de données puis attend les réponses venant de ce dernier

- modifier dynamiquement les interactions entre les modèles via l’extension DS-DEVS [Bar97];
- choisir parmi un ensemble de formalismes dédiés à la prise de décision;
- définir des automates à états finis;
- résoudre des équations différentielles à l’aide des méthodes Runge-Kutta ou Euler.

3.4.4 NetLogo-Levelscape

Netlogo [TW04] est une plateforme dédiée à la modélisation à base d’agents. La plateforme utilise un langage dédié appelé “logo”, qui est un langage de haut niveau comme celui de la plateforme GAMA. Une extension appelée Levelscape a été développée par [Hjo+20] et permet de construire des modèles à base d’agents multi-niveaux (ML-ABM). Comme l’indique [Mor13] dans sa revue, l’une des utilités des ML-ABM est de permettre le couplage de modèles. Levelscape permet alors d’appliquer ce type de modélisation dans NetLogo. [Hjo+20] présente d’ailleurs un exemple de couplage entre un modèle de proie-prédateur et un modèle de changement de climat. L’approche permet de relier des modèles à des échelles temporelles et spatiales différentes. Cependant, elle ne propose pas d’architecture dédiée comme des ordonnanceurs (ou orchestrateur) [Fia01]; [Cam+12]; [Has15] pour concilier les différentes échelles temporelles.

3.4.5 Model Coupling Toolkit (MCT)

MCT [LJO05] est une plateforme de multi-modélisation utilisée en océanographie, qui permet de coupler des modèles en les exécutant de façon indépendante (en parallèle) et échanger des données à travers un “coupler”. Chaque modèle utilise une bibliothèque de fonction appelée Message Passing Interface (MPI) qui est reliée au “coupler” afin de communiquer les données aux autres modèles. Grâce aux exécutions en parallèle des modèles et les conversions de données, MCT est un outil assez complet pour le couplage de modèles. Cependant, cette plateforme est implémentée en fortran, qui est un langage de programmation considéré comme obsolète [Huy17].

3.5 Synthèse

On reproche souvent aux approches de couplage de processus tel que DEVS, FMI, HLA ou AA4MM le manque de description sur l’aspect spatial des modèles [Has15]. Cependant, ces approches ont fait leur preuve dans leurs domaines d’application respective. En particulier l’approche DEVS et ses variantes (VLE

Records, CELL-DEVS, etc.) qui sont largement utilisées dans le monde de la modélisation et simulation.

Les plateformes de couplage offrent une possibilité d'extension afin d'améliorer les fonctionnalités déjà disponibles, mais requiert une certaine familiarité avec les langages de programmation orientés objets pour ce faire.

Les approches de couplage spatialisées à base d'agents, qui sont des approches relativement jeunes, ne prennent pas en compte la compétition de ressources, qui est un aspect important dans la modélisation des processus sol [Din+22b]. Cependant, ces approches possèdent des algorithmes utiles quant à la gestion des différentes échelles des modèles [Has15]. Nous allons explorer dans le prochain chapitre ce qui est de la coordination dans les approches de couplage de modèles à base d'agents, en passant par les différents types de coordination existants dans la théorie des systèmes multi-agents.

Chapitre 4

Coordination de modèles

Contenu du chapitre

4.1	Coordination par organisation	54
4.1.1	Coordination réactive	54
4.1.2	Coordination hiérarchique	54
4.2	Coordination par réglementation	55
4.3	Coordination par argumentation et négociation . . .	55
4.4	Synthèse : application des approches de coordination en couplage de modèles	56

Dans le cadre de cette thèse, nous nous intéressons particulièrement aux approches de couplage à base d’agents, en raison de leur potentiel à exprimer le phénomène d’émergence et la reproduction plus réaliste des phénomènes environnementaux. Le point commun entre ces approches, c’est qu’elles utilisent tous un ordonnanceur afin de synchroniser leurs modèles. Cependant, ce type de synchronisation n’est pas suffisante dès lors que l’on couple des modèles à travers l’espace. En effet, les actions simultanées des modèles ne sont pas gérées par l’ordonnanceur. L’approche `coupling4modeling` [Has15] par exemple, utilise une technique de priorisation afin d’exécuter les modèles utilisant les mêmes ressources de l’espace de manière simultanée. La théorie des systèmes multi-agents offre cependant plusieurs outils théoriques pour gérer ce type comportement.

“La notion d’interaction est cruciale aux SMA dû au fait que l’état de l’environnement à un certain point du temps résulte des opérations effectuées par les agents sur leur monde” [Fer95]. En effet, lorsqu’on crée un modèle ou plusieurs entités évoluent dans un même milieu, il est nécessaire de synchroniser leurs actions pour aboutir à une évolution cohérente de l’environnement. Une mauvaise synchronisation peut aboutir à des conflits dans un SMA [TB17]. Un conflit est défini par [Tes+02] comme un état sous lequel les actions des agents sont limitées ou réduites. Ainsi, ils peuvent se disputer une ressource, on parle de *conflits*

physiques, ou peuvent avoir différents points de vue dans une situation donnée, on parle dans ce cas de *conflit de connaissance*.

Nous nous intéressons aux approches de résolution de conflits dans l'optique d'améliorer les approches de couplage à base d'agents au niveau des interactions entre les sous-modèles.

4.1 Coordination par organisation

4.1.1 Coordination réactive

La coordination réactive s'appuie sur le fait que les agents se coordonnent par le biais de leur environnement de sorte que l'action d'un agent ne soit pas en contradiction avec d'autres.

Influence-Reaction [FM96] consiste à gérer les actions simultanées des agents grâce à l'état de leur environnement. Les auteurs proposent alors de diviser la phase d'action des agents en deux :

1. La phase d'influence qui se traduit comme l'intention des agents à modifier l'état de l'environnement.
2. La phase de réaction, ou la réponse de l'environnement par rapport aux influences.

De cette manière, quand plusieurs agents modifient l'état de l'environnement simultanément, l'état de ce dernier n'est calculé qu'après avoir décidé quelles influences ont été appliquées. Une fois, l'état du système calculé, les autres agents ajustent leurs prochaines actions par rapport à ce nouvel état.

L'approche IRM4MLS et la plateforme SIMILAR [Mor13] sont des applications de ce principe dans un environnement multi-niveaux.

On trouve aussi une forme de coordination réactive dans les approches comme DEVS [ZV93] et ses dérivées [QDR09]; [Ber+09], où les entités composant les modèles sont à l'écoute d'évènements pour calculer leurs états.

4.1.2 Coordination hiérarchique

Dans le cas où une entité gère l'exécution des agents, on parle alors de coordination hiérarchique. L'exemple la plus proche serait les orchestrateurs dans les approches de couplages à base d'agents [Fia01]; [SCC10]; [Has15]. Un planificateur appelé orchestrateur gère l'exécution des sous-modèles de sorte que ces derniers s'exécutent de manière cohérente en suivant des dates d'exécutions comme dans un calendrier.

Dans [Als+07] les auteurs proposent une architecture hiérarchique où les agents, membres d'un groupe, sont fédérés par des agents coordinateurs afin d'accéder à

des ressources ou à communiquer des informations. Cette approche permet de faire communiquer les demandes des agents entre différents groupes.

Les approches de couplage comme HLA [Awa+13] ou FMI [Blo12] où les interactions entre simulateurs sont gérées par un bus (respectivement RTI et FMU) peuvent aussi être considérés comme une coordination hiérarchique.

4.2 Coordination par réglementation

L'utilisation de règles ou lois régissant la coordination des agents ou architectures normatives [BTV06] est une forme de coordination plus avancée que celles développées précédemment. Une norme est définie par [BTV06] comme étant un ensemble de principes régissant un groupe d'individus pour qu'ils adoptent un comportement "acceptable".

L'architecture BOID [Bro+01]; [BDT01] ou Belief-Obligation-Intention-Desire, est une extension de l'architecture BDI ou Obligation désigne un ensemble de contraintes morales pouvant interférer sur le comportement d'un agent.

Normative KGP (Knowledge Goal Plan) [SST06] est un framework permettant aux agents de résoudre les conflits entre les objectifs personnels et sociaux. Normative KGP prend en compte les obligations ainsi que les interdictions¹.

Pour plus de détails sur les architectures normatives existantes, [Neu10] propose une classification d'un ensemble conséquent d'approches.

4.3 Coordination par argumentation et négociation

Les objectifs d'un agent peuvent entraver ceux des autres. Dans ce cas, les agents partagent des informations sur leurs tâches actuelles, afin de trouver un arrangement. Les agents doivent donc avoir un mécanisme de prise de décision pour argumenter avec d'autres agents pour négocier et ainsi résoudre les problèmes de conflits.

Ce type de coordination a pour objectif de communiquer des informations aux autres agents, afin de trouver un consensus ou un compromis si une situation de conflit survient [Oss08]. Les informations sont alors représentées de différentes manières pour faciliter la communication ou les prises de décisions. Parmi ces représentations, on retrouve, la sémantique [AS08], qui permet d'unifier la représentation des informations échangées par les agents. On retrouve aussi les mesures de confiances [CF98], une approche permettant aux agents de prendre des risques afin d'exécuter ou déléguer une tâche.

Puis l'argumentation, qui se définit comme étant l'initiation d'un processus de

¹ensemble d'action jugée prosaïque.

négociation pour aboutir ou non à un consensus. Des outils comme proposés dans [ADM07] proposent une représentation générique des processus d'argumentation.

4.4 Synthèse : application des approches de coordination en couplage de modèles

La nature hétérogène des sous-modèles utilisés dans les approches de couplage rend difficile l'application des approches de coordination comme la négociation, l'argumentation ou encore les réglementations. En effet, les sous-modèles proviennent souvent différentes disciplines et ne prennent pas forcément en compte certains aspects ontologiques comme les désirs, les croyances, les obligations ou les degrés de confiance. Cependant, il est tout à fait possible d'utiliser des agents négociateurs comme dans [Als+07] pour négocier à la place des sous-modèles.

La plupart des approches de couplage utilisent la coordination réactive ou hiérarchique. L'utilisation des méthodes d'exécution pessimistes² comme dans OSIRIS [Fia01] ou les orchestrateurs (à base d'évènements et/ou pas de temps) comme dans [ZV93]; [SCC10]; [Has+12]; [MK17] prouvent qu'un degré considérable de coordinations de modèles ont été faits. Cependant, en couplage de modèles spatialisés, les sous-modèles utilisent les ressources de l'espace. Dans le cas où ces sous-modèles utilisent ces ressources de manière simultanée, on se heurte à une problématique de compétition de ressources que les approches de couplage spatialisé de modèles existant ne peuvent pas gérer [Din+22b]. Cela ajoute une nouvelle contrainte aux approches de couplages spatialisés de modèles.

²les modèles ne s'exécutent que lorsqu'ils vérifient certaines conditions

Partie II

Contribution

Chapitre 5

Positionnement

Contenu du chapitre

5.1	Le paradigme agent pour encapsuler les modèles . . .	60
5.2	L'espace comme élément central du couplage	60
5.3	Un orchestrateur pour gérer les différentes tempo- ralités	61
5.4	Coordination des sous-modèles	62
5.4.1	Influence Reaction: Principe	63
5.4.2	Coordination multi-niveaux	64

Nous développons dans ce chapitre, notre positionnement par rapport à la littérature sur notre objectif principal :

Construire une approche de couplage de modèles à base d'agents centrée sur l'espace pour modéliser le fonctionnement écosystémique du sol.

Notre stratégie de couplage consiste à utiliser le paradigme agent pour encapsuler les différents processus du sol, puis coupler ces processus à travers l'espace (le sol) en leur permettant d'utiliser les ressources du sol. Le comportement de ces processus dépend de l'accessibilité aux ressources. Cependant les ressources du sol sont limitées. De ce fait, il est possible que des conflits de ressources surviennent lorsque celles-ci sont utilisées de manière simultanée alors que la quantité présente dans le sol est insuffisante pour subvenir aux demandes des processus (sous-modèles). Notre contribution réside dans la conception d'une approche permettant de résoudre ces conflits en utilisant l'espace comme un outil de calcul permettant de distribuer ces ressources aux sous-modèles. Pour cela, nous nous appuyons sur une théorie en système multi-agents appelée "influence-reaction" [FM96] afin de déterminer la manière dont les ressources sont utilisées par les sous-modèles en cas de conflit.

5.1 Le paradigme agent pour encapsuler les modèles

La modélisation des SMA est une pratique purement bottom-up, c’est à dire qu’on crée des agents interagissant dans un environnement puis on observe l’état du système résultant de leurs interactions (émergence). Les SMA sont donc une architecture distribuée où des entités indépendantes interagissent pour créer un phénomène d’émergence. Dans la même optique, le couplage de modèles regroupe plusieurs modèles existants et les met en interaction pour comprendre et étudier un phénomène. Il est alors naturel d’utiliser des agents pour “envelopper” ou “encapsuler” les sous-modèles et de les mettre en interaction dans un environnement pour décrire un phénomène afin de mieux représenter la réalité.

En utilisant ce paradigme, on acquiert plusieurs avantages :

- Une démarche de modélisation modulaire et distribuée : L’encapsulation des sous-modèles par des agents permet le remplacement d’un sous-modèle par un autre (modularité), on peut alors mesurer la performance de ce dernier sur le modèle-couplé ou la modifier à volonté.
- L’utilisation de puissants outils de modélisation dédiés au développement des SMA comme Netlogo ou Gama, qui fournissent un langage dédié plus accessible aux non-initiés à la programmation.
- L’accès à une large palette d’outil théorique [FM96]; [BTV06]; [MVD10] pour améliorer les approches de couplage existant et l’aspect interaction des modèles couplés qui en résulte.

5.2 L’espace comme élément central du couplage

Dans la réalité, l’espace est un élément partagé par les processus composant les phénomènes environnementaux. Face à la complexité croissante des nouvelles problématiques environnementales, l’espace devient un facteur important sur l’étude de ces phénomènes car les observations varient d’un endroit à un autre [Has15]. Dans les travaux comme [Pay+06a]; [Dav+07]; [Has15]; [Mar16], l’espace est l’élément central où les diverses interactions entre modèles se produisent. [Pay+06a]; [Dav+07] argumentent l’utilisation de l’espace en tant que facteur de couplage par le besoin d’un moyen d’intégration et de réutilisabilité des sous-modèles. [Has+12]; [Has15] complètent cet argument en introduisant des techniques de mise à l’échelle des sous-modèles afin d’observer l’effet des dynamiques à travers les différents niveaux de l’espace.

Ces travaux démontrent l’importance de l’espace en couplage de modèles. Cependant, ils ont été développés pour répondre à des problématiques d’urbanisation et partagent avec le sol cet aspect où les dynamiques opérantes sont spatialisées. De ce fait, nous utilisons l’espace comme facteur de couplage. Les sous-modèles communiqueront donc de manière indirecte en utilisant les ressources de

l'espace. C'est à dire que les sous-modèles modifient les ressources de l'espace (et modifient l'état de l'espace par la même occasion) et agissent en fonction de la disponibilité de ces ressources.

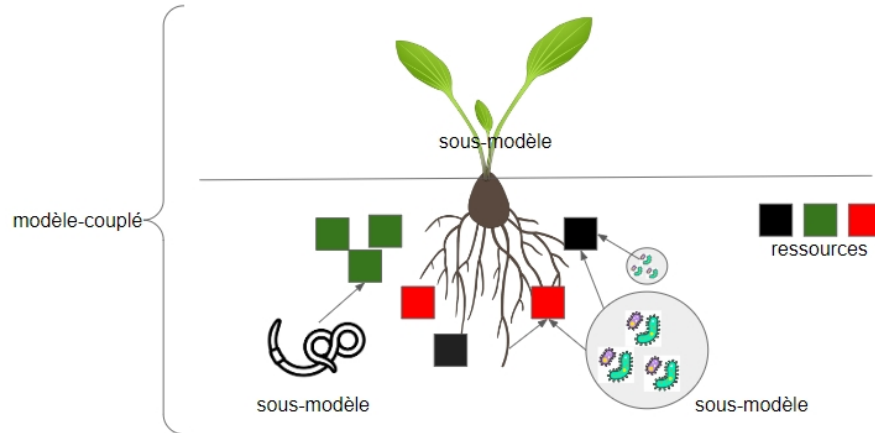


Figure 5.1: Les ressources de l'espace sont utilisées par les sous-modèles

5.3 Un orchestrateur pour gérer les différentes temporalités

Les sous-modèles et le modèle-couplé peuvent avoir des temporalités différentes. Bien que les sous-modèles soient indépendants, pour une bonne cohérence dans la simulation, il est nécessaire de synchroniser l'exécution des sous-modèles et du modèle-couplé.

Dans les plateformes de simulation telle que Gama [Tai+19a] ou NetLogo [TW04], une simulation avance d'un état à un autre en terme de cycle (ou tick dans NetLogo). Cependant en couplage de modèles, avant de passer d'un cycle à un autre, le modèle de simulation passe par plusieurs changements d'états dus à l'exécution des sous-modèles. La durée d'un cycle de simulation équivaut au pas de temps du modèle-couplé. Nous nous inspirons de la façon dont DEVS [ZV93] gère les temporalités de ses sous-modèles et d'un algorithme proposé par [Has15] où les sous-modèles sont exécutés de façon discrète tout au long de l'exécution du modèle-couplé comme dans la figure 5.2. Nous utilisons donc une représentation hybride du temps où le modèle-couplé avance à un pas de temps δ_{tm} , les sous-modèles avancent chacun en suivant leur pas de temps respectif δ_{ti} et l'état du système S_n est calculé à chaque changement d'état des sous-modèles et du modèle-couplé (figure 5.2).

L'orchestrateur que nous utiliserons s'inspirera de [ZV93]; [Has15]. Cependant, exécuter les modèles n'est pas son seul rôle, car il notifie également le système des modèles qui s'exécutent simultanément.

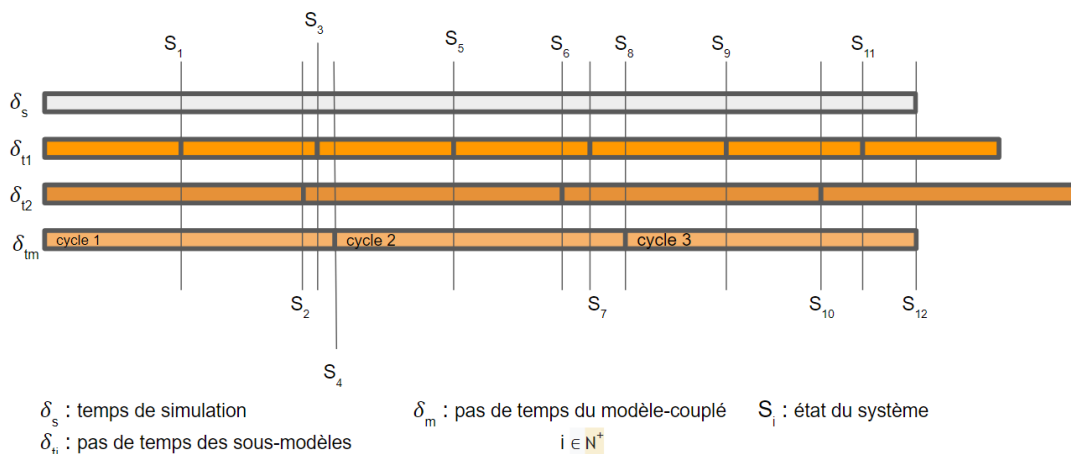


Figure 5.2: Modèle de temporalité hybride où l'état du système est calculé lorsque les sous-modèles ou le modèle-couplé s'exécutent

5.4 Coordination des sous-modèles

Comme les sous-modèles réagissent par rapport à la disponibilité des ressources de l'espace, il se peut que plusieurs sous-modèles les utilisent simultanément. Cela crée une compétition de ressources au niveau des sous-modèles. Par conséquent, plusieurs contraintes doivent être prises en compte par notre approche pour une bonne cohérence du modèle-couplé:

- Les ressources limitées : les ressources dans le sol sont limitées, certaines résultent d'un long processus de transformation par le biote du sol, consommés par certains êtres vivants à une grande densité de population ou se font rares en raison des différentes pratiques agricoles.
- L'utilisation simultanée des ressources : les sous-modèles peuvent avoir des temporalités différentes. Cependant il peut y avoir différentes instances du même sous-modèle utilisant la même ressource ou des sous-modèles de différentes temporalités utilisant la même ressource simultanément à un certain point du temps.
- Les conflits de ressources : en raison des deux points cités précédemment, les sous-modèles peuvent alors être en conflit de ressources lorsque plusieurs sous-modèles veulent accéder à une ressource limitée de manière simultanée. En conséquence, cela peut affecter le comportement des autres sous-modèles (car ils agissent en fonction de la disponibilité de la ressource) et ainsi perturber voire même fausser les observations et les résultats de simulations.

Dans son ouvrage sur les systèmes multi-agents, Jacques Ferber [Fer95] stipule que l'interaction est le plus grand atout des SMA mais aussi la source majeur de

ses maux. L'amélioration de cet aspect par des techniques de synchronisation conduit alors à un système plus cohérent et précis. De ce fait, nous allons donc utiliser une approche de coordination d'action afin de gérer et traiter ces conflits. Ce projet de thèse apporte donc une contribution dans le domaine du couplage de modèles.

5.4.1 Influence Reaction: Principe

Influence Reaction est une modalité d'action proposée par [FM96], dont l'objectif est de répondre à la problématique de la simultanéité des actions des agents situés.

L'état d'un environnement est modifié par le comportement des agents. Cependant, tous les comportements produits ne sont pas appliqués à l'environnement pour modifier son état [Mic07]. [FM96] proposent donc de diviser la phase d'action d'un agent en séparant ce qu'il fait (phase d'influence) et ce qui se passe dans son environnement (phase de réaction). En couplage spatialisé de modèles, l'influence se traduit donc comme l'intention d'un sous-modèle à modifier l'état de l'espace dans lequel il évolue. L'évolution de cet espace (passer d'un état à un autre) se traduit par la phase de réaction, où l'espace détermine quelles influences sont à l'origine du changement d'état.

Le changement de l'état d'un espace est défini dans le contexte de cette thèse comme la modification d'une ressource de l'espace.

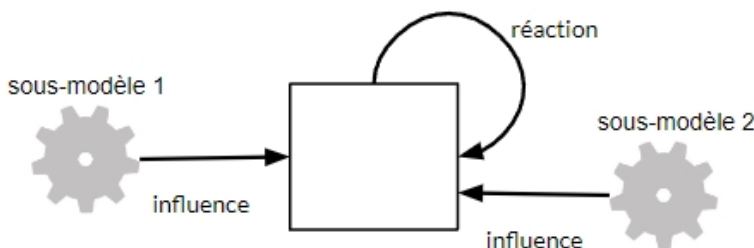


Figure 5.3: Illustration de Influence Réaction sur deux sous-modèles produisant des influences sur l'espace où ils évoluent.

Pour gérer les influences et les réactions, nous introduisons alors le concept de "Coordination de modèle", dont la fonction primaire est de définir la phase de réaction en situation de conflit. Pour cela, deux phases de traitements sont considérées à chaque production d'influences:

- la détection des conflits, qui se fait en association avec l'orchestrateur. Les sous-modèles s'exécutent selon leur pas de temps respectif, de ce fait il se peut que les instances du même sous-modèle sont induites en conflit ou quand différents sous-modèles s'exécutent à la même date et utilisent la même ressource.

- le traitement des conflits se fait de manière automatique pour déterminer la phase de réaction. Pour se faire, des stratégies doivent être définies pour chaque ressource de l'espace afin de déterminer comment les influences sont appliquées.

La coordination de modèles mentionnée ici n'est encore que la description du mécanisme au niveau d'une particule de l'espace. Si l'on s'intéresse à la coordination des sous-modèles dans une organisation multi-niveaux, on peut s'inspirer des approches de [Mic07]; [MVD10] sur une formalisation du principe de l'influence et réaction pour la simulation des SMA en multi-niveaux, afin de décrire le fonctionnement de la coordination de modèles dans une telle configuration.

5.4.2 Coordination multi-niveaux

Dans [Mor13], un niveau est défini comme étant un point de vue d'un système. Un modèle multi-niveaux sert donc à décrire différents points de vue d'un système. Dans le sol, le caractère multi-niveaux s'exprime par l'organisation de l'espace en plusieurs "couches" de différentes tailles comme dans le modèle PSF ou APSF [PBR00]; [Mar+08].

Multi-niveaux et multi-échelles sont cependant des termes parfois confondus. L'échelle désigne une résolution ou une étendue (souvent temporelle, spatiale ou de grandeur) faisant partie de l'aspect organisationnel décrit par un niveau. Un niveau décrit l'organisation et les interactions sur un point de vue d'un système dans le temps et dans l'espace. Pour donner une illustration concrète, l'APSF est un SMA multi-niveaux et multi-échelles qui décrit la structure du sol, l'arbre de canevas indique l'aspect organisationnel du sol tout en désignant quelles particules contiennent des particules plus petites (micro-agrégat, cavités de petites tailles...). En urbanisation, une ville, une région ou encore une commune décrivent un niveau, tandis que dire qu'une ville est plus grande qu'une autre, représente une étendue, cela décrit une échelle.

Afin de visualiser un point de vue particulier, il est indispensable de passer par des techniques d'agrégation ou de désagrégation. L'agrégation consiste à passer d'un niveau inférieur à un niveau supérieur (micro - > meso - > macro), tandis que la désagrégation décrit le processus inverse (macro - > meso - > micro) (figure 5.4).

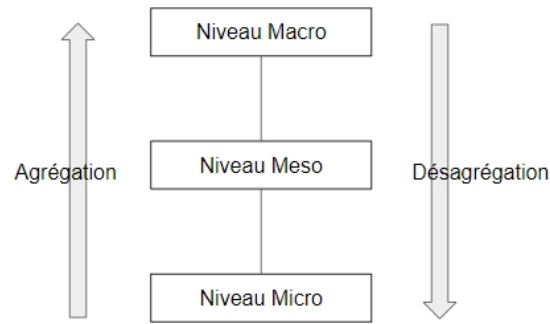


Figure 5.4: Passage d'un niveau à un autre

Il est alors possible de s'inspirer des travaux de [Mic07] où les phases d'influences sont émises au niveau micro et la phase de réaction est déterminée par la combinaison des influences au niveau macro, comme l'illustre la figure 5.5.

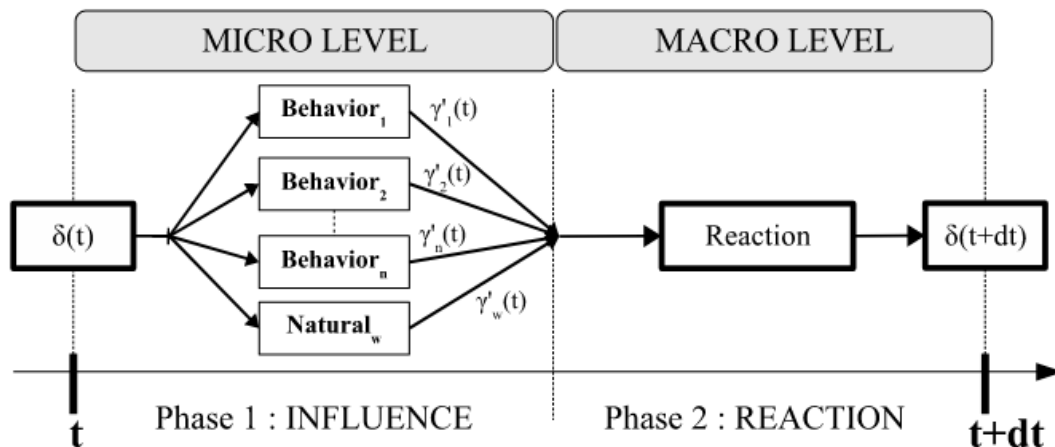


Figure 5.5: Dynamique du système dans IRM4S [Mic07]: les agents et l'environnement produisent des influences (respectivement $\gamma'_n(t)$ et $\gamma'_w(t)$) au niveau micro, puis la réaction est déterminée au niveau macro pour calculer l'état du système

En s'inspirant de ce modèle, déterminer la phase de réaction implique d'agréger les influences produites par les sous-modèles sur l'espace. Déterminer la phase de réaction sur une portion d'espace (contenant des sous-espaces) est donc une technique d'agrégation. Le processus inverse, la désagrégation, consiste alors à communiquer l'état du niveau supérieur grâce à une fonction permettant de distribuer l'état des ressources du niveau supérieur sur les niveaux inférieurs. De cette façon, il est possible de visualiser les changements à travers les différents niveaux du sol.

Chapitre 6

Architecture et implémentation de l'approche

Contenu du chapitre

6.1	Architecture de l'approche	68
6.1.1	Macro-Model	68
6.1.2	Micro-Model	68
6.1.3	Space	69
6.1.4	Modification List	69
6.1.5	Scheduler	69
6.1.6	Coordination Unit	69
6.1.7	Strategy	69
6.1.8	Scale Transfert	70
6.2	Implémentation	70
6.2.1	La plateforme Gama	70
6.2.2	Coupler les sous-modèles à travers l'espace	71
6.2.3	Gérer la temporalité des sous-modèles	71
6.2.4	Coordination des sous-modèles	73
6.2.5	Coordination dans un espace multi-niveaux et multi-échelle	76

Dans ce chapitre, nous concrétisons les positionnements cités dans la section 5 sous forme de diagramme UML ainsi que les algorithmes utilisés pour l'implémentation de notre approche de couplage et de la coordination des sous-modèles dans la plateforme gama.

des modifications sur l'espace, les sous-modèles peuvent envoyer des “influences” sur l'espace pour modifier une ressource, conformément au principe d'influence-reaction [FM96].

6.1.3 Space

L'espace est un environnement multi-niveaux et multi-échelles composé de “Sub-space” (des portions d'espace plus petites), elle contient les ressources utilisées par les sous-modèles et joue le rôle de facteur de couplage [Has+12]. Facteur de couplage signifie ici que les sous-modèles (micro-model dans le diagramme) communiquent à travers l'espace en agissant selon la disponibilité de la ressource.

6.1.4 Modification List

Cette classe se réfère à une liste regroupant toutes les influences émises par les sous-modèles sur la portion d'espace où ils se situent. Chaque **Sub-Space** possède une **Modification List** permettant de suivre et manipuler les influences envoyées par les sous-modèles.

6.1.5 Scheduler

Communément appelé Ordonnanceur ou Orchestrateur, la classe **Scheduler** a pour rôle de gérer les temps d'exécution de l'ensemble du système en se basant sur les pas de temps de chaque modèle (le modèle-couplé et les sous-modèles) et de l'espace.

6.1.6 Coordination Unit

L'unité de coordination permet de gérer les influences envoyées par les sous-modèles, cette classe filtre la **Modification List** grâce à des stratégies sélectionnées par l'utilisateur afin de définir quelles influences seront appliquées sur l'espace.

6.1.7 Strategy

La classe stratégie se définit comme étant une interface permettant d'implémenter plusieurs fonctions de coordination assignées à chaque ressource de l'espace pour filtrer la **Modification List**. Ici, nous avons trois types de fonctions de coordination: **Prioritization**, **Competition** et **Cooperation**.

La fonction **Prioritization** permet de filtrer la **Modification List** par rapport à un ordre de priorité assigné aux sous-modèles. L'ordre de priorité définit alors l'ordre d'accès aux ressources.

La fonction **Competition** permet de définir quel sous-modèle accapare la ressource disponible à l'instant t puis laisse le reste (s'il en a) aux autres sous-modèles.

la fonction `Cooperation` permet de distribuer la ressource disponible à l'instant t , en fonction du besoin de chaque sous-modèle.

6.1.8 Scale Transfert

Cette classe sert d'interface pour utiliser des méthodes d'agrégation ou de désagrégation afin d'observer l'évolution de l'espace à différent niveau. L'agrégation consiste à collecter l'ensemble des états des particules d'espaces au niveau inférieur pour déterminer l'état du niveau supérieur. Le processus inverse, la désagrégation, consiste à distribuer les ressources du niveau supérieur sur les particules d'espace au niveau inférieur. L'implémentation de ces méthodes est à la charge du modélisateur, car cela dépend fortement de la configuration ou l'implémentation de l'espace.

6.2 Implémentation

6.2.1 La plateforme Gama

Gama est une plateforme de modélisation et simulation dédiée aux systèmes multi-agents [Tai+19a]. Elle utilise un langage dédié appelé `gaml` (gama modeling language) dont la syntaxe s'inspire des langages orientés objets, tout en gardant une simplicité d'utilisation pour les utilisateurs non initiés à la programmation. La particularité de Gama réside dans la capacité à construire des modèles spatialisés grâce à une bibliothèque d'opérateurs pour manipuler les données spatiales.

Le méta-modèle de gama (figure 6.2), permet de définir un modèle composé d'une classe générique appelé `Species`, qui offre la possibilité de définir l'environnement, les agents et les expériences (plan de simulation).

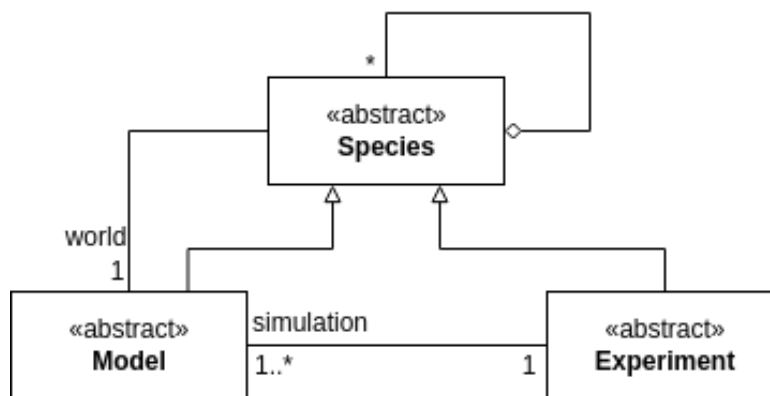


Figure 6.2: Méta-modèle de la plateforme Gama [Tai+19a]

La classe `Species` permet de définir les comportements, les attributs et les actions possibles des agents, de la même manière que la déclaration d'une classe dans les langages orientés objets.

Afin d’implémenter notre approche, nous avons utilisé une autre notion appelée `Skills`. `Skills` est une interface contenant un ensemble d’attributs et de fonctions permettant d’étendre les propriétés et méthodes (actions et comportements) des espèces auxquels il est rattaché. Grâce à cette notion du langage `gaml`, il a été possible d’implémenter les solutions que nous avançons dans notre approche.

6.2.2 Coupler les sous-modèles à travers l’espace

Afin d’associer un sous-modèle à un espace, nous avons développé dans notre extension, un skill appelé `coupled_model`. Ce skill permet, via la propriété “location” (qui définit la position d’un agent dans l’environnement), de placer un sous-modèle à une position définie par l’utilisateur et de définir les “influences” du sous-modèle en invoquant une fonction de la forme d’une requête, tout en indiquant la ressource et la quantité à modifier. De cette manière, en s’exécutant, l’influence du sous-modèle est enregistrée dans une liste associée à la portion d’espace où se trouve le sous-modèle. Cette procédure est donnée dans le code suivant :

```
species $model_name$ skills:[coupled_model]{
  //name of the current sub-space
  space particle;
  float income;

  init{
    //place un sous-modèle dans une particule
    //de l'espace
    do set_space_particle space:particle;

    //envoie une influence pour modifier
    //une ressource
    do request_modification parameter:"$ressource_name$"
    value:$value$;

    //reçoit la quantité de ressource modifiée
    income <- self.get_income(self, "$ressource_name$");
  }
}
```

6.2.3 Gérer la temporalité des sous-modèles

Les sous-modèles et le modèle-couplé possèdent leurs propres pas de temps. Le pas de temps du modèle-couplé est considéré comme référence pour définir un cycle de simulation. Cependant, l’état du système peut changer plusieurs fois au

cours d'un cycle de simulation en raison de à l'exécution des sous-modèles. Afin de modéliser ce mécanisme, nous avons repris un algorithme de [Has15], dans lequel un orchestrateur exécute de manière discrète plusieurs sous-modèles tout au long de l'exécution du modèle-couplé. Notre orchestrateur se démarque de celui de [Has15] par l'utilisation d'un module de coordination qui permet à chaque envoi d'influences, d'établir des stratégies de gestions d'influences au cas où un conflit serait détecté.

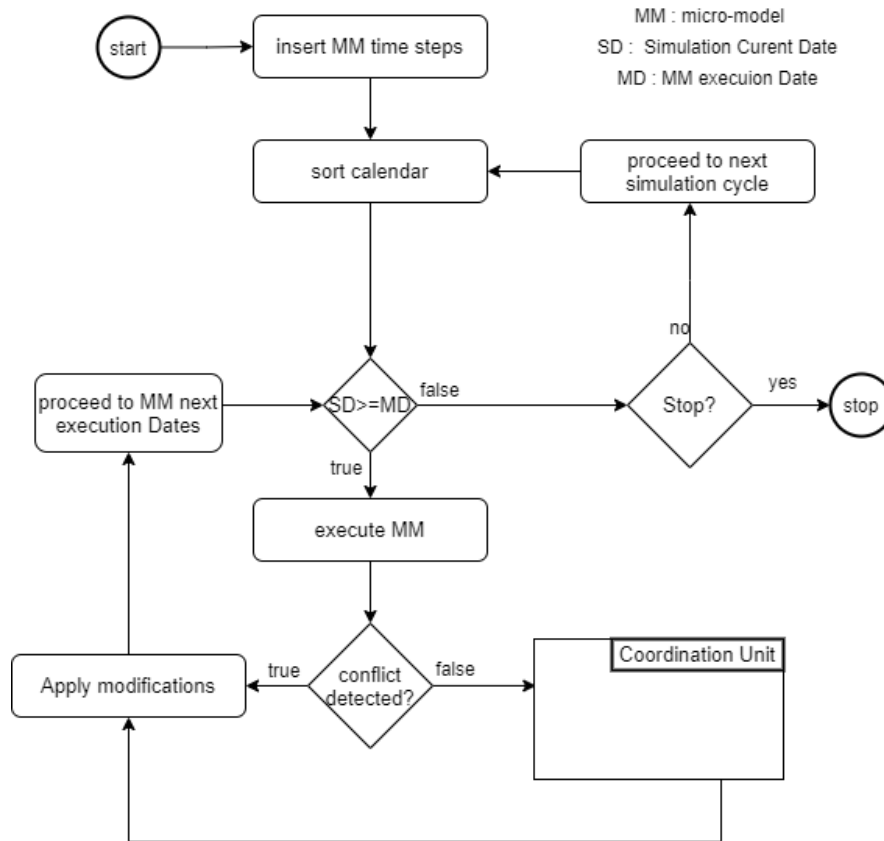


Figure 6.3: Algorithme de synchronisation des exécutions [Din+22b]

1. La première étape consiste à insérer les pas de temps respectifs de chaque sous-modèle dans un calendrier (une liste contenant les dates d'exécution des sous-modèles). La date de simulation (SD) est calculée à partir du pas de temps du modèle-couplé. La formule générale pour l'avancement du temps d'un modèle est $\Delta t'_m = \Delta t_m + \delta t_m$ où $\Delta t'_m$ indique la prochaine date d'exécution d'un modèle, Δt_m la dernière date d'exécution et δt_m le pas de temps du modèle. La date d'exécution du système (SD) et les sous-modèles sont alors calculés et triés dans un ordre croissant.
2. Les sous-modèles vérifiant la condition $SD \geq MD$ sont exécutés, puis les

dates d'exécution des sous-modèles (MD) sont recalculées et re-comparées à SD jusqu'à ce que la condition ne soit plus vérifiée.

3. Quand les sous-modèles sont exécutés, ils envoient les influences à la portion d'espace où ils se situent. Si plusieurs sous-modèles se situant dans la même portion de l'espace s'exécutent simultanément en modifiant la même ressource (détection de conflit), deux scénarios peuvent se produire. Si la ressource supporte toutes les influences, les changements sont opérés sur l'espace. Dans le cas contraire, la liste d'influence passe par un module de coordination ("Coordination Unit") avant d'appliquer les changements sur l'espace.
4. Lorsque la condition $SD \geq MD$ n'est plus vérifiée, on calcule la prochaine date SD puis revient à l'étape 1 en triant le calendrier.

L'implémentation d'un skill appelé "schedule" permet de définir les différentes temporalités des sous-modèles. Ce skill doit être attaché à l'environnement globale de la simulation appelé `global` dans `gama`.

```
global skills:[schedule] {

    float step_macro_model <- 2#days;
    //pas de temps du sous-modèle microbe
    float step_microbe_model <- 4#hours;
    //pas de temps du sous-modèle root
    float step_root_model <- 12#hours;

    //insere les pas de temps des sous-modèles
    do scheduler_insert process:microbe_model
    step:step_microbe_model ;
    do scheduler_insert process:root_model
    step:tep_root_model;

    //fonction qui calcule les dates d'exécutions
    reflex run_scheduler{
        do schedule;
    }
}
```

6.2.4 Coordination des sous-modèles

Le module de coordination a pour fonction principale de définir les influences appliquées à l'espace. La liste d'influence de chaque portion d'espace est examinée

afin de déterminer s'il y a conflit ou non. Dans l'exemple suivant, nous illustrons ce qu'est un conflit.

Exemple : Soient deux sous-modèles M_1 et M_2 , voulant modifier une ressource R dont la quantité est r , d'une valeur respective α_1 et α_2 et à date d'exécution respective δt_1 et δt_2 . Un conflit survient entre M_1 et M_2 lorsque:

$$\delta t_1 = \delta t_2 \text{ et } [r - (\alpha_1 + \alpha_2)] < 0.$$

Dans une forme plus généralisée, un conflit survient lorsqu'un ensemble de sous-modèle M_n s'exécutent à la même date δt ($\delta t = \delta t_0 = \delta t_1 = \dots = \delta t_n$) et que la quantité d'une ressource est insuffisante pour supporter les influences envoyées par les sous-modèles $[r - (\sum_{i=1}^n M_i)] < 0$.

$$\begin{cases} \delta t = \delta t_0 = \delta t_1 = \dots = \delta t_n & n \in \mathbb{N} \text{ et } t \in \mathbb{R}^+ \\ r - \sum_{i=1}^n M_i < 0 & i, n \in \mathbb{N} \text{ et } r \in \mathbb{R}^+ \end{cases} \quad (6.1)$$

Où n est le nombre de sous-modèles exécutés simultanément

Maintenant que le terme conflit est défini, nous détaillons par la suite les stratégies de coordination pour la résolution des conflits.

Le module de coordination a pour objectif de résoudre les conflits dans chaque sous-espace. Il travaille en association avec l'orchestrateur afin de capturer les envois simultanés d'influences. Le processus de résolution de conflits est illustré par la figure 6.4

Le principe du module de coordination est simple: pour chaque sous-espace, les influences envoyées simultanément sont enregistrées dans une liste. Si elle vérifie l'équation (6.1), le module de coordination applique une stratégie (préalablement choisie par le modélisateur) puis notifie les sous-modèles si les influences qu'ils ont envoyées ont été appliquées ou non. Dans le cas contraire (pas de conflit), les sous-modèles opèrent ensuite les changements dans leur sous-espace. L'algorithme se termine en redirigeant les prochaines instructions à l'orchestrateur (i.e calcul des prochaines dates d'exécution et application des changements aux sous-espaces concernés).

Les stratégies mentionnées ci-dessus consistent à filtrer l'accès aux ressources des sous-modèles induits en conflits. Les conflits surviennent lorsque les sous-modèles veulent consommer ou utiliser de la ressource de manière simultanée quand celle-ci ne présentent pas une quantité suffisante pour répondre à ces consommations. Pour se faire, nous avons implémenté trois stratégies de base pour la gestion des conflits:

1. Priorisation : organisation de manière séquentielle de l'accès aux ressources en attribuant une propriété appelée *priority* à chaque sous-modèle. La liste

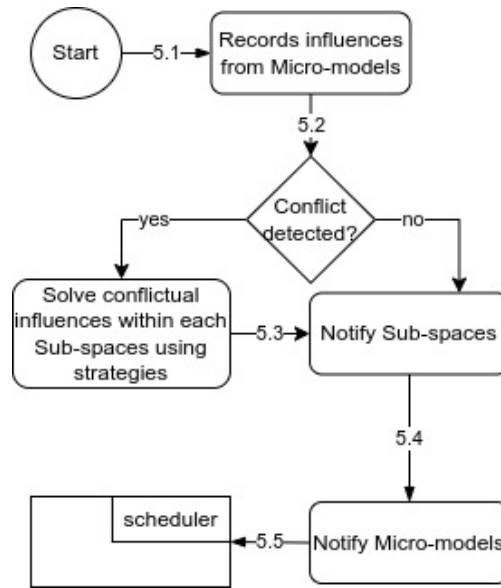


Figure 6.4: Algorithme de résolution de conflit [Din+22b]: les influences envoyées sont évaluées et traitées par des stratégies lorsqu'un conflit est détecté. Dans le cas contraire, les influences sont appliquées aux sous-espaces.

de modification sera ensuite filtrée en suivant un ordre ascendant de *priority*. Le sous-modèle en début de liste consomme alors la ressource puis laisse le reste aux autres sous-modèles ou accapare toute la ressource dans le cas où la quantité actuelle serait inférieure à la demande du sous-modèle.

2. Compétition : la stratégie de compétition permet de sélectionner un sous-modèle de sorte que celui-ci soit le premier à utiliser la ressource. Comme la priorisation, le sous-modèle sélectionné consomme toute la ressource si la quantité actuelle est inférieure à sa demande de consommation, ou laisse le reste aux autres sous-modèles s'il y en a.
3. Coopération : la stratégie de coopération consiste à diviser la ressource selon la demande de consommation de chaque sous-modèle. Soit r la quantité de ressource actuelle et α_i la quantité demandée par un sous-modèle; la quantité de ressource consommée (α'_i) par chaque sous-modèle est définie par la formule représentée par l'expression (6.2)

$$\alpha'_i = \frac{\alpha_i r}{\sum_{i=1}^n \alpha_i} \quad (6.2)$$

et la nouvelle quantité de ressource (r') est définie par l'expression (6.3) :

$$r' = \sum_{i=1}^n \alpha'_i \quad (6.3)$$

Les stratégies susmentionnées sont attribuées pour chaque ressource de l'espace, *i.e.*, les stratégies sont reliées aux ressources afin de résoudre les conflits à leur niveau et non au niveau des sous-modèles. En effet, il est difficile de prédire et d'établir des stratégies par rapport à la présence des sous-modèles dans une portion de l'espace, si la répartition spatiale des sous-modèles est faite de manière aléatoire.

```
species space {
  init{
    //utilise la stratégie de priorisation
    do define_coordinator_function parameter:"ressource_1"
    filter_using_model_attribute:"priority";

    //utilise la stratégie de coopération
    do define_coordinator_function parameter:"ressource_2"
    fair_distribution:true;

    //utilise la stratégie de compétition
    do define_coordinator_function parameter:"ressource_3"
    dominant_species:$model_name$;
  }
}
```

6.2.5 Coordination dans un espace multi-niveaux et multi-échelle

Par espace multi-niveaux et multi-échelles, nous nous référons à un espace composé de plusieurs sous-espaces (multi-niveaux) contenant des couches de plus petites tailles (multi-échelle). La coordination abordée dans 6.2.4 consiste à déterminer l'état d'une particule de l'espace (une couche de l'espace ne possédant aucune couche imbriquée) à partir des influences envoyées par les sous-modèles et les stratégies appliquées pour la gestion des conflits. Ici, nous tenons à expliquer le fonctionnement de notre approche dans un espace multi-niveaux. Pour cela, il faut déterminer des fonctions d'agrégation et de désagrégation pour afin de visualiser un niveau particulier de l'espace.

Dans une structure multi-niveaux, il est possible de réutiliser le principe d'influence et réaction pour déterminer comment les niveaux supérieurs et inférieurs s'influencent mutuellement, en reprenant l'implémentation de [Mic07]. La technique d'agrégation consiste donc à combiner l'état des niveaux inférieurs pour déterminer l'état du niveau supérieur. La technique de désagrégation, quant à elle, consiste à distribuer la valeur des variables d'états (les ressources) du niveau supérieur aux niveaux inférieurs. Pour cela, à chaque changement d'état sur un niveau, des influences sont émises par les niveaux inférieurs pour calculer l'état

du niveau supérieur et l'état des niveaux inférieurs peuvent être définis à partir de l'état du niveau supérieur [Din+22a]. La combinaison ou la redistribution des variables d'état de chaque niveau doit alors être définie par le modélisateur. Comme le montre la figure 6.5, la coordination de modèle opère au niveau des particules de l'espace. De ce fait, dans une structure multi-niveaux, il convient juste de déterminer comment la combinaison des influences au niveau inférieur et la distribution des variables d'état du niveau supérieur sont faites [Din+22a].

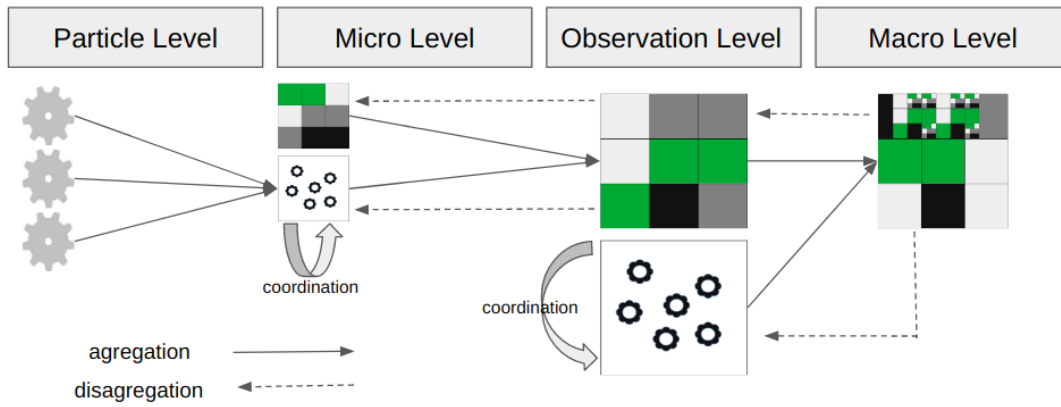


Figure 6.5: Coordination dans une structure multi-niveaux

Chapitre 7

Application sur la modélisation du fonctionnement du sol

Contenu du chapitre

7.1	Le projet CAMMiSolE	80
7.1.1	Collecte de données	80
7.1.2	Modèle de prédiction: ROT-MAT	82
7.1.3	Modèle Participatif	83
7.2	Sous-modèles	83
7.2.1	Modèle de Sol: APSF	84
7.2.2	Modèle de Vers de terre: SWORM	85
7.2.3	Modèles de Microbe	85
7.2.4	Racines	86
7.3	Simulation	86
7.3.1	Modèle de simulation	86
7.3.2	Résultats et Discussion	87
	Plan d'expérience	87
	Résultats	89
	Discussion	92

Dans ce chapitre, nous passons à l'évaluation de l'approche de couplage et de la coordination de modèle sur un modèle de décomposition de la matière organique. En premier lieu, nous détaillons quelques travaux de modélisation faits par l'équipe du projet, qui nous a mené à la construction de cette approche de couplage.

7.1 Le projet CAMMiSolE

Le projet CAMMiSolE : Effets du **C**hangement global en **A**frique de l'ouest et à **M**adagascar sur la diversité des **M**icroorganismes du **S**ol et ses conséquences sur les services **E**cosystémiques.

Contexte: L'agriculture est une activité très prometteuse pour le développement économique de certains pays de l'Afrique subsaharienne (Dont Madagascar) [Dia+07]. Cependant, son développement est freiné (au-delà les problèmes politiques) par deux principaux facteurs.

- L'agriculture principalement pluviale: l'agriculture à Madagascar est essentiellement pluviale. Elle est donc grandement menacée par le changement climatique. Le rendement agricole est rythmé par les saisons. Elle souffre également d'un manque de moyens pour la gestion de l'eau.
- L'accès limité aux moyens de fertilisation chimique constitue l'un des problèmes importants, car la majorité des agriculteurs sont de petits exploitants.

Objectif: l'objectif du projet CAMMiSolE est donc de combler ces lacunes par l'étude des processus du sol, puis d'intégrer ce savoir dans des outils de modélisation afin d'orienter au mieux les décideurs pour améliorer le rendement des agriculteurs face au changement global.

Dans un premier temps, l'équipe a fait une collecte de données auprès des agriculteurs en recueillant les pratiques agricoles des paysans, grâce à un jeu sérieux qui leur est présenté dans des feuilles de papier. À partir de cette enquête, un premier modèle de prédiction incluant les rotations de cultures a été créé pour simuler les rendements agricoles dans différents scénarios de stress environnemental. Ensuite, un modèle plus élaboré contenant deux interfaces représentant deux niveaux a été conçu en se basant sur les résultats du premier modèle. Au niveau macro, le jeu sérieux avec les rendements agricoles en guise sortie et au niveau micro les différents processus du sol qui influencent ces rendements. À partir de ce niveau micro, nous avons évalué les besoins en matière de couplage de modèles. L'approche de couplage construite dans ce projet de thèse contribue au projet CAMMiSolE en offrant un outil de modélisation et de simulation (à travers la plateforme Gama) pour décrire le fonctionnement du sol.

7.1.1 Collecte de données

Le recueil de données s'est déroulé à Imerintsiatosika, dans la région d'Itasy, à 35km d'Antananarivo. La création d'un jeu sérieux a permis à 17 agriculteurs de participer à la collecte de données. L'objectif du jeu est de connaître les pratiques agricoles des agriculteurs face à une situation de sécheresse et/ou un cas de manque de fertilisant pour chaque saison, sachant qu'une saison dure six mois.



Figure 7.1: Session de jeu sérieux

Règles du jeu: Le principe du jeu est simple. Un plateau de jeu est présenté à l'agriculteur. Il contient plusieurs types de terrain situés à différentes altitudes (rizière, plaine, colline, montagne) situés en ordonnée (axes y) et les saisons en abscisse (axes x). Ensuite, plusieurs coupons contenant les types de parcelles représentant les différents types de sol (tany mena (sol rouge), tany mainty (sol noir), tany mavo (sol jaunâtre)), cultures (légumineuses, tubercules et céréales) et fertilisants (fumiers, débris végétaux, composte) sont donnés à l'agriculteur pour qu'il les place sur le plateau de jeu. De cette manière, les enquêteurs peuvent déterminer les pratiques agricoles de l'agriculteur dans des situations de sécheresse ou/et manque de fertilisant.

Déroulement d'une session de jeu:

1. Les règles de jeu sont expliquées aux agriculteurs
2. L'agriculteur choisit le type de parcelle qu'il ou elle cultive en plaçant le coupon de parcelle sur le plateau du jeu.
3. L'agriculteur place les coupons de cultures et fertilisants pour présenter l'état actuel de leur culture (État initial du jeu).
4. L'agriculteur place ensuite les coupons de parcelle, cultures et fertilisants pour montrer les pratiques agricoles qu'ils utiliseront six mois après, dans des cas de sécheresse et/ou manque de fertilisant.

Grâce à cette enquête, il a été possible d'établir une base de données sur les pratiques agricoles de 17 agriculteurs à Imerintsiatosika. Un premier modèle

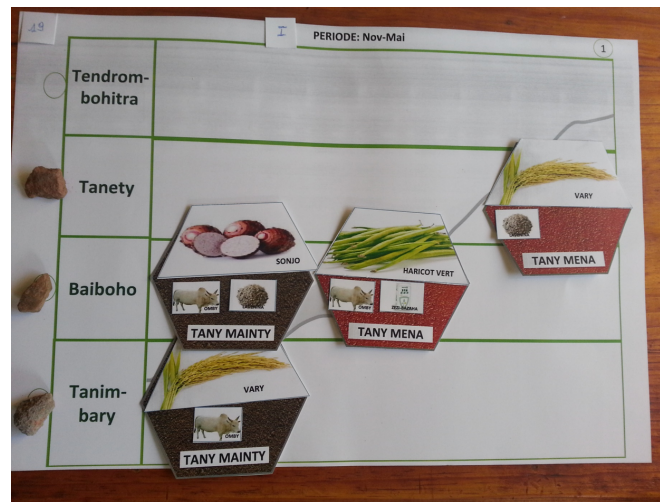


Figure 7.2: Résultat à la fin de session d'un jeu sérieux: Tanim-bary, Tanety, Baiboho et Tendrimbohitra représentent les niveaux où sont situées les parcelles cultivées

(ROT-MAT) de prédiction des rendements agricole a été conçu grâce à cette collecte de données.

7.1.2 Modèle de prédiction: ROT-MAT

ROT-MAT ou Rotation Matrix est le modèle de prédiction qui permet d'évaluer la pertinence (par rendement agricole) des pratiques agricoles des agriculteurs en cas de stress environnemental (sécheresses et manque de fertilisant). Il a été possible d'évaluer ces pratiques grâce aux travaux de Tombo Loïc Selmo [Loï16] sur la création d'un modèle à base d'agents pour prédire les rendements agricoles basés sur les résultats du jeu sérieux; et de Kanto Razanamalala [Raz17] sur la gestion de la fertilité des sols à Madagascar, qui a permis d'émettre des suggestions sur les rotations de cultures afin d'optimiser le rendement agricole.

Principe du modèle: Le modèle à base d'agents consiste à prédire le rendement agricole sur les scénarii de sécheresse et/ou manque de fertilisant. Le modèle a été implémenté sur la plateforme Gama [Tai+19a] et repose sur deux principaux éléments : un modèle à base d'agents représentant les agriculteurs ainsi que les parcelles à cultiver, puis un modèle de décision représenté par une matrice contenant la probabilité qu'une culture soit choisie par un agriculteur.

- Le modèle à base d'agents est constitué de deux principaux agents: l'agent agriculteur et l'agent parcelle. L'agent parcelle joue le rôle d'environnement en envoyant des stimuli à l'agent agriculteur, signalant qu'il doit faire une plantation ou de la récolte. L'agent agriculteur possède deux comportements. Le premier comportement consiste à plante un type de récolte

spécifique grâce à une matrice de décision (expliqué plus bas) lorsque l’agent parcelle envoie un signal de plantation. Le deuxième comportement consiste à récolter les plantes arrivées à maturation si la parcelle envoie un signal de maturation.

- Une matrice de décision ou *matrice de rotation* permet de déterminer la probabilité qu’un agriculteur choisisse une culture spécifique en saison¹ ou en contre-saison². Cette matrice a été créée à partir des données du jeu sérieux.

Le modèle ROT-MAT permet de prédire les pratiques agricoles des paysans, mais les effets sur le sol n’étaient pas pris en compte. Pour y remédier, l’équipe a conçu un autre modèle décrivant différents processus du sol en couplant plusieurs modèles représentant le biote du sol et leurs effets sur ce dernier.

7.1.3 Modèle Participatif

Un modèle plus avancé a été créé plus tard en intégrant une touche plus élaborée dans le modèle ROT-MAT. Ce modèle vise à faire une simulation participative en faisant interagir les scientifiques et les experts de domaines d’un côté et les agriculteurs de l’autre. De ce fait, deux interfaces ont été développées dans la plateforme Gama pour servir ces deux communautés. La première interface, conçu pour les agriculteurs, est l’implémentation du jeu sérieux. L’entrée du modèle est la même que le jeu sérieux (type de parcelle, culture, fertilisant) et la sortie (récolte) est déterminée par calcul et quantifié en nombre de sac de la culture choisie. La deuxième interface, pour les scientifiques et experts de domaines, consiste à calibrer un modèle de simulation du fonctionnement du sol (microbes, vers de terre, nématodes), couplé au modèle utilisé par l’agriculteur pour calculer leur rendement agricole.

Le modèle-couplé permet de faire une simulation participative. Cependant, le modèle utilisé par les experts de domaine avaient besoin d’amélioration, car les sous-modèles étaient couplés de façon ad-hoc, sans vraiment prendre en compte les contraintes de couplage et les possibles conflits de ressources, d’où l’élaboration de l’approche détaillée dans cette thèse.

7.2 Sous-modèles

Afin de tester notre approche de coordination, nous avons couplé à travers un modèle de sol inspiré de l’APSF [Mar+08], des vers de terres en reprenant la dynamique de SWORM [Bla+09], une population de microbes décomposeur et des agents représentant les racines du sol.

¹se réfère à la saison ou une culture est favorable à la plantation

²saison ou la plantation d’une culture est en dehors de la saison normale de culture

7.2.1 Modèle de Sol: APSF

L'approche APSF [Mar+08] permet de représenter l'hétérogénéité du sol, grâce à une structure pseudo-fractale, dans un cube composé d'autres cubes de plus petite taille pour représenter les particules du sol à des échelles différentes.

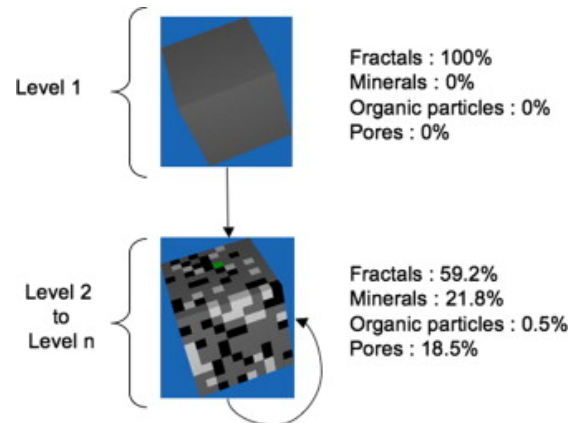


Figure 7.3: Image en 3D de l'APSF sur la plateforme MaDKit[GF00] (image tirée de [Mar+08])

Cette structure est générée à partir d'un arbre de canevas comme représenté dans la figure 7.4

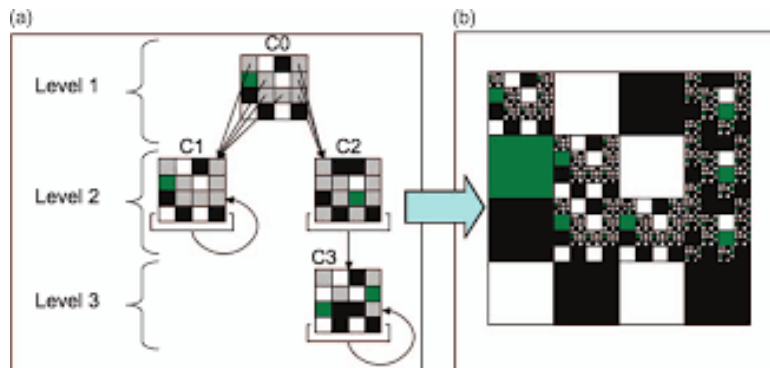


Figure 7.4: Canevas APSF: les canevas sont représentés par C0,C1,C2,C3 et sont alignés verticalement de sorte que ceux situés plus bas révèlent les patterns utilisés pour décrire les agrégats à des échelles plus petites (figure a). Dans la figure b, un sol en 2D construit en déployant différents canevas.

Le sol est généré sur une grille en reprenant la structure basique de l'APSF, les sous-espaces sont du type pore, matière organique (OM) ou matière minérale (MM). La matière organique est catégorisée en trois sous-types pour représenter les ressources du sol. La matière organique fraîche (raw_OM), la matière or-

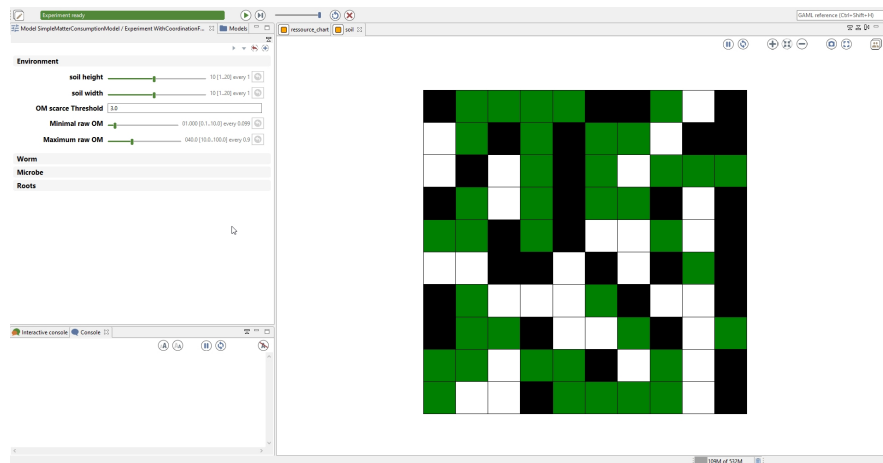


Figure 7.5: Représentation simplifiée du sol dans la plateforme gama

ganique décomposable (`decomp_OM`), la matière organique soluble (`soluble_OM`) et la matière minérale soluble (`soluble_MM`).

7.2.2 Modèle de Vers de terre: SWORM

Dans le modèle `sworm`, les vers de terre sont responsables de restructuration du sol en digérant la matière organique fraîche et en rejetant pendant leur passage un autre type de matière organique tout en activant l'activité microbienne dans cette portion de l'espace. Nous reproduisons ce comportement dans un modèle gama appelé `WormModel`. Dans ce modèle, un vers de terre ne possède qu'un comportement (`run()`) qu'il exécute à chaque itération. Dans ce comportement, il choisit trois actions possibles en fonction de la disposition de matière organique:

- `eat()`: pour consommer de la matière organique fraîche présente dans les sous-espaces du sol (`raw_OM`). Quand le sous-espace ne contient plus de matière organique fraîche, elle devient un pore et active un certain nombre d'agents `Microbe` dans ce sous-espace.
- `move()`: c'est la capacité du vers de terre à se déplacer de pore en pore à la recherche de matière organique fraîche.
- `deject()`: lorsque le vers de terre a fini de consommer la matière organique, elle rejette une certaine quantité de matière organique décomposable dans le pore ou il se situait précédemment.

7.2.3 Modèles de Microbe

Les vers de terre peuvent activer une colonie de microbes dans le sol en décomposant la matière organique fraîche. En présence de cette matière organique, les mi-

crobes entrent en activation et envoient des enzymes pour la décomposer. Cette dynamique est implémentée dans un modèle appelé `MicrobeModel`. Les agents microbes sont dispersés dans les pores (de façon homogène pour vérifier l'efficacité de l'approche de coordination). Le sous-modèle n'a qu'un seul comportement d'exécution `run()` et est exécuté à chaque pas de temps. Les actions possibles sont:

- `activate()`: permet de déclencher l'activité enzymatique des microbes lorsque la matière organique décomposable `decomp_OM` est disponible.
- `decompose()`: envoie des enzymes dans l'espace afin de décomposer la matière organique décomposable en matière organique soluble (`soluble_OM`).
- `feed()`: les microbes se nourrissent de matière organique soluble (`soluble_OM`) et la transforment en matière minérale soluble (`soluble_MM`).
- `feedMM()`: les microbes se nourrissent de matière minérale soluble (`soluble_MM`) en compétition avec les racines.

7.2.4 Racines

Les racines sont aussi présentes dans le sol sous l'aspect d'agents éparpillés dans l'espace. Les racines sont situées dans les pores et se nourrissent de nutriments (`soluble_MM`) présentes en compétition avec les microbes. L'implémentation de ce modèle est contenue dans un fichier appelé `RootModel`. Le modèle possède un comportement (`run()`) et une action possible(`feed()`).

7.3 Simulation

7.3.1 Modèle de simulation

Dans la figure 7.6, nous illustrons le modèle de simulation à implémenter dans la plateforme gama. Le modèle simplifié de décomposition de la matière organique peut être trouvé sur la plateforme GitHub ³.

Dans une situation de conflit, les vers de terres se disputent la matière organique fraîche. Nous avons implémenté le modèle pour que les vers de terres arrivés en premier sur une particule soient les premiers à se nourrir de la ressource. Dans le modèle de simulation, la ressource `raw_OM` est gérée par une fonction de priorisation en cas de conflit.

Les microbes coopèrent lorsqu'il s'agit de décomposer la matière organique décomposable (`decomp_OM`). Cette ressource sera gérée par le coordinateur avec

³lien du modèle de simulation: [modèle de simulation sous gama](#)

une fonction de coopération lors d'un conflit.

Les ressources `soluble_MM` et `soluble_OM` sont sujets à la compétition entre microbes et microbes pour la ressource `soluble_OM` et entre microbes et racines pour la ressource `soluble_MM`.

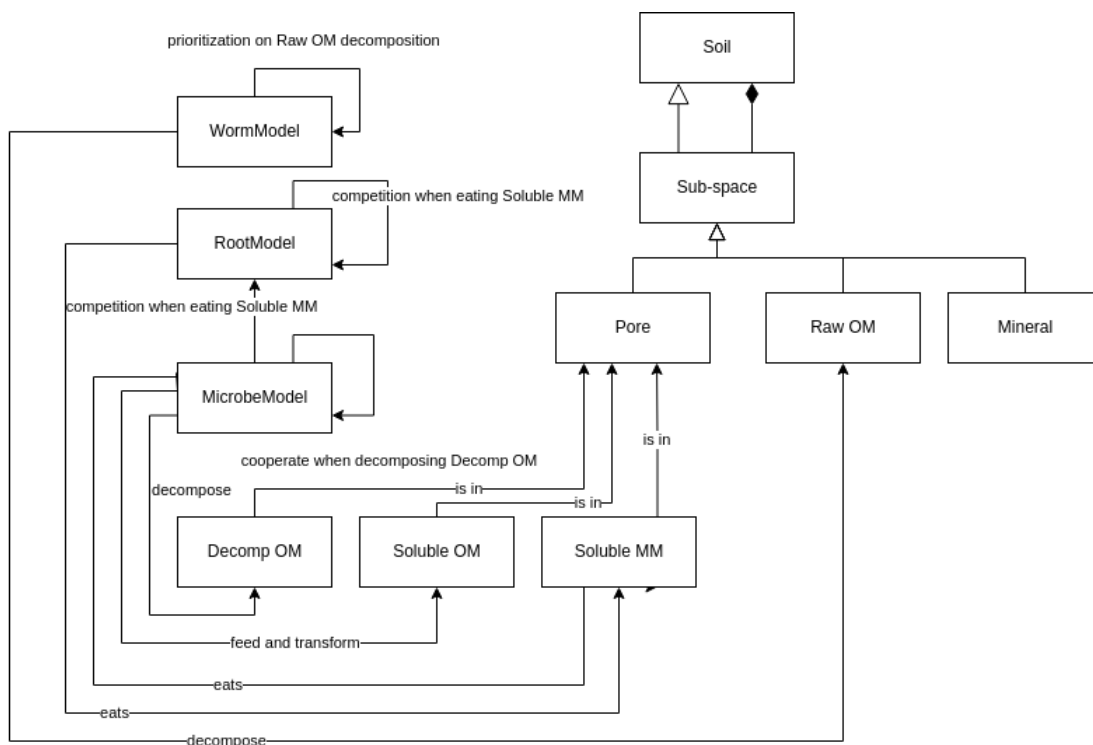


Figure 7.6: Le modèle-couplé à implémenter dans gama

7.3.2 Résultats et Discussion

Plan d'expérience

L'objectif de cette simulation est de déterminer l'efficacité de notre approche de couplage et en particulier l'efficacité de la coordination de modèles. Pour cela, nous avons lancé une série de tests sur le modèle-couplé en variant la taille du sol. Chaque simulation est évaluée sur une durée de 60 jours (temps de simulation). Comme le modèle est totalement déterministe, il est possible de comparer l'utilisation ou non des fonctions de coordination et évaluer les résultats à chaque modification de la taille de l'espace.

Comme le nombre de sous-modèles est déterminé par le nombre de pores dans l'espace, en augmentant la taille de l'espace, le nombre d'interactions et de conflits augmentent par conséquent et permet ainsi de faire une comparaison.

Cette comparaison est possible, car on simule l'utilisation d'un couplage spatialisé sans méthode de coordination de sous-modèles (les approches de couplage de la littérature) avec l'utilisation d'un couplage spatialisé ou il est possible de gérer conflits de ressources entre les sous-modèles. Cette comparaison est possible grâce à ce modèle déterministe de décomposition de la matière organique du sol.

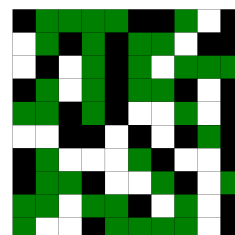
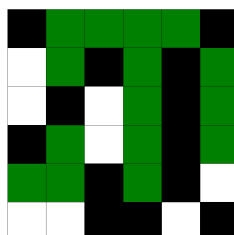
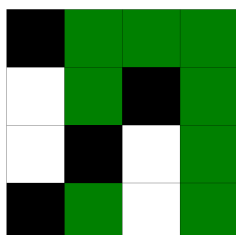


Figure 7.7: sol de taille 4x4 (25% de matières minérales et de pores et 50% de matières organiques)

Figure 7.8: sol de taille 6x6 (25% de pores, 33% de matières minérales et 42% de matières organiques)

Figure 7.9: sol de taille 10x10 (27% de pores, 32% de matières minérales et 41% de matières organiques)

Table 7.1: Paramètres initiaux

Nom du paramètre	Valeur
Taille du sol	4x4,6x6,10x10
Raw OM	40g/particule de type OM
pas de temps de la simulation	2 jours
Nombre de vers de terre	1 unité/ pore
pas de temps de WormModel	1 jour
Décomposition de Raw OM	1.3g/vers de terre
Nombre de colonie de Microbe	5 unité/pore
Consommation de decomp OM	0.03g/microbe
Consommation de Soluble OM(Microbe)	0.007g/microbe
Consommation de Soluble MM(Microbe)	0.009g/microbe
Pas de temps MicrobeModel	4 heures
Nombre de racine	2 unité/pore
Consommation de Soluble MM(Racine)	0.008g/racine
Pas de temps du RootModel	12 heures

Résultats

Dans les figures 7.10 (sans stratégie de coordination) et 7.11 (avec stratégie de coordination), nous observons la disponibilité des ressources du sol, ou la taille du sol est réduite à 16 particules (4 x 4) (25% de matières minérales et de pores et 50% de matière organique). On observe ici une légère différence dans la disponibilité des ressources.

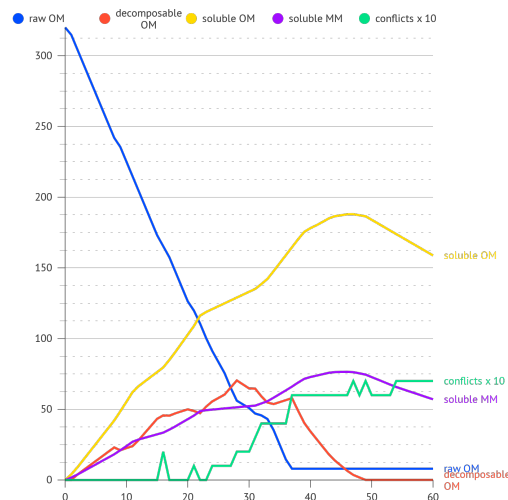
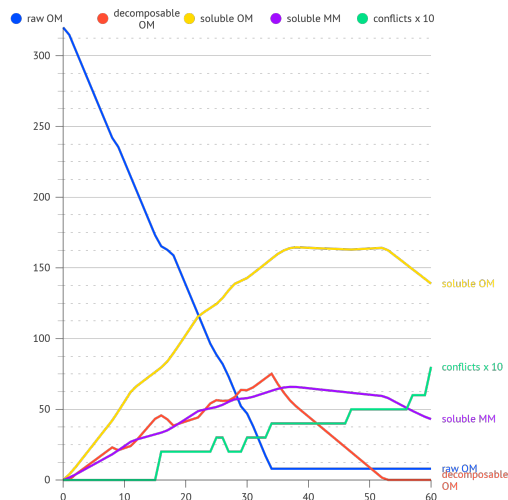


Figure 7.10: variation des ressources sans stratégie de coordination sur un sol de taille 4x4

Figure 7.11: variation des ressources avec stratégie de coordination sur un sol de taille 4x4

Dans les figures 7.12 (sans coordination) et 7.13 (avec stratégie de coordination) on constate un écart conséquent dans les résultats des deux simulations.

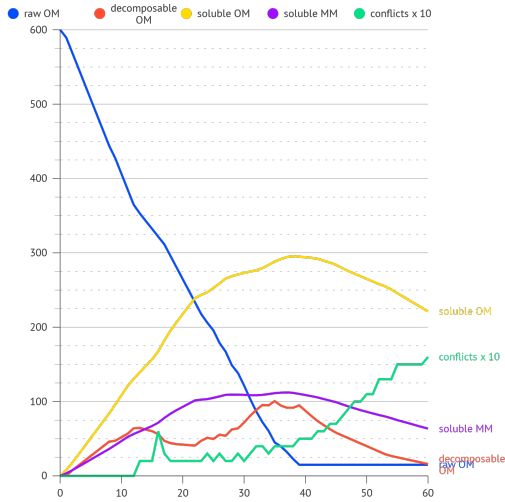


Figure 7.12: variation des ressources sans stratégie de coordination sur un sol de taille 6x6

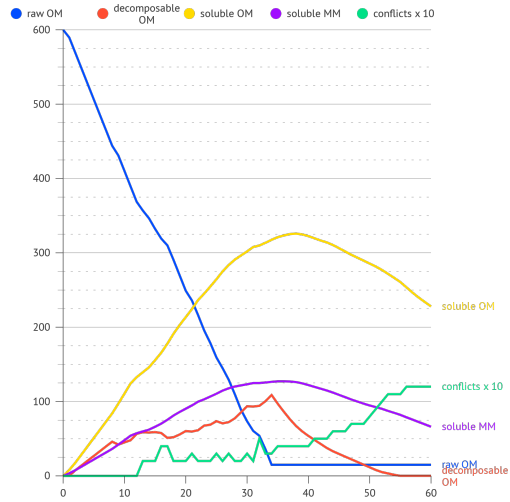


Figure 7.13: variation des ressources avec stratégie de coordination sur un sol de taille 6x6

Dans les figures 7.14 et 7.15, nous observons une différence significative de la disponibilité des ressources sur un sol de taille 10x10.

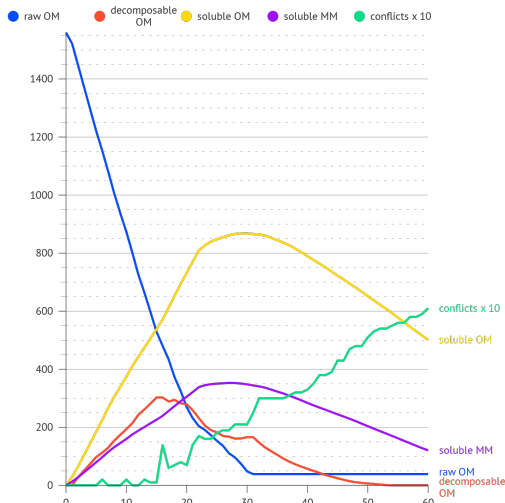


Figure 7.14: variation des ressources sans stratégie de coordination sur un sol de taille 10x10

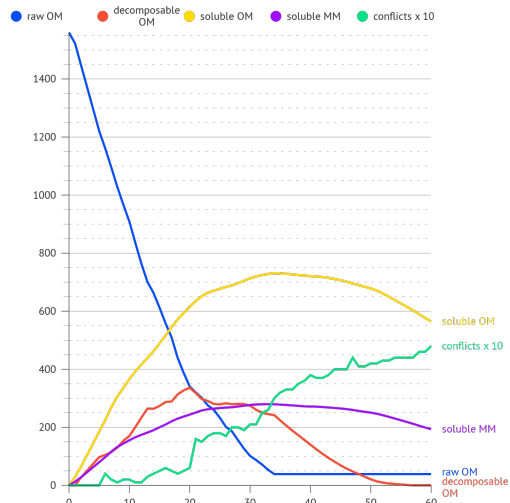


Figure 7.15: variation des ressources avec stratégie de coordination sur un sol de taille 10x10

La ressource `raw_OM` est disputée par les sous-modèles vers de terre et gérée par une stratégie de priorisation. Dans la figure 7.16, nous montrons la différence entre les deux simulations (avec et sans stratégie de coordination) sur le sol de

taille 10x10. Les microbes sont en coopération sur la ressource `decomp_OM`, les résultats des deux simulations sont comparés sur la figure 7.17.

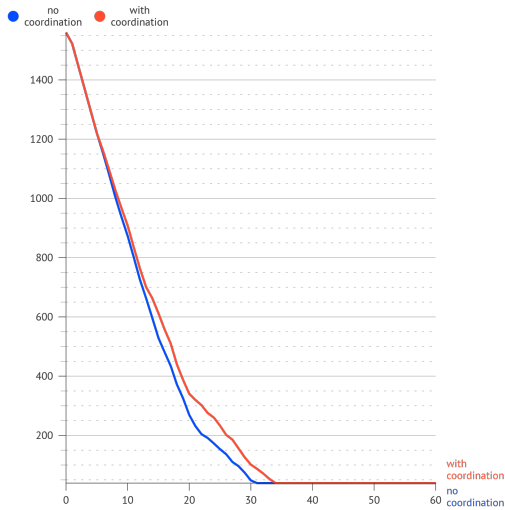


Figure 7.16: comparaison de la variation de la ressource `raw_OM` avec et sans coordination

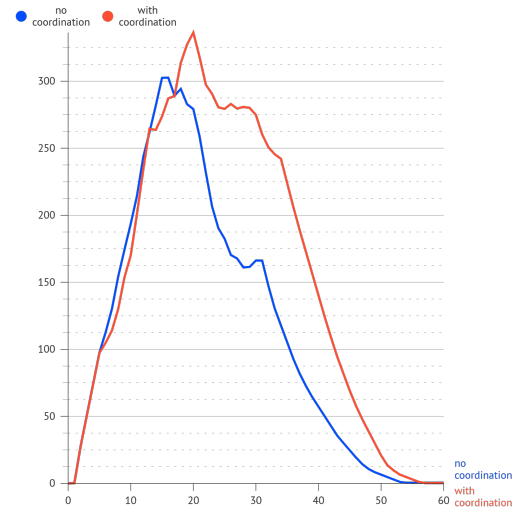


Figure 7.17: comparaison de la variation de la ressource `decomp_OM` avec et sans coordination

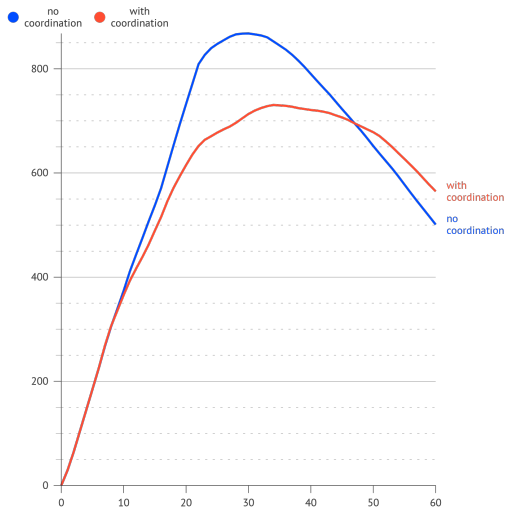


Figure 7.18: comparaison de la variation de la ressource `soluble_OM` avec et sans coordination

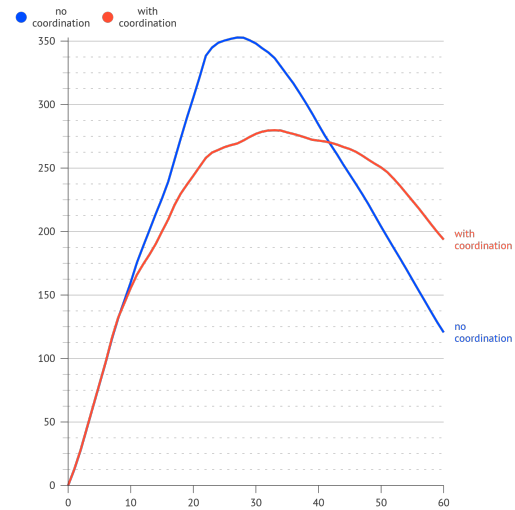


Figure 7.19: comparaison de la variation de la ressource `soluble_MM` avec et sans coordination

Dans les figures 7.16, 7.17, 7.18 et 7.19, nous observons les différences entre les deux simulations (avec et sans fonctions de coordination) sur les ressources

`soluble_OM` et `soluble_MM`. Les deux ressources sont gérées par une stratégie de compétition.

Le nombre de conflits détectés sur les deux simulations sont également différents (figure 7.20), expliquant la différence dans les résultats.

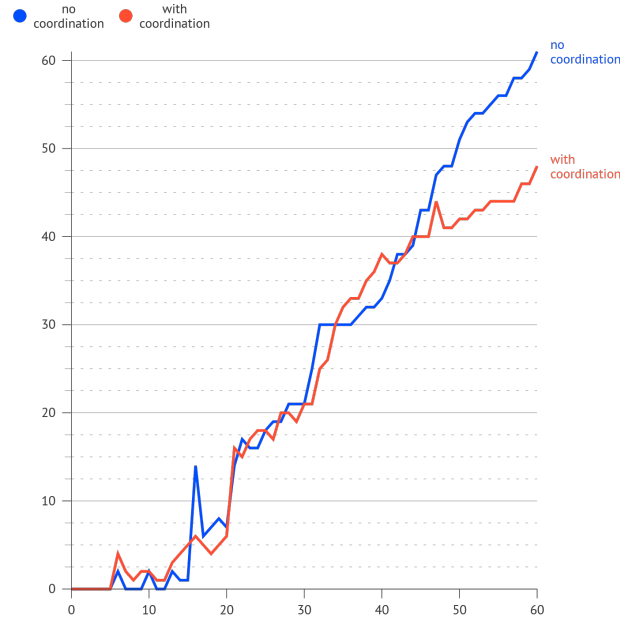


Figure 7.20: comparaison des conflits détectés avec et sans coordination

Discussion

Impacts des stratégies de coordination sur la simulation : Dans les résultats observés, nous constatons que les deux modèles de simulation ont le même comportement même en variant la taille de l'espace. Cependant, les données présentées dans les graphiques présentent des différences allant de "subtiles" (sol de taille 4x4) à "avec différence significative" (sols de taille 6x6 et 10x10). Ces différences sont dues à l'utilisation des stratégies de coordination, qui gèrent la distribution des ressources lorsque des conflits sont détectés. Nous avons aussi constaté qu'en augmentant la taille du sol, on augmente le nombre d'interactions, la simulation garde le même comportement, mais la comparaison de la quantité de ressources disponibles sur les deux simulations présente un écart significatif. Cet écart devient de plus en plus grand lorsqu'on augmente la taille de l'espace. Cela est dû au fait que lorsqu'on n'utilise pas de stratégie de coordination, les sous-modèles n'exploitent pas totalement les ressources dans l'espace. Comme les ressources `decomp_OM soluble_OM` et `soluble_MM` sont les résultats du raffinement de `raw_OM` par les sous-modèles, les stratégies utilisées pour calculer cette transformation peuvent grandement impacter la disponibilité de ces ressources. On peut

constater cela lorsqu'on compare la variation des ressources des deux simulations (figures 7.16, 7.17, 7.18, 7.19). Lorsque aucune stratégie de coordination n'est utilisée, les modèles détectent de la ressource, mais ne peuvent pas l'exploiter en raison du manque de contrôle ou de règles sur l'utilisation de cette ressource. Les résultats de l'application des différentes stratégies de coordination de modèles peuvent être visualisés au niveau des particules grâce à un fichier contenant les opérations effectuées.

Pertinence de l'approche de coordination des sous-modèles en couplage spatialisé de modèle : Les résultats obtenus suggèrent que la coordination des sous-modèles dans un contexte de couplage spatialisé de modèles constitue un moyen de définition de règles par rapport à l'interaction entre les modèles. À la différence des approches de couplages spatialisés cités dans le chapitre 3 qui utilisent un ordonnanceur pour synchroniser l'exécution de leurs sous-modèles, nous proposons une approche basée sur l'influence-réaction en collaboration avec un ordonnanceur. Cette approche permet de définir l'état de l'environnement en calculant la phase de réaction par des stratégies qui filtrent l'application des influences envoyées par les sous-modèles. Cette perspective a déjà été mentionnée dans [Pay+06b]; [Ral+19] ou ils utilisent un nouveau modèle de temporalité pour palier la limite des ordonnanceurs classiques utilisé dans les simulateurs.

Coordination dans un espace multi-niveaux et multi-échelles : Comme nous l'avons détaillé dans le chapitre 6, il est possible d'adapter cette approche sur un espace multi-niveaux tel que l'APSF [Mar+08]. La seule condition pour visualiser l'évolution du modèle-couplé à un niveau particulier nécessite des fonctions d'agrégation et de désagrégation afin de maintenir une certaine cohérence entre les niveaux. Pour cela, nous proposons une architecture permettant aux modélisateurs d'implémenter une approche basée sur l'influence réaction comme dans les travaux de [Mic07]; [MVD10].

Dans le cas où les échelles (les unités utilisées par les sous-modèles pour chaque ressource) utilisées par les sous-modèles sont différentes, une méthode de mise à l'échelle des données est indispensable afin de rendre l'espace et les sous-modèles interopérables. Une solution à ce problème a été proposée par une fonction de mise à l'échelle dans [Has15] où à chaque échange de données entre un sous-modèle et l'espace, les données sont mises à l'échelle de l'espace (si ce sont des données de sortie du sous-modèle) ou mise à l'échelle du sous-modèle (si ce sont des données d'entrées).

Conclusion

Dans cette thèse, nous avons abordé une approche de modélisation appelée couplage de modèles, dont la finalité est la représentation partiellement complète du sol et de ses processus. Étant donné que le sol est un espace hétérogène, ces processus y sont répartis à différents niveaux et échelles. Nous avons alors établi, grâce à une littérature abondante sur le couplage de modèle et le système multi-agents, une approche de couplage spatialisée, en reliant les différents modèles à coupler à travers l'espace, par les ressources qui y sont situées.

Nous avons découvert que ce type de couplage spatialisé provoque des imprécisions grâce à un premier essai de modélisation dans le projet CAMMiSolE. Ces imprécisions sont notamment dues à des conflits de ressources. Les approches de couplages spatialisés dans la littérature sont limitées lorsqu'il s'agit de gérer les actions simultanées sur une ressource du sol quand celle-ci ne supporte pas les demandes qui lui sont imposées. Pour cela, nous avons introduit le module de coordination, présent dans chaque sous-espace, qui a pour rôle de répartir l'attribution des ressources à chaque sous-modèle en cas de conflit. Le module de coordination se base sur une théorie d'action en système multi-agents appelée "influence reaction", en séparant l'état mental des agents (ce qu'ils veulent faire) et ce qui se passe sur leur environnement (ce qui est appliqué).

La phase d'influence permet de récupérer les demandes de ressources envoyées par les sous-modèles pour être traitées par des stratégies de coordination lorsque la phase de réaction est calculée au niveau de chaque sous-espace. Grâce à ces stratégies, l'espace restreint certaines actions des sous-modèles tout en tenant en compte la disponibilité des ressources. Cela affecte le comportement d'autres sous-modèles et par conséquent les résultats de simulations.

Nous avons testé l'efficacité de cette approche dans une version déterministe d'un modèle simplifié de sol. Plusieurs tests de comparaisons entre l'utilisation ou non du module de coordination ont été faites sur ce modèle. Les résultats suggèrent que l'utilisation du module de coordination donne une indication plus précise des variations de la disponibilité des ressources dans le temps.

Afin de faire correspondre notre approche à la structure hétérogène du sol, nous avons proposé, grâce au principe influence reaction, un autre module permettant d'observer les impacts entre les différents niveaux de l'espace. L'état des niveaux inférieurs est envoyé sous forme d'influences, pour ensuite être traité par

une fonction d'agrégation puis appliqué au niveau d'observation. Pour le niveau supérieur, une fonction désagrégante est appliquée pour répartir les éventuelles modifications de ressources aux niveaux inférieurs.

L'architecture de notre approche a été formalisée en UML, puis implémentée sur la plateforme GAMA afin de tester l'approche sur un modèle de fonctionnement du sol. Les résultats obtenus révèlent que notre approche donne une certaine flexibilité pour gérer la simultanéité des actions sur l'espace, par les sous-modèles, mais donne aussi des résultats plus précis grâce à la coordination des sous-modèles à différents niveaux. Cependant, elle se limite seulement aux modèles spatiaux ou les sous-modèles échangent à travers les ressources de l'espace. Il serait intéressant d'améliorer les stratégies de coordinations et implémenter des fonctions d'échanges entre les niveaux pour enrichir les résultats de cette thèse. Cela se fera par le maintien régulier du plugin développé sur la plateforme Gama.

Glossaire

Agrégation : Technique permettant de regrouper des données provenant de sources différentes.

Aspatial : qui n'est pas spatial ou relatif à l'espace.

Causalité : Relation entre une cause et son effet.

Changement global : Changement global des conditions climatiques.

Conflit : Un conflit est défini comme un état dans lequel les actions des entités impliquées sont limitées [Tes+02].

Couplage de modèle : Approche permettant de combiner, en un seul modèle, plusieurs modèles hétérogènes.

Désagrégation : Technique permettant de distribuer des données vers plusieurs sources.

Échelle : Une échelle définit une résolution ou une étendue.

Exécution pessimiste : Exécution d'un modèle de manière conditionnelle où seules les actions sûres sont exécutées.

Facteur de couplage : Intermédiaire permettant la communication entre les modèles à coupler.

Formalisme : Un formalisme permet d'homogénéiser une représentation grâce à un langage formel.

Fractal : Un fractal se définit comme étant la répétition d'une structure géométrique complexe à des échelles différentes.

Métamodèle : Modèle d'un modèle.

Modèle couplé : Modèle résultant d'un couplage, modèle composé de sous-modèles.

Modèle de simulation : Modèle spécifique à une plateforme de simulation.

Modèle : Abstraction de la réalité.

Niveau : Point de vue d'un système.

Orchestrateur, Scheduler : Entité arrangeant l'exécution d'un système et de ses composants.

Sous-modèle : Composant représentant un point de vue d'un système.

Système complexe : Système dans lequel se déroule un nombre important d'interactions, rendant le dénouement et la prédiction des états du système difficile.

Système multi-agent : Système dans lequel des entités virtuelles appelées

“agent” évoluent en interaction avec d’autres agents ou avec l’environnement ou ils évoluent.

Bibliographie

- [Als+07] W. Alshabi, S. Ramaswamy, M. Itmi, and H. Abdulrab. “Coordination, Cooperation and Conflict Resolution in Multi-Agent Systems”. en. In: *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*. Ed. by Tarek Sobh. Dordrecht: Springer Netherlands, 2007, pp. 495–500.
- [ADM07] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. “A unified and general framework for argumentation-based negotiation”. en. In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*. Honolulu, Hawaii: ACM Press, 2007, p. 1.
- [AS08] Manuel Atencia and Marco Schorlemmer. “I-SSA: Interaction-Situated Semantic Alignment”. en. In: *On the Move to Meaningful Internet Systems: OTM 2008*. Ed. by Robert Meersman and Zahir Tari. Vol. 5331. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 445–455.
- [SHW18] Bruno St-Aubin, Omar Hesham, and Gabriel Wainer. “A Cell-DEVS visualization and analysis platform”. In: July 2018, p. 14.
- [Awa+13] M. U. Awais, P. Palensky, A. Elsheikh, E. Widl, and S. Matthias. “The high level architecture RTI as a master to the functional mock-up interface components”. en. In: *2013 International Conference on Computing, Networking and Communications (ICNC)*. San Diego, CA: IEEE, Jan. 2013, pp. 315–320.
- [Ban+16] Arnaud Banos, Nathalie Corson, Benoit Gaudou, Vincent Laperrière, and Sébastien Rey Coyrehourcq. “Coupling micro and macro dynamics models on networks: Application to disease spread”. In: *Multi-Agent Based Simulation XVI: International Workshop, MABS 2015, Istanbul, Turkey, May 5, 2015, Revised Selected Papers 16*. Springer, 2016, pp. 19–33.
- [Ban+17] Arnaud Banos, Nathalie Corson, Christophe Lang, Nicolas Marilleau, and Patrick Taillandier. “Multiscale Modeling: Application to Traffic Flow”. en. In: *Agent-based Spatial Simulation with NetLogo, Volume 2*. Elsevier, 2017, pp. 37–62.

- [Bar97] Fernando J Barros. “Modeling formalisms for dynamic structure systems”. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 7.4 (1997), pp. 501–515.
- [Ber+09] Jacques-Eric Bergez, Patrick Chabrier, Frederick Garcia, Christian Gary, David Makowski, Gauthier Quesnel, E. Ramat, Helene Raynal, Nathalie Rouse, and Daniel Wallach. “RECORD: a new software platform to model and simulate cropping systems”. In: *Farming systems design 2009. International symposium on methodologies for integrated analysis of farm production systems*. Methodologies for integrated analysis of farm production systems. Monterey, United States, Aug. 2009, np.
- [BB09] Mike Bithell and James Brasington. “Coupling agent-based models of subsistence farming with individual-based forest models and dynamic models of water distribution”. en. In: *Environmental Modelling & Software* 24.2 (Feb. 2009), pp. 173–190.
- [Bla+09] E. Blanchart, N. Marilleau, J.-L. Chotte, A. Drogoul, E. Perrier, and Ch. Cambier. “SWORM: an agent-based model to simulate the effect of earthworms on soil structure”. en. In: *European Journal of Soil Science* 60.1 (Feb. 2009), pp. 13–21.
- [Blo12] T Blochwitz. “Functional Mockup Interface for Model Exchange and Co-Simulation”. en. In: (2012), p. 38.
- [BTV06] Guido Boella, Leendert van der Torre, and Harko Verhagen. “Introduction to normative multiagent systems”. en. In: *Computational & Mathematical Organization Theory* 12.2-3 (Oct. 2006), pp. 71–79.
- [Boo82] Grady Booch. “Object-oriented design”. In: *ACM SIGAda Ada Letters* 1.3 (1982), pp. 64–76.
- [BK00a] Jo Ellen Brandmeyer and Hassan A Karimi. “Coupling methodologies for environmental models”. en. In: *Environmental Modelling & Software* 15.5 (July 2000), pp. 479–488.
- [BK00b] Jo Ellen Brandmeyer and Hassan A Karimi. “Coupling methodologies for environmental models”. In: *Environmental Modelling & Software* 15.5 (July 2000), pp. 479–488.
- [Bra08] Fred Brauer. “Compartmental models in epidemiology”. In: *Mathematical epidemiology* (2008), pp. 19–79.
- [BDT01] J Broersen, M Dastani, and L van der Torre. “Wishful thinking”. In: *Proceedings of D NMR’01/2* (2001).

- [Bro+01] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert van der Torre. “The BOID Architecture: Conflicts between Beliefs, Obligations, Intentions and Desires”. In: *Proceedings of the Fifth International Conference on Autonomous Agents*. AGENTS ’01. Montreal, Quebec, Canada: Association for Computing Machinery, 2001, pp. 9–16.
- [Cam+12] Benjamin Camus, Julien Siebert, Christine Bourjot, and Vincent Chevrier. “Modélisation multi-niveaux dans AA4MM”. In: (2012).
- [CF98] C. Castelfranchi and R. Falcone. “Principles of trust for MAS: cognitive anatomy, social importance, and quantification”. In: *Proceedings International Conference on Multi Agent Systems (Cat. No.98EX160)*. 1998, pp. 72–79.
- [CEH96] Thomas W Charnock, John Elgy, and P Hedges. “Application of gis linked environment models over a large area”. In: *3rd international conference on integrating GIS and environmental modeling CD-ROM, NCGIA, UC Santa Barbara*. sn. 1996.
- [CZ94] Alex Chung Hen Chow and Bernard P Zeigler. “Parallel DEVS: A parallel, hierarchical, modular modeling formalism”. In: *Proceedings of Winter Simulation Conference*. IEEE. 1994, pp. 716–722.
- [DKW98] Judith S Dahmann, Frederick Kuhl, and Richard Weatherly. “Standards for simulation: As simple as possible but not simpler the high level architecture for simulation”. In: *Simulation* 71.6 (1998), pp. 378–387.
- [Dav+07] Daniel David, Denis Payet, Aurélie Botta, Gilles Lajoie, Sébastien Manglou, and Rémy Courdier. “Un couplage de dynamiques comportementales: le modèle DS pour l’aménagement du territoire.” In: *JFSMA* 7 (2007), pp. 129–138.
- [Dia+07] Xinshen Diao, Peter BR Hazell, Danielle Resnick, and James Thurlow. *The role of agriculture in development: Implications for Sub-Saharan Africa*. Vol. 153. Intl Food Policy Res Inst, 2007.
- [Din+22a] J.B Dinaharison, N.Marilleau, H.L Rakotonirainy, N. Corson, and J-P. Müller. “Toward A Multi-Leveled Coordination Of Sub-Models In Spatial Model Coupling”. In: *2nd conference GAMA Days*. 2022.
- [Din+22b] Jean Bienvenue Dinaharison, Nicolas Marilleau, Hasina Lalaina Rakotonirainy, Nathalie Corson, Laetitia Bernard, and Jean-Pierre Müller. “Agent-based spatial model coupling using a coordination unit”. In: *Proceedings of the 23rd Workshop ”From Objects to Agents”, Genova, Italy, September 1-3, 2022*. Ed. by Angelo Ferrando and Viviana Mascardi. Vol. 3261. CEUR Workshop Proceedings. CEUR-WS.org, 2022, pp. 164–175.

- [Dro+20] Alexis Drogoul, Patrick Taillandier, Benoit Gaudou, Marc Choisy, Kevin Chapuis, Nghi Huynh Quang, Doanh Nguyen-Ngoc, Damien Philippon, Arthur Brugière, and Pierre Larmande. “Designing social simulation to (seriously) support decision-making: COMOKIT, an agent-based modelling toolkit to analyse and compare the impacts of public health interventions against COVID-19”. In: *Review of Artificial Societies and Social Simulation-RofASSS* (2020).
- [DRP03] Raphaël Duboz, Éric Ramat, and Philippe Preux. “Scale Transfer Modeling: Using Emergent Computation for Coupling an Ordinary Differential Equation System with a Reactive Agent Model”. In: *Systems Analysis Modelling Simulation* 43.6 (June 2003), pp. 793–814.
- [Eri71] Erik Eriksson. “Compartment models and reservoir theory”. In: *Annual Review of Ecology and Systematics* 2.1 (1971), pp. 67–84.
- [Fer95] J. Ferber. “Les Systèmes multi-agents: vers une intelligence collective”. en. In: *undefined* (1995).
- [FMB05] Jacques Ferber, Fabien Michel, and José Báez. “AGRE: Integrating environments with organizations”. In: *Environments for Multi-Agent Systems: First International Workshop, E4MAS 2004, New York, NY, July 19, 2004, Revised Selected Papers 1*. Springer. 2005, pp. 48–56.
- [FM96] Jacques Ferber and Jean-Pierre Müller. “Influences and Reaction : a Model of Situated Multiagent Systems”. en. In: (1996), p. 9.
- [Fia01] Yawa Edem Fianyo. “Couplage de modèles à l’aide d’agents : le système OSIRIS”. Publication Title: <http://www.theses.fr>. thesis. Paris 9, Jan. 2001.
- [Fre+19] Enrico Fregnan, Tobias Baum, Fabio Palomba, and Alberto Bacchelli. “A survey on software coupling relations and tools”. In: *Information and Software Technology* 107 (2019), pp. 159–178.
- [Gar+03] P. Garnier, C. Néel, C. Aita, S. Recous, F. Lafolie, and B. Mary. “Modelling carbon and nitrogen dynamics in a bare soil with and without straw incorporation”. In: *European Journal of Soil Science* 54.3 (2003), pp. 555–568.
- [Gér+08] F Gérard, KU Mayer, MJ Hodson, and Jacques Ranger. “Modelling the biogeochemical cycle of silicon in soils: application to a temperate forest ecosystem”. In: *Geochimica et Cosmochimica Acta* 72.3 (2008), pp. 741–758.

- [GLH12] Javier Gil-Quijano, Thomas Louail, and Guillaume Hutzler. “From biological to urban cells: lessons from three multilevel agent-based models”. In: *Principles and Practice of Multi-Agent Systems: 13th International Conference, PRIMA 2010, Kolkata, India, November 12-15, 2010, Revised Selected Papers 13*. Springer. 2012, pp. 620–635.
- [GGW07] J. B. Gregersen, P. J. A. Gijsbers, and S. J. P. Westen. “OpenMI: Open modelling interface”. en. In: *Journal of Hydroinformatics* 9.3 (July 2007), pp. 175–191.
- [GF00] Olivier Gutknecht and Jacques Ferber. “The MadKit Agent Platform Architecture”. In: vol. 1887. July 2000.
- [HHM19] Q.K. Harpham, A. Hughes, and R.V. Moore. “Introductory overview: The OpenMI 2.0 standard for integrating numerical models”. In: *Environmental Modelling & Software* 122 (2019), p. 104549.
- [Has15] Inès Hassoumi. “Approche multi-agents de couplage de modèles pour la modélisation des systèmes complexes spatiaux : application à l’aménagement urbain de la ville de Métouia”. thesis. Paris 6, Jan. 2015.
- [Has+12] Ines Hassoumi, Christophe Lang, Nicolas Marilleau, Moncef Temani, Khaled Ghedira, and Jean Daniel Zucker. “Toward a Spatially-Centered Approach to Integrate Heterogeneous and Multi-scales Urban Component Models”. en. In: *Advances on Practical Applications of Agents and Multi-Agent Systems*. Ed. by Yves Demazeau, Jörg P. Müller, Juan M. Corchado Rodríguez, and Javier Bajo Pérez. Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, 2012, pp. 81–86.
- [Hjo+20] Arthur Hjorth, Bryan Head, Corey Brady, and Uri Wilensky. “LevelSpace: A NetLogo Extension for Multi-Level Agent-Based Modeling”. In: *Journal of Artificial Societies and Social Simulation* 23.1 (2020), p. 4.
- [HMK18] Elizabeth Hunter, Brian Mac Namee, and John D Kelleher. “A Comparison of Agent-Based Models and Equation Based Models for Infectious Disease Epidemiology.” In: *AICS*. 2018, pp. 33–44.
- [Huy+17] Nghi Quang Huynh, Arnaud Grignard, Hiep Xuan Huynh, Alexis Drogoul, et al. “Coupling equation based models and agent-based models: example of a multi-strains and switch SIR toy model”. In: *EAI Endorsed Transactions on Context-aware Systems and Applications* 4.11 (2017), e1–e1.
- [Huy17] Quang-Nghi Huynh. “CoModels, engineering dynamic compositions of coupled models to support the simulation of complex systems”. en. In: (2017), p. 124.

- [KM27] William Ogilvy Kermack and Anderson G McKendrick. “A contribution to the mathematical theory of epidemics”. In: *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721.
- [Kha+21] Farhan Mohammad Khan, Akshay Kumar, Harish Puppala, Gaurav Kumar, and Rajiv Gupta. “Projecting the criticality of COVID-19 transmission in India using GIS and machine learning methods”. In: *Journal of Safety Science and Resilience* 2.2 (2021), pp. 50–62.
- [LJO05] Jay Larson, Robert Jacob, and Everest Ong. “The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models”. en. In: *The International Journal of High Performance Computing Applications* 19.3 (Aug. 2005), pp. 277–292.
- [Lau06] Kung-Kiu Lau. “Software component models”. In: *Proceedings of the 28th international conference on Software engineering*. 2006, pp. 1081–1082.
- [Loi16] Selmo Tombo Loïc. “modélisation des pratiques agricoles dans les systèmes mixtes de la région centrale de Madagascar”. MA thesis. Montpellier, 34060: SupAgro Montpellier, Sept. 2016.
- [MG98] Pierre Marcenac and Sylvain Giroux. “Geamas: A generic architecture for agent-oriented simulations of complex processes”. In: *Applied Intelligence* 8.3 (1998), p. 247.
- [Mar+08] N. Marilleau, C. Cambier, A. Drogoul, J. -L. Chotte, E. Perrier, and E. Blanchart. “Multiscale MAS modelling to simulate the soil environment: Application to soil ecology”. In: *Simulation Modelling Practice and Theory* 16.7 (Aug. 2008), pp. 736–745.
- [Mar16] Nicolas Marilleau. “Approches distribuées à base d’agents pour modéliser et simuler les systèmes complexes spatialisés”. PhD thesis. Université Pierre & Marie Curie-Paris 6, 2016.
- [MLG18] Nicolas Marilleau, Christophe Lang, and Patrick Giraudoux. “Coupling agent-based with equation-based models to study spatially explicit megapopulation dynamics”. In: *Ecological Modelling* 384 (Sept. 2018), pp. 34–42.
- [Mas+07] D. Masse, C. Cambier, A. Brauman, S. Sall, K. Assigbetse, and J.-L. Chotte. “MIOR: an individual-based model for simulating the spatial patterns of soil organic matter microbial decomposition”. en. In: *European Journal of Soil Science* 58.5 (2007), pp. 1127–1135.
- [May+12] K. U. Mayer, R.T. Amos, S. Molins, and Frederic Gerard. “Reactive transport modeling in variably saturated media with MIN3P: basic model formulation and model enhancements”. In: *Groundwater Reactive Transport Models*. Bentham Science Publishers, 2012, np.

- [Mic07] Fabien Michel. “The IRM4S Model: The Influence/Reaction Principle for Multiagent Based Simulation”. In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*. AAMAS '07. Honolulu, Hawaii: Association for Computing Machinery, 2007.
- [Mor13] Gildas Morvan. “Multi-level agent-based modeling - A literature survey”. In: *arXiv:1205.0561 [cs]* (Nov. 2013). arXiv: 1205.0561.
- [MK17] Gildas Morvan and Yoann Kubera. “On time and consistency in multi-level agent-based simulations”. In: *arXiv:1703.02399 [cs]* (Mar. 2017). arXiv: 1703.02399.
- [MVD10] Gildas Morvan, Alexandre Veremme, and Daniel Dupont. “IRM4MLS: the influence reaction model for multi-level simulation”. In: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 2010, pp. 16–27.
- [Neu10] Martin Neumann. “A Classification of Normative Architectures”. In: *Simulating Interacting Agents and Social Phenomena*. Ed. by Keiki Takadama, Claudio Cioffi-Revilla, and Guillaume Deffuant. Tokyo: Springer Japan, 2010, pp. 3–18.
- [Oss08] Sascha Ossowski. “Coordination in Multi-Agent Systems: Towards a Technology of Agreement”. en. In: *Multiagent System Technologies*. Ed. by Ralph Bergmann, Gabriela Lindemann, Stefan Kirn, and Michal Pěchouček. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 2–12.
- [Pan95] Nicolai S Panikov. “Microbial growth kinetics”. In: (1995).
- [Pan+10] M. Pansu, L. Sarmiento, M. A. Rujano, M. Ablan, D. Acevedo, and P. Bottner. “Modeling organic transformations by microorganisms of soils in six contrasting ecosystems: Validation of the MOMOS model”. en. In: *Global Biogeochemical Cycles* 24.1 (2010).
- [Pay+06a] Denis Payet, Remy Courdier, Nicolas Sebastien, and Tiana Ralambondrainy. “Environment as support for simplification, reuse and integration of processes in spatial MAS”. en. In: *2006 IEEE International Conference on Information Reuse & Integration*. Waikoloa Village, HI, USA: IEEE, Sept. 2006, pp. 127–131.
- [Pay+06b] Denis Payet, Rémy Courdier, Tiana Ralambondrainy, and Nicolas Sébastien. “Le modèle à Temporalité: pour un équilibre entre adéquation et optimisation du temps dans les simulations agent.” In: *JFSMA*. 2006, pp. 63–76.

- [PBR10] E. M. A. Perrier, N. R. A. Bird, and T. B. Rieutord. “Percolation properties of 3-D multiscale pore networks: how connectivity controls soil filtration processes”. In: *Biogeosciences* 7.10 (2010), pp. 3177–3186.
- [PBR00] Edith Perrier, Nigel Bird, and Michel Rieu. “Generalizing the fractal model of soil structure: the pore—solid fractal approach”. en. In: *Developments in Soil Science*. Ed. by Y. Pachepsky, J. W. Crawford, and W. J. Rawls. Vol. 27. Fractals in Soil Science. Elsevier, Jan. 2000, pp. 47–74.
- [PM11] Sébastien Picault and Philippe Mathieu. “An Interaction-Oriented Model for Multi-Scale Simulation”. In: *IJCAI’2011 – Barcelona (Spain) – July, 16-22 2011*. Ed. by T. Walsh. Spain: AAAI Press, 2011, pp. 332–337.
- [PMK10] Sébastien Picault, Philippe Mathieu, and Yoann Kubera. *PADAWAN, un modèle multi-échelles pour la simulation orientée interactions*. Pages: 204. Jan. 2010.
- [QDR09] Gauthier Quesnel, Raphaël Duboz, and Éric Ramat. “The Virtual Laboratory Environment—An operational framework for multi-modelling, simulation and analysis of complex dynamical systems”. In: *Simulation Modelling Practice and Theory* 17.4 (2009), pp. 641–653.
- [Ral+19] Tahina Ralitera, Denis Payet, Nathan Aky, and Rémy Courdier. “The Temporality Model Time Scheduling Approach: A Practical Application”. In: *Multi-Agent-Based Simulation XIX*. Ed. by Paul Davidsson and Harko Verhagen. Cham: Springer International Publishing, 2019, pp. 115–125.
- [Raz17] Kanto Razanamalala. “Priming effect : vers un outil de gestion de la fertilité des sols cultivés à Madagascar”. 2017NSAM0047. PhD thesis. 2017.
- [RVO07] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. “Give agents their artifacts: the A&A approach for engineering working environments in MAS”. In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 2007, pp. 1–3.
- [SST06] Fariba Sadri, Kostas Stathis, and Francesca Toni. “Normative KGP agents”. English. In: *Computational Mathematical Organization Theory* 12.2-3 (2006), pp. 101–126.
- [Sel+15] Bran Selic, Conrad Bock, Steve Cook, Pete Rivett, Tom Rutt, Ed Seidewitz, and Doug Tolbert. “OMG Unified Modeling Language (Version 2.5)”. In: (Mar. 2015).

- [Sho93] Yoav Shoham. “Agent-oriented programming”. In: *Artificial intelligence* 60.1 (1993), pp. 51–92.
- [SCC10] Julien Siebert, Laurent Ciarletta, and Vincent Chevrier. “Agents and artefacts for multiple models co-evolution. Building complex system simulation as a set of interacting models”. en. In: (2010), p. 9.
- [Tai+19a] Patrick Taillandier, Benoit Gaudou, Arnaud Grignard, Quang-Nghi Huynh, Nicolas Marilleau, Philippe Caillou, Damien Philippon, and Alexis Drogoul. “Building, composing and experimenting complex spatial models with the GAMA platform”. In: *GeoInformatica* 23.2 (2019), pp. 299–322.
- [Tai+19b] Patrick Taillandier, Arnaud Grignard, Nicolas Marilleau, Damien Philippon, Quang-Nghi Huynh, Benoit Gaudou, and Alexis Drogoul. “Participatory Modeling and Simulation with the GAMA Platform”. en. In: *Journal of Artificial Societies and Social Simulation* 22.2 (2019), p. 3.
- [TB17] Alicia Tang and Ghusoon Basheer. “Validating a Novel Conflict Resolution Strategy Selection Method (ConfRSSM) Via Multi-Agent Simulation”. In: *International Journal of Advanced Computer Science and Applications* 8 (Sept. 2017), pp. 185–194.
- [Tes+02] Catherine Tessier, Laurent Chaudron, Heinz-Jürgen Müller, and Gerhard Weiss, eds. *Conflicting Agents: Conflict Management in Multi-Agent Systems*. en. Vol. 1. Multiagent Systems, Artificial Societies, And Simulated Organizations. Boston, MA: Springer US, 2002.
- [TW04] Seth Tisue and Uri Wilensky. “Netlogo: A simple environment for modeling complexity”. In: *International conference on complex systems*. Vol. 21. Boston, MA. 2004, pp. 16–21.
- [VSR98] H Van Dyke Parunak, Robert Savit, and Rick L Riolo. “Agent-based modeling vs. equation-based modeling: A case study and users’ guide”. In: *Multi-Agent Systems and Agent-Based Simulation: First International Workshop, MABS’98, Paris, France, July 4-6, 1998. Proceedings 1*. Springer. 1998, pp. 10–25.
- [Ver+16] H. Vereecken, A. Schnepf, J.w. Hopmans, M. Javaux, D. Or, T. Roose, J. Vanderborght, M.h. Young, W. Amelung, M. Aitkenhead, S.d. Allison, S. Assouline, P. Baveye, M. Berli, N. Brüggemann, P. Finke, M. Flury, T. Gaiser, G. Govers, T. Ghezzehei, P. Hallett, H.j. Hendricks Franssen, J. Heppell, R. Horn, J.a. Huisman, D. Jacques, F. Jonard, S. Kollet, F. Lafolie, K. Lamorski, D. Leitner, A. McBratney, B. Minasny, C. Montzka, W. Nowak, Y. Pachepsky, J. Padarian, N. Romano, K. Roth, Y. Rothfuss, E.c. Rowe, A. Schwen, J. Šimůnek, A. Tiktak, J. Van Dam, S.e.a.t.m. van der Zee, H.j. Vogel, J.a. Vrugt,

- T. Wöhling, and I.m. Young. “Modeling Soil Processes: Review, Key Challenges, and New Perspectives”. en. In: *Vadose Zone Journal* 15.5 (2016), vzt2015.09.0131.
- [WG01] Gabriel Wainer and Norbert Giambiasi. “Timed Cell-DEVS: modeling and simulation of cell spaces”. In: *Discrete event modeling and simulation technologies*. Springer, 2001, pp. 187–214.
- [WC01] Michael Wooldridge and Paolo Ciancarini. “Agent-oriented software engineering: The state of the art”. In: *Lecture notes in computer science* (2001), pp. 1–28.
- [ZV93] B.P. Zeigler and S. Vahie. “Devs Formalism and Methodology: Unity of Conception/diversity of Application”. In: *Proceedings of 1993 Winter Simulation Conference - (WSC '93)*. Dec. 1993, pp. 573–579.