



# Pavages réguliers et modélisation des dynamiques spatiales à base de graphes d'interaction : conception, implémentation, application

Mathieu Castets

## ► To cite this version:

Mathieu Castets. Pavages réguliers et modélisation des dynamiques spatiales à base de graphes d'interaction : conception, implémentation, application. Modélisation et simulation. Université de Montpellier, 2015. Français. <tel-01315063>

**HAL Id: tel-01315063**

**<https://hal.archives-ouvertes.fr/tel-01315063>**

Submitted on 12 May 2016

**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.

# THÈSE

Pour obtenir le grade de  
Docteur

Délivré par l'Université de Montpellier

Préparée au sein de l'école doctorale **I2S**  
Et de l'unité de recherche **TETIS**

Spécialité: **Informatique**

Présentée par **Mathieu Castets**

**Pavages réguliers et  
modélisation des dynamiques  
spatiales à base de graphes  
d'interaction : conception,  
implémentation, application**

Soutenue le 15 décembre 2015 devant le jury composé de

Mr. Pascal PONCELET	Université de Montpellier	Directeur de thèse
Mme Florence LE BER	ENGEEES	Rapportrice
Mr. Rémy COURDIER	Université de la Réunion	Rapporteur
Mme Marianne HUCHARD	Université de Montpellier	Examinatrice
Mr. Christophe PROISY	IRD UMR AMAP	Examineur
Mr. Danny LO SEEN	CIRAD UMR TETIS	Encadrant
Mr. Pascal DEGENNE	CIRAD UMR TETIS	Invité









## Résumé

La modélisation et la simulation de dynamiques spatiales, en particulier pour l'étude de l'évolution de paysages ou de problématiques environnementales pose la question de l'intégration des différentes formes de représentation de l'espace au sein d'un même modèle. Ocelet est une approche de modélisation de dynamiques spatiales basée sur le concept original de graphe d'interaction. Le graphe porte à la fois la structure d'une relation entre entités d'un modèle et la sémantique décrivant son évolution. Les relations entre entités spatiales sont ici traduites en graphes d'interactions et ce sont ces graphes que l'on fait évoluer lors d'une simulation. Les concepts à la base d'Ocelet peuvent potentiellement manipuler les deux formes de représentation spatiale connues, celle aux contours définis (format vecteur) ou la discrétisation en grille régulière (format raster). Le format vecteur est déjà intégré dans la première version d'Ocelet. L'intégration du format raster et la combinaison des deux restaient à étudier et à réaliser. L'objectif de la thèse est d'abord étudier les problématiques liées à l'intégration des champs continus et leur représentation discrétisée en pavage régulier, à la fois dans le langage Ocelet et dans les concepts sur lesquels il repose. Il a fallu notamment prendre en compte les aspects dynamiques de cette intégration, et étudier les transitions entre données géographiques de différentes formes et graphe d'interactions à l'aide de concepts formalisés. Il s'est agi ensuite de réaliser l'implémentation de ces concepts dans la plateforme de modélisation Ocelet, en adaptant à la fois son compilateur et son moteur d'exécution. Enfin, ces nouveaux concepts et outils ont été mis à l'épreuve dans trois cas d'application très différents : deux modèles sur l'île de la Réunion, le premier simulant le ruissellement dans le bassin versant de la Ravine Saint Gilles s'écoulant vers la Côte Ouest de l'île, l'autre simulant la diffusion de plantes invasives dans les plaines des hauts à l'intérieur du Parc National de La Réunion. Le dernier cas décrit la spatialisation d'un modèle de culture et est appliqué ici pour simuler les rendements de cultures céréalières sur l'ensemble de l'Afrique de l'Ouest, dans le contexte d'un système d'alerte précoce de suivi des cultures à l'échelle régionale.

## Summary

The modelling and simulation of spatial dynamics, particularly for studying landscape changes or environmental issues, raises the question of integrating different forms of spatial representation within the same model. Ocelet is an approach for modelling spatial dynamics based on the original concept of interaction graph. Such a graph holds both the structure of a relation between entities of a model and the semantics describing its evolution. The relationships between spatial entities are here translated into interaction graphs and these graphs are made to evolve during a simulation. The concepts on which Ocelet is based can potentially handle two known forms of spatial representation : shapes with contours (vector format) or regular grid cells (raster). The vector format is already integrated in the first version of Ocelet. The integration of raster and the combination of the two remained to be studied and carried out. The aim of the thesis is to first study the issues related to the integration of continuous fields and their representation by regular tiling, both in the Ocelet language and the concepts on which it is based. The dynamic aspects of this integration had to be taken into account and transitions between different forms of geographic data and interaction graphs had to be studied in the light of the concepts formalized. The concepts were then implemented in the Ocelet modelling platform, with the adaptation of both its compiler and runtime. Finally, these new concepts and tools were tested in three very different cases : two models on Reunion Island, the first simulating runoff in Ravine Saint Gilles watershed in the West Coast of the island, the other simulating the spread of invasive plants in the high plains inside the Reunion National Park. The last case describes the spatialisation of a crop model and is applied here to simulate the cereal crop yields in West Africa, in the context of an early warning system for regional crop monitoring.

## Remerciements

Je tiens particulièrement à remercier Danny Lo Seen et Pascal Degenne, ces trois années sur la modélisation des dynamiques spatiales ont été passionnantes, on ne peut espérer meilleur sujet. Je vous remercie de m'avoir accordé une grande liberté scientifique, d'avoir favorisé des discussions sur des thématiques qui m'étaient inconnues ainsi que votre accompagnement et votre soutien tout le long de cette thèse. J'espère pouvoir continuer à travailler à vos côtés.

Je tiens à remercier mon directeur de thèse Pascal Poncelet, pour la confiance qu'il m'a accordée, ses conseils avisés et son encadrement durant ces trois années.

Merci à Florence Le Ber et Rémy Courdier pour avoir accepté d'évaluer mes travaux de recherches en tant que rapporteurs, ainsi que Marianne Huchard et Christophe Proisy pour avoir accepté d'être examinateurs. Je vous remercie, pour l'intérêt que vous avez porté à mes travaux.

Un remerciement particulier pour Cédric Gaucherel qui m'a ouvert la voie de la recherche et de la science il y a de cela cinq ans et les discussions enrichissantes que nous avons (aurons) pu avoir.

Je tiens à remercier tous les collègues de la maison de la télédétection qui font de cette unité un espace de convivialité et de travail exceptionnels. Le directeur de l'UMR TETIS Jean-Philippe Tonneau, mon collègue de bureau Christian qui m'a supporté (ou l'inverse !) pendant trois ans, Agnès, Valou, Anja, Louise, Camille, Xavier, Bastien et Isa. Merci.

Merci à mes parents pouvoir m'avoir encouragé durant cette thèse et de m'avoir donné l'envie de travailler dans le domaine de la recherche.

J'exprime toute ma gratitude à mes amis, pour leurs soutiens et les moments épiques passés ensemble : Romuald, Eric, Franck, Micka et Julien ainsi qu'à Jules, GroB, Yann, Vincent, Sophia, Jimmy et Diane.

Merci pour l'accueil chaleureux à l'île de la Réunion par Daniel et l'équipe Artists : Pierre, Elodie, JB.

Je tiens enfin à remercier les financeurs de mes travaux, l'Agence Nationale de la Recherche à travers le projet DESCARTES (ANR 11-AGRO-002-01 Descartes <http://www.projet-descartes.fr/>). Ce travail a aussi bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'avenir pour le projet GEOSUD portant la référence ANR-10-EQPX-20.







# Table des matières

<b>1</b>	<b>Introduction générale</b>	<b>15</b>
1.1	Objectifs de la thèse . . . . .	17
1.2	Contributions . . . . .	18
1.3	Plan de la thèse . . . . .	20
<b>2</b>	<b>Modélisation des dynamiques spatiales</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Modélisation des systèmes . . . . .	24
2.3	Différentes approches de modélisation . . . . .	25
2.3.1	Dynamique des systèmes . . . . .	25
2.3.2	Modélisation Individus-centrée . . . . .	26
2.3.3	Autres . . . . .	29
2.4	Approche Ocelet . . . . .	30
2.5	Utilisation de graphes dans la modélisation spatiale . . . . .	31
2.6	Discussion . . . . .	33
<b>3</b>	<b>Conception</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Définitions . . . . .	36
3.2.1	Elements de la théorie des graphes . . . . .	37
3.2.2	Concepts généraux d'Ocelet . . . . .	39
3.2.3	Concepts du langage Ocelet . . . . .	40
3.2.4	Représentation spatiale par objets vecteurs . . . . .	44
3.3	Le format Raster . . . . .	46
3.4	Pavages réguliers . . . . .	48
3.5	Cellules . . . . .	51
3.6	Pavages réguliers et graphes d'interaction . . . . .	52
3.6.1	Entités cellulaires et graphe d'interaction . . . . .	53
3.6.2	Entités cellulaires, géométriques, sans représentation spatiale et graphe d'interaction . . . . .	55
3.6.3	Entités cellulaires de différents types et graphe d'interaction . . . . .	59
3.7	Conclusion . . . . .	60

<b>4</b>	<b>Implémentation</b>	<b>63</b>
4.1	Introduction . . . . .	63
4.2	Développements sur le langage . . . . .	64
4.3	Gestion des données matricielles . . . . .	65
4.4	Lecture et écriture des données matricielles . . . . .	66
4.4.1	Image raster . . . . .	66
4.4.2	Série temporelle d'images . . . . .	67
4.5	Implémentation et gestion de graphes . . . . .	69
4.5.1	Graphe entre entités cellulaires de même type . . . . .	71
4.5.2	Graphes entre entités cellulaires et entités géométriques ou sans représentation spatiale . . . . .	81
4.5.3	Graphes entre entités cellulaires de types différents . . . . .	86
4.6	Conclusion . . . . .	90
<b>5</b>	<b>Cas d'application</b>	<b>91</b>
5.1	Introduction . . . . .	92
5.2	Modèle de ruissellement . . . . .	93
5.2.1	Objectifs du modèle . . . . .	93
5.2.2	Entités, variables d'état . . . . .	94
5.2.3	Processus et scénarios . . . . .	96
5.2.4	Principes de conception . . . . .	97
5.2.5	Initialisation . . . . .	100
5.2.6	Données d'entrée . . . . .	101
5.2.7	Résultats et discussion . . . . .	103
5.3	Modèle de diffusion de plantes invasives . . . . .	105
5.3.1	Objectifs du modèle . . . . .	106
5.3.2	Entités, variables d'état . . . . .	106
5.3.3	Processus et scénarios . . . . .	107
5.3.4	Principes de conception . . . . .	108
5.3.5	Initialisation . . . . .	110
5.3.6	Données d'entrée . . . . .	111
5.3.7	Résultats et discussion . . . . .	112
5.4	Système d'alerte précoce en sécurité alimentaire . . . . .	115
5.4.1	Objectifs du modèle . . . . .	115
5.4.2	Entités, variables d'état . . . . .	116
5.4.3	Processus et scénarios . . . . .	117
5.4.4	Principes de conception . . . . .	118
5.4.5	Initialisation . . . . .	120
5.4.6	Données d'entrée . . . . .	121
5.4.7	Résultats et discussion . . . . .	122
5.5	Conclusion sur les cas d'application . . . . .	128

<b>6 Conclusion générale</b>	<b>131</b>
6.1 Le formalisme de graphe d'interaction étendu aux pavages réguliers .	132
6.2 La plateforme de Ocelet enrichie de la capacité à incorporer de la donnée raster . . . . .	133
6.3 Des concepts et outils mis à l'épreuve dans différents cas d'application	135
6.4 Perspectives et futurs développements . . . . .	137
<b>Appendices</b>	<b>159</b>
<b>A Environnement de développement d'Ocelet (OMP)</b>	<b>161</b>
<b>B Principales Méthodes</b>	<b>163</b>
B.1 Méthodes de la propriété "Cell" . . . . .	163
B.2 Méthodes des graphes d'interaction . . . . .	164
B.3 Méthodes des datafacers "RasterFile" et "TemporalSeriesFile" . . . .	165
<b>C Intégration de plusieurs formes de représentations spatiales dans un modèle de simulation</b>	<b>167</b>
<b>D Integrating raster and vector spatial representations with interac- tion graphs for multi-scale environmental simulations</b>	<b>173</b>



# Chapitre 1

## Introduction générale

De nombreux défis sociétaux aujourd'hui (e.g. l'atténuation et l'adaptation aux changements climatiques, la sécurité alimentaire et l'agriculture durable, la préservation de la biodiversité et des ressources naturelles) soulèvent des problèmes qui relèvent des sciences de l'environnement mais qui doivent être étudiés en lien avec les activités humaines (IPCC, 2014 ; FAO FIDA et PAM, 2014 ; SCDB, 2014). Le terme "Land Change Science"<sup>1</sup> (LCS) est maintenant utilisé pour désigner cette science qui vise à comprendre les dynamiques humaines et environnementales à l'origine des changements d'occupation et d'utilisation des terres et qui nécessitent l'intégration des sciences sociales, naturelles et de l'information géographique (Rindfuss et al., 2004 ; Magliocca et al., 2015). Eminemment complexes, les problèmes soulevés ont pour caractéristique de ne pouvoir être traités de manière sectorielle. Ils requièrent plutôt une approche interdisciplinaire car nécessitant l'échange de connaissances et de méthodes et la coopération entre champs disciplinaires (Morin, 1994). La modélisation et l'utilisation de modèles concernent maintenant la plupart des disciplines scientifiques et deviennent un moyen de dialogue entre disciplines. Ainsi, "la modélisation devient une méthodologie de plus en plus incontournable dans le processus d'acquisition des connaissances." (Schmidt-Lainé and Pavé, 2002).

De manière générale un modèle est une représentation plus ou moins simplifiée d'un objet ou d'un système. Par exemple, une carte géographique est un modèle, et l'échelle de la carte, par la sélection des informations qui doivent y figurer, détermine le niveau de simplification. Il faut donc déterminer les motivations et les buts du modèle à établir en ayant conscience qu'il reste une simplification. Trois caractéristiques principales des modèles sont relevées par Popper (1973) : i) il doit avoir une ressemblance avec le système réel ; ii) il doit constituer une simplification du système réel et iii) il représente une idéalisation du système réel. Les modèles peuvent avoir différentes finalités. Ils peuvent être une représentation de l'existant en termes de visualisation, telle une carte ou une maquette. Ils peuvent servir à reproduire des

---

1. Terme pour lequel il n'existe pas encore de traduction unanime en français.

phénomènes d'un système dans le but de les comprendre ; la description de processus et l'observation de résultats pouvant amener à mieux cerner certains phénomènes (Kuipers, 1994). Ils peuvent aussi avoir pour but de faire une projection dans le futur (Minsky, 1988). Enfin la méthode expérimentale utilise parfois des modèles pour tester la validité d'une hypothèse lorsque les expérimentations ne peuvent être envisagées en grandeur nature (Develay, 1989 ; Grelley, 2012).

On parle de modélisation ou simulation informatique lorsque les modèles sont construits comme des programmes informatiques pour permettre la réalisation d'expériences virtuelles. Cette pratique s'est fortement répandue ces dernières décennies dans tous les domaines de la recherche scientifique, notamment en sciences de l'environnement, comme l'atteste le nombre de publications dans les revues scientifiques spécialisées telles que : *Ecological Modelling*, *Environmental Modelling and Software* ou *Environmental Informatics*. Schmidt-Lainé and Pavé (2002) relevaient déjà que l'informatique avait largement contribué à l'extension de la modélisation, d'abord par ses capacités de calcul numérique, mais de plus en plus en apportant ses propres méthodes de modélisation telles que les systèmes multi-agents (Ferber, 1995), les réseaux de neurones artificiels (Touzet, 1992), les algorithmes génétiques (Mitchell et al., 1996), les automates cellulaires (Von Neumann, 1966) ou les langages formels (Lindenmayer and Rozenberg, 1972).

Dans une revue récente du domaine de la modélisation de dynamiques paysagères Gaucherel et al. (2014) appelle au développement d'un vrai cadre théorique cohérent pour assoir la discipline. Malgré l'absence de ce cadre théorique, de nombreuses approches de modélisation se sont développées pour lesquelles il n'existe pas vraiment de classification satisfaisante. Il est usuel de plutôt donner leurs principales caractéristiques : dynamique ou statique ; spatialement implicite ou explicite ; analytique ou de simulation ; temps continu ou temps discret ; mécaniste, empirique ou à base de processus. Certaines approches ont réussi à émerger en se basant sur des communautés très actives d'utilisateurs, par exemple : SME (Maxwell and Costanza, 1997) ; CORMAS (Bousquet et al., 1998) ; NetLogo (Wilensky, 1999) ; SELES (Fall and Fall, 2001) ; DYPAL (Gaucherel et al., 2012) ; GAMA (Grignard et al., 2013). Les modèles développés en Land Change Science, les "Land Change Models" sont généralement dynamiques et spatialement explicites : la configuration spatiale de l'occupation et de l'utilisation du sol affecte – et est affectée par – les processus à l'œuvre qui couplent dynamiques environnementales et dynamiques humaines à différentes échelles de temps et d'espace (Treml et al., 2007).

Dans sa revue des modèles de changement d'utilisation des terres, Agarwal et al. (2002) aborde la complexité spatiale des modèles en distinguant parmi les modèles spatialement explicites ceux qui sont spatialement représentatifs et ceux qui sont spatialement interactifs. Les premiers sont capables de bien représenter les entités géographiques dans l'espace mais celles-ci ne peuvent interagir, comme peuvent le

faire les entités géographiques dans les modèles spatialement interactifs. La possibilité de faire interagir les entités est fortement contrainte par les formes de représentation spatiales (modes "raster" : cellules et "vecteur" : points, lignes, polygones) adoptées par un modèle donné. Dans la grande majorité des cas, l'interaction spatiale s'utilise en mode "raster" et résulte du voisinage implicite des cellules qui peuvent s'influencer mutuellement. Les interactions entre d'autres formes de représentation (lignes, polygones) ne sont pas implicites et doivent être explicitées par le modélisateur. Ceci explique que peu de modèles spatialement interactifs fonctionnent en mode "vecteur".

L'approche Ocelet a été développée avec comme motivation de donner au modélisateur une plus grande liberté pour décrire les structures, en particulier spatiales, sur lesquelles des interactions peuvent porter (Degenne, 2012). Le concept de graphe d'interaction, qui est au cœur d'Ocelet, a été élaboré à partir de la structure de graphe, pour tirer parti de la flexibilité qu'elle procure, et en l'enrichissant de la sémantique appropriée pour pouvoir exprimer de manière unique des relations spatiales, fonctionnelles, sociales ou hiérarchiques. Les modèles sont construits à l'aide d'un langage métier (Ocelet) dans un environnement de développement de modèles de dynamiques spatiales (OMP : Ocelet Modelling Platform). La version 1.0 disponible en ligne à ce jour (septembre 2015) ne permet de modéliser qu'avec des entités dont la représentation spatiale est une géométrie, "vecteur". Même si dès le départ le fait de ne pas imposer un choix a priori du mode de représentation de l'espace était jugé important (Degenne et al., 2009). La possibilité de construire des modèles en utilisant les deux modes "raster" et "vecteur" avec le même formalisme Ocelet à base de graphes d'interaction a vite soulevé des questions méthodologiques qui demandaient à être traitées spécifiquement.

## 1.1 Objectifs de la thèse

1. Concevoir l'intégration du mode raster dans le langage Ocelet et l'extension du formalisme de graphe d'interaction aux espaces découpés régulièrement : par un pavage régulier ;
2. Implémenter dans la plateforme de modélisation Ocelet les développements logiciels nécessaires permettant de construire des modèles où interagissent des entités spatiales "vecteur" et "raster" ;
3. Réaliser des modèles appliqués à différents domaines thématiques pour vérifier la cohérence conceptuelle et l'efficacité d'exécution de l'application sur des cas concrets.

## 1.2 Contributions

Pour répondre aux objectifs de la thèse, les travaux sont organisés en ces différentes contributions :

**Conception** : L'intégration d'une représentation spatiale supplémentaire (pavage régulier) dans le formalisme d'Ocelet doit se faire en respectant les concepts présents et déjà utilisés avec une représentation spatiale vectorielle. En particulier, nous souhaitons pouvoir utiliser des entités dans des graphes d'interaction quelles que soient leurs représentations spatiales, et pouvoir les utiliser ensemble dans un même modèle.

En tenant compte de la définition même d'un pavage régulier, notamment sur sa constitution en éléments de même forme, nous proposons de donner la possibilité à une entité d'avoir comme propriété une nouvelle forme de représentation spatiale qui correspond à un élément du pavage : la cellule. Un pavage régulier est alors composé d'un ensemble d'entités cellulaires. Nous avons ensuite déterminé les différentes structures de graphes d'interaction induites par la représentation spatiale d'une entité par une cellule :

- Un graphe d'interaction qui tient compte de la structure régulière du pavage. Les entités cellulaires d'un même type peuvent alors interagir par voisinage ou distance de voisinage.
- Un graphe d'interaction entre des entités cellulaires et des entités non cellulaires. Dans le cas où les entités non cellulaires sont représentées spatialement par du vecteur, la structure du graphe d'interaction tient alors compte de la correspondance spatiale entre ces entités et les entités cellulaires.
- Un graphe d'interaction entre des entités cellulaires de différents types, permettant de lier des entités avec des cellules de formes ou de résolution différentes.

**Implémentation** : Une fois le concept de graphe d'interaction étendu à la représentation spatiale des entités par des pavages réguliers, il s'agissait de l'implémenter dans le langage Ocelet et dans le moteur d'exécution. Nous proposons une simulation de parcours de graphe qui permet de ne pas contenir le graphe complet en mémoire, en raison de la taille potentiellement très importante du graphe, exprimée en nombre de sommets et d'arêtes. Nous avons ainsi proposé des parcours simulés pour chaque structure de graphe d'interaction mentionnées ci-dessus :

- Le parcours simulé d'un graphe d'interaction sur des entités cellulaires de même type. Les arêtes sont simulées en s'appuyant sur les graphes de Cayley (Cayley, 1878). Chaque arête étant ainsi simulée et les positions des sommets déduites des positions des cellules sur le pavage.
- Le parcours simulé d'un graphe entre entités cellulaires et entités non cellulaires repose sur une structure qui permet d'indexer la position des entités cellulaires et de faire le lien avec les autres entités. Lorsque les entités non cellulaires ont une géométrie, cette structure d'indexation est construite par correspondance

spatiale entre la géométrie et les cellules. Dans l'autre cas, la construction est explicitée par le modélisateur.

- Le parcours simulé d'un graphe entre entités cellulaires de différents types : le parcours des arêtes est simulé par correspondance spatiale entre les entités cellulaires de formes ou résolutions différentes.

Les fonctions d'interaction peuvent ainsi être appliquées sur chaque arête simulée. Des implémentations supplémentaires ont dû tenir compte de la simultanéité de l'application de ces fonctions lors des parcours simulés. Nous avons ensuite considéré l'aspect gestion des données en permettant d'intégrer des images au format raster ainsi que des séries temporelles d'images. Les implémentations ont tenu compte du format matriciel des données raster pour gérer les entités cellulaires dans le langage : ces entités, contrairement aux autres, ne sont plus instanciées, les propriétés de toutes ces entités sont gérées au format matriciel. L'accès aux propriétés se fait alors suivant la position de leurs cellules dans la matrice.

**Cas d'application :** Les concepts développés et l'implémentation effectuée ont été testés sur trois cas d'applications de modélisation des dynamiques spatiales :

- Un modèle de ruissellement sur le bassin versant de Saint Gilles, à l'île de la Réunion qui met en avant la gestion combinée de représentations "vecteur" et "raster". Des processus de ruissellement sont définis sur des entités cellulaires, traduisant des flux de proche en proche ainsi que sur des entités géométriques linéaires traduisant des flux sur les routes et les ravines. Nous avons présenté une comparaison des débits observés et simulés pour deux simulations : la première lors de l'épisode cyclonique Dina en 2002, et l'autre, sur une période plus étalée en 2009.
- Un modèle sur la diffusion d'une plante invasive à l'île de la Réunion : l'*Ulex europaeus*. Une diffusion de proche en proche a été appliquée sur des entités cellulaires qui se base sur des indices issus d'images de télédétection. Nous présentons des résultats visuels de la simulation de la diffusion de l'espèce entre l'année 2004 et l'année 2012.
- Un modèle permettant de spatialiser la simulation des rendements de cultures céréalières sur l'Afrique de l'Ouest. Le modèle de culture SARRA-H simule actuellement les bilans hydrique et carboné, et la phénologie d'une culture à la parcelle. Nous testons ici la possibilité d'utiliser Ocelet pour déployer le modèle sur une grande zone découpée en cellules, en prenant en compte l'hétérogénéité de la zone dans les paramètres d'entrée du modèle. Le modèle spatialisé est en cours d'intégration à un système d'alerte précoce (SAP) au Centre régional AGRHYMET (Niger). La publication du bulletin mensuel sur la satisfaction des besoins en eau du 30 juin 2015 fait mention des résultats issus de l'utilisation de ce modèle par nos partenaires d'AGRHYMET.

## 1.3 Plan de la thèse

Le corps de cette thèse est organisé en quatre chapitres :

### **Chapitre 2 : Modélisation des dynamiques spatiales**

Ce chapitre situe l'approche Ocelet par rapport aux grandes familles de méthodes et d'outils existants dans le domaine de la modélisation de dynamiques spatiales. Nous décrivons comment les graphes d'interaction peuvent être utilisés pour modéliser les systèmes dynamiques composés d'entités, dont certaines spatiales, qui sont en interaction.

### **Chapitre 3 : Conception de l'intégration de pavages réguliers dans Ocelet**

Dans ce chapitre, nous abordons conceptuellement l'intégration d'une représentation spatiale régulière : par pavages réguliers, et de tenir compte de cette structure sous le formalisme des graphes d'interaction. Le format de données le plus connu de ce type de représentation est le format raster, composé d'éléments qui ont la même forme : les pixels, qui dans ce cas ont une forme carrée. En tenant compte de la gestion des données de ce format, nous proposons d'inclure d'autres formes qui peuvent constituer un pavage régulier : les triangles équilatéraux et les hexagones réguliers. La prise en compte des pavages réguliers fait apparaître plusieurs structures de graphes implicites que nous proposons d'étudier : a) les graphes de voisinage entre des entités représentées spatialement par des cellules, b) les graphes entre des entités représentées spatialement par des cellules et des entités géométriques "vecteur", ou sans représentation spatiale et c) les graphes entre entités représentées par des cellules de résolutions ou de formes différentes couvrant une même zone géographique.

### **Chapitre 4 : Implémentation du Raster dans Ocelet**

Ce chapitre concerne les implémentations opérées dans le moteur d'exécution d'Ocelet pour rendre ces types de graphes manipulables dans l'environnement de développement d'Ocelet (OMP). Pour tenir compte du formalisme du langage, ainsi que de la taille de plus en plus grande des images raster (e.g. images satellite), des algorithmes spécifiques de parcours de graphes d'entités représentées spatialement par une cellule ont été développés. Pour chaque type de graphes énumérés précédemment, les algorithmes employés ainsi que des exemples d'utilisation avec le langage sont décrits.

### **Chapitre 5 : Cas d'application**

Les nouvelles capacités de modélisation ont été testées sur trois cas d'applications décrits dans ce chapitre. Elles sont issues de domaines thématiques très différents :

i) la modélisation du ruissellement de l'eau dans le bassin versant de Saint Gilles situé à l'île de la Réunion, ii) la modélisation de la diffusion d'une espèce invasive de plante dans les hauts des plaines de l'Ouest de l'île de la Réunion et, iii) la spatialisation d'un modèle de simulation des rendements de cultures céréalières en Afrique de l'Ouest, une catégorie de modèle qui est qualifiée de Système d'Alerte Précoce (SAP).

Enfin, nous concluons en commentant les apports de ce travail en termes de capacités nouvelles d'expressions par le modélisateur, et nous présentons les perspectives associées aux concepts, aux implémentations et aux cas d'application proposés.



# Chapitre 2

## Modélisation des dynamiques spatiales

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>23</b>
<b>2.2</b>	<b>Modélisation des systèmes</b>	<b>24</b>
<b>2.3</b>	<b>Différentes approches de modélisation</b>	<b>25</b>
2.3.1	Dynamique des systèmes	25
2.3.2	Modélisation Individus-centrée	26
2.3.3	Autres	29
<b>2.4</b>	<b>Approche Ocelet</b>	<b>30</b>
<b>2.5</b>	<b>Utilisation de graphes dans la modélisation spatiale</b>	<b>31</b>
<b>2.6</b>	<b>Discussion</b>	<b>33</b>

---

### 2.1 Introduction

Parmi les différentes finalités de la modélisation informatique, celle de support de processus d'acquisition et de formalisation de connaissances occupe une place importante dans l'activité de recherche. La modélisation est ainsi associée à des expérimentations, des observations, et des simulations pour permettre de confirmer certaines connaissances ou de réfuter certaines hypothèses sur le système étudié (Pavé, 2005). Comme nous l'avons indiqué en introduction, dans cette thèse, nous nous intéressons plus particulièrement aux modèles de dynamiques spatiales où sont représentés les processus à l'œuvre dans un espace géographique et qui participent à engendrer les changements observés que l'on veut étudier. Les modèles de dynamiques spatiales prennent en compte les changements référant à l'espace selon certaines temporalités (Ruas and Sanders, 2015). Dans ce chapitre nous revenons sur la problématique de la modélisation des dynamiques spatiales et rappelons les concepts associés à l'approche systémique. Cette présentation nous permet de mettre

en évidence les critères à prendre en compte dans notre contexte. Dans la section 2.3, nous décrivons les principales approches existantes pour modéliser les dynamiques spatiales. Récemment une nouvelle approche, Ocelet, a été proposée pour modéliser les dynamiques spatiales au travers de graphes à structure dynamique. Cette approche étant particulièrement adaptée à notre problématique, et la suite de cette thèse étant basée sur elle, nous introduisons dans les sections 2.4 et 2.5 ses principales caractéristiques (une présentation plus détaillée de l'approche est proposée dans le chapitre 3). Enfin, nous concluons ce chapitre par une discussion.

## 2.2 Modélisation des systèmes

Au départ proche du concept d'ensemble, le sens de "système" a évolué dans de nombreuses disciplines scientifiques pour maintenant exprimer l'idée que l'ensemble constituait un tout et ne pouvait se réduire à être étudié au travers de ses parties prises indépendamment. C'est le biologiste Von Bertalanffy (1950) qui a proposé la Théorie Générale des Systèmes dans laquelle il postule qu'un système doit être décrit dans son ensemble, en prenant en compte l'organisation des éléments qui le constituent ainsi que leurs interactions. C'est son travail qui est considéré encore aujourd'hui comme le fondement de l'**approche systémique** (Le Moigne, 2006). Durand (2013) propose en quatre modules un processus de modélisation systémique, pragmatique et souple, qui peut être utilisé dans de nombreux domaines :

1. *Définir le projet.* Ce module implique de répondre à deux questions : Quelle est la finalité de l'exercice de modélisation ? Quelles sont les frontières à fixer à cet exercice ? Cela permet d'être clair sur les objectifs à atteindre, et de décider ce qui est considéré comme faisant partie du système ou comme étant extérieur au système.
2. *Définir le système.* Identifier et définir les éléments qui vont constituer le modèle. Décrire comment les éléments sont reliés entre eux : quels éléments interagissant avec qui. Caractériser les interactions, et décider si elles peuvent être à sens unique ou à double sens. On obtient ainsi la structure du système.
3. *Déterminer le comportement du modèle.* Il s'agit ici de décrire le fonctionnement du système, en définissant les caractéristiques des éléments au regard de leur rôle dans le système, et les interactions qui vont faire évoluer le système. Les éléments peuvent être invariants (éléments considérés comme stable dans le modèle), variables (capable de changer selon les processus prévus dans le modèle) ou être des contraintes internes ou externes au système. Ce module contient la phase de validation qui permet de vérifier si le modèle reflète suffisamment la réalité pour être utilisé pour les objectifs définis dans le premier module.
4. *Utiliser le modèle.* C'est la phase de simulation, où en faisant varier les paramètres et les contraintes, il est permis de tirer des enseignements sur le système. Elle repose généralement sur la définition de scénarios (e.g. optimiste,

pessimiste, passif, volontariste) dont les simulations peuvent aider à la prise de décision.

Dans notre contexte de modélisation de dynamiques spatiales il est nécessaire de mieux définir ou de prendre en compte un certain nombre de contraintes, notamment dans les modules 2, 3 et 4. En effet, ces modules doivent tenir compte de la représentation spatiale et dynamique des éléments du système, des relations et interactions entre eux, mais également des scénarios lors des simulations. Pour tenir compte de la représentation spatiale, le modèle doit attribuer une forme de représentation à ses éléments. Parmi les formats de représentation, les plus utilisés sont le format raster et le format vecteur :

1. Pour le format raster, l'espace est découpé en éléments réguliers (comme des pixels pour une image) en se basant sur des données matricielles. Les éléments d'un modèle sont alors représentés par une cellule.
2. Pour le format vecteur, l'espace est représenté par des "objets". Les éléments d'un modèle sont alors représentés par des géométries.

Dans la section suivante, nous présentons les grandes familles d'approches de modélisation informatique qui remplissent certaines des conditions pour réaliser des modèles de dynamiques spatiales selon une approche systémique.

## 2.3 Différentes approches de modélisation

Les moyens de modélisation en sciences de l'environnement ont fait l'objet d'articles de revue par le passé (Baker, 1989 ; Ford, 1999 ; Agarwal et al., 2002 ; Bousquet and Le Page, 2004 ; Van Ittersum et al., 2008) et plus récemment (Degenne, 2012 ; Kelly et al., 2013 ; Laniak et al., 2013). Dans la suite de cette section, nous présentons les grandes familles de "méthodes" de modélisation en précisant leur fondement ainsi que les outils permettant de les mettre en œuvre. Nous abordons ainsi les méthodes qui relèvent de la "Dynamique des systèmes", les méthodes "Individus-centrées" et les "Autres".

### 2.3.1 Dynamique des systèmes

Après avoir élargi le champ d'application de la systémique à la dynamique industrielle dans les années 1960, Forrester (1968) élabore la dynamique générale des systèmes. Ses travaux permettent la formalisation mathématique d'un système dynamique et introduisent des concepts comme boucles de rétroaction, flux (évolution temporelle d'une variable du système), réservoirs (stocks) où s'accumulent ces variables, et effets retard. La construction d'un modèle est aidée par l'utilisation d'un formalisme graphique où les stocks, flux et boucles de rétroaction sont représentés par des diagrammes à base d'icônes comme le présente la figure 2.1. On parle alors de modèles "stock-flux". Avec la démocratisation des moyens de calculs dans les

années 1990, des outils sont développés pour modéliser selon ce paradigme, en permettant la résolution numérique des équations mathématiques décrivant les systèmes dynamiques. Les plus connus de ces outils sont Stella (Costanza, 1987) et Vensim (Eberlein and Peterson, 1992). Longtemps utilisés dans les milieux académiques et de la recherche, ce sont maintenant des logiciels commerciaux (iThink/Stella®, Vensim®) parmi d'autres (e.g. Powersim, InsightMaker).

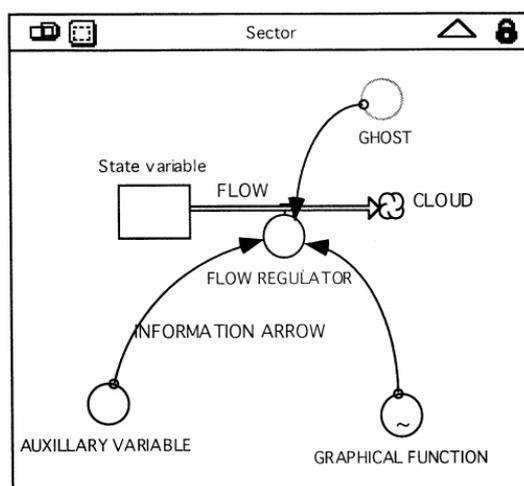


FIGURE 2.1 – Structure composée d'éléments permettant l'échange de matière par flux avec Stella (Costanza and Gottlieb, 1998).

Modéliser des dynamiques spatiales selon les principes de la dynamique des systèmes se heurte à la difficulté de décomposer un système incluant un espace géographique en éléments ou sous-systèmes répartis dans cet espace. Une solution proposée par le Spatial Modelling Environment (SME), (Costanza and Voinov, 2004), est de décomposer l'espace en cellules, d'inclure dans chaque cellule un modèle de type stock-flux, et de permettre aux cellules voisines d'échanger des flux. Une autre solution est proposée dans Simile (Muetzelfeldt and Massheder, 2003) où il est possible de décomposer le système en éléments contenant chacun un modèle stock-flux, mais en incluant des fonctions de désagrégation. Même si Simile n'est pas considéré comme pouvant faire des modèles "spatialement explicites", il est permis de donner aux éléments du modèle des coordonnées de position dans une représentation de l'espace.

### 2.3.2 Modélisation Individus-centrée

A l'opposé de la dynamique des systèmes considérée comme approche "descendante" où un système est décrit dans son ensemble, puis éventuellement décomposé en sous-systèmes, l'approche "individus-centrée" peut être considérée comme "ascendante", où ce sont les comportements des éléments du système qui sont décrits

(Degenne, 2012). C'est en simulant les interactions d'un grand nombre de ces éléments que le comportement global du système se dégage. Les deux formes de modélisation qui relèvent de cette approche sont les "automates cellulaires" (AC) et les "systèmes multi-agents" (SMA).

### *Automates cellulaires*

En 1960, les travaux de Von Neumann (1966) sur les systèmes auto répliatifs mènent à la création du "copieur et constructeur universel" qui n'est autre que le premier automate cellulaire. Un AC est défini par :

1. un réseau de cellules dans un espace donné
2. un ensemble d'états possibles dans lesquels peuvent être les cellules
3. une ou plusieurs règles pour décrire comment une cellule change d'état à un instant  $t$ , en tenant compte de l'état de la cellule, et des états de ses voisins au temps  $t-1$

Dans un espace à 2 dimensions, une cellule peut avoir 4 (voisinage de Von Neumann) ou 8 voisins (voisinage de Moore). Ces voisinages correspondent par exemple à Nord, Sud, Est, Ouest et dans le cas de 8 voisins Nord-Est, Nord-Ouest, etc. Le Jeu de la vie (Gardner, 1970) utilise par exemple cette représentation pour montrer qu'il est possible via des règles très simples de produire des états du système non prévisibles par rapport à l'état initial. Ce phénomène peut notamment être utilisé pour la modélisation de systèmes complexes. La Figure 2.2 présente différents motifs de cellules répétitifs qui peuvent être obtenus via le jeu de la vie en considérant qu'une cellule possède 8 voisins.

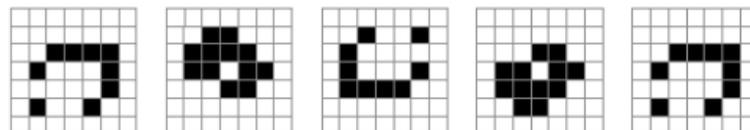


FIGURE 2.2 – Les différents motifs du jeu de la vie.

Les AC ont beaucoup été utilisés pour étudier les dynamiques urbaines Spacelle (Langlois, 2005) ; (Langlois and Phipps, 1997) ; RIKS (Engelen et al., 1995) ; SLEUTH (Clarke et al., 1997) et autres systèmes socio-environnementaux (Engelen et al., 1995) ; DINAMICA (Soares-Filho et al., 2002). Relativement simples à mettre en œuvre, les modèles à base d'automates cellulaires souffrent cependant de certaines limites importantes : les règles de transition sont les mêmes pour toutes les cellules, et la taille identique des cellules rend difficile la prise en compte de différentes échelles.

### Systemes multi-agents

Lors de travaux sur les systèmes distribués en intelligence artificielle le concept d'*acteur* a d'abord émergé pour permettre aux entités d'un système de communiquer (échanger des messages) et d'agir en fonction des messages reçus (Hewitt, 1976). Les acteurs sont reliés entre eux pour représenter les relations ou les modes de communication qui vont permettre de spécifier la méthode de changement de leurs états. D'autres recherches, toujours en intelligence artificielle distribuée, ont ensuite amené à faire évoluer le concept d'acteur en *agent* et *système multi-agents* (SMA) dont le terme est proposé par Gasser et al. (1987). Les agents ont comme les acteurs la capacité d'échanger des messages et d'agir en conséquence, mais ont aussi des objectifs propres et une certaine autonomie dans leur capacité de prise de décision en tenant compte de leur environnement comme présenté dans la figure 2.3. Le but est de faire collaborer un ensemble d'agents définis comme autonomes organisés en société pour résoudre collectivement un problème (Drogoul, 1993).

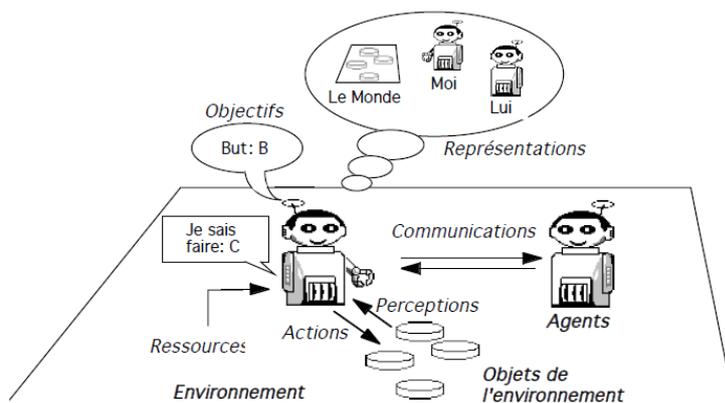


FIGURE 2.3 – Les différents modes de communication des agents, entre eux et leur environnement (Ferber, 1995).

Les agents peuvent avoir des propriétés de différents niveaux de sophistication (autonomie, réactivité, communication, aptitude sociale, pro-activité, intelligence, capacité d'apprentissage, mobilité, etc..). On distingue les agents réactifs – qui ont une vision locale de leur environnement et un comportement déterminé par rapport à la réception de stimuli élémentaires – des autres agents mieux dotés, dits cognitifs. Le domaine d'application de la modélisation des systèmes multi-agents est devenu très large, et un très bon aperçu est donné dans l'ouvrage de Ferber (1995). Dans l'article faisant l'état de l'art de la discipline appliquée à la gestion des écosystèmes, Bousquet and Le Page (2004) prônent des recherches qui doivent aller au-delà du rôle des individus pour étudier plus en profondeur les différentes formes d'organisations (spatial, en réseau ou hiérarchiques) et les interactions entre ces différents niveaux d'organisation.

Deux approches utilisant les systèmes multiagents pour modéliser les dynamiques spatiales sont aujourd'hui particulièrement utilisées : CORMAS (Bousquet et al., 1998) et NetLogo (Wilensky, 1999). Précisons toutefois qu'il existe de nombreuses autres approches qui même si elles ne sont pas initialement définies pour modéliser la dynamique spatiale peuvent être utilisées : MadKit (Gutknecht and Ferber, 2000), Mimosa (Müller, 2010), Swarm (Minar et al., 1996), GEAMAS-NG (David et al., 2011) ou encore RePast Symphony (North et al., 2005).

Dans ces différentes approches, la prise en compte de l'espace est faite le plus souvent en utilisant une grille de cellules carrées pour représenter un espace 2D. Certaines approches autorisent aussi l'utilisation de couches "vecteur" à travers une extension ou un couplage faible avec un Système d'Information Géographique (SIG). La plateforme GAMA (Taillandier et al., 2014) apparue récemment offre le compromis entre les méthodes trop complexes pour être utilisées par les non-informaticiens (e.g. RePast Symphony) et celles trop limitées pour développer des modèles intégrant un grand nombre d'agents (e.g. NetLogo). GAMA offre en outre des capacités avancées pour la gestion de l'espace, avec notamment la prise en compte de données géographiques vectorielles et rasters. Nous reviendrons sur cet aspect dans la discussion de ce chapitre.

### 2.3.3 Autres

Peu d'outils permettent de modéliser selon les deux paradigmes "dynamiques des systèmes" et "systèmes multi-agents" à la fois. Le seul à notre connaissance, Anylogic (Borshchev and Filippov, 2004), propose des simulations qui intègrent ces deux approches. C'est un logiciel commercial qui semble assez riche en fonctionnalités, mais qui n'est pas suffisamment transparent, notamment pour l'une des finalités les plus importantes : l'acquisition de la connaissance sur le système modélisé. Il existe d'autres approches permettant de modéliser des dynamiques spatiales qui ne s'apparentent pas aux deux paradigmes présentés ci-dessus. Sans être exhaustifs, nous présentons quelques approches utilisées dans les domaines d'applications qui nous concernent.

Certaines d'entre elles font le choix de prendre en compte l'espace en utilisant une grille de cellules raster, sans pour autant qu'elles soient des automates cellulaires ou des agents. Les contributions méthodologiques portent sur la manière d'associer un changement d'utilisation du sol (une des propriétés d'une cellule) aux facteurs moteurs de changements. CLUE (Veldkamp and Fresco, 1996), CLUE-S (Verburg et al., 2002), LANDIS (Mladenoff, 2004) et SELES (Fall and Fall, 2001) sont des outils connus et utilisés qui ont cette caractéristique. Ces outils continuent bien sûr d'évoluer en s'améliorant. SELES, par exemple, l'un des plus aboutis, couple gestion environnementale et SIG raster et vecteur, et utilise un langage métier pour décrire

les processus d'origine naturelle et humaine.

Cependant la plupart de ces approches utilisent pour la modélisation un langage informatique généraliste (C++ ou Java). Elles possèdent bien entendu la puissance d'expression de ces langages mais sont tout à fait pénalisantes pour les experts qui souhaitent modéliser leur système car nécessitant trop de connaissances informatiques. En effet, modéliser directement à partir d'un langage informatique généraliste n'est envisageable que si le modélisateur a une bonne maîtrise de la programmation dans ce langage. C'est le choix d'OpenFluid (Fabre et al., 2010) qui est une plateforme logicielle développée en C++ dédiée à la modélisation du fonctionnement des paysages complexes, et qui met l'accent sur les flux entre entités spatiales. L'espace est représenté par un ensemble d'unités spatiales gérées par un graphe. Une autre option intéressante est de fournir une bibliothèque contenant des structures de données et des fonctions pour les manipuler (API : Application Programming Interface) comme dans APILand (Boussard et al., 2010) utilisable avec le langage Java.

Enfin le langage L1 (Gaucherel et al., 2006b) et la plateforme DYPAL permettent de simuler la dynamique des paysages selon sept règles génériques traduisant les transformations de l'état des éléments du paysage (ou patches) en tenant compte des relations de voisinage par un graphe. Ces règles se séparent en deux catégories : une catégorie dite de composition changeant l'état des propriétés de l'élément et des règles de configuration agissant sur les géométries des éléments qui sont au nombre de six : union, division, dilatation, érosion, disparition et apparition d'éléments. Cette approche a été déclinée sous la forme d'une grammaire formelle, un langage pour décrire l'évolution du paysage (Gaucherel et al., 2012) comme des successions de cultures en utilisant des automates (Houet et al., 2014).

## 2.4 Approche Ocelet

Contrairement aux approches précédentes, l'approche Ocelet privilégie la notion de graphes d'interaction pour modéliser les dynamiques spatiales. Dans cette section, nous présentons ce mécanisme. Une présentation plus complète de l'approche Ocelet est proposée dans le chapitre 3.

Ocelet est d'abord un **langage métier** qui s'utilise dans un environnement logiciel dédié à la simulation de dynamiques spatiales (Degenne, 2012). Le choix d'un langage métier a été motivé par le fait qu'il offre le meilleur compromis entre capacité d'expression et facilité d'utilisation. Son formalisme repose sur le concept de **graphes d'interaction** où chaque sommet du graphe représente un élément du système. Les arêtes du graphe servent à traduire les interactions entre les éléments reliés par application de fonctions portées par les arêtes. Les graphes permettent de représenter de manière unique différentes formes de relations (spatiales, fonction-

nelles, hiérarchiques, sociales) qu'il peut y avoir entre les entités du système. Un modèle en Ocelet se spécifie selon trois niveaux :

- **Le niveau des individus** : Il s'agit de la description des éléments du système à manipuler. Dans Ocelet le concept d'*entité* est utilisé pour représenter ces éléments. Ils possèdent des propriétés qui caractérisent leurs états, ainsi que des fonctions (services) qui sont utilisées pour interroger l'entité sur son état, ou pour modifier cet état.
- **Le niveau des interactions** : Le terme *relation* traduit les échanges entre les entités, permettant ainsi de spécifier les changements d'états entre les entités. Le concept de graphe est ici utilisé, une entité étant un sommet du graphe et une interaction étant portée sur l'arête entre deux entités. Ce type de graphe, le graphe d'interaction, est défini formellement dans Ocelet.
- **Le niveau du système et de sa dynamique** : Le *scénario* constitue la phase opérationnelle du modèle, ainsi que la phase d'initialisation d'une simulation. C'est à ce niveau que se construisent les graphes d'interaction et se font les appels des fonctions d'interactions.

Enfin Ocelet facilite la construction de graphes avec de nombreuses entités par interfacement direct avec les données (SIG, tableur, base de données) via le concept de *datafacer*.

Par rapport aux approches de modélisation présentées ci-dessus, Ocelet se situerait dans "Autres", même si elle serait capable de modéliser des boucles de rétroaction ou des agents réactifs. Plusieurs des choix faits pour Ocelet l'ont aussi été dans d'autres approches, comme l'usage d'un langage métier, ou la prise en compte à la fois de données raster et vecteur pour représenter l'espace. Mais l'originalité principale de l'approche Ocelet est de centrer la modélisation sur les interactions, notamment spatiales, et de pouvoir décrire toutes les interactions (spatiales, fonctionnelles, sociales, hiérarchiques) présentes dans un modèle avec un seul et même concept, le graphe d'interaction. Dans la suite nous présentons simplement l'utilisation de graphes en modélisation spatiale et décrivons comment l'extension du concept vers le graphe d'interaction permet de modéliser les dynamiques spatiales. Une description plus complète est donnée par Degenne (2012).

## 2.5 Utilisation de graphes dans la modélisation spatiale

En écologie du paysage, les graphes sont largement utilisés pour représenter les éléments d'un paysage (Risser et al., 1984). Dans leurs travaux utilisant la photo-interprétation d'images de télédétection, Cantwell and Forman (1993) proposent une méthode de construction de graphe qui prend en compte la configuration spatiale des éléments du paysage (patchs qui peuvent être séparés par des frontières ou connectés par des corridors) et les interactions qu'il peut y avoir entre ces éléments, permettant

de représenter les flux. Cette méthode permet ainsi de comparer des paysages en se basant sur la configuration spatiale des éléments et leurs interactions. Ce type de construction a été testé sur 25 paysages différents et les auteurs ont trouvé qu'un nombre restreint de motifs pouvaient les représenter tous. La figure 2.4 représente les différents motifs de graphes retenus par les auteurs.

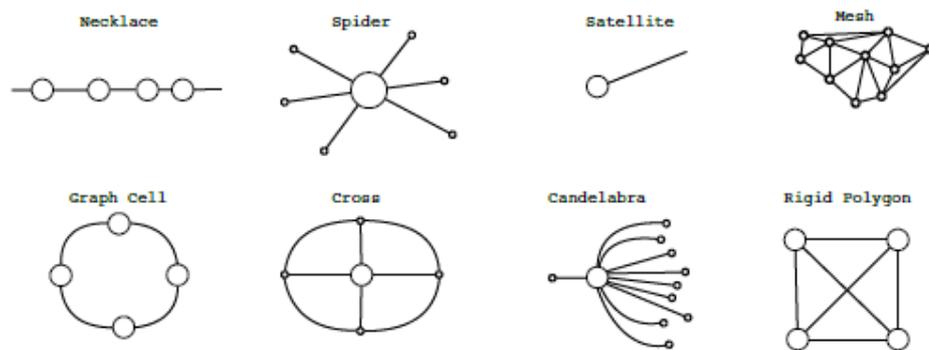


FIGURE 2.4 – Les différents motifs de graphes de paysages identifiés par Cantwell and Forman (1993)

Suite à ces travaux, et d'autres plus récents (Urban and Keitt, 2001), Fall et al. (2007) introduit le *Spatial Graph Theory* où les structures spatiales sont prises en compte explicitement dans leur référentiel géographique. Ainsi, dans un *graphe spatial*, un sommet du graphe n'est pas un objet adimensionnel comme dans la figure 2.4, mais possède bien une surface, une forme, et est positionné dans un espace géographique. Les graphes restent néanmoins utilisés essentiellement pour l'étude de connectivités entre habitats, et non pas pour l'étude de dynamiques paysagères.

Degenne et al. (2010) introduit le concept de graphe d'interaction, qui définit non seulement *qui* est en relation (la structure du graphe), mais aussi *comment* les éléments sont liés (le comportement du graphe). Ainsi, un graphe d'interaction est un graphe auquel est attachée la sémantique nécessaire pour spécifier ce qui se passe lorsque les éléments sont en interaction : le type d'information échangée, les actions que peuvent réaliser les unes sur les autres, les effets qu'ont les interactions sur les éléments et les arêtes concernées. La figure 2.5 présente différents types de relations dans un même espace géographique : des relations de voisinage entre parcelles en (a) ou entre les parcelles et la rivière en (b) ; des relations "sociales" entre acteurs en (c) et des relations "fonctionnelles" entre agriculteurs et les parcelles sur lesquelles ils agissent en (d). Les arêtes de ces graphes portent des fonctions qui peuvent être activées pour modifier l'état des entités connectées.

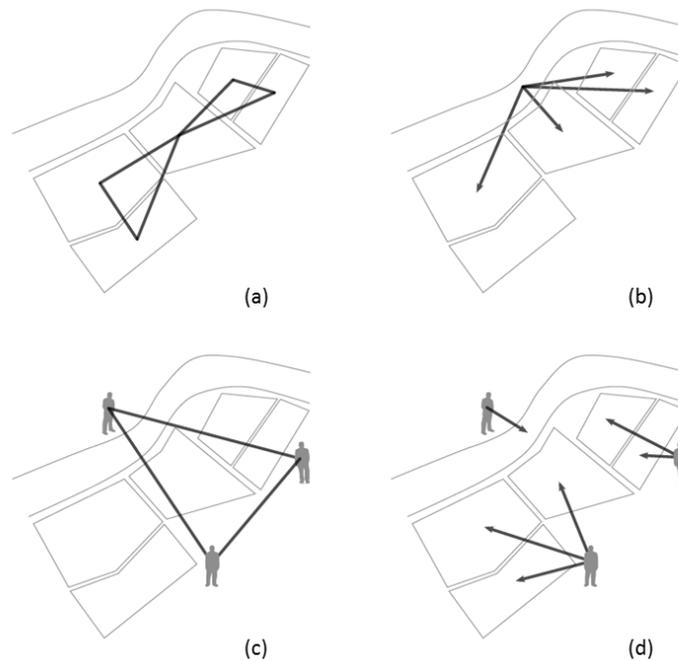


FIGURE 2.5 – Différents types d’interaction (a et b : spatiale ; c : sociale ; d : fonctionnelle) entre les éléments d’un paysage (Degenne, 2012).

## 2.6 Discussion

Dans ce chapitre nous avons tenté de situer l’approche Ocelet parmi les principales approches en vigueur aujourd’hui qui servent à la modélisation de dynamiques spatiales. Ocelet a fait le choix d’un langage métier, offrant par cela une grande capacité d’expression aux modélisateurs, et permet de développer des modèles dans un environnement logiciel qui comprend un compilateur ainsi qu’un moteur d’exécution. Si la plupart des autres approches représentent l’espace sous la forme d’une grille de cellules régulières, quelques-unes dont Ocelet permettent de gérer l’espace avec des objets dont la géométrie est définie par une représentation vecteur. Rares sont les approches combinant efficacement les deux. Les systèmes d’informations géographiques ont depuis longtemps intégré ces deux types de représentation de l’espace. Leur gestion se fait par l’utilisation de couches thématiques d’information géographique. La manipulation de ces données se fait généralement soit dans un but d’analyse spatiale, soit à des fins de représentation cartographique. Les SIG ont en effet rapidement montré leurs limites lorsqu’il s’agissait de gérer le temps. Certaines approches de modélisation ont cependant développé une gestion de l’espace en se couplant avec les SIG. L’espace n’est dans ce cas pas inclus conceptuellement dans l’approche, la gestion des données spatiales restant du domaine du SIG.

La principale caractéristique d’Ocelet est de modéliser les dynamiques spatiales

avec des graphes d'interaction – des graphes dynamiques à structure dynamique enrichis de fonctions d'interaction qui permettent de décrire les processus élémentaires propres à chaque type de relation. Dans l'implémentation d'Ocelet actuelle (version 1) n'est prise en compte que la représentation vecteur des éléments. Il est à noter cependant que le concept de graphe d'interaction est applicable tel quel à une représentation en cellules raster. Les relations de voisinage seraient alors implicites, mais pourraient ne pas se cantonner qu'à des voisines immédiates, comme c'est souvent le cas. Des possibilités d'utilisation conjointe raster-vecteur viennent aussi immédiatement à l'esprit, comme par exemple celle de représenter une hétérogénéité (raster) dans une parcelle de culture (polygone), ou encore de prendre en compte le modelé du terrain à partir d'un modèle numérique d'altitude (raster), et surtout l'ouverture vers la source de données spatiales la plus utilisée qu'est la télédétection.

Dans la suite de ce mémoire, notre objectif est donc de répondre aux questions suivantes :

*Quelles sont les conséquences de l'intégration du format raster ?* Intuitivement ce format peut avoir une granularité beaucoup plus fine que celle qui existe actuellement. Les pixels d'une image satellite, par exemple, peuvent correspondre à des cellules qui dans le modèle peuvent être mises en interaction avec les autres.

*Quelles sont les conséquences sur les graphes d'interaction ?* Les mécanismes de graphe d'interaction ont montré qu'ils offraient toutes les facilités pour le modélisateur de spécifier son modèle. Cependant ces dernières s'intéressaient à des descriptions de plus haut niveau. L'un des problèmes dans ce cadre est de gérer de la manière la plus transparente possible des représentations à des niveaux de granularité très différentes : données vectorielles et données raster.

*Quelles implications sur l'implémentation ?* L'importation des sources de données raster telles que les images satellites dans un modèle engendre des graphes d'interaction de taille conséquente. Il est donc indispensable de proposer des mécanismes de manipulation qui soient efficaces pour que la modélisation et les simulations ne soient pas pénalisées par le traitement des graphes associés. Nous répondons à ces questions dans les chapitres suivants et dans un premier temps nous nous intéressons à la prise en compte de la représentation régulière de l'espace dans le formalisme d'Ocelet.

# Chapitre 3

## Conception

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>35</b>
<b>3.2</b>	<b>Définitions</b>	<b>36</b>
3.2.1	Elements de la théorie des graphes	37
3.2.2	Concepts généraux d'Ocelet	39
3.2.3	Concepts du langage Ocelet	40
3.2.4	Représentation spatiale par objets vecteurs	44
<b>3.3</b>	<b>Le format Raster</b>	<b>46</b>
<b>3.4</b>	<b>Pavages réguliers</b>	<b>48</b>
<b>3.5</b>	<b>Cellules</b>	<b>51</b>
<b>3.6</b>	<b>Pavages réguliers et graphes d'interaction</b>	<b>52</b>
3.6.1	Entités cellulaires et graphe d'interaction	53
3.6.2	Entités cellulaires, géométriques, sans représentation spatiale et graphe d'interaction	55
3.6.3	Entités cellulaires de différents types et graphe d'interaction	59
<b>3.7</b>	<b>Conclusion</b>	<b>60</b>

---

### 3.1 Introduction

Nous proposons dans ce chapitre d'élargir les possibilités de représentation spatiale du langage Ocelet en intégrant la forme de représentation régulière (format de données raster), qui, avec la forme géométrique (format de données vecteur) sont les deux types de représentations spatiales les plus couramment utilisés (Longley et al., 2015) en modélisation. Le choix de l'un ou l'autre de ces formats est souvent déterminé à priori, indépendamment de la nature du système à modéliser, mais il serait préférable que le modélisateur n'ait pas à être contraint par un tel choix pour élaborer ses modèles (Couclelis, 1992).

Nous avons dû considérer l'intégration du format raster en cohérence avec les concepts du langage selon trois aspects : (i) la forme de représentation de l'espace pour décrire les entités, (ii) les différentes formes de relations spatiales entre ces entités et (iii) la façon de traiter les processus que l'on fait porter par ces relations.

Le premier aspect concerne donc la représentation spatiale des entités en jeu dans un modèle. A travers l'usage de pavages réguliers c'est au sens large que le format raster est pris en compte, ce qui offre la possibilité de représenter et manipuler différentes natures de champs continus (discretisés régulièrement) mais aussi d'intégrer des structures de données de type images (notamment de télédétection) qui sont de plus en plus largement utilisées dans les modèles.

Les relations entre entités d'un modèle peuvent être basées sur une forme de voisinage spatial pouvant faire appel à la notion de distance, l'adjacence, la superposition, ou bien d'autres mesures spatiales que l'on peut imaginer. La première version d'Ocelet propose des outils qui facilitent la construction de tels liens à partir d'opérateurs spatiaux sur des représentations vecteur des entités manipulées. Les liens ainsi créés sont explicites c'est-à-dire qu'une fois construits il sont conservés avec leurs identités propres et ils peuvent être traités indépendamment des structures de données qui ont servi à les construire. Nous montrons dans ce chapitre que le fait de manipuler des structures spatialement régulières permet de construire des relations spatiales présentant des propriétés intéressantes. En particulier on peut dans bien des cas s'affranchir d'une création explicite des arêtes du graphe et donner accès à différents cas de graphes implicites.

La façon de traiter les processus portés par des graphes implicites réguliers n'est abordée que succinctement dans ce chapitre mais exposée plus en détail dans le chapitre 4.

Après un rappel sur quelques éléments de la théorie des graphes et les concepts de base d'Ocelet, nous présentons ici les résultats de notre étude sur les deux premiers aspects que nous venons de citer : la représentation spatiale au format raster et la façon dont nous l'avons rendue possible en tenant compte des différentes formes de relations spatiales susceptibles de porter des interactions.

## 3.2 Définitions

Dans cette section nous présentons le formalisme du langage, les concepts et méthodes associés à Ocelet. Le formalisme reposant sur le principe de graphes d'interaction, nous commençons par rappeler, dans un premier temps, des éléments de définitions sur la théorie des graphes.

### 3.2.1 Elements de la théorie des graphes

Les notions données sur la théorie des graphes sont principalement utilisées dans les parties suivantes à propos de la conception de l'intégration et la gestion de la représentation régulière de l'espace. Pour plus de détails sur la théorie des graphes le lecteur pourra se référer à (Berge, 1970 ; White, 1973 ; Diestel, 2005).

Un graphe  $G$  est composé de deux ensembles : l'ensemble des sommets, noté  $X(G) = \{x_1, x_2, \dots, x_n\}$  et l'ensemble des arêtes du graphe  $U$  tel que  $U(G) = \{u_1, u_2, \dots, u_n\}$ . Le graphe se note  $G(X, U)$ . L'ordre  $|G|$  d'un graphe est défini comme étant le nombre de sommets de  $G$  :  $|G| = |X(G)|$ .

Pour une arête  $u_1 = (x_1, x_2)$ ,  $u_1$  est appelée arête incidente à  $x_1$  et  $x_2$ . L'extrémité initiale de l'arête est représentée par  $x_1$  et l'extrémité terminale de l'arête est représentée par  $x_2$ . Un graphe dont les sommets des arêtes sont distingués par extrémité est dit orienté. Sans distinction des sommets aux extrémités d'une arête, le graphe est dit non-orienté.

Les structures et les composantes de graphes peuvent être comparables, on peut par exemple comparer le nombre de sommets ou encore le nombre d'arêtes entre deux graphes. Il y a équivalence de graphe lorsqu'un graphe  $G(X, U)$  et un graphe  $G'(X', U')$  vérifient les conditions :  $X(G) = X'(G')$  et  $U(G) = U'(G')$ .

Il existe de nombreuses structures de graphes selon, par exemple, les classes (ou types) de sommets, leurs représentations ou encore les méthodes de connexion pour relier les sommets avec des arêtes. Nous proposons de présenter quelques-unes de ces structures remarquables :

Un graphe  $G(X, U)$  est dit  $k$  - parties lorsque les sommets sont partitionnés en  $k$  ensembles distincts  $X_1, X_2, \dots, X_k$  et si l'extrémité des arêtes sont des éléments d'ensembles distincts. Par exemple, pour  $k = 2$ , un graphe est appelé biparti ou encore pour  $k = 3$  triparti. Lorsque tous les sommets d'un graphe sont connectés à tous les autres, il s'agit d'un graphe complet. La figure 3.1 illustre des exemples de graphes biparti, triparti et complet. Une chaîne d'arêtes se définit par un ensemble d'arêtes non orientées. Un graphe  $G(X, U)$  est dit connexe si pour chaque sommet  $\{x_1, x_2, \dots, x_n\}$  il existe une chaîne d'arêtes  $\{u_1, u_2, \dots, u_n\}$  permettant d'atteindre chacun de ces sommets à partir d'un sommet précédent. Une chaîne d'arêtes orientées avec laquelle on peut parcourir les extrémités en suivant l'orientation est appelée chemin.

La représentation des graphes peut aussi être caractérisée en donnant une position aux sommets sur un plan euclidien et si une arête relie deux sommets (dans le cas où une arête relie plus de deux sommets, il s'agit d'un hypergraphe, que nous ne traiterons pas dans cette thèse). Le graphe est dit planaire s'il est possible d'avoir

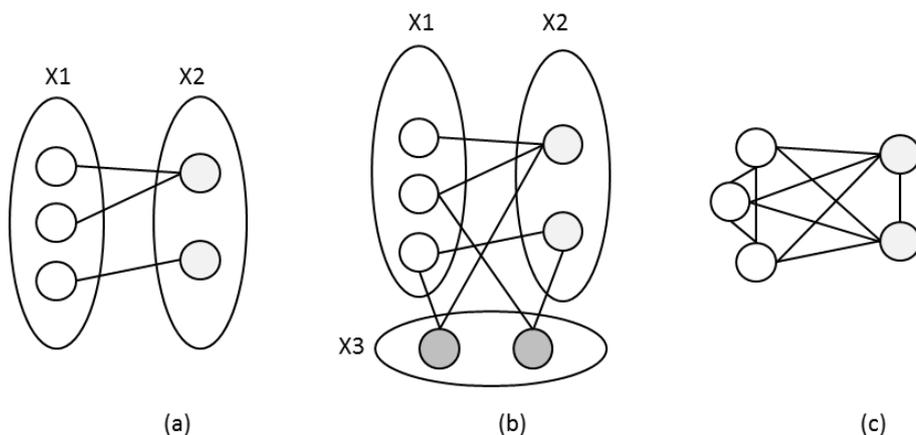


FIGURE 3.1 – Un graphe biparti entre les sommets appartenant aux ensembles  $X_1$  et  $X_2$  (a). Un graphe triparti entre les éléments appartenant aux ensembles  $X_1$ ,  $X_2$  et  $X_3$  en (b). Un graphe complet (c).

une représentation de ce graphe sans avoir d'intersection entre ses arêtes, comme illustré par la figure 3.2 (a) où l'intersection peut être supprimée en déplaçant le sommet  $x_1$  ou le sommet  $x_4$ . Un graphe est dit planaire topologique, illustré par la figure 3.2 (b), en rajoutant les conditions suivantes à un graphe planaire : (i) Les arêtes doivent avoir des extrémités différentes. (ii) Les arêtes ne doivent pas croiser d'autres sommets ou d'autres arêtes.

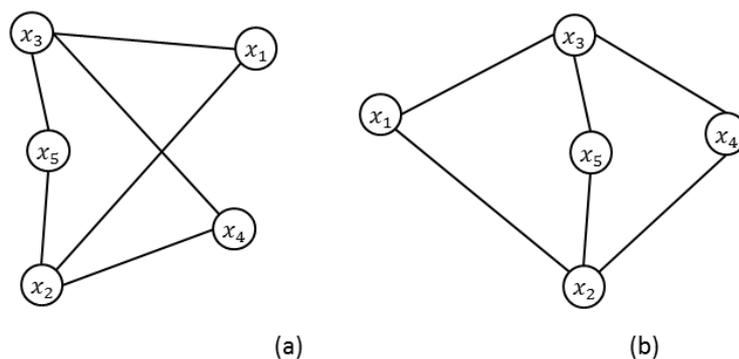


FIGURE 3.2 – Un graphe planaire en (a), il est possible d'avoir une représentation de ce graphe sans intersection entre les arêtes. Un graphe planaire topologique (b).

Les caractéristiques des graphes sont nombreuses, nous n'avons abordé ici que les éléments qui sont nécessaires à la compréhension de la suite de ce document, en particulier ce qui concerne les définitions sur le formalisme d'Ocelet.

### 3.2.2 Concepts généraux d'Ocelet

La modélisation avec Ocelet favorise une représentation systémique dans laquelle on cherche à relier entre elles toutes les entités susceptibles d'interagir les unes avec les autres puis à faire évoluer l'ensemble en réalisant ces interactions dans un certain ordre. L'ensemble de ces entités et des liens que l'on peut établir entre elles constituent un graphe dans lequel toutes les formes de relations du modèle sont présentes ensemble. Pour décrire la nature de ces interactions, le formalisme d'Ocelet suppose que l'on considère séparément les sous-graphes constitués de relations d'une même nature. Une entité pouvant appartenir à plusieurs de ces sous-graphes.

A titre d'exemple, imaginons un modèle constitué de trois types d'entités :

- Des communes avec leurs limites administratives que nous noterons  $C$ . Chaque commune individuelle étant indicée dans un modèle :  $c_1, c_2, \dots$
- Des parcelles cadastrales que nous noterons  $P$ . Chaque parcelle sera elle aussi indicée :  $p_1, p_2, \dots$
- Des exploitations agricoles que nous noterons  $E$ . Chaque exploitation sera indicée de la même façon.

Un modèle complet pourrait être construit sous la forme d'un graphe comme celui de la figure 3.3

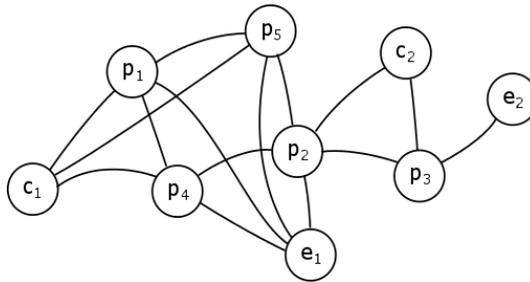


FIGURE 3.3 – Un modèle complet représenté par un graphe. Différents types d'éléments sont représentés par les sommets : communes, parcelles cadastrales, exploitations.

Avec Ocelet les interactions possibles vont en fait être regroupées selon les types de sous-graphes comme dans la figure 3.4 où l'on distingue 3.4 (a) le sous-graphe reliant les communes aux parcelles qui en dépendent, 3.4 (b) le sous-graphe reliant les parcelles aux exploitations auxquelles elles appartiennent, et 3.4 (c) le sous-graphe de voisinage spatial des parcelles.

Pour chacun de ces sous-graphes on sait que toutes les arêtes sont du même type : la nature des extrémités est toujours la même. Cette propriété nous permet de faire porter aux arêtes des fonctions d'interaction dont la sémantique est spécifique aux entités reliées.

Le moteur d'exécution d'Ocelet garantit que lorsque l'on exécute une fonction d'interaction sur les arêtes d'un graphe, le résultat (le changement d'état des entités et des liens) sera le même quel que soit l'ordre dans lequel on aura parcouru les

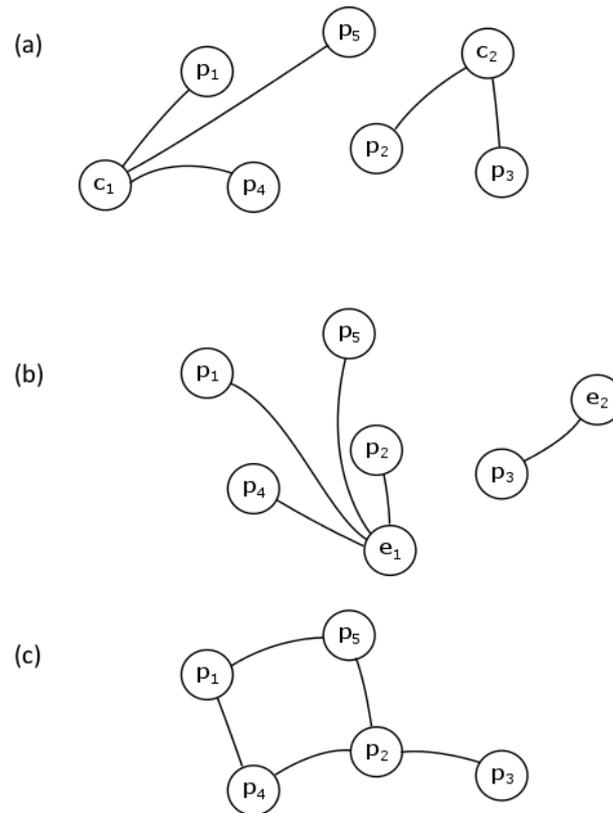


FIGURE 3.4 – Décomposition du graphe du modèle complet en plusieurs sous-graphes : un sous-graphe entre les communes et les parcelles qui en dépendent (a), un sous-graphe entre les exploitations et leurs parcelles (b) et le sous-graphe de voisinage entre les parcelles (c).

arêtes. Cela revient à dire que la fonction d'interaction est exécutée simultanément sur toutes les arêtes concernées. Nous verrons plus loin que cela peut dans certains cas entraîner des incertitudes sur l'affectation de propriétés, mais Ocelet fournit au modélisateur le moyen de lever ces incertitudes (à l'aide de *fonctions d'agrégation*).

Nous donnons par la suite des définitions formelles sur les principes de conception de modèles avec Ocelet.

### 3.2.3 Concepts du langage Ocelet

#### Entité

L'entité représente un élément du système que l'on cherche à modéliser. On notera  $E\{e_1, e_2, \dots, e_n\}$  l'ensemble des entités. Les entités sont catégorisées en types, celles d'un même type constituant alors un sous-ensemble de  $E$ . L'état d'une entité est caractérisé par ses propriétés à un temps donné. Chacune des propriétés appartenant à un domaine  $D$  de valeurs, par exemple une propriété réelle ayant un

domaine  $D = \mathbb{R}$ . Des fonctions nommées "service" en Ocelet permettent de consulter (fonctions de consultation) ou encore de modifier ces propriétés (fonctions de transition).

**Définition 1.** Une entité  $e$  est constituée d'un  $p$ -uplet de propriétés  $P\{\alpha, \beta, \gamma, \dots\}$ . On note  $e_t$  l'état d'une entité au temps  $t$  qui est donné par les valeurs de ses propriétés :  $P_t\{\alpha_t, \beta_t, \gamma_t, \dots\}$ . L'état se traduit par le produit cartésien des domaines des propriétés :  $D_e = D_\alpha \times D_\beta \times \dots$

Pour traduire la dynamique sur les propriétés, celles-ci sont données par la définition 2 suivante :

**Définition 2.** Soit  $e$  une entité définie par une propriété  $\alpha$  alors :

$\alpha_n$  représente l'état d'une propriété au pas de temps  $n$

$\alpha_m^n$  représente la succession d'états pris par une propriété entre les pas de temps  $m$  et  $n$ .

$\vec{\alpha}$  représente l'ensemble de l'historique des états pris par la propriété  $\alpha$  au cours de la simulation.

En considérant l'état de l'entité  $e$  alors :

$e_n$  représente l'état de l'entité au pas de temps  $n$

$e_m^n$  représente la succession d'états pris par l'entité entre les pas de temps  $m$  et  $n$

$\vec{e}$  représente l'ensemble de l'historique des états pris par l'entité au cours de la simulation

Nous pouvons classer les types de propriétés en trois catégories :

- les types primitifs : entiers, réels et booléens
- les types composés : chaînes de caractères, listes, tables de hachages, Structure (une structure étant composée de un ou plusieurs types différents)
- les types de représentations spatiales : les géométries dont la définition est donnée dans la partie suivante

### Fonctions sur entité

Plusieurs types de fonctions peuvent être définis sur les propriétés des entités d'un modèle : (i) les fonctions de consultation permettant d'obtenir de l'information sur l'état d'une entité et (ii) les fonctions de transition permettant de faire évoluer cet état.

**Définition 3.** Une fonction de consultation  $f_c$  sur une entité  $e$  permet de lire l'état de cette entité sans en modifier son état. Soit  $D_m$  le domaine de valeurs possibles du résultat de la fonction de consultation, cette fonction est définie par  $f_c : D_e \mapsto D_m$ .

**Définition 4.** Une fonction de transition élémentaire  $f$  notée sur une entité  $e$  décrit le passage de l'état des propriétés de l'ensemble ou sous ensemble de propriétés  $P\{\alpha, \beta, \gamma, \dots\}$  du temps  $t$  au temps  $t + 1$ . Une fonction de transition  $f_t$  est constituée d'une ou plusieurs fonctions de transition élémentaires ainsi que d'une ou plusieurs fonctions de consultation.

Des exemples  $f$  et  $g$  de fonctions de transition d'états s'écrivent alors :

$$\begin{aligned} f : D_\alpha &\mapsto D_\alpha, & \alpha_{n+1} &= f(\alpha_n) \\ g : D_\alpha \times D_\beta &\mapsto D_\alpha, & \alpha_{n+1} &= g(\alpha_n, \beta_n) \end{aligned} \quad (3.1)$$

## Relation et graphe d'interaction

Une ou plusieurs structures de graphes peuvent être utilisées pour spécifier quelles sont les entités qui vont être impliquées dans des interactions. Le concept de graphe d'interaction ou relation sont ici définis pour représenter de telles structures. La relation porte la description du type d'entité ainsi que des interactions devant être appliquées.

**Définition 5.** Soit  $E\{e_1, e_2, \dots, e_n\}$  l'ensemble des entités. Il y a interaction entre une entité  $e$  appartenant à  $E$  et une ou plusieurs autres entités distinctes appartenant à  $E$  si une ou plusieurs fonctions de transition  $f_t$  affectent l'état d'au moins une propriété de  $e$  en faisant intervenir une propriété d'une autre entité appartenant à  $E$ . Cette fonction est notée fonction d'interaction :  $f_i$ . Un ensemble de fonctions d'interaction est noté  $F_i$ .

**Définition 6.** Le concept de relation porte l'ensemble des fonctions d'interaction  $F_i$  et est noté :  $R = (A, F_i)$  avec  $A = (E1\{e_{1_1}, e_{1_2}, \dots, e_{1_n}\}, E2\{e_{2_1}, e_{2_2}, \dots, e_{2_n}\})$  un couple d'ensembles d'entités sur lesquelles appliquer les fonctions d'interaction. Nous étudierons uniquement les interactions entre deux ensembles, distincts ou non d'entités. La structure portant les liens entre les entités peut être construite par un graphe où les sommets sont les entités et les arêtes représentent les liens. L'instanciation d'une relation est appelée graphe d'interaction, les liens du graphe portant alors les fonctions d'interaction. Un tel graphe se note  $\Gamma = (A, U, F_i)$  avec  $U = (u_1, u_2, \dots, u_3)$  une famille dont les éléments sont issus du produit cartésien  $E1 \times E2$ .

La figure 3.5 présente un exemple de graphes sur lesquels appliquer une fonction d'interaction.

## Prise en compte d'affectation multiple et fonctions d'agrégations de valeurs

Le fait d'utiliser des graphes pour appliquer des fonctions sur les arêtes implique de tenir compte des multiples valeurs pouvant venir affecter une propriété d'entité si celle-ci est connectée avec plusieurs autres entités. Pour traduire le fait que les

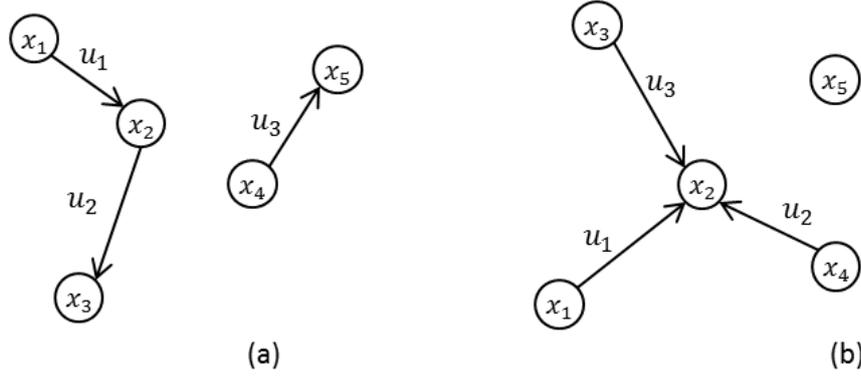


FIGURE 3.5 – Exemple de graphes sur lesquels appliquer des fonctions d'interaction.

interactions sont appliquées simultanément sur l'ensemble des arêtes, les valeurs des propriétés d'une entité ne sont changées qu'une fois le parcours des arêtes terminé.

$$f(\alpha, \gamma, \beta) \Rightarrow \begin{cases} \alpha_{n+1} = \frac{1}{3}(2 \times \alpha_n + \beta_n) \\ \beta_{n+1} = \beta_n - \gamma_n \end{cases} \quad (3.2)$$

Soit  $f(\alpha, \gamma, \beta)$  la fonction d'interaction donnée par l'équation 3.2, en l'appliquant au graphe de la figure 3.5 (a), nous obtenons les changements d'états donnés par les équations 3.3.

$$\begin{aligned} \text{Sur } u_1 : f(x_1 : \alpha, x_1 : \gamma, x_2 : \beta) &\Rightarrow \begin{cases} x_1 : \alpha_{n+1} = \frac{1}{3}(2 \times x_1 : \alpha_n + x_2 : \beta_n) \\ x_2 : \beta_{n+1} = x_2 : \beta_n - x_1 : \gamma_n \end{cases} \\ \text{Sur } u_2 : f(x_2 : \alpha, x_2 : \gamma, x_3 : \beta) &\Rightarrow \begin{cases} x_2 : \alpha_{n+1} = \frac{1}{3}(2 \times x_2 : \alpha_n + x_3 : \beta_n) \\ x_3 : \beta_{n+1} = x_3 : \beta_n - x_2 : \gamma_n \end{cases} \\ \text{Sur } u_3 : f(x_4 : \alpha, x_4 : \gamma, x_5 : \beta) &\Rightarrow \begin{cases} x_4 : \alpha_{n+1} = \frac{1}{3}(2 \times x_4 : \alpha_n + x_5 : \beta_n) \\ x_5 : \beta_{n+1} = x_5 : \beta_n - x_4 : \gamma_n \end{cases} \end{aligned} \quad (3.3)$$

Lorsqu'un sommet a une multiplicité d'arêtes, plusieurs valeurs vont affecter une propriété. Ces valeurs sont nommées "valeurs candidates" et sont gardées en mémoire, il revient au modélisateur de spécifier comment affecter une propriété à partir de ces valeurs candidates au moyen de fonctions d'agrégation.

**Définition 7.** Une fonction d'agrégation  $h$  est définie sur l'ensemble des valeurs issues d'une fonction d'interaction qui affectent une propriété de l'entité. Par exemple nous pouvons définir la fonction  $h$  sur une propriété par :

$$h : D_\alpha^k \mapsto D_\alpha, \quad k \in N \quad (3.4)$$

Les fonctions d'agrégation servent à traiter les valeurs candidates et à affecter la propriété d'une entité par une valeur unique. Ocelet propose des fonctions de bases (moyenne, maximum d'une série de valeurs, etc) mais le modélisateur peut définir ses propres fonctions d'agrégation.

## Datafacer et accès aux sources de données

Dans le but de faciliter l'intégration de données dans les modèles en Ocelet, le concept de datafacer a été défini pour construire des ensembles d'entités en lisant des données dans des fichiers ou des bases de données, et exporter des ensembles d'entités dans des fichiers ou des bases de données. Des premiers datafacers ont été développés permettant l'accès aux bases de données PostGreSQL, des fichiers de type CSV (Comma Separated Value) ou encore des fichiers textes. Dans le but de pouvoir intégrer des données spatiales, un datafacer permet de lire et écrire des fichiers du format ESRI Shapefile. Ces derniers facilitent la création d'entités comportant au moins un attribut spatial, cet attribut étant initialisé à partir des données de géométries contenues dans le fichier Shapefile ou la table PostGIS. Les autres propriétés sont initialisées à partir des colonnes attributaires non spatiales.

## Scenario

La construction d'un modèle Ocelet prend en compte les définitions précédentes d'entité, de relation et de datafacer. Ce type de définition est déclaratif et c'est dans le scénario qu'il faut les instancier. C'est aussi dans le scénario que sont définis les enchaînements des opérations comme l'appel des interactions en tenant compte de la description de la dynamique du système. Les opérations peuvent être :

- Import ou export de données avec un datafacer
- Création et instanciation d'entités
- Création et instanciation de relations
- Connexion d'entités dans un graphe d'interaction
- Appel de fonctions d'interaction
- Des fonctions de contrôle comme une boucle permettant de décrire la dynamique du modèle

Si le concept de graphe d'interaction peut être appliqué pour représenter n'importe quel élément d'un système, la spatialisation des entités se fait en attribuant une géométrie (format vecteur) en tant que propriété des entités.

### 3.2.4 Représentation spatiale par objets vecteurs

La première version du langage Ocelet permet de caractériser spatialement des entités avec des géométries, nous proposons d'en donner une définition : Un "objet" géométrique se définit par un ensemble de points dont les coordonnées appartiennent à l'espace des réels sur un plan euclidien à partir d'un système de coordonnées projetées. Les types primitifs géométriques sont au nombre de trois : le point, la ligne et le polygone. Les géométries de ces types sont dites "primitives". Les types multiples, constitués de plusieurs géométries primitives d'un même type étant le multi-points, multiligne et multi polygones. Nous nommerons géométrie un "objet" défini par un type, primitif ou multiple de géométrie. Une entité est appelée *entité géométrique* si une de ses propriétés est une géométrie.

Plusieurs types d'opérations sont applicables sur une géométrie, par exemple des opérations de translation, rotation, calculs de distance, du centre de gravité etc. Les opérations booléennes sur les géométries (*ET*, *OU*, *NON*, *XOR*) qui pour une géométrie *A* et une géométrie *B* correspondent aux opérations suivantes :

1. *A ET B* : intersection de *A* et *B*, figure 3.6(2)
2. *A OU B* : union de *B* et *A*, figure 3.6(3)
3. *A NON B* : différence de *A* par rapport à *B*, figure 3.6(4)
4. *B NON A* : différence de *B* par rapport à *A*, figure 3.6(5)
5. *A XOR B* : différence symétrique entre *A* et *B*, figure 3.6(6)

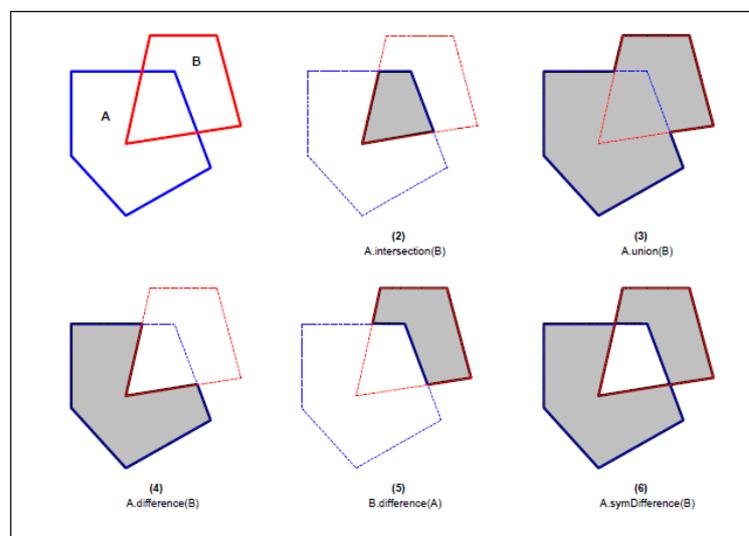


FIGURE 3.6 – Les différentes opérations booléennes applicables sur deux géométries *A* et *B* (Vivid Solutions, 2003)

L'utilisation des formes géométriques et des opérations applicables sur celles-ci se font avec la bibliothèque JTS (Vivid Solutions, 2003). Ces fonctions sur les géométries sont utilisables de la même manière que les fonctions de consultation ou de transition définies précédemment, en tant que propriété d'une entité. Les fonctions de consultation peuvent alors servir à connaître l'aire d'une géométrie ou sa distance par rapport à un point par exemple. Les fonctions de transition peuvent modifier les coordonnées des points de la propriété géométrique, modifiant la forme de l'entité, par l'utilisation d'une fonction booléenne ou encore une fonction définie spécifiquement dans un modèle.

Les sources de données permettant d'apporter une caractéristique spatiale à une entité d'un modèle sont multiples. Les principales sources de données utilisées dans un système d'information géographique sont sous la forme de couches vectorielles ou

raster (Longley et al., 2015). La gestion des géométries en tant que propriété permet dans ce cas de manipuler de l'information venant de couches vecteurs. Dans les SIG, on peut manipuler un ensemble de couches, soit pour les combiner afin de produire une carte, soit à des fins d'analyses spatiales par exemple. La construction de ces objets vecteurs se fait selon différentes méthodes, il peut s'agir d'un découpage de l'espace selon un point de vue, un expert peut se baser sur une image satellite et en extraire des formes qu'il juge nécessaires à son étude. Les objets vecteurs peuvent aussi venir d'un processus de segmentation et classification d'image, dans ce cas les objets sont construits en assistant l'expert qui décrit des règles pour construire la forme de ces objets géométriques par rapport à une image. Nous pouvons noter qu'un espace occupé par ces objets n'est pas forcément entièrement recouvert, comme le présente la figure 3.7 (b). Un espace peut aussi être construit selon des méthodes comme un diagramme de Voronoï ou la triangulation de Delaunay qui sont notamment utilisées par les modèles neutres (Gaucherel et al., 2006a). Il s'agit dans ce cas de construire des polygones par rapport à une série de points comme le présente la figure 3.7 (a).

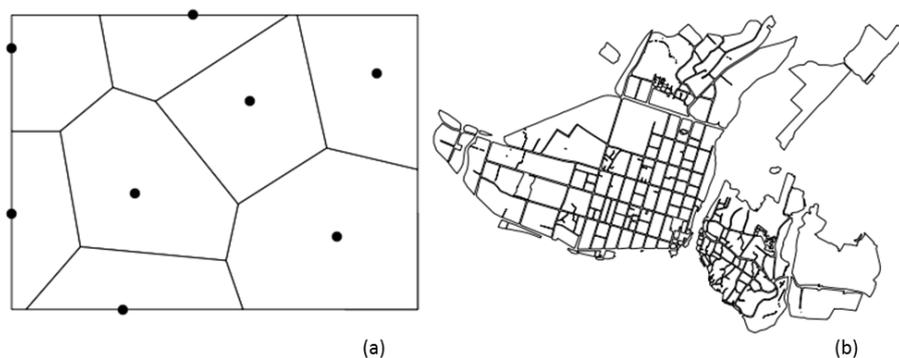


FIGURE 3.7 – Représentation de l'espace par objets vecteurs. Un diagramme de Voronoï (a). Des polygones caractérisant un cadastre (b).

Ces principales définitions ont permis de présenter les concepts de base d'Ocelet : entité, relation, datafacer et scénario. Nous avons aussi vu le type géométrique que la version actuelle d'Ocelet est capable de manipuler pour attribuer une représentation spatiale aux entités. Dans la section suivante, nous présentons une autre forme de représentation régulière, le format raster.

### 3.3 Le format Raster

Le format raster est une structure de données sous une forme matricielle. L'unité de base de ce format est le pixel (picture element) qui permet de caractériser un point de l'image. Un pixel est défini généralement par une forme carrée (parfois rectangulaire), une résolution et des valeurs caractérisant l'espace qu'il recouvre.

Pour  $n$  valeurs d'un pixel, le raster sera composé de  $n$  matrices aussi appelées bandes de l'image. La dimension du raster se note  $l \times h$  qui détermine respectivement le nombre de colonnes de pixels et le nombre de lignes de pixels, le nombre total de pixels de l'image étant le produit des deux.

Le type d'image le plus connu comporte trois bandes pour les trois composantes de couleurs : rouge verte et bleue servant à représenter une image en couleur selon la combinaison des nuances. En matière d'imagerie en télédétection, de nombreux types de valeurs peuvent être utilisés pour représenter la valeur d'une zone représentée par un pixel. Par exemple un modèle numérique de terrain va permettre de connaître l'altitude sur un pixel de l'image. D'autres images peuvent contenir des indices de végétations, des indices de températures etc.

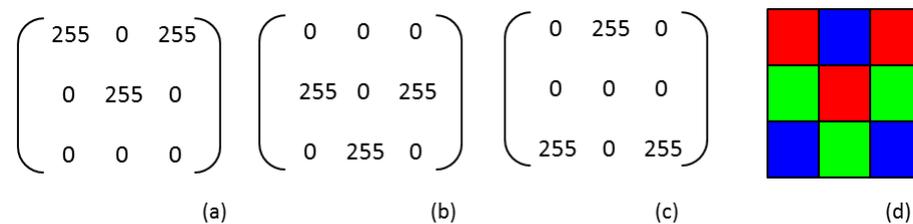


FIGURE 3.8 – Exemple de représentation d'une image par une matrice 3x3 avec trois bandes qui contiennent les composantes des couleurs rouge, verte et bleue.

Ce type de format permet de caractériser un espace régulier en attribuant une forme régulière pour chaque valeur de la matrice. La forme carrée du pixel d'une image est directement issue du format du capteur. Celle-ci est la plus largement répandue mais nous pouvons aussi noter que des capteurs, moins répandus, permettent d'obtenir de l'information avec une forme hexagonale (Delbrück, 1993). L'avantage principal de ce format est sa méthode de stockage des valeurs sur une matrice ainsi que l'apport d'un type d'information régulier sur une zone. Cette matrice est donnée par la définition 8. Le concept de datafacer est utilisé en Ocelet pour intégrer des données dans un modèle. Nous nommons "RasterFile" le datafacer spécifique qui permet de lire et écrire des données de ce type dans un modèle.

**Définition 8.** *La matrice de largeur  $l$ , de hauteur  $h$  et de nombre de bandes  $b$  sera notée  $M_{(l,h,b)}$ . Une bande de la matrice contient un même type d'information sur l'ensemble des pixels.*

Si la dimension spatiale des images apporte une information à un instant donné, les séries temporelles d'images apportent une dimension temporelle supplémentaire. Celles-ci sont largement utilisées dans le domaine de l'analyse spatiale et permettent par exemple de suivre les dynamiques du paysage (agricole, urbanisation, gestion des risques etc). L'intérêt de leur utilisation dans un modèle est alors de pouvoir obtenir de l'information spatiale à des temps donnés. La figure 3.9 présente une série fictive d'images décennales sur des données pluviométriques qui peuvent être utilisées à

l'intérieur d'un modèle pour amener de l'information sur chaque pixel et par décade par exemple.

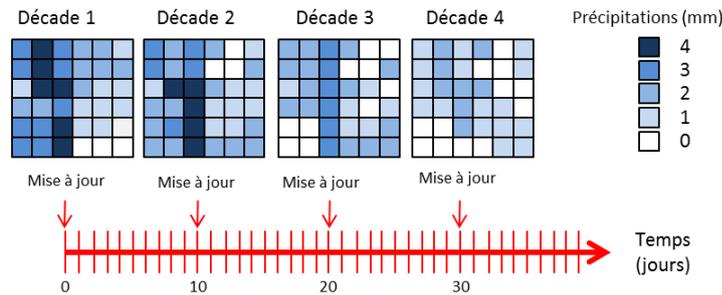


FIGURE 3.9 – Représentation d'une série temporelle contenant les données en précipitations et la possibilité de mettre à jour l'information spatiale à chaque décade lors de la simulation.

Comme pour la gestion des données raster dans un modèle en Ocelet, nous nommons "TemporalSeriesFile" le datafacer permettant de lire et écrire des données venant d'une série temporelle d'images. En respectant la définition 8, nous donnons en définition 9 une matrice avec la dimension temporelle supplémentaire.

**Définition 9.** Une série temporelle se note  $M_{n(l,h,b)}^m$  avec  $n$  la première matrice de la série et  $m$  la dernière matrice de la série.

Les données raster au format matriciel ne contiennent cependant que les valeurs des pixels, sans préciser leur forme géométrique. Pour en avoir une représentation, il faut leur définir une forme qui convienne à ce type de format, par exemple spécifier que ce sont des pixels de forme carrée. Les datafacers permettent uniquement de lire ou écrire des données. Il reste à définir comment utiliser ces données dans un modèle. Les sections suivantes présentent la méthode de gestion de ces données pour permettre de manipuler de l'information régulière avec le formalisme d'Ocelet.

### 3.4 Pavages réguliers

Une portion de l'espace euclidien peut être représentée par un pavage constitué d'éléments finis appelés tuiles, représentés par un polygone. La surface de l'espace est complètement recouverte par ces tuiles formant des structures répétitives. Un pavage est dit régulier si les tuiles ont la même forme, et si elles partagent un côté de même longueur (Grünbaum and Shephard, 1987). Pour respecter ces conditions, seules trois formes de polygones sont possibles : le carré, le triangle équilatéral et l'hexagone régulier. Des pavages avec ces formes sont présentés avec la figure 3.10. Nous nommerons par la suite une tuile par cellule. Un pavage régulier présente plusieurs caractéristiques qui facilitent l'échange d'information si l'on considère l'aspect

spatial de ces formes. En effet, la forme régulière implique que la représentation spatiale des cellules est la même pour toutes ces cellules. De plus la distance de voisinage et le nombre de voisins des cellules sont égaux (sans tenir compte des éléments au bord d'un espace fini).

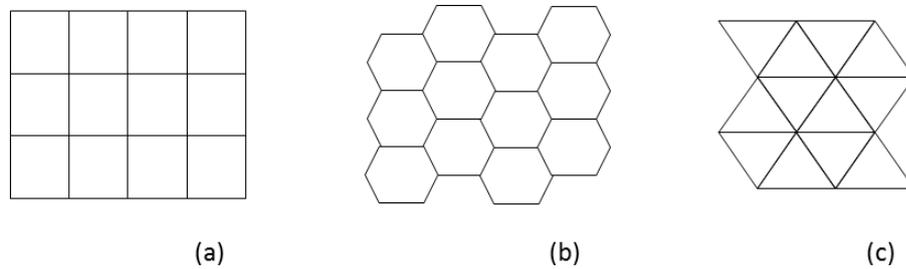


FIGURE 3.10 – Les différents pavages réguliers avec des formes géométriques régulières : carrées (a), hexagonales (b) et triangulaires (c).

D'autres pavages sont à noter comme les pavages semi-réguliers constitués de plusieurs formes de polygones avec une structure répétitive ou encore des pavages pentagonaux comme l'illustre la figure 3.11.

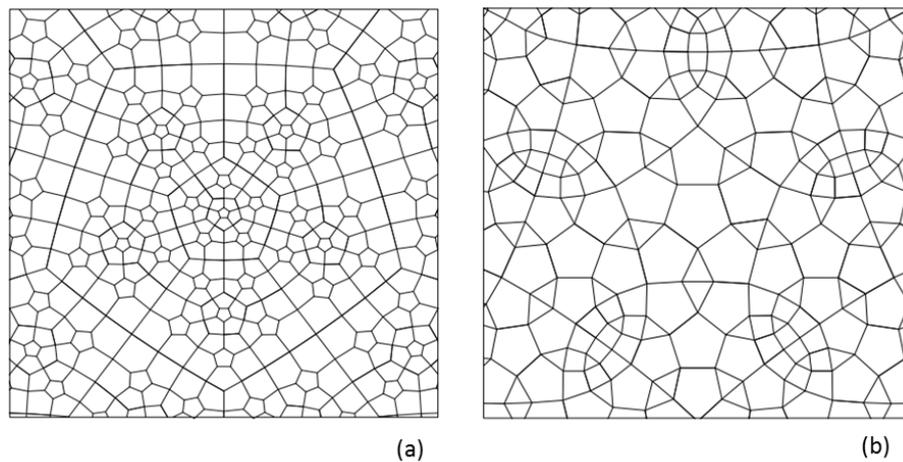


FIGURE 3.11 – Un pavage semi-régulier de plusieurs types de pentagones (a). Un pavage semi-régulier composé de pentagones, triangles et quadrilatères (b) (Bowers and Stephenson, 1997).

Comme nous l'avons vu dans le chapitre 2, de nombreuses approches utilisent des pavages réguliers : les automates cellulaires sont majoritairement simulés sur un pavage composé de carrés mais sont aussi définissables sur des pavages hexagonaux ou triangulaires par exemple. Nous avons aussi vu que c'était le cas pour des plateformes se basant sur des systèmes multi-agents comme NetLogo.

Nous analysons à présent comment intégrer les pavages réguliers dans le formalisme d'Ocelet en tenant compte de leurs caractéristiques :

- La régularité du pavage permet d'indexer les cellules et donc d'y accéder avec une forme d'adressage facile à manipuler.
- Il est possible de construire implicitement des arêtes entre une cellule et ses voisines grâce à cet adressage, et l'on peut même définir plusieurs formes de voisinage.
- La forme de chaque cellule n'a pas besoin d'être conservée en mémoire puisqu'elles sont toutes identiques.
- La régularité de la structure de graphe ainsi obtenue permet aussi certaines optimisations pour synchroniser les changements d'états lors de l'application d'une fonction d'interaction.

La première version d'Ocelet traite le cas général dans lequel tous les graphes qui sont manipulés sont des graphes explicitement construits. Chaque arête de ces graphes est donc construite puis conservée en mémoire par le moteur d'exécution d'Ocelet. Dans le cas des pavages réguliers d'un espace à deux dimensions, nous sommes dans la situation d'un ensemble de cellules doté d'une relation d'ordre spatial à deux dimensions qui présente plusieurs propriétés intéressantes. Nous proposons d'en tenir compte pour répondre à plusieurs objectifs et conditions :

- Une entité représentée par une géométrie doit pouvoir accéder à une information venant d'un pavage régulier par correspondance spatiale. La réciproque doit aussi être vérifiée.
- Une cellule doit pouvoir accéder aux propriétés de ses voisins.
- Un ou plusieurs pavages réguliers doivent pouvoir être utilisés dans un même modèle, quelles que soient la taille et la forme des cellules.
- Si plusieurs pavages réguliers sont définis dans un modèle et s'ils sont en correspondance spatiale, une cellule d'un pavage doit pouvoir avoir accès à l'information venant d'une cellule d'un autre pavage si elles se correspondent spatialement.
- Des formats de données apportent de l'information spatiale sur chaque cellule d'un pavage. Ceux-ci doivent pouvoir être utilisés à l'intérieur d'un modèle.

En considérant ces objectifs et conditions, nous retenons les choix suivants : une entité peut avoir une seule forme de représentation, soit une géométrie, soit par l'élément minimal d'un pavage, une cellule. L'ensemble des entités d'un même type, caractérisées par une cellule sont alors les constituants d'un pavage régulier. Les autres propriétés d'une entité spatialement représentée par une cellule caractérisent l'espace du pavage en un point. La définition de types différents d'entités ayant une cellule en propriété représente ainsi la définition de plusieurs pavages. L'état des entités doit pouvoir alors changer par l'application de fonctions d'interaction. Pour respecter le formalisme d'Ocelet et les définitions proposées dans la section 3.2, ces fonctions d'interaction doivent être appliquées sur les arêtes reliant les entités. Celles-ci doivent donc faire partie d'une structure de graphe qui sera différente selon les types d'entités :

- Un graphe où les sommets sont des entités d'un même type et qui ont une propriété "cellule".

- Un graphe où les sommets sont des entités ayant une propriété "cellule" et des entités non caractérisées par une cellule.
- Un graphe où les sommets sont des entités de types différents et ayant une cellule comme propriété.

La forme des cellules, leur position dans l'espace régulier / euclidien, permettent de proposer des structures de graphes implicites que nous décrivons dans les sections suivantes. Nous donnons premièrement, l'implication de la définition d'une nouvelle propriété caractérisant la forme d'une entité dans le langage.

## 3.5 Cellules

En Ocelet, la dynamique du modèle est traduite par un changement d'état des entités lorsque celles-ci sont en interaction. Les entités peuvent être caractérisées par une forme de représentation, qui, dans la première version du langage est un type de géométrie (présentée en section 3.2.4) qui caractérise une portion de l'espace d'une zone d'étude. Cette zone est alors définie par l'ensemble des entités géométriques et l'information contenue dans les propriétés de ces entités. La représentation d'une zone par un pavage régulier nous a conduit à pouvoir caractériser spatialement une entité par l'élément minimal d'un pavage : la cellule. La cellule devient alors la deuxième forme de représentation de l'information géographique dans Ocelet et vient caractériser une propriété spatiale d'une entité.

**Définition 10.** *La propriété cellule a trois caractéristiques : (i) Une forme carrée, triangulaire ou hexagonale. Les entités, pour représenter un pavage ont toutes la même forme. (ii) Une cellule comporte une dimension. Les cellules étant des formes régulières, la longueur d'un côté suffit à définir cette dimension. (iii) Une position dans le pavage régulier qui se traduit par une position matricielle, et une position sur l'espace euclidien déduite de la position matricielle.*

Une *entité cellulaire* est une entité dont une de ses propriétés est une cellule. Elle rejoint la définition 1 donnée sur les entités d'Ocelet. Cependant il faut rajouter une contrainte liée à la définition de l'espace représenté par un pavage régulier : la propriété cellule de l'entité ne peut subir aucune modification. Cette contrainte se justifie par le fait que si la forme d'une cellule est modifiée, le pavage n'est plus régulier. Seules les autres propriétés d'une entité peuvent donc évoluer au cours du temps. Les fonctions de consultation, de transition et d'interaction (définitions 3, 4 et 5) sont applicables sur ces propriétés. Une entité spatiale ne peut avoir qu'une seule forme de représentation : soit une géométrie, soit une cellule.

Si la forme d'une cellule ne peut être modifiée, celle-ci peut être utilisée avec une fonction de consultation en tant que géométrie. La forme géométrique d'une cellule est obtenue selon ses caractéristiques : type de la forme, position et dimension. La géométrie de la cellule peut alors être utilisée pour appliquer les opérations géométriques décrites dans la section 3.2.3. L'intégration de cette propriété dans

le langage passe par la définition d'un nouveau type de propriété dont le mot clef est "cell". La définition d'une entité avec ce type de propriété implique que chaque élément du pavage est une entité cellulaire d'un même type.

**Définition 11.** Soit  $Ec\{ec_1, ec_2, \dots, ec_n\}$  l'ensemble des entités cellulaires. Soit  $M_{(l,h,b)}$  la matrice donnée par la définition 8. Le nombre de bandes de la matrice est égal au nombre de propriétés  $n = |\{\alpha, \beta, \gamma, \dots\}|$  des entités cellulaires appartenant à  $Ec$ . Nous noterons l'ensemble des bandes de la matrice  $B\{b_\alpha, b_\beta, b_\gamma, \dots\}$ . La valeur d'une propriété  $\alpha$  de l'entité  $ec$  est contenue sur la matrice à cette position sur la bande. Ainsi la valeur de la propriété est donnée directement par  $M(x, y, b_\alpha)$ . Les valeurs des propriétés de l'ensemble d'entités  $Ec$  au temps  $t$  sont donc contenues dans la matrice au même temps que nous notons  $M_{t(l,h,b)}$ . Avec la définition 2 et la définition 9, la succession d'états pour l'ensemble des entités cellulaires est notée  $Ec_n^m$  et correspond à  $M_{n(l,h,b)}^m$ . L'ensemble de l'historique des états, noté  $\vec{Ec}$  correspond à  $M_{(l,h,b)}^{\vec{}}$ .

**Définition 12.** Une fonction de consultation  $f_c$  ou de transition  $f_t$  sur une entité cellulaire  $ec$  agissant sur les propriétés  $P\{\alpha, \beta, \gamma, \dots\}$  agissent donc sur la matrice pour les bandes  $B\{b_\alpha, b_\beta, b_\gamma, \dots\}$ .

Cette section a mis en avant la définition d'un nouveau type de propriété spatiale ainsi que son utilisation formelle en Ocelet pour caractériser spatialement une entité. Nous présentons par la suite la gestion d'un ensemble d'entités caractérisées par cette propriété qui constitue alors un pavage régulier.

### 3.6 Pavages réguliers et graphes d'interaction

Nous avons vu qu'une entité cellulaire est définie par l'usage d'une nouvelle forme de propriété spatiale de forme régulière : la cellule. L'ensemble des entités cellulaires d'un même type de cellule (même forme, même dimension) constitue alors un pavage régulier :

**Définition 13.** Nous notons  $Pv(M_{(l,h,b)}, Ec)$  un pavage régulier d'entités  $Ec$ . La forme de l'élément minimal du pavage est donnée par la forme de la propriété "cell" du type d'entité  $Ec$ . L'ensemble des informations des entités est contenu dans la matrice  $M_{(l,h,b)}$ . Ainsi chaque définition d'un type d'entité cellulaire correspond à un pavage régulier.

La description de la dynamique du modèle passe par un enchaînement de fonctions d'interaction entre les entités connectées dans un graphe. Nous abordons les différents cas de graphes d'interaction qu'il est possible de concevoir lorsque ceux-ci contiennent des entités appartenant à un pavage régulier.

### 3.6.1 Entités cellulaires et graphe d'interaction

Pour décrire un système, Ocelet utilise des relations représentées par des graphes où chaque sommet est une entité. Il appartient généralement au modélisateur de construire ses propres graphes d'entités. Du fait de la structure d'un pavage régulier, nous proposons une construction implicite d'un graphe entre toutes les entités de même type qui ont une cellule en propriété. Ainsi l'utilisateur manipule directement un graphe d'entités cellulaires qui sont connectées à leurs cellules voisines selon leurs formes (triangle, carrée ou hexagonale). Nous pouvons distinguer plusieurs types de voisinage. Le premier détermine les cellules voisines lorsque celles-ci partagent un côté en commun, aussi connu sous le nom de voisinage de Von Neumann. Le deuxième pouvant tenir compte du voisinage des cellules si celles-ci partagent un sommet, aussi connu sous le nom de voisinage de Moore. Notons que dans les cas des cellules hexagonales, seuls leurs côtés sont partagés, comme le présente la figure 3.12.

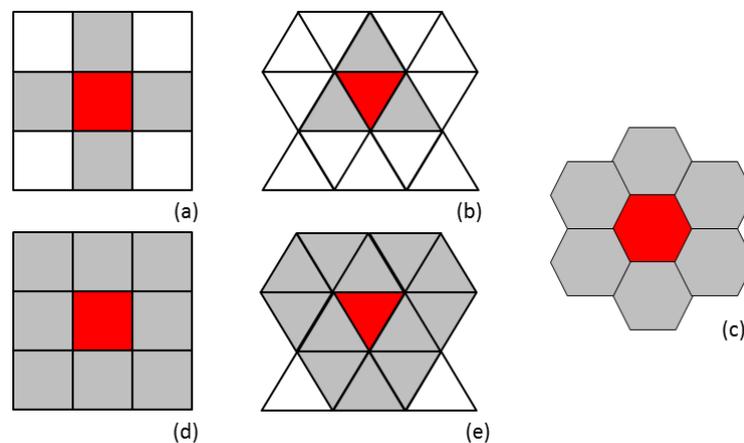


FIGURE 3.12 – Le voisinage de Von Neumann pour des cellules carrées (a), triangulaires (b) et hexagonales (c). Le voisinage de Moore pour des cellules carrées (d) et triangulaires (e).

Plusieurs métriques de distances comme la distance de Manhattan (Girard and Girard, 2010) et la distance de Tchebychev (Cantrell, 2000) permettent de faire la correspondance entre une distance euclidienne et une distance sur un espace régulier. Le voisinage de Von Neumann correspond à une distance de Manhattan de 1 et le voisinage de Moore une distance de Tchebychev de 1 par exemple.

Pour des entités  $e_i$  et  $e_j$  dont les propriétés  $\alpha$  sont des cellules de coordonnées  $(x_i, y_i)$ , la distance de Manhattan correspond à l'équation 3.5 et la distance de Tchebychev à l'équation 3.6.

$$d(e_1, e_2) = |(e_1 : \alpha : x_1) - (e_2 : \alpha : x_2)| + |(e_1 : \alpha : y_1) - (e_2 : \alpha : y_2)| \quad (3.5)$$

$$d(e_i, e_i) = \max_{i \in (0, n)} (|(e_i : \alpha : x_i), (e_i : \alpha : y_i)) - ((e_i : \alpha : x_i), (e_i : \alpha : y_i))|) \quad (3.6)$$

Nous proposons ainsi de tenir compte de ces distances notées  $S$  suivant le type de voisinage des graphes aussi connues sous le nom de voisinage de Moore étendu ou voisinage de Von Neumann étendu. A ces distances nous rajoutons celle décrivant le rayon d'un cercle dont le centre est la cellule concernée.

Nous notons  $Vois(T, S)$  le type de voisinage avec  $T = \{Moore, VonNeumann, Cercle\}$  et  $S$  la distance de voisinage. Chaque type de voisinage implique une gestion par graphe différente. Le nombre de voisins évoluant alors pour une distance  $S$  :  $(2S + 1)^2 - 1$  pour une distance de Tchebychev et  $2S(S + 1) + 1$  pour une distance de Manhattan. La figure 3.13 présente les voisins pour une distance ou rayon de 2 pour la cellule centrale.

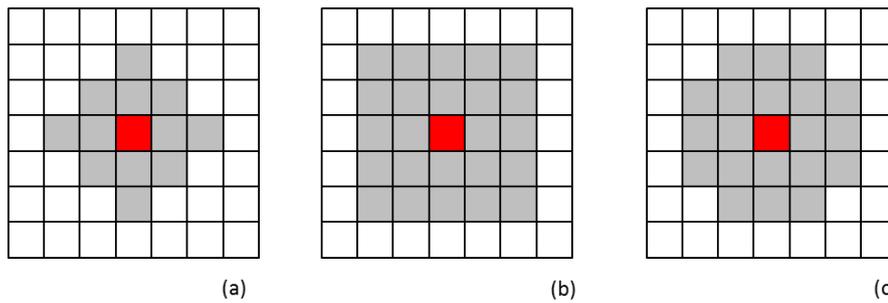


FIGURE 3.13 – Représentations des voisins pour une cellule centrale en rouge : en (a) un voisinage de Von Neumann de distance 2, en (b) un voisinage de Moore de distance 2 et en (c) un voisinage selon un cercle de rayon 2.

Un graphe d'interaction d'entités cellulaires d'un même type est donc constitué d'un pavage régulier, d'une spécification sur le type de voisinage et d'un ensemble de fonctions d'interaction permettant de changer l'état des propriétés des entités  $Ec$  sur la matrice.

**Définition 14.** Soit  $Pv$  un pavage régulier donné en définition 13 constitué d'entités  $Ec$ . Soit  $Fi$  l'ensemble des fonctions d'interaction prenant en compte les propriétés des entités appartenant à  $Ec$ . Un graphe d'interaction selon un type de voisinage se note d'après la définition 6 :  $\Gamma = (Pv(M_{(l,h,b)}, Ec), Vois(T, S), Fi)$ .

Ce type de graphe a des propriétés caractéristiques prédéfinies avec par exemple : l'ordre du graphe qui est obtenu par le produit du nombre de colonnes et du nombre

de lignes de la matrice  $M_{(l,h,b)}$  du pavage. Le graphe est topologique planaire pour un voisinage de Von Neumann avec une distance de 1.

La figure 3.14 présente un pavage carré basé sur des données fictives d'un modèle numérique de terrain. L'image a une dimension de  $6 \times 6$  pixels, chaque pixel représentant une valeur d'élévation de 0 à 5 mètres (figure 3.14 (a)). A partir de ce pavage, on peut déduire un graphe implicite d'ordre 36. La représentation du graphe en utilisant le voisinage de Von Neumann est ici utilisé.

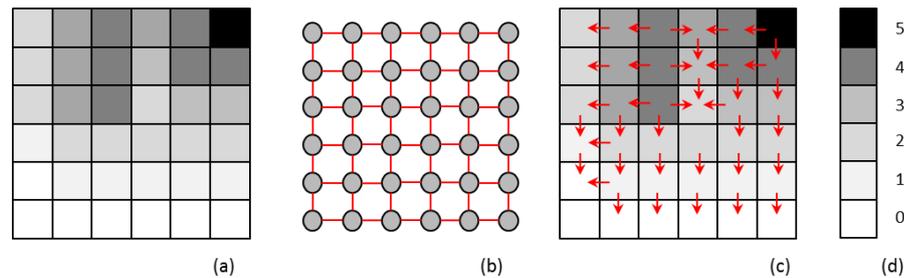


FIGURE 3.14 – En (a) la matrice de dimension  $6 \times 6$  contenant les valeurs d'altitude correspondant à l'échelle en (d). En (b) l'ensemble des arêtes sur cette matrice avec un voisinage de Von Neumann. En (c) l'orientation des arêtes selon l'altitude des cellules.

Le graphe proposé applique les interactions sur chaque entité, les arêtes étant non orientées ou bi-directionnelles (figure 3.14 (b)). Les interactions peuvent inclure des fonctions de tests qui vont indiquer quelle entité est affectée par un changement par rapport aux entités voisines. Dans notre exemple nous pouvons considérer que chaque entité va influencer une entité voisine selon son altitude si celle-ci est surélevée (figure 3.14 (c)). Nous considérons que ce sont des cellules d'un mètre de résolution. L'entité caractérisée par la cellule peut avoir d'autres propriétés comme une quantité d'eau que nous décidons d'ajouter à notre exemple avec une valeur de 4mm par  $m^2$ . La figure 3.15 montre la transformation de cette quantité d'eau en appliquant l'interaction proposée de la figure 3.14 (c).

Ce type de graphe propose de manipuler des entités cellulaires d'un type représentant une zone par un pavage régulier. La prochaine section décrit la manipulation des entités cellulaires avec Ocelet avec des entités pouvant être représentées ou non par une géométrie.

### 3.6.2 Entités cellulaires, géométriques, sans représentation spatiale et graphe d'interaction

Si la gestion de graphe d'entités cellulaires permet d'aborder l'aspect spatial régulier, il est peu adapté pour représenter des entités pouvant caractériser une portion de l'espace correspondant à des objets dont la taille peut dépasser largement celle d'une cellule (et inversement). Le format vecteur, manipulé depuis longtemps

4	4	4	4	4	4
4	4	4	4	4	4
4	4	4	4	4	4
4	4	4	4	4	4
4	4	4	4	4	4
4	4	4	4	4	4

8	2	0	6	2	0
8	2	0	8	2	2
8	1,3	0	7,3	4	2
6	2	1,3	0	2	4
6	2	4	4	4	4
4	6	8	8	8	8

(a) (b)

FIGURE 3.15 – En (a) l'état initial  $t = 1$  des entités cellulaires avec une valeur pour les propriétés "eau" de 4. En (b) le résultat au temps  $t = 2$ .

dans les systèmes d'information géographique, s'avère dans ce cas plus approprié (des limites administratives comme un parcellaire représenté par des polygones, des formations du paysage comme des rivières représentées par des linéaires).

La construction de ces objets du paysage se fait souvent de deux manières : l'une étant issue de l'analyse et de processus d'extraction comme la segmentation et la classification d'images. L'autre revenant directement à l'expert qui va construire ses objets selon sa connaissance et selon les buts de son étude. Si ces objets ont une signification pour l'expert, la trace des informations ayant été utilisées pour les définir n'est pas conservée. L'information servant à les construire est généralement une image dont les valeurs sont directement issues d'un capteur (embarqué par un satellite par exemple). Le capteur va de plus renvoyer un ou plusieurs types de données hétérogènes sur une zone. Des informations venant d'autres types de capteurs pouvant être utilisées a posteriori dans le modèle, celles-ci n'auraient alors plus aucun lien avec les objets construits. La figure 3.16 présente un exemple fictif qui montre la construction de trois objets vecteurs à partir d'une autre couche de données régulières.

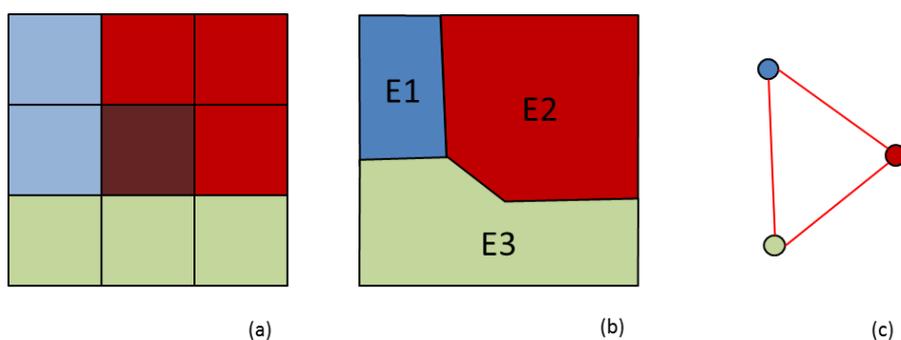


FIGURE 3.16 – Schéma de la représentation d'un espace régulier en (a) et la représentation en objets, ou géométries en (b). Le graphe de voisinage sur les géométries est présenté en (c).

Considérons qu'il s'agit de surfaces d'un type donné avec en rouge une surface urbanisée, en bleu une surface en eau et en vert une surface végétalisée. L'interprétation des données régulières (figure 3.16 (a)) peut conduire à la construction d'objets homogènes définis par une géométrie. Ces géométries caractérisent spatialement trois entités  $E1$ ,  $E2$  et  $E3$  (figure 3.16 (b)). Le graphe de voisinage par adjacence de ces objets étant représenté par la figure 3.16 (c). Si le but du modèle est de calculer un échange selon l'altitude de ces objets il faudrait alors pouvoir leur attribuer une valeur en altitude venant d'une autre couche de données en correspondance spatiale. Nous cherchons à utiliser le formalisme d'Ocelet pour manipuler des entités "objet" d'un modèle pouvant être reliées aux entités cellulaires régulières afin de retrouver les données hétérogènes correspondantes et les utiliser dans la définition de processus de modélisation.

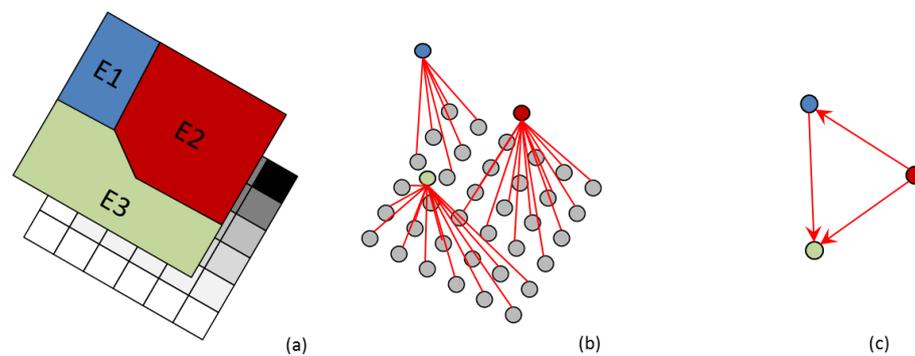


FIGURE 3.17 – En (a) les couches superposées spatialement d'entités géométriques avec des entités cellulaires. En (b) le graphe permettant à ces entités d'interagir. En (c) le graphe orienté d'entités géométriques après avoir obtenu l'information sur l'altitude venant des entités cellulaires.

La relation entre des entités géométriques et des entités cellulaires se construit aussi de manière implicite en faisant correspondre spatialement les géométries et les cellules des entités comme l'illustre la figure 3.17 (b).

**Définition 15.** Soit un pavage régulier donné par la définition 13 constitué d'entités  $E_c$  et soit  $E_g$  l'ensemble des entités géométriques d'un type. Un graphe d'interaction entre les entités cellulaires  $E_c$  et  $E_g$  se note :  $\Gamma = (Pv(M_{(l,h,b)}, E_c), U, Fi)$  avec  $U = E_c \times E_g$  le produit cartésien déduit de la correspondance spatiale entre les entités cellulaires et les entités géométriques. Ce type de graphe est biparti.

Pour montrer le contraste qui existe entre des processus sur différentes formes de représentation de l'espace, nous schématisons un échange de quantités d'eau sur les entités caractérisées par une géométrie. Celles-ci auront une altitude moyenne déduite de l'espace régulier. A partir de cette donnée nous pouvons orienter les interactions dans le graphe de voisinage sur les objets (figure 3.17 (c)). Ce premier exemple montre comment des valeurs hétérogènes peuvent aider à définir des comportements d'entités représentées par des objets. Cependant l'exemple montre aussi

que les interactions entre les entités géométriques deviennent floues et perdent de l'intérêt comparé aux entités cellulaires, plus fines selon certains cas. Ce phénomène est précisé avec la figure 3.18.

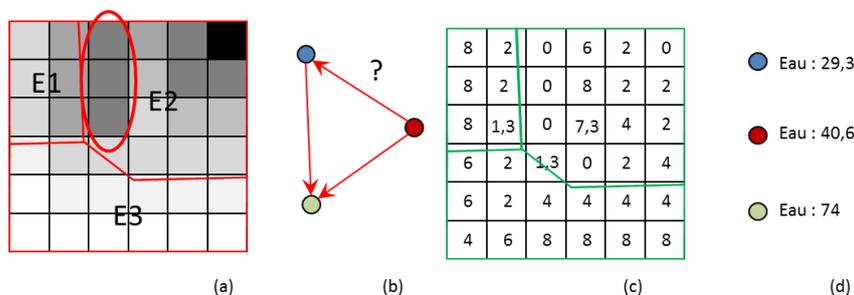


FIGURE 3.18 – En (a) des flux qui ont un chemin différent qui n'est pas traduit sur les entités géométriques (b). Le calcul de flux au niveau des entités cellulaires permet de palier ce problème et de rapporter le résultat sur les entités géométriques correspondantes en (c) et (d).

Nous voyons ainsi que les interactions entre entités géométriques ne tiennent pas compte de l'hétérogénéité des valeurs de l'altitude. En effet si c'était le cas l'interaction entre les entités  $E1$  et  $E2$  ne pourrait traduire qu'un échange partiel. La gestion des interactions entre des entités cellulaires développée dans la section précédente permet de gérer cette hétérogénéité. Les interactions entre entités cellulaires et entités géométriques permettent dans un deuxième temps d'affecter les valeurs issues de l'interaction au niveau de l'espace d'un pavage vers l'espace représenté par des géométries(cf. figure 3.18 (c) et figure 3.18 (d)).

Le type de relation qui permet de décrire des interactions entre entités cellulaires et entités géométriques est un bon support pour prendre en compte le changement de représentation. Dans certains cas le modélisateur est amené à définir des entités sans représentation spatiale. La création de relations incluant ce type d'entité est alors explicite et il revient au modélisateur de spécifier les méthodes de connexions entre ces entités et des entités cellulaires. La définition de ce type de graphe d'interaction rejoint la définition 15. Le produit cartésien entre les entités du graphe étant alors explicite.

La figure 3.19 illustre comment attribuer aux entités cellulaires une entité sans représentation. Dans ce cas une station pluviométrique permettant d'attribuer la quantité de pluie des entités cellulaires à un temps donné en utilisant une fonction d'interaction.

Cette partie concernait la gestion des entités cellulaires vers des entités ayant une représentation géométrique ou sans représentation spatiale en tenant compte du formalisme d'Ocelet et des graphes d'interaction. La section suivante décrit la conceptualisation pour gérer des graphes entre entités cellulaires de types différents.

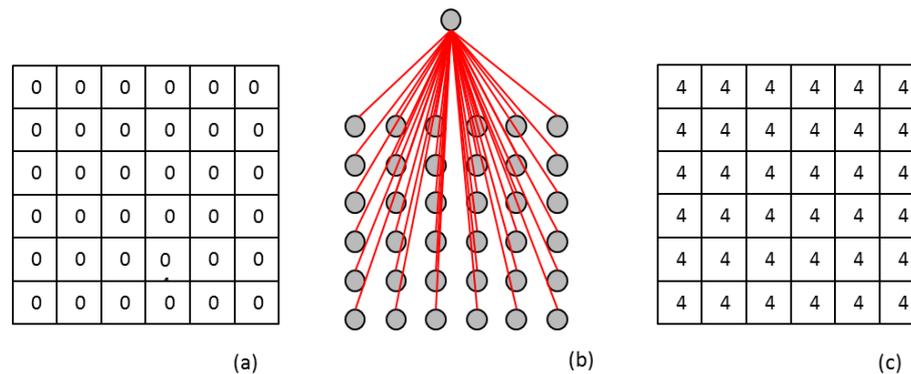


FIGURE 3.19 – En (a) la matrice représentant les entités cellulaires et leurs valeurs en eau. En (b) un graphe entre les entités cellulaires et une entité sans représentation spatiale. Une interaction entre la cellule sans représentation spatiale et les entités cellulaires permet d'affecter une valeur en eau pour toutes les entités cellulaires connectées (c).

### 3.6.3 Entités cellulaires de différents types et graphe d'interaction

Si les capteurs embarqués permettent d'obtenir de l'information de différents types, le type et l'altitude du capteur détermine aussi la résolution de l'image. La taille et la valeur du pixel donnent ainsi la mesure d'une valeur déterminée par le capteur et les conditions de prise de vue pour une zone donnée. Les informations venant d'images de résolutions différentes devaient alors pouvoir être utilisées dans le formalisme d'Ocelet. En plus de pouvoir manipuler de l'information de différentes résolutions (ou différentes échelles spatiales), des processus de modélisation doivent pouvoir être définis aussi à des échelles différentes. Si des processus sur plusieurs espaces réguliers d'échelles différentes sont possibles avec des relations sur chacun de ces espaces, il faut étudier comment le lien entre de tels espaces peut être réalisé. Dans cette optique, un graphe implicite est proposé entre les entités cellulaires d'une couche vers les entités cellulaires d'une autre couche, ces couches pouvant avoir des résolutions ainsi que des formes différentes, les arêtes du graphe relient ces entités par correspondance spatiale.

**Définition 16.** Soit  $Pv$  un pavage régulier d'entités cellulaires  $Ec$  et  $Pv'$  un autre pavage régulier d'entités cellulaires  $Ec'$  différentes de  $Ec$  suivant la définition 13. Un graphe d'interaction entre les entités cellulaires  $Ec$  et  $Ec'$  se note :  
 $\Gamma = (\{Pv(M_{(l,h,b)}, Ec), Pv'(M'_{(l',h',b')}, Ec')\}, U, Fi)$  avec  $U = Ec \times Ec'$  le produit cartésien déduit de la correspondance spatiale entre les entités cellulaires de types différents. Ce type de graphe est biparti.

La figure 3.20 (a) présente deux couches de résolutions différentes dont la couche représentant le modèle numérique de terrain, de résolution plus fine ainsi que la

couche ayant servi à déterminer les objets de la partie précédente. Les entités cellulaires sont connectées implicitement dans un graphe de relation dans la figure 3.20 (b). Des interactions peuvent alors être définies pour spécifier le changement d'états entre ces entités, une interaction simple sur la moyenne des valeurs du modèle numérique de terrain permet d'attribuer une altitude moyenne aux entités cellulaires de résolution moins fine (figure 3.20 (c)).

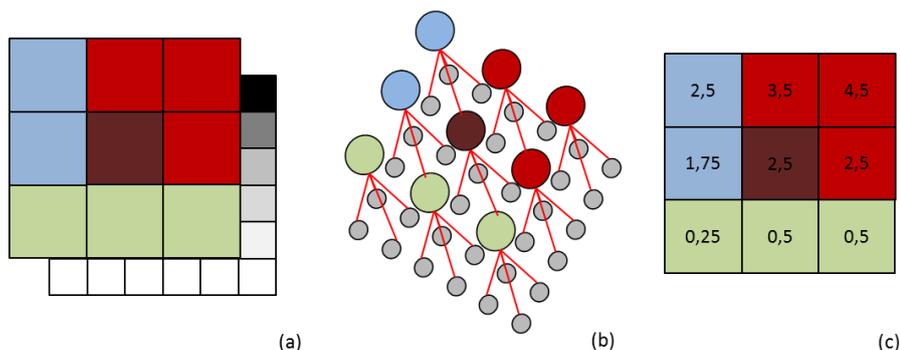


FIGURE 3.20 – En (a) deux couches d'entités cellulaires de résolutions différentes. En (b) le graphe permettant de lancer des interactions sur ces entités. En (c) l'affectation des propriétés "mnt" des cellules de résolution la plus fine sur les entités cellulaires de la deuxième couche par interaction.

Si la figure 3.20 montre une simple attribution de valeurs, des processus plus complexes peuvent être définis dans une interaction. De plus ceux-ci peuvent être appelés après avoir lancé des interactions sur chacune de ces couches comme présenté dans la section 3.6.1 et les quantités d'eau calculées dans le premier exemple pourraient venir affecter l'état de la couche moins fine.

Ces sections ont ainsi permis de présenter la manipulation d'entités cellulaires dans le formalisme d'Ocelet en termes de gestion de relations et de graphes d'interaction.

### 3.7 Conclusion

Un choix est généralement fait entre les deux formats de représentation de l'espace, régulière ou par des géométries (format raster ou format vecteur). Ocelet propose une méthode générique de conceptualisation des modèles à base de graphes représentant les relations entre les éléments d'un système : les entités, les arêtes des graphes permettant de définir des fonctions traduisant les interactions entre ces entités. Dans le cadre de la modélisation des dynamiques spatiales, les entités peuvent être caractérisées par une représentation spatiale. Si la première version du langage Ocelet a pris en compte la représentation des entités d'un modèle en les caractérisant par une géométrie, il restait à intégrer la gestion de la représentation par un espace découpé régulièrement : par un pavage régulier.

Après avoir défini formellement les concepts d'Ocelet et donné des définitions sur les éléments du langage pour concevoir un modèle, nous avons abordé les définitions pour le format raster ainsi que sur la représentation d'un espace par un pavage régulier. En étudiant plus particulièrement la forme élémentaire et répétitive d'un pavage, nous avons proposé d'en tenir compte en définissant une nouvelle propriété permettant de caractériser une entité : la cellule qui correspond à la représentation d'un élément du pavage et qui, pour constituer un pavage régulier, ne peut correspondre qu'à trois formes régulières : le carré, l'hexagone régulier et le triangle équilatéral. En tenant compte de cette forme de représentation et du formalisme des graphes d'interaction nous avons pu déterminer plusieurs structures de graphes implicites sur lesquels appliquer des fonctions d'interaction et faire évoluer l'état des entités de ces graphes :

- La première structure de graphe traduisant les interactions entre entités représentées par une cellule appartenant au même pavage. Le voisinage étant déterminé implicitement à partir de la forme de la cellule.
- La deuxième structure de graphe traduisant les interactions entre entités représentées par des cellules avec des entités représentées par des géométries ou sans représentation spatiale. Ce type de graphe permet ainsi de lier les entités de différentes représentations et de pouvoir y appliquer des fonctions d'interaction.
- La troisième et dernière structure de graphe permet de faire le lien entre des entités cellulaires de différents pavages sur une même zone. Ce type de graphe, implicite par la correspondance spatiale entre les entités permet d'appliquer des fonctions d'interaction sur des entités ayant une représentation régulière mais de formes ou de résolutions différentes.

Nous avons ainsi proposé différents graphes d'interactions dont la structure est implicitement construite selon les formes de représentation de l'espace. La conception d'un modèle en Ocelet se fait ainsi toujours en manipulant des graphes, en y appliquant des fonctions d'interaction, quelle que soit la caractéristique spatiale de l'entité. Si nous avons montré que le formalisme d'Ocelet convenait pour pouvoir manipuler deux formes de représentation de l'espace. La construction des modèles avec le langage Ocelet en tenant compte de l'intégration formelle des pavages réguliers sous le formalisme d'Ocelet a fait l'objet de développements particuliers qui sont traités dans le chapitre suivant.



# Chapitre 4

## Implémentation

### Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>63</b>
<b>4.2</b>	<b>Développements sur le langage</b>	<b>64</b>
<b>4.3</b>	<b>Gestion des données matricielles</b>	<b>65</b>
<b>4.4</b>	<b>Lecture et écriture des données matricielles</b>	<b>66</b>
4.4.1	Image raster	66
4.4.2	Série temporelle d'images	67
<b>4.5</b>	<b>Implémentation et gestion de graphes</b>	<b>69</b>
4.5.1	Graphe entre entités cellulaires de même type	71
4.5.2	Graphes entre entités cellulaires et entités géométriques ou sans représentation spatiale	81
4.5.3	Graphes entre entités cellulaires de types différents	86
<b>4.6</b>	<b>Conclusion</b>	<b>90</b>

---

### 4.1 Introduction

Ce chapitre est consacré aux méthodes d'implémentation que nous avons développées pour permettre la gestion des pavages réguliers en Ocelet. Participer au développement d'un langage de programmation requiert une familiarisation avec les méthodes de construction du langage, de l'analyse de syntaxe, en passant par l'analyse sémantique jusqu'à la génération de code permettant de compiler un modèle en Ocelet. L'intégration de la forme de représentation avec des formes régulières amène à de nombreux développements dans le langage Ocelet, allant de la manière d'utiliser les éléments de cet espace jusqu'à la manipulation de l'ensemble de ces éléments en utilisant des graphes. La définition d'interaction entre les entités appartenant à de tels graphes nécessite des algorithmes particuliers pour chaque cas de graphes d'interaction présentés dans le chapitre 3 :

- Le premier étant un graphe entre les entités cellulaires d'un même pavage régulier.
- Le deuxième étant un graphe entre les entités représentées par des cellules et les entités représentées par des formes géométriques.
- Le dernier graphe étant sur des entités cellulaires de couches différentes.

Si l'implémentation effectuée sur ces graphes permet de manipuler les entités cellulaires dans un modèle Ocelet, d'autres développements ont été menés pour faciliter la lecture de données raster externes :

- un datafacier *RasterFile* permettant de lire une image qui va permettre d'alimenter en données raster les graphes d'entités cellulaires.
- un datafacier *TemporalSeriesFile* permettant de mettre à jour les données matricielles d'un graphe avec des données contenues dans une série de fichiers au format raster.

La première section donne des éléments pour effectuer ces implémentations dans le compilateur du langage Ocelet, les sections suivantes concernent les implémentations effectuées pour la gestion des pavages réguliers par le langage ainsi que les différents graphes énumérés précédemment.

## 4.2 Développements sur le langage

Les premiers travaux sur le langage Ocelet sont décrits par Degenne (2012). Le langage, comme beaucoup de langages métiers, s'appuie sur un langage généraliste, dans le cas présent il s'agit de Java. Le compilateur d'Ocelet passe par une phase d'analyse lexicale qui consiste à découper le code écrit en Ocelet en lexèmes, ou unités lexicales. Ces unités lexicales sont ensuite traitées par un analyseur syntaxique qui va les représenter en une structure syntaxique sous forme arborescente (AST : Abstract Syntax Tree). La sémantique du programme est ensuite analysée en effectuant un contrôle de typage sur chaque sommet de cet arbre syntaxique. Celui-ci est ensuite transformé en arbre syntaxique Java, ce qui permet de générer les classes Java représentant le code décrit en Ocelet. Le code généré lors de cette dernière étape s'appuie largement sur un ensemble de classes prédéfinies qui constituent le moteur d'exécution.

L'exécution d'un programme revient alors finalement à exécuter un programme en Java dont une partie des classes a été générée et une autre partie se trouve dans le moteur d'exécution.

La première version du compilateur d'Ocelet reposait sur le compilateur de compilateur Tatoon (Cervelle et al., 2006) développé à l'Université Paris-Est, Marne-La-Vallée. Depuis, la solution a été migrée et repose sur l'environnement de développement **xtext** (Bettini, 2013). Celui-ci propose un ensemble d'outils pour la création de langages métiers qui simplifie les phases d'analyses lexicales et syntaxiques (figure

4.1) facilitant ainsi la production d'un environnement de développement à partir du module RCP (Rich Client Platform) d'Eclipse (Rubel, 2006). L'OMP (Ocelet Modelling Platform, annexe A) est l'environnement qui permet de faciliter la programmation de modèles en langage Ocelet : compilation, gestion des fichiers sources, lancement de simulations etc.

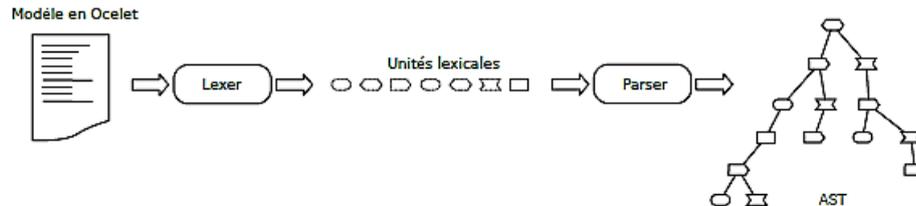


FIGURE 4.1 – Schéma des phases d'analyses lexicales et syntaxiques.

Nous n'avons pas eu besoin de modifier la grammaire du langage Ocelet pour réaliser les travaux de cette thèse. Les analyseurs lexicaux et syntaxiques n'ont donc pas été modifiés. Par contre le moteur d'exécution a été enrichi avec un ensemble de classes nécessaires à l'ajout du type de propriété *Cell*, et à la gestion de différentes formes de graphes d'interaction basés sur des structures de pavages réguliers.

Nous décrivons principalement dans les sections qui suivent, les ajouts effectués au moteur d'exécution ainsi que les algorithmes développés pour traduire les parcours de différentes formes de graphes.

### 4.3 Gestion des données matricielles

L'exécution d'une fonction d'interaction se fait sur l'ensemble des arêtes d'un graphe de relation. Lors de l'appel d'une telle fonction, le moteur d'exécution d'Ocelet parcourt toutes les arêtes d'un graphe de relation et applique la fonction définie. Toutes les valeurs affectées à des propriétés par la fonction sont gardées en mémoire par les entités. Le changement d'état effectif des entités n'intervient que lors d'une phase de synchronisation à la fin du parcours des arêtes. La résolution, la taille des images ainsi que les types de données impliquent de devoir gérer des graphes avec un très grand nombre de sommets et d'arêtes. La manipulation de plusieurs couches raster en instanciant une classe d'entité pour chaque cellule ainsi que pour les arêtes risque de s'avérer rapidement excessif en consommation d'espace mémoire. Pour répondre à ce type de problème, nous avons simulé l'instanciation des entités et des arêtes. Dans le cas des entités cellulaires, chaque propriété correspond à la valeur contenue sur une bande de la matrice à la position de la cellule de cette entité. Ainsi pour une matrice  $4 \times 4$ , soit 16 entités cellulaires, la définition de deux propriétés pour ce type d'entité est représentée par la figure 4.2.

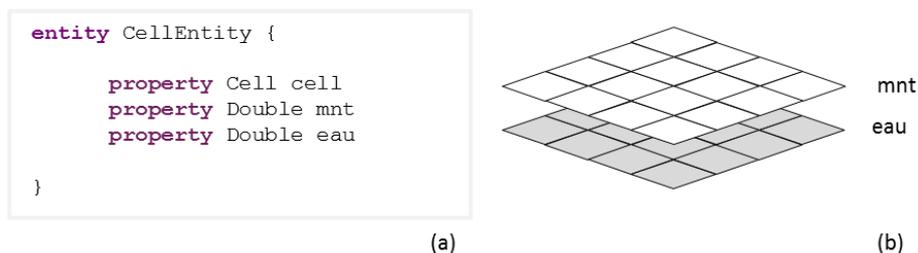


FIGURE 4.2 – Définition d’une entité cellulaire en Ocelet, caractérisée par une propriété "cell" (a). Cette déclaration correspond à la création de deux bandes de matrice pour deux propriétés de l’entité (b).

Lorsque l’utilisateur définit une entité avec une propriété cellule, une bande de la matrice est créée. La taille de la matrice est spécifiée lors de la lecture de l’image au format raster ou par génération sur une zone donnée. Nous partons du principe qu’une seule instance d’entité cellulaire permet de simuler des entités. Il s’agit du même objet en mémoire mais les caractéristiques des propriétés changent dès lors que la position en  $x$  et  $y$  de sa propriété cellule change. Cette gestion des cellules entraîne une limitation quant à leur utilisation. Il n’est dès lors pas possible de stocker des entités cellulaires dans une liste par exemple. Les méthodes utilisables par le modélisateur sur la propriété "cell" sont données en annexe B.1.

Nous verrons que cette approche est adaptable pour appliquer des fonctions d’interaction sur ces entités. Dans ce cas, de tels graphes, où les sommets sont composés en totalité ou en partie d’entités cellulaires, doivent avoir un parcours simulé que nous détaillerons dans les sections suivantes. Nous verrons dans un premier temps comment faciliter la lecture des images au format raster pour assigner les valeurs des bandes de l’image aux propriétés des entités cellulaires d’un modèle.

## 4.4 Lecture et écriture des données matricielles

### 4.4.1 Image raster

Le concept de datafacer a été introduit dans Ocelet pour aider le modélisateur à intégrer des données et à générer des entités. Il permet de lire des sources de données telles que des Shapefiles, CSV ou encore accéder à des bases de données. L’intégration des données raster dans Ocelet passe donc par la création d’au moins un datafacer permettant de lire les différents formats d’images. Pour ce faire nous utilisons la bibliothèque Java GDAL (Warmerdam, 2008) qui permet de gérer les différents formats d’images (geotiff, formats propriétaires, etc). Au lieu de retourner

une liste d'entités comme les autres datafacers, le datafacer *RasterFile* retourne une matrice de valeurs où chaque bande correspond à un ensemble ou sous ensemble de propriétés d'une entité cellulaire. La figure 4.3 donne exemple de définition d'un datafacer pouvant lire un fichier raster dont les valeurs seront attribuées aux entités cellulaires. Il est important de noter que cette définition de datafacer est totalement cohérente avec l'usage des autres datafacers du langage. Un modélisateur habitué à manipuler Ocelet n'aura aucun mal à comprendre comment faire appel à *RasterFile* dans ce contexte.

```
datafacer MyRaster{  
  
    store RasterFile("data/myDTM.tif", "EPSG:2975")  
  
    match CellEntity {  
        mnt : "0"  
    }  
}
```

FIGURE 4.3 – Définition d'un datafacer permettant de lire des données raster en Ocelet.

Pour gérer la lecture des différents formats d'images, la classe *RasterFile* est créée dans le moteur d'exécution d'Ocelet. Cette classe est héritée lorsque le mot clé *RasterFile* est utilisé à la suite de *store* (figure 4.3), la classe générée étant nommée selon l'identifiant spécifié à la suite de datafacer. Dans cet exemple (figure 4.3), la classe "MyRaster" sera générée et héritera de la classe *RasterFile* du moteur d'exécution. Des méthodes supplémentaires sont générées dans cette classe pour faire la correspondance entre les données de l'image raster et la matrice qui contient les valeurs de propriétés des entités cellulaires. Cette correspondance est ici spécifiée avec le mot clef "match". Dans la figure 4.3 les propriétés "mnt" des entités auront la valeur correspondante à la bande 0 de l'image. Ainsi en utilisant ce datafacer, on obtient directement la matrice contenant des valeurs pouvant être attribuées à partir d'une image sur une zone spécifique. La matrice sert ensuite à alimenter une relation (graphe) pour que le modélisateur puisse la manipuler avec les entités cellulaires directement dans le modèle. La figure 4.4 présente un modèle numérique de terrain sur l'île de la Réunion (a), son découpage en zone sur une matrice de  $3 \times 3$  (b) et les valeurs des propriétés des entités cellulaires sur la matrice (c).

#### 4.4.2 Série temporelle d'images

Une série temporelle d'images correspond à un ensemble d'images caractérisant une zone sur plusieurs intervalles de temps. On peut ainsi avoir des séries temporelles d'images caractérisant la couverture de végétation tout au long d'une année pour un pas de temps donné. Les enjeux de l'intégration des séries temporelles est premièrement d'apporter de l'information temporelle supplémentaire mais peut aussi servir à forcer un modèle selon les données observées. Pour cela le datafacer

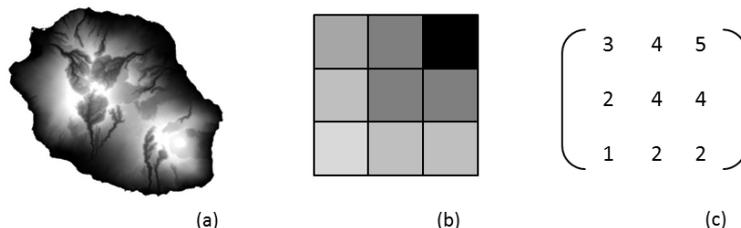


FIGURE 4.4 – Un modèle numérique de terrain en (a) dont les données sont extraites pour créer une matrice de  $3 \times 3$ , la zone de la matrice doit être spécifiée dans le modèle. En (c) le stockage des valeurs correspondantes pour les propriétés "mnt" des entités cellulaires.

*TemporalSeriesFile* a été développé permettant de lire un répertoire contenant la série d'images dont la définition en Ocelet est présentée par la figure 4.5.

```

datafacer Temporal{
    store TemporalSeriesFile("data/serie")

    match CellEntity {
        mnt : "0"
    }
}

```

FIGURE 4.5 – Définition d'un datafacer permettant de lire des séries temporelles d'images.

Les méthodes utilisables dans un modèle avec les datafacers sont données en annexe B.3. Ce datafacer va faciliter la lecture de plusieurs images et à mettre à jour les données d'une matrice par correspondance avec les entités cellulaires passée en paramètre (mot clef "match"). Son utilisation en Ocelet est présentée par la figure 4.6, en se rapportant à la définition présentée par la figure 4.5.

L'implémentation de ce type de datafacer permet la mise à jour de données issues de séries temporelles. Elle permet aussi d'exporter l'état à un temps donné d'une partie ou de l'ensemble des propriétés des entités cellulaires appartenant au même type. L'export correspond alors à une image raster contenant un nombre de bandes égal au nombre de propriétés de ce type d'entité. Après avoir déterminé comment attribuer aux entités les formes et les données spatiales régulières nous abordons dans la section suivante les implémentations pour faire évoluer l'état des entités avec des fonctions d'interaction.

```

scenario Model{
    let temporal = Temporal{}
    let myRaster = MyRaster{}
    let cells = Cells{}
    cells.connect(myRaster.readAll)
    while(temporal.hasNext){
        temporal.update
        temporal.next
    }
}

```

FIGURE 4.6 – Exemple d'utilisation du datafacer de série temporelle d'images et de la mise à jour des propriétés "mnt" des entités cellulaires contenues dans la relation "cells".

## 4.5 Implémentation et gestion de graphes

Avec Ocelet, le modélisateur manipule des graphes d'entités par le biais des relations. Nous avons identifié trois cas de figure nécessitant un traitement spécifique pour la réalisation de relations : (i) la régularité des entités cellulaires donne la possibilité de construire un graphe reliant les cellules d'une même couche raster, à travers différentes définitions de voisinage, (ii) on peut établir des liens entre des entités possédant une géométrie dont le contour est défini en format vecteur et les cellules d'un raster se trouvant sur la zone couverte par ces entités, (iii) entre deux raster distincts ayant éventuellement des résolutions spatiales différentes, on peut avoir besoin de relier les cellules qui se recouvrent spatialement. La figure 4.7 présente un exemple de définition des différentes relations pouvant gérer les entités avec des cellules selon ces trois cas de figure.

```

(a) relation Cells<CellEntity c1, CellEntity c2>{
    }

(b) relation Cells<Plot plot, CellEntity c>{
    }

(c) relation Cells2<CellEntity c1, CellEntity2 c2>{
    }

```

FIGURE 4.7 – (a) Relation entre des entités cellulaires de même type. (b) Relation entre entités géométriques et cellulaires. (c) Relation entre entités cellulaires de types différents.

Pour chaque type de ces relations une classe Java est générée et va hériter de la classe appropriée contenue dans le moteur d'exécution d'Ocelet dont une partie des classes de graphes est illustrée par la figure 4.8. Le choix de la classe parente se fait

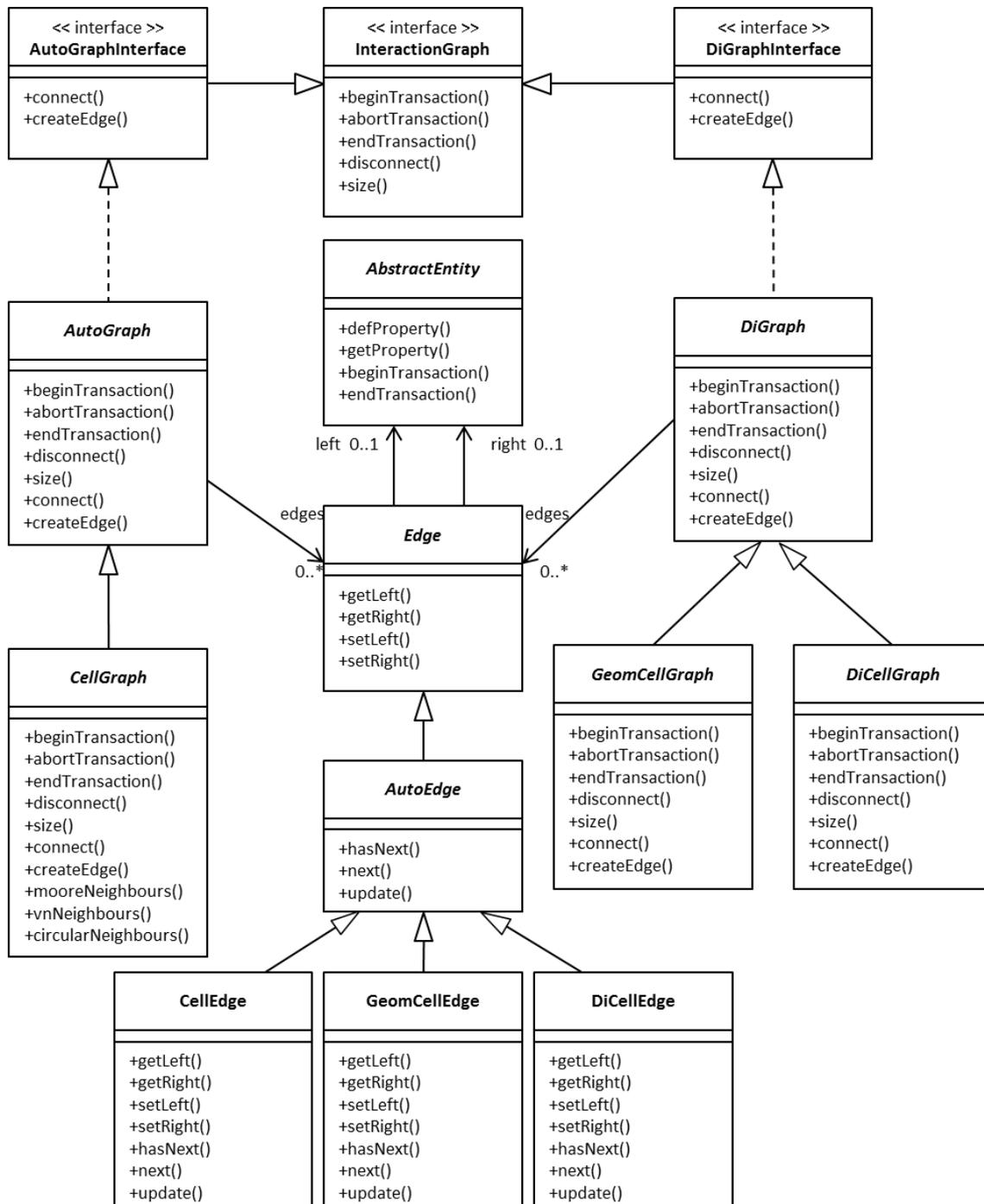


FIGURE 4.8 – Diagramme UML des classes de graphes dans le moteur d'exécution d'Ocelet.

lors de l'analyse sémantique qui identifie le type des propriétés des entités passées en paramètres de la relation. Le moteur d'exécution contient des classes parentes pour les différents types de graphes. Les méthodes utilisables par le modélisateur sur ces relations sont données en annexe B.2.

Une méthode héritée particulière est celle qui correspond au parcours des arêtes du graphe de la relation. Cette méthode agit différemment selon les types d'entités de la relation. La méthode de base d'Ocelet fonctionne en trois phases :

- (i) Ouverture des transactions sur les entités : les valeurs affectées aux propriétés des entités par une fonction d'interaction vont pouvoir être stockées en tant que valeurs candidates. L'affectation effective étant reportée à la fin de la transaction.
- (ii) Parcours de toutes les arêtes du graphe : pour chaque arête, la fonction d'interaction est appliquée sur les deux sommets (entités) aux extrémités de l'arête.
- (iii) Fermeture des transactions sur les entités : les valeurs candidates sont affectées à leurs propriétés, soit en prenant une valeur au hasard, soit en appliquant une fonction d'agrégation si celle-ci est définie par le modélisateur.

Ces étapes sont nécessaires lorsque la structure de graphe est inconnue, le graphe d'interaction étant construit explicitement. L'ouverture et la fermeture des transactions correspondent à la phase de synchronisation des entités. La régularité des entités cellulaires permet d'une part de proposer une construction de graphes implicites, et d'autre part de développer des méthodes adaptées de parcours de graphes et de synchronisations des valeurs candidates. Nous décrivons ces aspects dans les sections suivantes.

### 4.5.1 Graphe entre entités cellulaires de même type

Le choix de gérer un espace régulier avec des entités cellulaires en se basant sur une matrice de données permet de développer des algorithmes qui vont "simuler" le parcours de graphe. Dans ce cas la relation garde deux instances d'entités ainsi qu'une instance d'arête en mémoire. Les propriétés d'une entité cellulaire pouvant être récupérées en connaissant sa position sur la matrice, une arête est simulée entre cette entité cellulaire et une entité voisine dont la position est déduite. Ainsi c'est toujours la même instance d'entité qui est manipulée mais l'accès à ses propriétés change. A chaque nouveau voisin, la position de l'arête est mise à jour et les fonctions d'interaction peuvent alors être lancées. Le parcours des arêtes est différent selon la forme des cellules, mais pour ce type de graphe, ce parcours peut être pour toutes les formes assimilé à ce que l'on retrouve dans les automates cellulaires. L'algorithme que nous avons implémenté reproduit la simulation du parcours basé sur un graphe de Cayley (Cayley, 1878). White (1973) en présente les notions qui ont été reprises par Róka (1999, 1994) lors de ses travaux sur la simulation d'automates cellulaires par des graphes de Cayley :

Soit  $G$  un groupe avec  $G = \{g_1, g_2, \dots, g_n\}$ , un sous ensemble de son ensemble d'éléments  $X$ . Un mot  $w$  sur  $G$  est un produit fini  $f_1, f_2 \dots f_n$  où chaque  $f_i$  appartient à l'ensemble  $GU G^{-1}$ . Si chaque élément de  $G$  peut être exprimé en un mot de  $G$  alors  $g_1, g_2 \dots$  sont des générateurs pour  $G$ . Une relation est une égalité entre deux mots de  $G$ . Si  $G$  est généré par  $g_1, g_2 \dots$  et si chaque relation dans  $G$  peut être dérivée des relations  $p = p', q = q', r = r' \dots$  alors on peut écrire d'après Róka (1994) :

$$G = \langle g_1, g_2 \dots | p = p', r = r' \dots \rangle (= \langle G | R \rangle) \quad (4.1)$$

Róka (1999) démontre que la prise en compte des générateurs sans leurs inverses correspond à un graphe de Cayley sur un automate cellulaire. Les générateurs du groupe vont déterminer les sommets du graphe en indiquant la position d'un voisin. Ainsi un voisinage de Von Neumann est traduit par le groupe et ses générateurs définis en 4.2, un voisinage de Moore par 4.3, un voisinage sur des triangles par 4.4 et un voisinage sur des hexagones par 4.5.

$$G = \langle a, b | ab = ba \rangle \quad (4.2)$$

$$G = \langle a, b, c, d | ab = ba, c = ab, d = a^{-1}b \rangle \quad (4.3)$$

$$G = \langle a, b, c | a^2 = 1, b^2 = 1, c^2 = 1, (abc)^2 = 1 \rangle \quad (4.4)$$

$$G = \langle a, b, c | ab = ba, ac = ca, bc = cb, abc = 1 \rangle \quad (4.5)$$

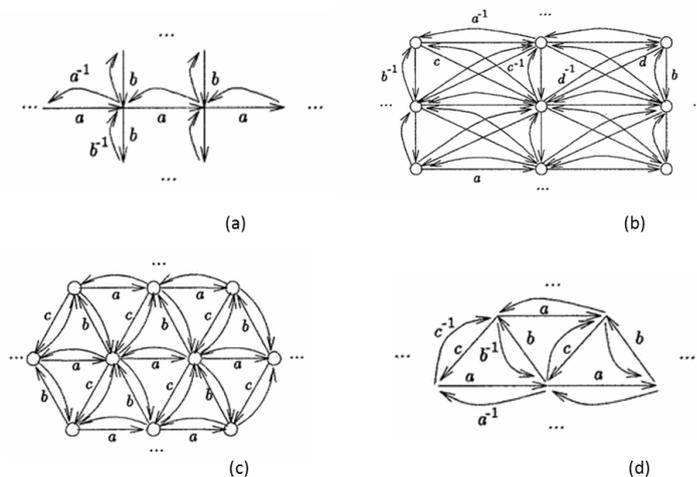


FIGURE 4.9 – Représentation du voisinage d'un automate cellulaire. Pour des cellules carrées avec en (a) le voisinage de Von Neumann, et en (b), le voisinage de Moore. Le voisinage pour des cellules triangulaires en (c) et hexagonales en (d) (Róka, 1999)

La figure 4.9 (a) représente les générateurs  $a, b$  pour un voisinage de Von Neumann de distance 1, pour le groupe 4.2. La figure 4.9 (b) présente les générateurs  $a, b, c$  et  $d$  pour un voisinage de Moore de distance 1 pour le groupe 4.3.

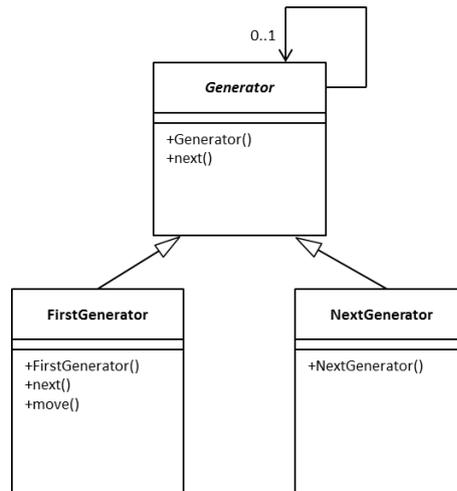


FIGURE 4.10 – Diagramme de classes des générateurs

La structure du graphe sur les cellules n'étant pas stockée, elle est simulée en appliquant à chaque cellule les générateurs correspondant à un graphe de Cayley sur un type de voisinage défini. Une arête d'un graphe est alors simulée par rapport à un générateur du groupe. Le nombre de générateurs varie selon la forme de la cellule mais aussi selon le type et la longueur des distances de Tchebybchev, Manhattan ou rayon du cercle évoquées en section 3.6.1 du chapitre précédent.

La construction des générateurs du groupe se fait à l'initialisation du graphe. Les générateurs sont liés sous la forme d'une liste chaînée présentant une récursivité entre le dernier et le premier élément. Cette liste est parcourue pour chaque cellule de la matrice. Deux types de générateurs sont construits : le premier de la classe "FirstGenerator" présenté sur la figure 4.10 met à jour la position de la cellule sur laquelle les générateurs de voisinage sont appliqués. Ce premier générateur indique aussi la position du premier voisin. Les autres générateurs de la classe "NextGenerator" (figure 4.10) indiquent la position des voisins suivants. Le dernier générateur est enfin relié au premier. Chaque générateur contient une position en  $x$  et  $y$ . Ces

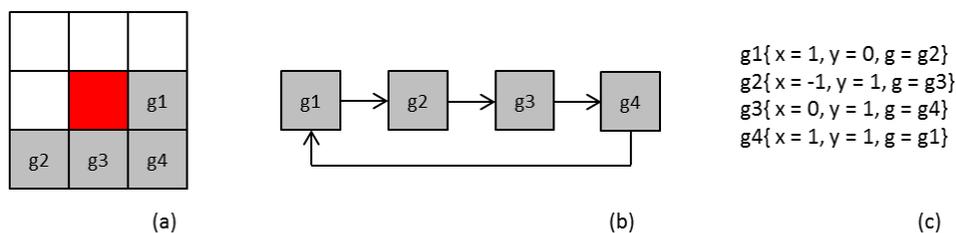


FIGURE 4.11 – En (a) les voisins qui sont simulés par les générateurs pour un voisinage de Moore de distance 1. En (b) la liste chaînée de générateurs permettant de simuler les arêtes d'un graphe. En (c) les valeurs des positions des générateurs pour mettre à jour la position des entités cellulaires.

positions, ajoutées à la position de l'entité cellulaire centrale donnent les coordonnées des entités voisines sur la matrice. La première étape consiste à créer la liste chaînée de générateurs suivant le type de voisinage, la forme des cellules ainsi que la distance de voisinage. Pour une cellule et un voisinage de Moore de distance de 1 nous obtenons les voisins représentés dans la figure 4.11 (a) en ayant une liste chaînée de générateurs représentée par la figure 4.11 (b).

L'algorithme 1 correspond au voisinage de Moore sur des cellules carrées de distance  $S$ . La méthode "move()" est seulement utilisée dans la méthode "next()" par le pre-

---

**Algorithm 1** mooreGeneratorFactory( $S$ )

---

**Require:**  $S$  distance de voisinage

**Ensure:**  $g1$  premier élément de la liste chaînée de générateurs  $\{g1, g2, \dots, gn\}$

```

 $g1 \leftarrow newFirstGenerator(x = 1, y = 0)$ 
 $G \leftarrow g1$ 
if  $S > 1$  then
  for  $i = 2$  to  $S$  do
     $gx \leftarrow newNextGenerator(x = i, y = 0)$ 
     $g1.g \leftarrow gx$ 
     $g1 \leftarrow gx$ 
  end for
end if
for  $j = 1$  to  $S$  do
  for  $i = -S$  to  $S$  do
     $gx \leftarrow NextGenerator(x = i, y = j)$ 
     $g1.g \leftarrow gx$ 
     $g1 \leftarrow gx$ 
  end for
end for
 $G.g \leftarrow g1$ 
return  $g1$ 

```

---

mier générateur, il permet d'incrémenter la position de l'entité cellulaire à partir de laquelle seront recherchés les voisins. La méthode "next()" permet de donner la position d'une entité voisine. La position est calculée par les générateurs en fonction de la position de la cellule concernée. Il faut effectuer un test sur la position de l'entité voisine pour vérifier que celle-ci soit bien comprise dans la matrice. Les entités voisines simulées peuvent en effet être en dehors du domaine pour les cellules sur le bord de la matrice.

L'application de ces générateurs est possible seulement en considérant que les arêtes du graphe sont bidirectionnelles. Le passage sur un voisin d'une cellule ne se fai-

sant qu'une seule fois. L'application des générateurs sur une seule cellule ne simule cependant pas le parcours de tous ses voisins. Pour un voisinage de Moore de distance 1, les générateurs d'une cellule ne simulent que quatre arêtes (figure 4.12 (a)). Les quatre arêtes manquantes sont simulées lors du passage sur les autres entités cellulaires précédentes (figure 4.12 (b)). La liste chaînée de générateurs est ainsi ap-

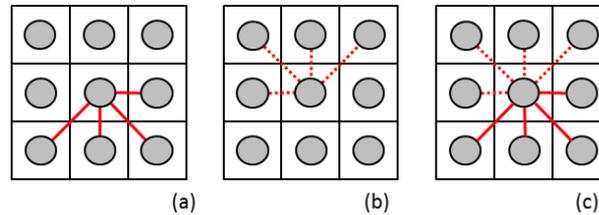


FIGURE 4.12 – (a) Parcours des générateurs sans leurs inverses pour l'entité cellulaire située au milieu. (b) Parcours des arêtes simulées par les générateurs lors du passage sur les entités précédentes. (c) Parcours total simulé.

pliquée sur chaque entité cellulaire. Le parcours des entités cellulaires se fait ligne par ligne, en partant de la première ligne de la matrice. L'algorithme 2 présente ce parcours. Le moteur d'exécution d'Ocelet applique une fonction d'interaction sur chaque arête. Dans le cas présent où l'on fait usage des générateurs, une seule instance d'arête est créée et maintenue durant le parcours de graphe. Seule la position de ses sommets (entités) aux extrémités change, ce qui simule une nouvelle arête à chaque générateur valide. Le stockage temporaire des variables issues d'une fonction d'interaction se fait selon le parcours de la matrice : l'ensemble des valeurs candidates est conservé tant que la simulation du parcours des arêtes n'est pas terminée sur une ligne. L'utilisation des générateurs implique de créer une ligne d'une taille équivalente à la largeur de la matrice, chaque élément de la ligne contient les valeurs candidates, la position sur cette ligne correspondant à la position d'une cellule. Ainsi,  $S + 1$  lignes sont créées pour une distance  $S$  de voisinage. Les lignes sont stockées par ordre de parcours de la matrice. La fonction d'interaction applique les valeurs sur ces lignes au lieu d'écrire sur les propriétés mêmes des cellules contenues dans la matrice (figure 4.13 (b)). Les propriétés des cellules sont affectées à la fin du parcours de la première ligne en appliquant une fonction d'agrégation (figure 4.13 (c)). Enfin, les valeurs candidates de cette ligne sont réinitialisées. Cette ligne devient la dernière ligne en ordonnée, la deuxième ligne devenant la première.

Soit  $C_i\{c_1, c_2, \dots, c_n\}$  l'ensemble composé des valeurs candidates issues d'une fonction d'interaction à la position en abscisse  $i$ . Un élément  $c$  de  $C_i$  contient les ensembles correspondant aux valeurs candidates des propriétés  $P\{\alpha, \beta, \gamma, \dots\}$ . Soit  $\theta_j\{C_1, C_2, \dots, C_l\}$  représentant une ligne contenant les ensembles des valeurs candidates avec  $l$  le nombre de colonnes de la matrice. Nous appelons  $St\{\theta_1, \dots, \theta_{S+1}\}$  la structure composée de  $S + 1$  lignes, avec  $S$  la distance de voisinage.

L'ouverture d'une transaction correspond à l'initialisation de la structure  $St$ . Contrairement au fonctionnement initial de la fermeture des transactions par Ocelet, le

**Algorithm 2** Méthode next**Require:** x1,y1 position de la cellule de la première entité

x2,y2 position de la cellule de l'entité voisine

G générateur donné par mooreGeneratorFactory(S)

width : nombre de colonnes de la matrice

height : nombre de lignes de la matrice

**Ensure:** mise à jour des positions x1,y1 et x2,y2**if** x1 = width - 1 **then**x1  $\leftarrow$  0y1  $\leftarrow$  y1 + 1**else**x1  $\leftarrow$  x1 + 1**end if****if** x1 + x < width et y1 + y < height et x1 + x > 0 et y1 + y > 0 **then**x2  $\leftarrow$  x1 + xy2  $\leftarrow$  y1 + yG  $\leftarrow$  G.g**else**G  $\leftarrow$  G.g

G.next()

**end if**

premier ensemble  $\theta_1$  est synchronisé lorsqu'un générateur de classe *NextGenerator* change de position de la cellule principale en  $y$ . Les valeurs de cet ensemble sont ré-initialisées, les ensembles  $\theta_j$  sont permutés tels que  $\theta_1 = \theta_{S+1}$  et pour  $j < S+1$ ,  $\theta_j = \theta_{j+1}$  (figure 4.13 (d)). L'utilisation d'une matrice qui contient les données par position et par bande permet une utilisation implicite des cellules au format carré. Pour tenir compte des autres formes de cellules, la forme triangulaire ou hexagonale, le parcours de graphe doit simuler les voisins de telles formes tout en se basant sur l'accès aux données sur la matrice. Pour la forme hexagonale, celle-ci correspond à un décalage de colonnes paires et impaires que nous pouvons reproduire sur la matrice. Pour la forme triangulaire, leur représentation sur une matrice est aussi reproductible en tenant compte de la parité, comme le présente la figure 4.14.

Le parcours entre des cellules de formes différentes par rapport au carré doit cependant être modifié. Les cellules hexagonales, pour une distance de voisinage de 1, ont en effet six voisins et les générateurs de voisinages doivent être adaptés en conséquence. Nous gardons la même méthode de parcours de la matrice, ligne par ligne en appliquant des générateurs pour déterminer les voisins de la cellule courante. Un exemple de parcours de graphe pour les cellules hexagonales est donné avec la figure 4.15.

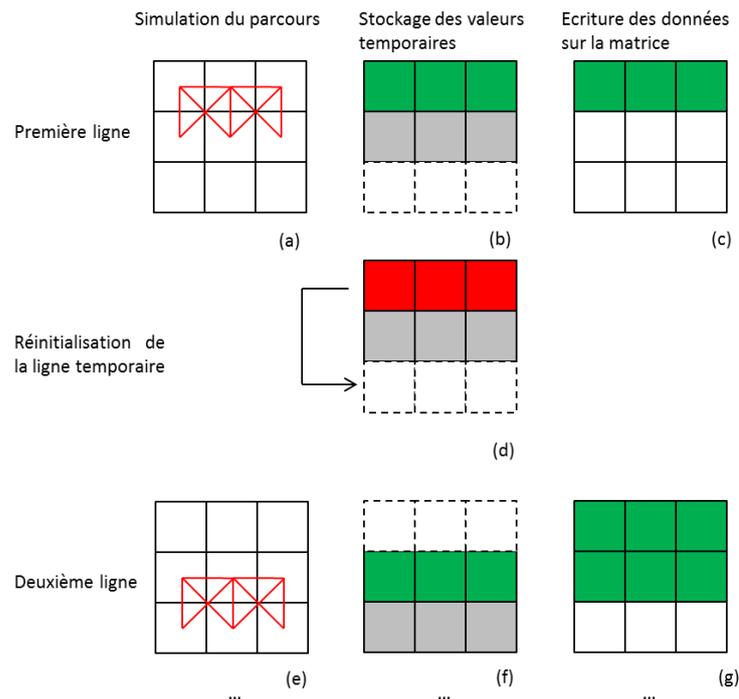


FIGURE 4.13 – Méthode de stockage des valeurs temporaires sur la structure  $St$  en (b), (d) et (f) et leurs synchronisations à la fin du parcours de la première ligne en (c) et de la deuxième ligne en (g). Le parcours des voisins est simulé pour la première ligne en (a) et en (e) sur la deuxième ligne.

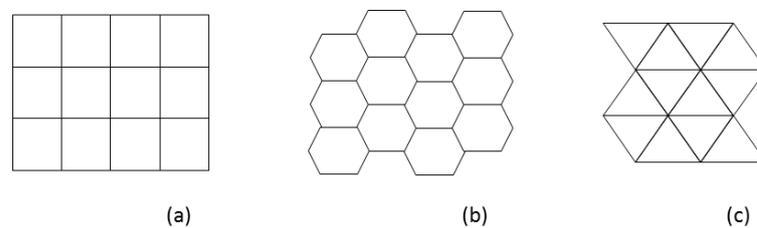


FIGURE 4.14 – Différents pavages constitués d'entités cellulaires hexagonales en (b) ou triangulaires en (c) qui peuvent se rapporter à une matrice carrée en (a) par décalage.

Comme le voisinage change selon la parité des colonnes, la liste chaînée de générateurs pour une cellule ne suffit plus, contrairement aux cellules carrées. Dans ce cas la liste chaînée contient les générateurs de deux cellules successives sur la même ligne. La figure 4.16 présente la représentation des générateurs sur la matrice lorsque la cellule est sur une colonne paire et sur une colonne impaire. Les générateurs  $g1$  et  $g4$  de la figure 4.16 (b) correspondent à la classe "NextGenerator" de la figure 4.10 pour spécifier le changement de position de la cellule sur laquelle est simulé le

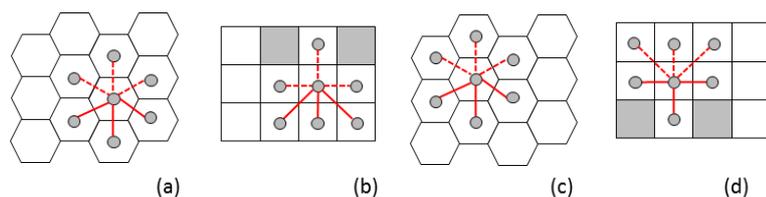


FIGURE 4.15 – Les arêtes en pointillées simulent le parcours effectué par les générateurs sur les lignes précédentes. Les figures (a) et (b) présentent les générateurs d'une cellule hexagonale d'une colonne paire. Les figures (c) et (d) présentent les générateurs d'une cellule hexagonale de colonnes impaires.

voisinage. Dans cet exemple, les générateurs ne doivent pas avoir d'inverse dans un voisinage sur la forme hexagonale, bien que ce soit le cas lorsqu'ils sont rapportés sur la matrice (par exemple  $g_4$  et  $g_5$  de la figure 4.16). La disposition des générateurs présentée permet de garder un nombre de lignes en mémoire pour la synchronisation égale à la distance de voisinage  $S + 1$ .

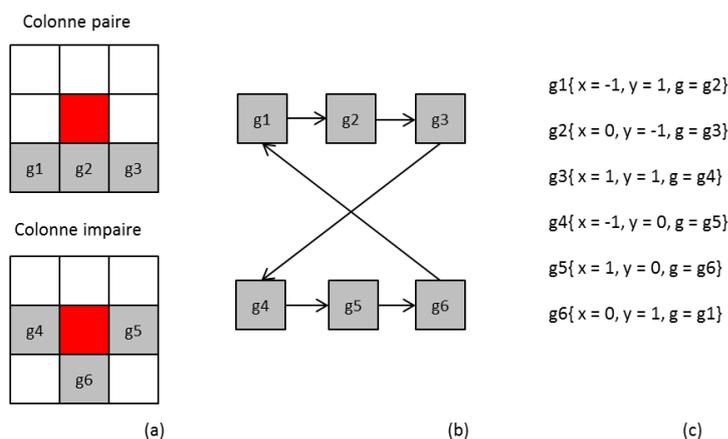


FIGURE 4.16 – La figure (a) présente les voisins d'une cellule hexagonale sur une colonne paire et impaire et leurs positions sur la matrice. La figure (b) présente la liste chaînée de générateurs. En (c) la valeur des positions des générateurs qui serviront à mettre à jour la position des cellules voisines.

Le parcours des entités cellulaires ayant une forme triangulaire est proche du parcours avec les cellules hexagonales et fonctionne aussi selon les colonnes paires ou impaires de la matrice. La figure 4.17 donne un exemple pour une distance  $S = 1$ , un seul générateur est appliqué sur les colonnes impaires et deux générateurs sont appliqués pour les colonnes paires.

Ce type de relation contient donc les générateurs à appliquer lorsque les interactions sont appelées. Une fois les générateurs définis, le parcours de graphe est le même quel que soit la forme sauf pour la forme triangulaire ou un échange de

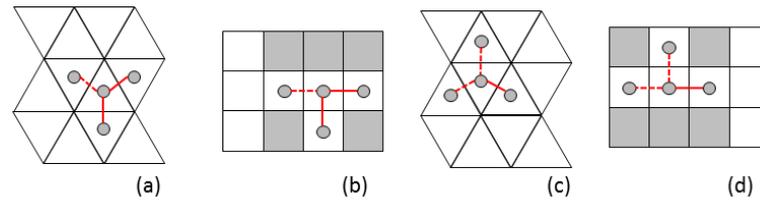


FIGURE 4.17 – Les arêtes en pointillées simulent le parcours effectué par les générateurs sur les lignes précédentes. Les figures (a) et (b) présentent les générateurs d'une cellule triangulaire d'une colonne paire. Les figures (c) et (d) présentent les générateurs d'une cellule triangulaire de colonnes impaires.

générateur de type "NextGenerator" doit avoir lieu à chaque changement de ligne puisqu'un triangle n'a pas la même orientation selon les lignes à la même colonne. Pour simuler une nouvelle arête du graphe, la méthode "nextGenerator" est appelée. La liste chaînée de générateurs permet de simuler la prochaine arête en mettant à jour les positions des entités cellulaires automatiquement. La simulation du parcours d'arêtes permet d'appliquer les fonctions d'interactions classiquement avec Ocelet : à chaque arête la fonction d'interaction est appliquée sur les entités aux extrémités de l'arête.

La figure 4.18 présente la définition d'une relation entre entités cellulaires de même type. Après instantiation de la relation, l'appel de l'interaction va lancer le parcours des cellules. Chaque propriété "eau" des entités cellulaires va recevoir les valeurs des propriétés "eau" de leurs entités cellulaires voisines. Le mot clé "agg" à la fin de l'interaction indique quel type de fonction d'agrégation est appliqué sur les valeurs candidates. Dans la figure 4.18, nous donnons un exemple avec une fonction d'agrégation intégrée de base dans Ocelet, la fonction Mean qui va renvoyer la moyenne de ces valeurs.

```

relation Cells<CellEntity c1, CellEntity c2> {

    interaction moyenne{
        c1.eau = c2.eau
        c2.eau = c1.eau
    } agg {
        c1.eau << Mean
        c2.eau << Mean
    }
}

```

FIGURE 4.18 – Définition d'une relation entre des entités cellulaires de même type en Ocelet. La fonction d'interaction traduit un échange d'eau selon le voisinage de chaque cellule.

L'algorithme 3 simule l'ensemble des arêtes d'un graphe entre entités cellulaires

de même type, une arête est simulée lorsque la position  $(x1, y1)$  de l'entité de type  $E1$  et la position de  $(x2, y2)$  de l'entité de type  $E2$  sont mises à jour par les générateurs.

---

**Algorithm 3** Parcours de graphe d'interaction d'entités cellulaires
 

---

**Require:**  $M_{t(l,h,b)}$  matrice de données

$F_i$  fonction d'interaction

$St\{\theta_1, \dots, \theta_{S+1}\}$  Structure pour stocker les valeurs candidates

$H\{h_\alpha, h_\beta, h_\gamma, \dots\}$  ensemble des fonctions d'agrégation

$g1$  premier générateur venant de l'algorithme `generatorFactory(S)`

$e_1, e_2$  instance d'entités dont les positions de leurs propriétés cellulaires vont changer

$P\{\alpha, \beta, \gamma, \dots\}$  la définition des types de propriétés d'une entité cellulaire

**Ensure:**  $M_{t+1(l,h,b)}$  matrice de données au temps  $t + 1$

$x1 \leftarrow e1.cell.x$

$y1 \leftarrow e1.cell.y$

$x2 \leftarrow e2.cell.x$

$y2 \leftarrow e2.cell.y$

$cpt \leftarrow 0$

**while**  $x1 < l - 2$  **and**  $y1 < h - 1$  **and**  $x2 < l - 1$  **and**  $y2 < h$  **do**

**if**  $y1 \neq cpt$  **then**

**for**  $C_x$  in  $\theta_1$  **do**

**for**  $p$  in  $P\{\alpha, \beta, \gamma, \dots\}$  **do**

$v\{v_\alpha, v_\beta, \dots\}$  ensemble contenant les valeurs candidates de la propriété de type  $p$  dans  $C_x$

$M_{(x,cpt,p)} \leftarrow h_p(v_p)$

        réinitialiser  $v$

**end for**

**end for**

    déplacer  $\theta_1$  à la fin de  $St\{\theta_1, \dots, \theta_{S+1}\}$

$cpt \leftarrow cpt + 1$

**end if**

$g1.next(x1, y1, x2, y2)$

  appliquer  $f_i$  sur  $v\{v_\alpha, v_\beta, \dots\}$  de  $C_{x1}$  élément de  $\theta_{y1-cpt}$

  appliquer  $f_i$  sur  $v\{v_\alpha, v_\beta, \dots\}$  de  $C_{x2}$  élément de  $\theta_{y2-cpt}$

**end while**

---

Cette implémentation rend possible l'utilisation de graphes d'entités cellulaires sans garder les arêtes en mémoire, la section suivante concerne l'implémentation effectuée pour les graphes entre les entités cellulaires et des entités non cellulaires.

## 4.5.2 Graphes entre entités cellulaires et entités géométriques ou sans représentation spatiale

### Graphe entre entités cellulaires et entités géométriques

Le deuxième type de graphe manipulable avec l'intégration des pavages réguliers dans Ocelet est le graphe entre entités cellulaires et des entités géométriques. Ce type de graphe est aussi implicite. En effet il est possible de passer d'une forme de représentation géométrique vers une forme cellulaire en discrétisant la géométrie selon la résolution des cellules. Le cas inverse, de cellules à vecteur est plus complexe. Les objets "géométriques" étant généralement issu d'un processus de segmentation ou d'une construction par l'expert nous ne traiterons pas du cas de construction d'objets géométriques à partir de cellules. Nous laissons cependant la possibilité de construire des géométries en appliquant des fonctions booléennes sur la forme géométrique des cellules mais celles-ci doivent être spécifiées par le modélisateur. De nombreux algorithmes existent pour extraire les pixels pour une géométrie donnée, les suivants ont été retenus et sont utilisés en tant que méthode *cellsExtract* :

- Points : une simple conversion par transformation affine donnant la position de la cellule.
- Ligne : algorithme de Bresenham (Bresenham, 1965).
- Polygone : de nombreux algorithmes existent pour "rasteriser" un polygone, la méthode des triangles hiérarchiques (Jiménez et al., 2009) ou encore le calcul de point à l'intérieur d'un polygone (Yang et al., 2010). Parmi ces méthodes la plus simple et la plus robuste reste la méthode du "scan line" (Ferguson, 1963).

L'irrégularité des formes possibles pour les entités géométriques implique deux cas de figure avec (i) une géométrie peut être superposée à plusieurs cellules et (ii) une cellule peut être superposée à plusieurs géométries. Ces deux conditions doivent être prises en compte pour le parcours et la simulation des arêtes d'un graphe permettant de faire les liens entre des entités cellulaires et des entités géométriques. En effet une entité géométrique ou une entité cellulaire peuvent avoir plusieurs arêtes incidentes et donc plusieurs valeurs candidates à synchroniser lors d'une fonction d'interaction. Les entités cellulaires n'étant pas instanciées, nous devons garder la trace des positions des cellules correspondantes à une géométrie après avoir lancé la méthode d'extraction de cellule suivant le type de géométrie.

La structure permettant de simuler le graphe entre des entités de représentation différentes doit permettre de synchroniser les valeurs candidates sur les entités cellulaires. La synchronisation des valeurs pour les entités géométriques pouvant se faire en utilisant la méthode inhérente à Ocelet : en parcourant toutes les entités géométriques à la fin du parcours du graphe. Nous proposons la structure *CellIndex* qui permet d'indexer la position des cellules. Cette structure est un graphe orienté 4-parties avec :

- le premier ensemble  $K\{k\}$  composé d'un seul sommet sur lequel commence le

parcours de la structure.

- un ensemble  $X\{x_1, x_2, \dots, x_n\}$  composé de sommets correspondant à la position en  $x$  des entités cellulaires extraites.
- un ensemble  $Y\{y_1, y_2, \dots, y_n\}$  composé de sommets correspondant à la position en  $y$  des entités cellulaires extraites.
- un ensemble  $Eg\{e_1, e_2, \dots, e_n\}$  composé de sommets correspondant aux entités géométriques.

Le parcours de cette structure commence sur le sommet  $k$ . Les sommets appartenant à  $X\{x_1, x_2, \dots, x_n\}$  reliés à  $k$  et orientés de  $k$  vers  $X$  déterminent la position en abscisse des entités cellulaires pour  $x_n < l$  avec  $l$ , le nombre de colonnes de la matrice. Les sommets appartenant à  $Y\{y_1, y_2, \dots, y_n\}$  reliés à  $X$  et orientés de  $X$  vers  $Y$  déterminent la position en ordonnée des entités cellulaires pour  $y_n < h$ , avec  $h$  le nombre de lignes de la matrice. Ainsi le couple  $(x, y)$  avec  $x \in X, y \in Y$  indique la position d'une entité cellulaire sur la matrice. Enfin les sommets appartenant à  $Eg\{eg_1, \dots, eg_2, \dots, eg_n\}$  déterminent la ou les entités géométriques correspondant spatialement au couple  $(x, y)$ . Les éléments de  $Eg$  sont reliés à un élément de  $Y$  et sont orientés de  $Y$  vers  $Eg$ . Une arête du graphe d'interaction correspond donc au triplet  $\{x, y, eg\}$  avec  $x \in X, y \in Y, e \in Eg$ . Cette structure est représentée par la figure 4.19.

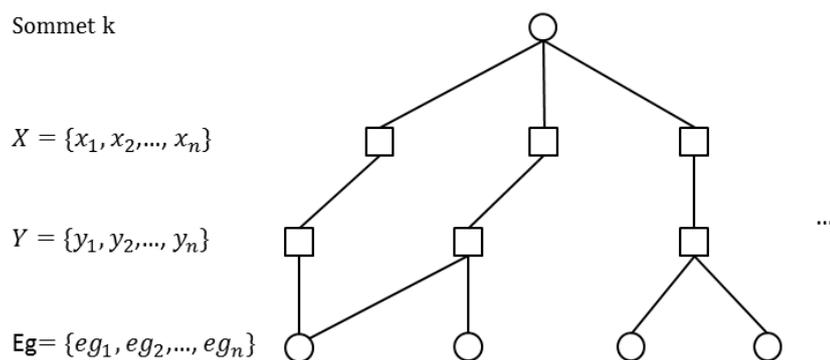


FIGURE 4.19 – La structure *CellIndex* construite pour stocker les positions des cellules extraites sur des géométries. Un couple de sommets  $(x, y)$  avec  $x \in X$  et  $y \in Y$  correspond à la position de l'entité cellulaire sur la matrice. Les entités géométriques correspondant spatialement à cette entité sont reliées à un sommet appartenant à  $Y$ .

La construction de cette structure est appliquée lorsqu'un ensemble d'entités géométriques et un ensemble d'entités cellulaires sont connectés dans une relation. Les positions des entités cellulaires sont déterminées à partir de la méthode d'extraction de cellules en parcourant l'ensemble des entités géométriques. L'algorithme 4 correspond à la construction de la structure.

La simulation du parcours de graphe entre entités cellulaires et entités géométriques va s'établir sur le parcours de la structure *CellIndex*. Le parcours débute sur

---

**Algorithm 4** createCellIndex

---

**Require:**  $Eg\{eg_1, eg_2, \dots, eg_n\}$  entités géométriques**Ensure:**  $k$  premier sommet de la structure *CellIndex*

```

initialiser CellIndex avec le premier sommet  $k$ 
for  $eg$  in  $Eg$  do
  for  $(x, y)$  in cellsExtract do
    if  $x$  est un sommet de  $k$  then
      if  $y$  est un sommet de  $x$  then
        if  $eg$  n'est pas un sommet de  $y$  then
           $y \leftarrow eg$  comme sommet
        end if
      else
         $y \leftarrow eg$  comme sommet
         $x \leftarrow y$  comme sommet
      end if
    else
       $y \leftarrow eg$  comme sommet
       $x \leftarrow y$  comme sommet
       $k \leftarrow x$  comme sommet
    end if
  end for
end for
retourner  $k$ 

```

---

le sommet  $k$  à partir duquel chaque sommet connecté appartenant à l'ensemble  $X$  indique la position en abscisse des cellules. Chaque sommet appartenant à  $Y$  connecté à un sommet  $x$  indique la position en ordonnée  $y$ . Chaque sommet appartenant à  $Eg$  connecté aux sommets de  $Y$  sont des entités géométriques. Une arête du graphe est simulée avec, aux extrémités avec (i) une entité cellulaire de position donnée par le sommet en abscisse et le sommet en ordonnée et (ii) une entité géométrique donnée par le sommet connecté au sommet en ordonnée. Une fonction d'interaction  $f_i$  peut alors être appliquée sur ces deux entités. Les valeurs candidates pour l'entité cellulaire doivent être stockées dans un ensemble  $v\{v_\alpha, v_\beta, v_\gamma, \dots\}$  correspondant aux valeurs candidates des propriétés  $P\{\alpha, \beta, \gamma, \dots\}$ . Les fonctions d'agrégation, si elles sont définies, sont appliquées sur ces valeurs lorsque le parcours de l'ensemble des entités géométriques est terminé sur le sommet qui représente alors une entité cellulaire de position  $(x, y)$ . L'algorithme 5 présente la simulation du parcours de graphe entre des entités cellulaires et des entités géométriques.

La définition d'une relation entre ces deux types d'entités est présentée avec la figure 4.20 en Ocelet.

---

**Algorithm 5** Parcours de graphe entre entités cellulaires et géométriques
 

---

**Require:**  $Eg\{eg_1, eg_2, \dots, eg_n\}$  entités géométriques

$M_{(l,h,b)}$  la matrice de données

$F_i$  ensemble des fonctions d'interaction

$H\{h_\alpha, h_\beta, h_\gamma, \dots\}$  ensemble des fonctions d'agrégation

$ec$ , instance d'entité cellulaire

$P\{\alpha, \beta, \gamma, \dots\}$  la définition des type de propriétés d'une entité cellulaire

$k$  premier sommet de  $CellIndex$  donné par  $createCellIndex$

**Ensure:**  $M_{t+1(l,h,b)}$  matrice de données au temps  $t + 1$

$Eg\{eg_1, eg_2, \dots, eg_n\}$  entités géométriques au temps  $t + 1$

initialiser  $v\{v_\alpha, v_\beta, \dots\}$  ensemble pour les valeurs candidates de l'entité  $ec$

ouvrir les transactions sur  $EG$

**while**  $k$  a un sommet **do**

$x \leftarrow k$  : prochain sommet

**while**  $x$  a un sommet **do**

$y \leftarrow x$  : prochain sommet

$ec.cell \leftarrow (x, y)$

**while**  $y$  a un sommet **do**

$eg \leftarrow y$  : prochain sommet

      appliquer  $F_i$  sur  $eg$  et  $ec$

      stocker le résultat de  $F_i$  dans  $v\{v_\alpha, v_\beta, \dots\}$

**end while**

**for**  $p : P\{\alpha, \beta, \gamma, \dots\}$  **do**

$M_{(x,y,p)} \leftarrow h_p(v_p)$

    réinitialiser  $v_p$

**end for**

**end while**

**end while**

Fermer les transactions sur  $EG$

---

```

relation PlotCells<Plot plot, CellEntity c> {

    interaction setMnt{
        plot.mnt = c.mnt
    } agg {
        plot.mnt << Mean
    }
}

```

FIGURE 4.20 – Exemple de définition d’une relation entre entités géométriques "Plot" et entités cellulaires "CellEntity". La fonction d’interaction "setMnt" permet d’attribuer les valeurs des propriétés "mnt" des entités cellulaires et de les attribuer aux propriétés "mnt" des entités "Plot". La fonction d’agrégation calcule la moyenne sur les valeurs venant des entités cellulaires.

Dans l’exemple présenté par la figure 4.21, seules les propriétés des entités géométriques sont affectées par les propriétés des entités ayant une cellule. L’exemple reprend la définition de la relation présentée par la figure 4.20.

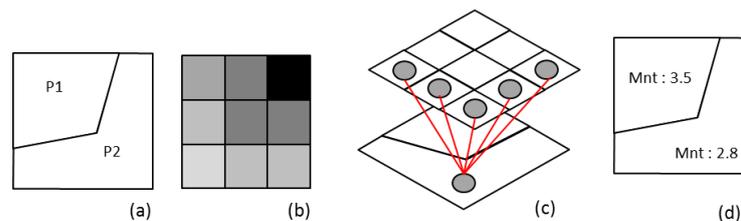


FIGURE 4.21 – (a) Les entités géométriques (polygones). (b) Les entités cellulaires. (c) Les connexions entre l’entité géométrique *P2* et les entités cellulaires correspondantes. (d) Résultat de l’interaction.

Lors de la définition d’une relation entre une entité cellulaire et une entité géométrique, le langage va générer une classe en Java qui va hériter de la classe `Geom-CellGraph`. Il revient cependant au modélisateur d’utiliser des données qui sont correspondantes spatialement.

### Graphe entre entités cellulaires et entités sans représentation spatiale

Si la création de graphes entre entités géométriques et entités cellulaires est implicite, nous laissons la possibilité de définir des graphes entre entités sans représentation spatiale et entités cellulaires. Dans ce cas il faut spécifier la manière dont sont connectées ces entités. La création du graphe se fait dans le scénario d’Ocelet et la méthode de connexion doit être spécifiée. La même structure d’indexation est utilisée et l’algorithme simulant le parcours de graphe sur cette structure est le même

que celui effectué pour les entités ayant une géométrie en propriété. Les sommets de la structure permettant de simuler une arête avec un sommet correspondant à la position en  $(x, y)$  de l'entité cellulaire. Le sommet connecté au sommet  $y$  est alors l'entité sans représentation. Ces sommets sont construits explicitement par le modélisateur en utilisant la méthode `connect(ec, e)` avec `ec` une instance d'entité cellulaire et `e` une instance d'entité sans représentation. Un exemple d'appel de cette méthode ainsi que le graphe obtenu est présenté avec la figure 4.22.

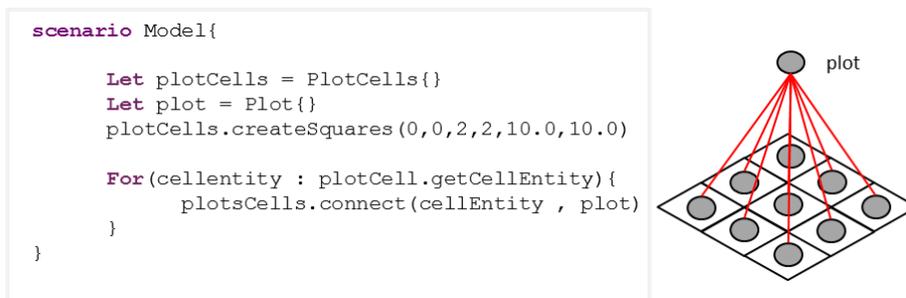


FIGURE 4.22 – Présentation d'un code en Ocelet pour connecter explicitement une entité avec des entités cellulaires et la représentation schématique du graphe obtenu.

Dans la figure 4.22, le code Ocelet se situe au niveau du scénario. Deux instances sont créées, l'instance de la relation et une instance d'entité "Plot" dans laquelle aucune propriété n'est initialisée. Celle-ci n'a donc pas de forme de représentation spatiale, la méthode de connexion générale entre une matrice et une liste d'entités vectorielles n'étant pas utilisable. Il faut donc spécifier comment connecter ces entités. La méthode `createSquares()` génère une matrice de 3 x3 avec des cellules carrées de résolution de 10 mètres. La boucle va ensuite connecter toutes les entités cellulaires à l'entité "plot". La connexion est ici très simple mais des tests supplémentaires sur les propriétés des entités peuvent être spécifiés pour ne connecter que les entités sélectionnées selon leurs propriétés.

Nous avons vu dans cette section comment simuler le parcours de graphe entre des entités cellulaires et les autres entités en tenant compte du fait que les entités cellulaires ne sont pas instanciées. La section suivante présente l'implémentation effectuée pour gérer des graphes entre des entités cellulaires de types différents.

### 4.5.3 Graphes entre entités cellulaires de types différents

Nous avons vu qu'une image pouvait caractériser une zone selon divers indices comme l'altimétrie, la température etc. Ces indices sont cependant spécifiés pour une résolution donnée. Les sources de données n'étant pas forcément homogènes, il convient de proposer au modélisateur la gestion de graphes d'entités cellulaires différentes, celles-ci pouvant avoir des résolutions différentes ou encore des formes différentes. Dès lors les entités du modèle pourront échanger des informations, toujours en utilisant des interactions sur les graphes d'entités cellulaires en ayant des

résolutions spatiales différentes. La construction de graphe entre entités ayant une résolution différente sera effective dès lors que le modélisateur choisit de connecter deux matrices d'entités différentes. Ce graphe n'est pas construit directement en mémoire. La relation va aussi simuler le parcours de ces graphes en gardant en mémoire deux instances d'entités cellulaires, une de chaque type.

### Parcours de graphe entre couches d'entités cellulaires de même résolution

La simulation du parcours de graphe se fait sur la matrice, ligne par ligne, et, pour chaque ligne, les colonnes sont parcourues. A chaque changement de colonne les positions des deux entités cellulaires sont mises à jour et l'interaction est appliquée sur ces deux entités. La définition d'une relation et d'une fonction d'interaction entre des entités cellulaires de types différents est présentée par la figure 4.23.

```

Relation Cells2<CellEntity c1, CellEntity2 c2>{
    interaction setMnt(){
        c2.mnt = c1.mnt
    } agg {
        c2.mnt << Mean
    }
}

```

FIGURE 4.23 – Définition d'une relation en Ocelet entre des entités cellulaires de types différents : "CellEntity" et "CellEntity2". La fonction d'interaction entre ces entités permet de donner une valeur à la propriété "mnt" des entités "CellEntity2" à partir des entités "CellEntity" correspondantes.

La figure 4.24 est un exemple de graphe obtenu par la simulation du parcours entre des entités cellulaires de même forme et de même résolution. L'exemple reprend la fonction d'interaction présentée par la figure 4.23.

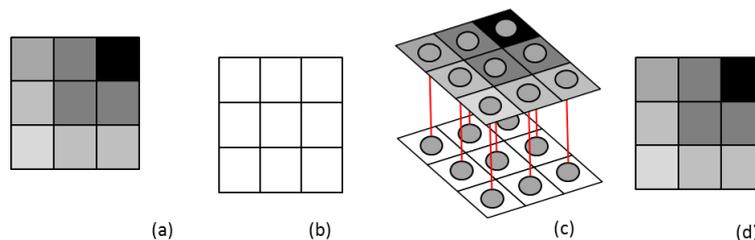


FIGURE 4.24 – En (a) la première couche d'entités cellulaires correspondant au rôle "CellEntity". En (b) la deuxième couche correspondant au rôle "CellEntity2". En (c) Le graphe final simulé entre les deux couches. En (d) le résultat de l'application de l'interaction.

Il revient au modélisateur de choisir les matrices ayant une correspondance spatiale, l'algorithme 6 ne cherchant, avec cette méthode, que les entités cellulaires d'une couche différente selon leurs positions matricielles.

---

**Algorithm 6** Parcours de graphe d'interaction d'entités cellulaires différentes
 

---

**Require:**  $M_{t(l,h,b)}$  matrice correspondant aux entités  $Ec$

$M'_{t(l,h,b')}$  matrice correspondant aux entités  $Ec'$

$Fi$  ensemble des fonctions d'interaction

$H\{h_\alpha, h_\beta, h_\gamma, \dots\}$  ensemble des fonctions d'agrégation sur  $Ec$

$H'\{h'_{\alpha'}, h'_{\beta'}, h'_{\gamma'}, \dots\}$  ensemble des fonctions d'agrégation sur  $Ec'$

$ec, ec'$  instances d'entités cellulaires

$(x1, y1), (x2, y2)$  position matricielle des propriétés cellules de  $ec$  et  $ec'$

$P\{\alpha, \beta, \gamma, \dots\}$  la définition des types de propriétés de  $Ec$

$P'\{\alpha', \beta', \gamma', \dots\}$  la définition des types de propriétés de  $Ec'$

**Ensure:**  $M_{t+1(l,h,b)}$  matrice de données au temps  $t + 1$

$M'_{t+1(l,h,b')}$  matrice de données au temps  $t + 1$

initialiser  $v\{v_\alpha, v_\beta, \dots\}$

initialiser  $v'\{v'_{\alpha'}, v'_{\beta'}, \dots\}$

$(x1, y1) \leftarrow (0, 0)$

$(x2, y2) \leftarrow (0, 0)$

**while**  $x1 < l - 1$  **and**  $y1 < h - 1$  **do**

$v \leftarrow Fi(ec, ec')$ , résultat de l'interaction sur les propriétés  $P\{\alpha, \beta, \gamma, \dots\}$

$v' \leftarrow Fi(ec, ec')$ , résultat de l'interaction sur les propriétés  $P'\{\alpha', \beta', \gamma', \dots\}$

**for**  $p$  in  $P\{\alpha, \beta, \gamma, \dots\}$  **do**

$M_{(x1,y1,p)} \leftarrow h_p(v_p)$

réinitialiser  $v_p$

**end for**

**for**  $p'$  in  $P'\{\alpha', \beta', \gamma', \dots\}$  **do**

$M'_{(x2,y2,p')} \leftarrow h'_{p'}(v'_{p'})$

réinitialiser  $v'_{p'}$

**end for**

**if**  $x1 = l - 1$  **then**

$x1 \leftarrow 0$

$y1 \leftarrow y1 + 1$

$x2 \leftarrow 0$

$y2 \leftarrow y2 + 1$

**else**

$x1 \leftarrow x1 + 1$

$x2 \leftarrow x2 + 1$

**end if**

**end while**

---

L'algorithme initialise la position des cellules des deux instances d'entités cellulaires de type différent à 0. Ces positions sont ensuite incrémentées : en abscisse tant que la fin de la ligne n'est pas atteinte, ou en ordonnée en réinitialisant la position en abscisse lorsque la fin de la ligne est atteinte. Avant l'incrémentation des positions, les fonctions d'interaction sont appliquées sur les entités cellulaires ainsi que les fonctions d'agrégation sur leurs propriétés. Le parcours de l'algorithme prend fin lorsque la dernière position matricielle est atteinte.

### Parcours de graphe entre couche d'entités cellulaires de résolutions différentes

La simulation des arêtes est différente si les résolutions entre les couches de cellules diffèrent. Le parcours se base sur les cellules de la matrice ayant la résolution la plus fine. Nous avons décidé d'utiliser un algorithme robuste mais qui implique une perte de précision lorsque les résolutions des deux couches amènent à un recouvrement partiel des cellules : la mise à jour des coordonnées des entités cellules de résolution moins fine se fait en utilisant une transformation des coordonnées matricielles. Cette transformation implique que la cellule la plus proche à l'arrondi est choisie. Par exemple, dans le cas où une cellule recouvre une cellule à 60% et une autre cellule à 40% de la distance du centre de la cellule, c'est celle ayant le plus haut recouvrement qui est sélectionnée. La figure 4.25 présente deux couches de cellules n'ayant pas la même résolution et le graphe obtenu par transformation de coordonnées entre ces cellules.

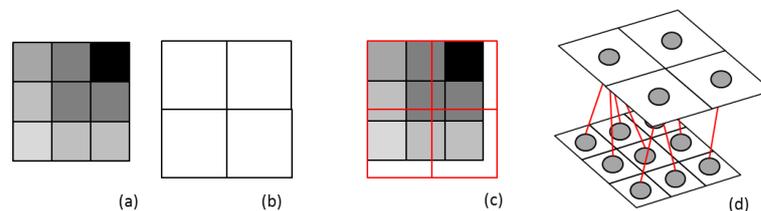


FIGURE 4.25 – Deux couches d'entités cellulaires de résolutions différentes en (a) et (b). La superposition des couches peut inclure un décalage selon la différence de résolution. Le graphe obtenu se base sur le plus proche voisin (d).

L'algorithme est en partie identique à celui sur des entités cellulaires de type différent mais de même résolution. La différence majeure est au niveau du changement de position des cellules qui n'est plus une simple incrémentation mais tient compte de la résolution des cellules. De plus la méthode de stockage des valeurs candidates pour les entités cellulaires de résolution moins fine se fait par ligne, jusqu'à ce que le parcours change de ligne sur ces entités.

Ces types de graphes permettent au modélisateur de gérer des pavages réguliers et de pouvoir les manipuler dans Ocelet quelque soient leurs résolutions ou encore la forme de leurs cellules.

## 4.6 Conclusion

Le langage Ocelet a pour ambition de libérer le modélisateur de la gestion des contraintes spatiales par la manipulation de graphes. L'intégration de la représentation spatiale régulière dans le langage a nécessité une implémentation permettant l'incorporation de données raster de manière simplifiée dans le modèle en développant les datafacers appropriés. Pour respecter le formalisme du langage, une nouvelle propriété cellule a été intégrée permettant de caractériser la représentation d'une entité. Les autres développements majeurs ont permis de proposer directement des graphes implicites de voisinages ou par correspondance spatiale utilisables dans le modèle qui sont au nombre de trois.

- Un graphe de voisinage entre les cellules d'une même couche. Le type de voisinage dépendant alors de la forme de la cellule, carrée, hexagonale ou triangulaire. Elle dépend aussi du type de distance utilisée pour former le voisinage : distance de Tchebychev, Manhattan ou encore le rayon d'un cercle.
- Un graphe traduisant le changement de représentation entre des entités cellulaires et des entités vectorielles. Ce type de graphe est aussi utilisable avec des entités sans représentation spatiale. Dans ce cas les connexions entre les entités sont explicitées lors de la construction du modèle.
- Un graphe traduisant le changement d'échelle entre des couches différentes d'entités cellulaires.

Les implémentations effectuées ont suivi les développements conceptuels du chapitre 3. Pour les tester, nous proposons dans le chapitre suivant trois cas d'application de modélisation de dynamiques spatiales.

# Chapitre 5

## Cas d'application

### Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>92</b>
<b>5.2</b>	<b>Modèle de ruissellement</b>	<b>93</b>
5.2.1	Objectifs du modèle	93
5.2.2	Entités, variables d'état	94
5.2.3	Processus et scénarios	96
5.2.4	Principes de conception	97
5.2.5	Initialisation	100
5.2.6	Données d'entrée	101
5.2.7	Résultats et discussion	103
<b>5.3</b>	<b>Modèle de diffusion de plantes invasives</b>	<b>105</b>
5.3.1	Objectifs du modèle	106
5.3.2	Entités, variables d'état	106
5.3.3	Processus et scénarios	107
5.3.4	Principes de conception	108
5.3.5	Initialisation	110
5.3.6	Données d'entrée	111
5.3.7	Résultats et discussion	112
<b>5.4</b>	<b>Système d'alerte précoce en sécurité alimentaire</b>	<b>115</b>
5.4.1	Objectifs du modèle	115
5.4.2	Entités, variables d'état	116
5.4.3	Processus et scénarios	117
5.4.4	Principes de conception	118
5.4.5	Initialisation	120
5.4.6	Données d'entrée	121
5.4.7	Résultats et discussion	122
<b>5.5</b>	<b>Conclusion sur les cas d'application</b>	<b>128</b>

---

## 5.1 Introduction

Ce chapitre présente des cas d'application pour modéliser des dynamiques dans les paysages en utilisant les méthodes développées. Nous avons voulu tester notre approche sur trois cas provenant de thématiques très différentes dans lesquels il semblait pertinent d'utiliser des données raster et vecteur combinées. Même si ces modèles ont été développés en collaboration avec des experts de domaine, en utilisant leurs connaissances des principaux processus inclus dans les modèles, cet exercice n'a pas pour objectif de réaliser des modèles "scientifiquement validés". L'objectif ici est bien de démontrer les nouvelles capacités d'expression en modélisation sur des cas d'application concrets. Cependant nous menons l'exercice à son terme, c'est-à-dire, nous effectuons les tests de validation nécessaires pour montrer que les modèles simulent bien les processus décrits, sachant que ces modèles ne sont que des premières versions de modèles que les experts des domaines peuvent améliorer ultérieurement.

Deux des modèles présentés ci-après ont leurs zones d'étude à la Réunion, île volcanique située dans l'Océan indien. Le premier modèle simule le ruissellement dans le bassin versant de Saint Gilles à l'Ouest de l'île, lors d'un épisode cyclonique (Dina 2002) et sur une période plus étalée d'une année. Le deuxième modèle simule la diffusion d'une plante invasive exotique l'ajonc d'Europe (*Ulex europaeus*) en zone d'altitude à l'intérieur du Parc Naturel, en prenant en compte un important incendie qui a eu lieu en 2010. Le dernier modèle permet de suivre les rendements pour divers types de cultures de céréales sèches présents en Afrique de l'Ouest. L'intégration du pavage régulier dans Ocelet permet de spatialiser un modèle de culture en le faisant fonctionner dans des conditions différentes pour chaque cellule du pavage. Le modèle de culture calcule des bilans carboné et hydrique du couvert végétal, en prenant en compte les cycles phénologiques. L'intérêt de notre approche étant la possibilité d'utiliser des images et séries temporelles d'images satellite.

Pour décrire ces modèles nous suivons les préconisations du protocole *ODD* (Grimm et al., 2010) développé à l'origine pour normaliser la description de modèles multi-agents. Même si nos modèles ne sont pas basés sur ce paradigme, ils présentent suffisamment de caractéristiques communes pour que l'on puisse s'appuyer sur *ODD* pour les décrire. Ainsi pour chaque modèle, sont présentés son objectif, les entités du modèle ainsi que les variables d'états, les processus et les scénarios principaux, le principe de conception, des éléments d'initialisation des modèles, et les données d'entrées utilisées. Nous ne traitons pas l'aspect "sous-modèle", prévu dans le protocole *ODD* mais non transposable à Ocelet. Nous terminons la description des modèles par une partie discussion des résultats de simulation ainsi que des perspectives sur ces modèles.

## 5.2 Modèle de ruissellement

Un bassin versant est une délimitation naturelle du territoire en fonction du relief et des lignes de crêtes. L'ensemble de l'eau précipitée sur ce territoire est drainé vers un point de sortie, l'exutoire. L'eau va se frayer un chemin vers l'exutoire par ruissellement de surface et par infiltration dans les sols vers les nappes souterraines, tout en s'évaporant en partie. Le ruissellement de l'eau en surface va suivre une orientation inhérente au relief et l'eau s'infiltrera plus ou moins selon les types de sols, l'occupation du sol, etc. Lorsque le territoire est restructuré, modifiant les surfaces et l'occupation du sol, le ruissellement et l'infiltration seront aussi changés, ainsi que les variations de débit à l'exutoire. En particulier, l'urbanisation entraîne l'imperméabilisation des surfaces, accélérant le ruissellement et provoquant des montées d'eau rapides en aval.

C'est notamment le cas sur l'île de la Réunion. Le relief de l'île est très escarpé, et les versants volcaniques sont façonnés par des ravines qui ont des caractéristiques hydrologiques particulières (Raunet, 1991 ; Barcelo, 1996). Le climat de l'île est aussi complexe. S'il est généralement classé comme de type tropical humide, de très grandes différences microclimatiques existent en raison de la structure topographique de l'île. Celle-ci est aussi sur la route des cyclones qui surviennent généralement en fin et début d'année, apportant une pluviométrie très importante pendant quelques jours d'affilée. Avec une population de plus de 800.000 habitants, en croissance d'environ 1,5 % chaque année, l'urbanisation est un processus bien visible.

Les caractéristiques particulières du fonctionnement hydrologique de l'île font que les modèles classiques de l'hydrologie sont inadaptés (Barcelo, 1996). Le comportement des ravines réunionnaises reste mal connu alors qu'elles façonnent l'ensemble des bassins versants qui sont généralement de petite taille et à flanc de colline. La grande majorité des ravines restent sèches presque tout le long de l'année. Ces caractéristiques expliquent un temps de concentration (la durée nécessaire pour une goutte d'eau de parcourir le chemin d'un point du bassin versant jusqu'à l'exutoire) court et des crues soudaines. Ce phénomène est accentué par l'urbanisation non règlementaire où l'eau pluviale collectée est envoyée directement dans les ravines.

### 5.2.1 Objectifs du modèle

La modélisation du ruissellement dans le contexte réunionnais nous a semblé intéressant pour mettre à l'épreuve la capacité de manipuler des entités linéaires (routes, ravines), surfaciques (parcelles) et cellulaires (modèle numérique de terrain) dans un même modèle. Nous nous intéressons plus particulièrement au bassin versant de Saint Gilles, en nous basant sur deux études à des échelles temporelles très différentes, l'une s'est intéressée à l'épisode cyclonique Dina de 2002 (Roques,

2013) et l'autre, à une période plus étalée sur l'année 2009 (Anh Dao, 2015). Ces deux études se basent sur un modèle construit avec HEC-HMS (HMS, 2010, 2000) où le BV de Saint Gilles est décomposé en 7 sous-bassins caractérisés chacun par des paramètres moyens. La conception de notre modèle intègre lui la connaissance des processus de ruissellement à deux échelles plus fines : l'une mettant en avant les flux d'eau entre entités linéaires telles que les routes et les ravines, et l'autre opérant sur un pavage régulier où chaque élément du pavage peut échanger des quantités d'eau avec ses voisins. Contrairement au modèle précédent ce présent modèle permettrait de prendre en compte explicitement les effets de l'urbanisation et le comportement des ravines dans le ruissellement.

### 5.2.2 Entités, variables d'état

**CellEntity** : l'espace du bassin versant étudié est divisé pour obtenir un pavage régulier. Chaque entité du pavage est représentée par une cellule carrée de 20m de résolution. Ce type d'entité est caractérisé par des propriétés que nous pouvons séparer en deux catégories : (i) les propriétés statiques apportant l'information spatiale, (ii) les propriétés variables, ou variables d'états caractérisant l'évolution de l'état de cette entité. Un total de 229 072 entités de ce type est géré lors de la simulation.

Propriétés statiques :

- Altitude moyenne de la cellule (en m)
- NDVI
- Volume du réservoir d'eau en sous-sol (en m<sup>3</sup>)
- Vitesse d'infiltration de la surface vers le réservoir (en m<sup>3</sup> par pas de temps)
- Vitesse de vidange du réservoir (en m<sup>3</sup> par pas de temps)
- Taux de ruissellement local (qui dépend de l'occupation du sol et de la valeur en NDVI)
- Station de mesure pluviométrique la plus proche

Propriétés d'état :

- Volume d'eau de surface résultant du bilan hydrique (en m<sup>3</sup>)
- Volume d'eau entrant sur la cellule au cours du pas de temps (en m<sup>3</sup>)
- Volume d'eau quittant la cellule au cours du pas de temps (en m<sup>3</sup>)
- Volume d'eau présent dans le réservoir en sous-sol (en m<sup>3</sup>)

**Linear** : les entités qui représentent deux types d'objets du paysage sur lesquels un processus de ruissellement est appliqué : les routes ou les ravines. Leur représentation est une géométrie linéaire. Celle-ci est caractérisée par l'altitude moyenne qui est attribuée par une relation avec les entités MntCell. Cette altitude sert à savoir quelle entité est en amont d'une autre. Un total de 4190 entités de ce type est géré

lors de la simulation.

Propriétés statiques :

- Altitude moyenne (en m)
- Taux de ruissellement local (qui dépend du type de linéaire)
- Type de linéaire (route ou ravine)

Propriétés d'état :

- Volume d'eau de surface résultant du bilan hydrique des cellules (en m<sup>3</sup>)
- Volume d'eau entrant sur le linéaire au cours du pas de temps (en m<sup>3</sup>)
- Volume d'eau quittant le linéaire au cours du pas de temps (en m<sup>3</sup>)

**MntCell** : entité représentée par une cellule carrée de 10m de résolution. Elle est caractérisée par une propriété donnant l'information sur l'altitude en mètre d'un point de l'espace. Cette entité sert uniquement à apporter l'information en altitude aux entités "CellEntity". Un total de 903 100 entités de ce type est géré lors de la simulation.

**NdviCell** : entité représentée par une cellule carrée de 10m de résolution. Elle est caractérisée par une propriété donnant l'information sur la valeur en NDVI d'un point de l'espace. Cette entité sert uniquement à apporter l'information en NDVI aux entités "CellEntity". Un total de 903 100 entités de ce type est géré lors de la simulation.

**Plot** : entité représentée par un polygone. Elle est caractérisée par une propriété qui spécifie le type de surface : urbanisée ou végétalisée. Un total de 5292 entités de ce type est géré lors de la simulation.

**Node** : une entité représentée par une géométrie de type point. Elle sert à répartir les flux entre les entités "Linear" selon leur position en amont ou en aval du nœud. Une entité "Node" est créée à chaque intersection géométrique entre les entités linéaires. Un total de 3311 entités de ce type est géré lors de la simulation.

**RainStation** : les précipitations sont enregistrées aux stations pluviométriques suivantes : Tan Rouge, Piton Maido, Petite France, La Saline les Hauts et l'Ermitage. Nous avons utilisé la méthode des polygones de Thiessen pour appliquer la répartition des précipitations sur les entités. Chaque station pluviométrique est ainsi représentée par un polygone. Nous avons donc 5 entités de ce type correspondant aux 5 stations pluviométriques.

**Soil** : l'entité est représentée par une géométrie de type polygone. Elle indique

le type de sol des zones qui vont servir à attribuer un réservoir aux entités "CellEntity". Les types de sols sont : Andic Podzol, Silandic andosol, Andic Cambisol, Haplic Cambisol, Leptic Phaeozem et Leptic Cambisol. Un total de 123 entités de ce type est géré lors de la simulation.

### 5.2.3 Processus et scénarios

Les processus ont pour objectif de montrer que l'on peut modéliser des phénomènes à différentes échelles en utilisant différentes formes de représentations spatiales. Le premier processus de ruissellement est traduit sur les entités "CellEntity". Leur quantité d'eau va évoluer au cours du temps selon plusieurs principes :

- Une quantité d'eau est attribuée aux entités venant des précipitations
- Cette quantité va ruisseler si les précipitations sont supérieures à un seuil ou si le réservoir d'eau du sol de l'entité est saturé.
- Le phénomène d'infiltration est appliqué et va tenir compte du réservoir d'eau. Ce réservoir a un volume en eau attribué selon le type sol.
- La quantité d'eau infiltrée est déterminée selon le type de surface de l'entité. Pour cela nous utilisons d'une part l'information venant des entités "Plot" pour savoir si la surface est urbanisée ou végétalisée et d'autre part une fonction de seuil en tenant compte de l'indice de végétation (NDVI) donné par les entités "NdviCell".
- L'eau infiltrée est ensuite attribuée au réservoir. Si le réservoir est saturé ou si les précipitations dépassent le seuil de ruissellement, une quantité d'eau pouvant ruisseler est déterminée.
- Enfin l'eau ruisselle de cellule en cellule en fonction des pentes entre cellules. Les pentes sont obtenues à partir du modèle numérique de terrain.

Le deuxième processus de ruissellement est lancé sur les entités vectorielles linéaires connectées avec leurs entités nœuds correspondant à la jonction entre chacune d'entre elles. Dans ce modèle chaque géométrie linéaire est un segment.

- La quantité d'eau obtenue après le calcul de flux d'eau au niveau cellulaire est attribuée aux entités linéaires respectives.
- La quantité d'eau reçue par chaque linéaire est attribuée au nœud directement connecté au linéaire en aval de celui-ci.
- La quantité d'eau récupérée par le nœud est ensuite redistribuée sur les entités linéaires en aval des nœuds.
- Finalement une fonction d'infiltration est appliquée sur la quantité d'eau des linéaires.

### 5.2.4 Principes de conception

Les processus sont lancés dans le scénario par l'appel à des interactions dans cet ordre : (i) l'interaction de flux au niveau des entités "CellEntity" et (ii) la mise à jour de la quantité d'eau des entités "CellEntity" vers la couche d'entités linéaires. Enfin le processus de flux sur les entités linéaires est lancé. Le pas de temps du modèle est de 2 min sur 5 jours. Pour ce modèle nous mettons à jour les précipitations sur l'ensemble des cellules. Nous ramenons simplement la valeur horaire des précipitations en divisant par le nombre de pas de temps en une heure.

La zone d'étude est divisée régulièrement en entités "CellEntity". Nous calculons le ruissellement entre ces entités cellulaires ainsi que le bilan hydrique sur chaque entité à des pas de temps réguliers. Le bilan hydrique représente la différence entre l'eau qui arrive sur une entité (pluie, ruissellement provenant de cellules en amont) et qui quitte la même entité (ruissellement vers des entités en aval, infiltration).

Nous avons en cela repris les principes de base du modèle HMS (Hydrological Modelling System) mis au point au Laboratoire de Géoscience de l'Université de La Réunion mais en simplifiant certains de ses aspects. En particulier, dans cette version du modèle, nous n'avons pas pris en compte l'eau qui est retenue par la végétation et évacuée par évapotranspiration. Mais le modèle présente l'intérêt de pouvoir agir sur chaque élément de l'espace à l'intérieur de la zone d'étude de façon à tester l'influence d'une action en un lieu précis, et aussi de pouvoir produire des représentations cartographiques dynamiques de l'évolution du ruissellement en chaque point. Ce modèle est donc le résultat d'un compromis entre le niveau de complexité et le nombre de processus à prendre en compte d'une part, et l'objectif de produire un outil sur lequel on puisse agir sur des lieux précis et observer l'influence de ces actions sur la distribution spatiale et temporelle du ruissellement d'autre part.

Pour simuler le fait que l'urbanisation, et en particulier la présence de routes, a un impact sur le ruissellement, et que les ravines représentent des vecteurs de flux d'eau, nous incluons le phénomène de ruissellement sur les linéaires. Nous construisons une couche d'entités vectorielles qui ont une géométrie linéaire et qui échangeront des quantités d'eau selon leurs pentes. Le graphe d'entités vectorielles est construit explicitement dans le modèle, et pour répartir des flux, nous ajoutons à ce graphe une entité "Node" qui va redistribuer l'eau selon les linéaires en aval du nœud, l'eau arrivant sur les nœuds venant des linéaires en amont. La quantité d'eau de ces linéaires provient directement de la quantité d'eau venant de leurs cellules respectives calculée avec le processus de transfert d'eau entre cellules.

**CellFlow** : il s'agit d'une relation de voisinage immédiat entre les entités "CellEntity". Le type de voisinage utilisé ici est le voisinage de Moore. Les fonctions d'interaction de cette relation décrivent les échanges possibles de quantité d'eau selon

la pente entre entités voisines représentés par la figure 5.1. La fonction d'infiltration

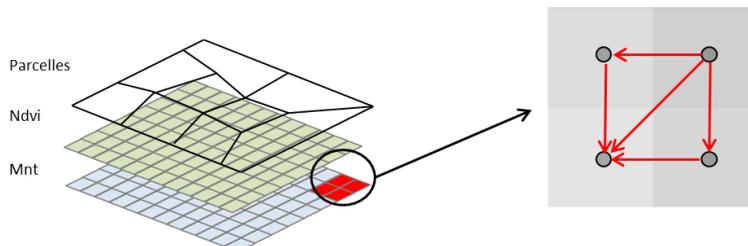


FIGURE 5.1 – Représentation de diverses couches d'information telles que le modèle numérique de terrain, une couche de NDVI ainsi que le parcellaire. Les flux sont représentés en tenant compte de l'altitude.

tient compte d'un seuil à partir duquel les précipitations ruissellent entièrement. Si les précipitations sont en dessous de ce seuil, l'eau infiltrée est calculée selon le type de surface ainsi qu'un pourcentage par rapport à la valeur en NDVI de la cellule. Une fois la fonction d'infiltration appliquée à l'ensemble des entités "CellEntity", nous appliquons une interaction traduisant les flux de proche en proche dont une définition simplifiée en Ocelet est donnée par la figure 5.2.

```

relation CellFlow<CellEntity c1, CellEntity c2> {
    interaction flow(){
        c1.water = f(c2.water, c1.mnt, c2.mnt)
        c2.water = f(c1.water, c1.mnt, c2.mnt)
    } agg{
        c1.water << Sum
        c2.water << Sum
    }
}

```

FIGURE 5.2 – Définition de la relation et de la fonction d'interaction traduisant les flux d'eau entre les entités cellulaires.

**CellPlot** : relation entre les entités "Plot" et les entités "CellEntity", reliées par correspondance spatiale. L'interaction de cette relation n'est pas utilisée lors de la dynamique du modèle : elle sert principalement à attribuer un type de surface, urbanisée ou végétalisée aux entités "CellEntity". Cette interaction pourrait être appelée de manière dynamique si nous tenions compte d'un changement du type de sol des entités "Plot".

**CellLinear** : relation entre les entités linéaires "Linear" représentant les routes et les ravines permettant de les connecter aux entités cellulaires "CellEntity" qui leur correspondent spatialement sur le pavage régulier. La quantité d'eau des entités linéaires est mise à jour après l'interaction de flux de la relation "CellFlow", schématisée par la figure 5.3 (b). La mise à jour de la quantité d'eau entre les entités

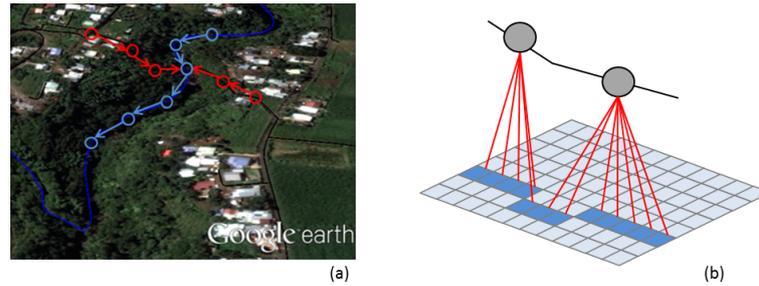


FIGURE 5.3 – Représentation du graphe de flux entre les entités linéaires et leurs nœuds d'intersection (a). Représentation du graphe entre les entités linéaires et les entités cellulaires (b).

cellules et les entités linéaires se lance par une interaction après le processus de flux entre les cellules à l'aide d'une relation connectant les linéaires vers leurs entités cellulaires respectives. La définition de la relation entre ces entités est présentée par la figure 5.4.

```

relation CellLinear<CellEntity c, Linear linear> {
    interaction watertoLinear() {
        linear.water = c.water
    } agg linearFlow() {
        linear.water << Sum
    }
}

```

FIGURE 5.4 – Définition de la relation en Ocelet et de l'interaction traduisant les échanges d'eau entre les entités cellulaires et les entités linéaires.

**LinearNode** : relation entre les entités linéaires et leurs nœuds de jonction définie en Ocelet par la figure 5.5 et schématisée par la figure 5.3 (a). Les nœuds sont calculés par intersection entre les linéaires. Ces nœuds servent à redistribuer l'eau des linéaires en amont vers les linéaires en aval.

**RainToCell** : relation permettant d'attribuer la valeur des précipitations aux entités "CellEntity" correspondant spatialement aux entités "RainStation". Les données horaires des précipitations de chaque station pluviométrique sont ramenées au pas de temps de 2 min. Celles-ci sont attribuées aux entités "CellEntity" avec l'interaction décrite par la figure 5.6.

**MntToCellEntity** : cette relation ne contient qu'une seule fonction d'interaction qui permet de calculer l'altitude des entités "CellEntity" de 20m de résolution en calculant la moyenne des valeurs en altitude des entités "MntCell" d'une résolution de 10m.

```

relation LinearNode<Linear linear, Node node> {

    interaction linearFlow(){
        if(node.aval){
            node.water = linear.water
        } else {
            linear.water = node.water
        }
    } agg linearFlow(){
        linear.water << Sum
        node.water << Sum
    }

}

```

FIGURE 5.5 – Définition de la relation entre les entités linéaires et leurs nœuds à leurs intersections. La fonction d'interaction traduit les échanges de quantités d'eau entre les linéaires en amont ou en aval d'un nœud.

```

relation RainToCell<CellEntity c, RainStation r> {

    interaction setPrecipitation(Integer day){
        c.rain = r.getPrecipitation(day)
    }

}

```

FIGURE 5.6 – Définition en Ocelet de la relation permettant d'attribuer la valeur des précipitations vers les entités "CellEntity".

**NdviToCellEntity** : de même que la relation précédente, celle-ci sert à attribuer une valeur moyenne en NDVI aux entités "CellEntity" en prenant les valeurs en NDVI des entités "NdviCell".

## 5.2.5 Initialisation

### Phase d'instanciation

La phase d'initialisation du modèle comprend l'instanciation de toutes les relations, ainsi que leurs connexions, par exemple entre les entités "CellEntity" et les entités "Linear". C'est dans cette phase que les entités cellulaires sont initialisées et dont les valeurs en altitude ou indice de NDVI sont attribuées avec les datafacers *RasterFile*.

### Contrôle d'une simulation

C'est dans la première partie du scénario que sont affectées des valeurs aux paramètres du modèle présentés par la figure 5.7. On trouve en particulier la définition de la résolution des entités cellulaires sur lesquelles le calcul de flux d'eau a lieu,

les paramètres d'infiltration qui tiennent compte du type de sol selon la valeur en NDVI, le taux d'infiltration appliqué sur les linéaires, le pas de temps horaire et le seuil de ruissellement selon les précipitations.

```

scenario WaterFlow{
    //paramètres de contrôle
    fix ndvi_Inf = 0.45 // Facteur d'infiltration selon la valeur en NDVI
    fix road_Inf = 0.0 // Facteur d'infiltration pour les linéaires « routes »
    fix gully_Inf = 0.6 // Facteur d'infiltration pour les linéaires « ravines »
    fix urban_Inf = 0,0 // Facteur d'infiltration pour les « Plot » de type urbains
    fix crop_Inf = 0,5 // Facteur d'infiltration pour les « Plot » de type végétalisé
    ...
}

```

FIGURE 5.7 – Définition des paramètres de contrôles des taux d'infiltrations selon la valeur en NDVI ou encore des types de surface des entités.

### 5.2.6 Données d'entrée

**Zone d'étude :** la zone d'étude est délimitée par un polygone représentant le bassin versant de Saint Gilles, situé sur le territoire de la côte Ouest de la Réunion (figure 5.8).



FIGURE 5.8 – La zone d'étude se situe à l'Ouest de l'île de la Réunion, la limite (polygone rouge) du bassin versant de Saint Gilles présente des variations d'altitude allant de 150 à 2000 mètres.

**Couches du modèle numérique de terrain et de ndvi :** les couches du modèle numérique de terrain (figure 5.9 (e)) et du NDVI (figure 5.9 (f)) sont au format

raster. Celles-ci sont intégrées dans le modèle en utilisant le datafacer *RasterFile* pour obtenir les entités "MntCell" et "NDVICell", celles-ci servant à attribuer les valeurs aux entités "CellEntity".

**Stations pluviométriques** : nous utilisons les données horaires des précipita-

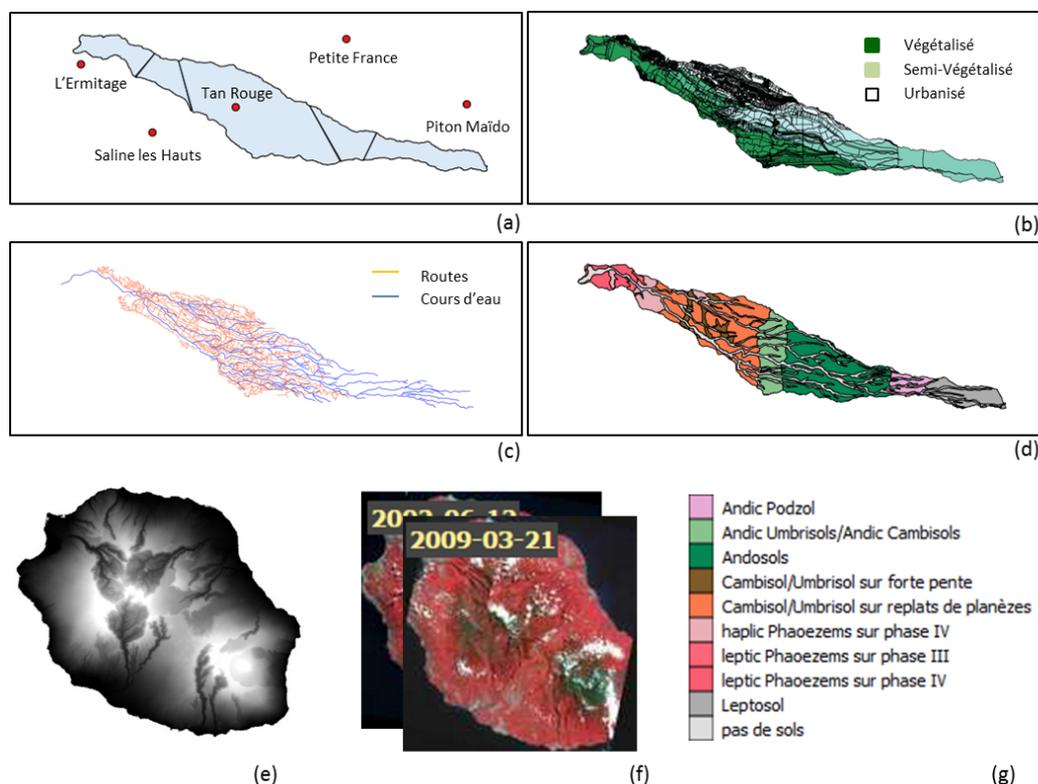


FIGURE 5.9 – Les différentes couches de données du modèle : les polygones de Thiessen (a) ; la couche du parcellaire (b) ; le réseau de routes et des cours d'eau (c) et la carte pédologique(d). Les différentes couches raster : avec le MNT (e) et les couches NDVI de l'image de 2002 pour l'épisode cyclonique et celle de 2009 pour la simulation sur l'année 2009 (f). Les différents types de sols (g) correspondant à la carte (d).

tions de cinq stations : l'Ermitage, Petite France, Piton Maïdo, Saline les Hauts et la station du Tan rouge. Nous répartissons spatialement ces données en utilisant la méthode des polygones de Thiessen (figure 5.9 (a)).

**Carte pédologique** : la carte pédologique (Feder and Bourgeon, 2009) utilisée est au format vecteur et caractérise les sols (figure 5.9 (d)). Cette caractérisation permet d'attribuer une profondeur pour le réservoir ainsi qu'une vitesse de ruissellement.

**Parcellaire, routes et ravines** : ces données sont au format vecteurs et contiennent des géométries : des polygones pour le parcellaire (figure 5.9 (b)) qui sont attribués aux entités "Plot" et des lignes pour les routes et les ravines (figure 5.9 (c)) qui sont attribuées aux entités "Linear".

Le tableau 5.1 présente un récapitulatif des données d'entrée.

Nom	Format	Résolution	Source	Pas de temps
Parcellaire	Vecteur	NA	DDE Réunion	NA
Linéaire	Vecteur	NA	BD TOPO	NA
Carte pédologique	Vecteur	NA	(Feder and Bourgeon, 2009)	NA
Polygones de Tiessen	Vecteur	NA	NA	NA
Couche MNT	Raster	10m	IGN	NA
Couches NDVI	Raster	10m	SPOT5	NA
Pluviométrie	CSV	NA	Météo France	horaire

TABLE 5.1 – Récapitulatif des données d'entrée du modèle du ruissellement.

### 5.2.7 Résultats et discussion

#### Simulation des débits d'eau lors de l'épisode cyclonique Dina de 2002

Pour la simulation de cet épisode, nous nous basons sur les précipitations données par la station du Tan Rouge (figure 5.10 (a)), ces précipitations sont affectées aux entités de la zone. Les données horaires sont divisées pour obtenir les valeurs applicables au pas de temps de deux minutes. la figure 5.10 (b) présente les débits simulés à l'exutoire du bassin versant ainsi que les débits observés.

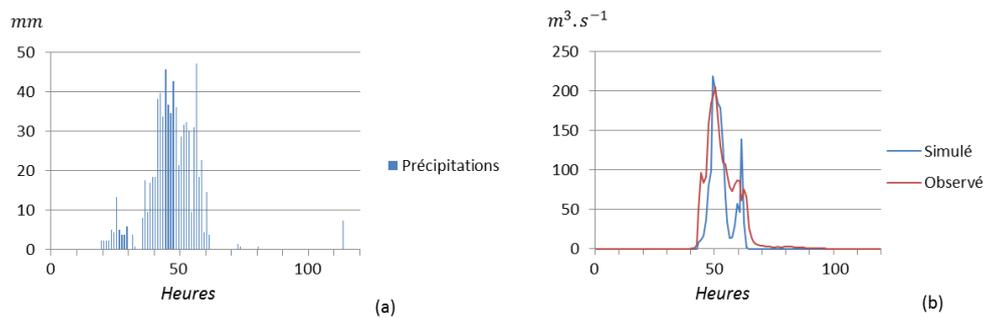


FIGURE 5.10 – En (a) les données des précipitations horaires venant de la station du Tan Rouge, utilisées pour simuler les débits. Comparaison des débits simulés et observés à l'exutoire lors de l'épisode cyclonique Dina de 2002 (b).

### Simulation du ruissellement sur l'année 2009

Pour la simulation sur l'ensemble de l'année de 2009, nous avons gardé les mêmes processus que ceux utilisés pour l'épisode cyclonique. Nous avons cependant utilisés

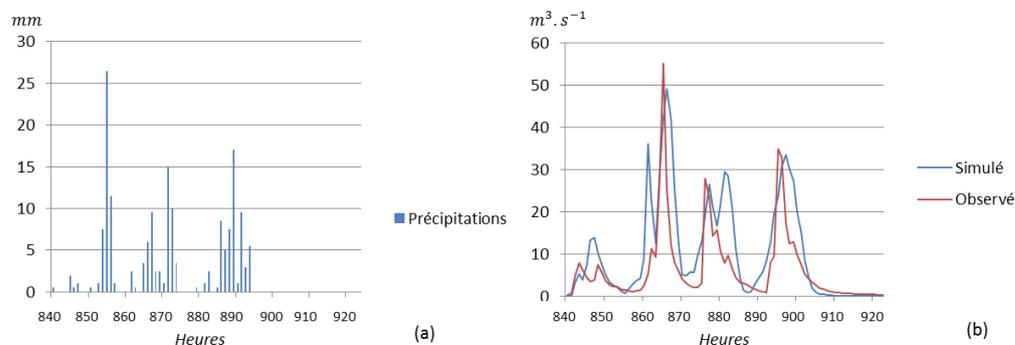


FIGURE 5.11 – En (a) les données des précipitations horaires venant de la station du Tan Rouge. Comparaison des débits simulés et observés à l'exutoire le 05/02/2009 (b).

les données pluviométriques de cinq stations et les avons spatialisées avec la méthode des polygones de Thiessen.

Pour garder des temps de calcul raisonnables sur une année au pas de temps de deux minutes, les calculs sont lancés lorsqu'il y a des épisodes pluvieux sur une station de plus de 10 heures. La figure 5.11 (b) présente deux courbes sur les débits observés et simulés, le 05/02/2009 (840ème heure) où un enchaînement de plusieurs épisodes pluvieux a eu lieu 5.11 (a).

Les résultats du modèle sont satisfaisants sur les valeurs des débits observés et simulés. Nous avons définis les fonctions d'infiltrations, et la gestion du stock d'eau des réservoirs en nous basant sur les dires d'experts. Celles-ci sont améliorables. Le résultat sur la méthode montre que l'on peut multiplier la définition de processus que ce soit sur des entités cellulaires ou des entités linéaires, ou les deux. Nous avons aussi montré que nous pouvions utiliser une multitude de sources de données avec plusieurs couches rasters et vecteurs.

Ce modèle ouvre des voies notamment en termes de couplage de modèle. Il serait en effet intéressant de coupler ce modèle dans lequel l'infiltration est influencée par l'urbanisation avec un modèle d'expansion de l'urbanisation par exemple ou encore un modèle sur les pratiques agricoles.

## 5.3 Modèle de diffusion de plantes invasives

Dans les hauteurs de l'île de la Réunion, l'*Ulex europaeus* (figure 5.12), plus connu sous le nom d'ajonc d'Europe, a fait son apparition il y a quelques années. Il s'agit d'une espèce végétale de la famille des *Fabaceae*, aussi appelé genêt épineux. Celle-ci est naturellement présente en Europe, notamment dans les paysages bretons.



FIGURE 5.12 – Photo de l'*Ulex europaeus* à l'île de la Réunion.

Si l'île se situe en milieu tropical, le climat dans les hauteurs est plus frais, ce que semble particulièrement affectionner le genêt. L'espèce n'est cependant pas naturellement présente sur l'île, celle-ci a été introduite. Bien que le mode d'introduction reste inconnu, le foyer d'apparition serait situé dans la Plaine des Cafres (Atlan and Tarayre, 2012), un plateau proche du centre de l'île. Les conditions climatiques étant favorables, l'espèce s'est répandue dans les hauteurs, et s'est développée en gagnant du terrain sur les autres espèces végétales. Telles sont les caractéristiques d'une espèce envahissante. Celle-ci fait partie des 100 espèces les plus envahissantes au monde (Lowe et al., 2000). Dans un écosystème, les espèces envahissantes ont des impacts écologiques, avec une influence sur les autres espèces et leurs habitats. Dans un milieu insulaire, elles sont considérées comme la première cause d'extinction sur les espèces déjà présentes dans le milieu (Reaser et al., 2007).

Comment gérer ce type d'espèce ? Quel est son mode de fonctionnement ? A quel point représente-t-elle une menace ? Pour répondre à ces questions, plusieurs démarches sont possibles. La zone de couverture de l'espèce est grande. Mettre en équations la progression de l'espèce dans la région semble difficile. Pour y arriver, une démarche à explorer est de modéliser avec les connaissances actuelles le fonctionnement de l'organisme, d'étudier son développement, sa croissance et sa diffusion sur les terres.

### 5.3.1 Objectifs du modèle

Nous proposons de développer un modèle de diffusion de l'ajonc d'Europe en nous basant sur le récent travail de Commins (2015). Cette étude apporte des éléments de compréhension sur la croissance et l'expansion de l'*Ulex europaeus* dans les plaines des hauts de l'Ouest de la Réunion. La méthode proposée pour gérer les pavages réguliers en faisant interagir les éléments selon leurs voisinages est ici utilisée pour traduire la diffusion de graines de l'espèce de proche en proche pour une surface colonisée. Plusieurs types de données sont utilisés, des indices de végétation nous indiquant quels seraient les foyers potentiels pour l'*Ulex europaeus*, un modèle numérique de terrain, avec lequel l'information en altitude nous permet d'orienter la diffusion. Après la dissémination des graines intervient la phase de croissance des plantes. Celles-ci donneront à leur tour des graines. Il a été relevé que les incendies ont un impact non négligeable sur le développement de l'espèce. Dans un premier temps un incendie va réduire le nombre d'individus, mais par la suite, la croissance du genêt est plus importante car l'espace est moins encombré. C'est la dynamique de ces phénomènes que nous proposons de décrire en simulant la diffusion et la croissance de l'*Ulex europaeus* dans les plaines des hauts de l'Ouest. Nous nous basons sur des données indiquant les zones colonisées par l'espèce en 2004. En y appliquant les processus sur les éléments que nous décrivons par la suite nous pouvons observer la dynamique de propagation, ou encore, comparer les résultats avec une carte des zones peuplées par l'espèce en 2012.

### 5.3.2 Entités, variables d'état

**UlexCell** : l'espace de la zone d'étude est discrétisé en entités cellulaires d'une résolution de 5m. Chaque entité de ce type est caractérisée par des propriétés que nous pouvons séparer en deux catégories : (i) les propriétés statiques caractérisant la portion de l'espace de la cellule, (ii) les propriétés variables, ou variables d'états caractérisant l'évolution de l'état d'une cellule. Pour une résolution de 5 mètres, un total de 2 211 120 entités de ce type est géré lors de la simulation.

Propriétés statiques

- Altitude de l'entité (en m)
- NDVI, indice de végétation par différence normalisée.

Propriétés d'état

- Densité de présence (de 0 à 5)

**MntCell** : entité représentée par une cellule carrée de 10m de résolution. Elle est caractérisée par une propriété donnant l'information sur l'altitude en mètre d'un

point de l'espace. Cette entité sert uniquement à apporter l'information en altitude aux entités "UlexCell". Un total de 550 122 entités de ce type est géré lors de la simulation.

**NDVICell** : ces entités contiennent la valeur en NDVI pour une résolution de 10m. Cette information est mise à jour à chaque pas de temps venant d'une série temporelle d'images satellite contenant les valeurs en NDVI de 2004 à 2012. Un total de 550 122 entités de ce type est géré lors de la simulation.

**Coverage** : entités ayant un polygone comme représentation correspondant à la surface de plantes observées en 2004 et en 2012.

**StudyArea** : aire d'étude totale représentée par un polygone.

**FireZone** : entité représentée par une géométrie spécifiant la zone d'incendie de 2010.

La dynamique du modèle est traduite par un enchaînement de processus pour simuler les phénomènes de diffusion et de croissance des plantes selon plusieurs facteurs qui sont traités dans la section suivante.

### 5.3.3 Processus et scénarios

Le processus de diffusion de proche en proche est traduit au niveau cellulaire à une résolution de 5 mètres sur l'ensemble de la zone des plaines des hauts de l'Ouest de l'île de La Réunion. Cette résolution peut cependant varier dans les paramètres de contrôle de la simulation que nous présentons dans la partie 5.3.5. Les entités "UlexCell" ont une quantité de graines ainsi qu'une surface colonisée qui évoluent au cours du temps :

- Une surface couverte par la plante est attribuée aux cellules. Cette surface étant calculée à partir de données vectorielles par intersection géométrique avec la forme des cellules.
- La production de graines sur une cellule est calculée à chaque pas de temps en fonction de la surface de plante.
- Le processus de croissance des plantes est ensuite appliqué et dépend de plusieurs facteurs : un pourcentage de graines vont évoluer en individus adultes tandis qu'un pourcentage va disparaître.
- Au pas de temps suivant, les individus adultes vont à leur tour pouvoir diffuser des graines.

Pour montrer l'impact des incendies sur la densité des plantes nous faisons intervenir un processus événementiel pour l'année 2010 uniquement. Il consiste simplement à réduire la densité des surfaces des plantes des entités "UlexCell" dans la zone où l'incendie a été observé.

### 5.3.4 Principes de conception

L'ensemble de la zone d'étude est divisé pour obtenir un pavage d'entités "UlexCell". Nous calculons la diffusion de graines ainsi que la croissance des plantes sur chaque entité au pas de temps annuel. Nous proposons ici de décrire les principales relations entre les entités du modèle.

**MntToUlexCell** : permet d'établir un lien entre les entités "MntCell" et les entités "UlexCell". Les entités "MntCell" contiennent une valeur en altitude venant du modèle numérique de terrain de 10 mètres de résolution. Cette relation ne contient qu'une seule fonction d'interaction qui permet de calculer l'altitude moyenne selon la résolution des cellules des entités "UlexCell" de chaque cellule et d'affecter la propriété correspondante.

**NDVIToUlexCell** : permet d'établir le lien entre les entités "NDVICell" contenant les valeurs en NDVI et les entités "UlexCell". Cette relation ne contient aussi qu'une seule fonction d'interaction (figure 5.13) qui permet d'attribuer la valeur en NDVI à chaque pas de temps. La valeur en NDVI des entités "NDVICell" est attribuée par un datafacer de série temporelle d'images. L'interaction est appelée à chaque pas de temps pour mettre à jour la valeur en NDVI des entités "UlexCell".

```

relation NDVIToUlexCell<NDVICell n, UlexCell u> {
    interaction setNdvi() {
        u.ndvi = n.ndvi
    } agg linearFlow() {
        u.ndvi << Mean
    }
}

```

FIGURE 5.13 – Relation entre les entités "NDVICELL" et les entités "UlexCell" pour attribuer une valeur en NDVI aux entités "UlexCell".

**Neighbours** : il s'agit d'une relation de voisinage entre les entités "UlexCell". La relation contient une interaction traduisant la diffusion de proche en proche comme présenté sur la figure 5.14 (a) et sur la figure 5.14 (c). La quantité diffusée tient compte de la pente entre deux entités ainsi que de la couverture de surface végétale. Dans ce modèle nous proposons de faire varier la distance de voisinage entre 1 et 6 (distance de Tchebychev), la quantité de graines diffusée tient compte de la distance séparant les entités.

**AreaToUlex** : relation entre la zone d'étude représentée par un polygone et les entités "UlexCell". L'interaction applique un processus de croissance des plantes sur chaque entité "UlexCell". Ce processus tient compte du nombre de graines que contient l'entité ainsi que deux seuils sur la valeur en NDVI des entités cellulaires.

La croissance des plantes est ainsi possible seulement s'il s'agit d'un milieu partiellement végétalisé (seuil supérieur à 0.45) et semi ouvert (seuil inférieur à 0.75) comme présenté sur la figure 5.14 (b).

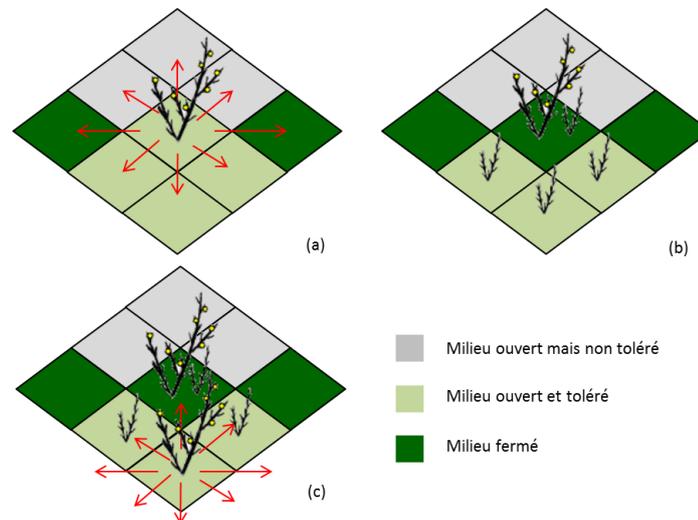


FIGURE 5.14 – Schémas de la diffusion de proche en proche des graines (a) et (c) pour un voisinage de distance de 1. Le processus de croissance des plantes est appliqué lorsque le milieu est ouvert et toléré par l'espèce (b).

**UlexFire** : c'est une relation entre les entités "UlexCell" en correspondance spatiale avec l'entité "FireZone". L'interaction est appelée une fois lorsque le pas de temps de la simulation arrive à l'année 2010. Toutes les entités "UlexCell" ont alors une décroissance sur la densité de plantes.

```

relation CoverageToCell<UlexCell c, Coverage cov> {
    interaction setCoverage() {
        let inter = c.cell.toPolygon.intersection(cov.polygon)
        c.coverage = inter.area
    } agg {
        c.coverage << Sum
    }
}

```

FIGURE 5.15 – Relation dont l'interaction permet de définir une aire de couverture aux entités "UlexCell" par intersection géométrique entre la cellule des entités et le polygone des entités "Coverage".

**CoverageToCell** : relation entre les entités "UlexCell" et les entités "Coverage". Les entités "Coverage" étant des polygones correspondant à la surface de colonisation des plantes étudiées, la surface de plante est attribuée aux entités "UlexCell"

en utilisant une interaction calculant leurs intersections géométriques (figure 5.15). Dans cette interaction l'aire des intersections est cumulée par la fonction d'agrégation "Sum" entre les polygones spécifiant la zone des plantes avec l'intersection de la forme géométrique des cellules.

### 5.3.5 Initialisation

#### Paramètres de contrôle d'une simulation

Nous avons spécifié dans le modèle plusieurs paramètres de contrôles présentés par la figure 5.16.

```
scenario UlexSpread{
  //paramètres de contrôle
  fix NDVI_MIN = 0.45 // Seuil minimum en NDVI pour la croissance
  fix NDVI_MAX = 0.75 // Seuil maximal en NDVI pour la croissance
  fix GROWTH = 0.2 // Facteur de croissance des plantes
  fix RESOLUTION = 5.0 // Résolution des cellules des entités UlexCell
  fix neighbourDistance = 1.0 // distance de voisinage (distance de
  Tchebychev)
  ...
}
```

FIGURE 5.16 – Les paramètres de contrôles définis dans le scénario en Ocelet.

Les paramètres permettent de contrôler les seuils en NDVI pour définir si le milieu est propice pour une graine qui va alors évoluer en individu adulte. Si c'est le cas le paramètre "GROWTH" de croissance est appliqué et détermine la densité d'individu obtenu au temps  $t+1$  par rapport à la densité d'individus au temps  $t$ . Des paramètres de contrôle sur l'espace permettent ensuite de spécifier la résolution des entités "UlexCell" ainsi que le nombre de voisins pris en compte dans la relation de diffusion.

#### Phase d'instanciation

En plus des paramètres de contrôle, la phase d'initialisation comprend l'instanciation des relations ainsi que de la lecture des données d'entrée en utilisant les datafacers raster et/ou vecteurs. Nous décrivons ces données dans la partie suivante.

### 5.3.6 Données d'entrée

Les données d'entrée du modèle sont sous plusieurs formats. La zone d'étude est délimitée par un polygone (figure 5.17).



FIGURE 5.17 – Zone d'étude délimitée par un polygone, en rouge.

Nous pouvons attribuer une densité de l'espèce *Ulex europaeus* en utilisant un couche vectorielle de couverture de l'espèce sur la zone en 2004 (Triolo and Hoff, 2006) présentée par la figure 5.18 (a). Les densités de l'espèce simulées sont ensuite comparées avec la couche de couverture de 2012 (Payet, 2012) présentée par la figure 5.18 (b). Nous utilisons une couche vectorielle délimitant l'étendue de l'incendie de 2010 (figure 5.18 (c)). Enfin nous utilisons des couches raster comme le modèle numérique de terrain pour influencer la diffusion de graines selon la pente, figure 5.18 (d). Pour caractériser un milieu couvert ou semi couvert en végétation, nous utilisons une série temporelle d'images satellite contenant la valeur en NDVI pour chaque année de 2004 à 2012 (figure 5.18 (e)).

Le tableau 5.2 présente un récapitulatif des données d'entrée.

Nom	Format	Résolution	Source	Pas de temps
Couverture UE 2004	Vecteur	NA	(Triolo and Hoff, 2006)	NA
Couverture UE 2012	Vecteur	NA	(Payet, 2012)	NA
Couverture Feu 2010	Vecteur	NA	IGN(2010)	NA
Zone d'étude	Vecteur	NA	(Commins, 2015)	NA
Couche MNT	Raster	10m	IGN	NA
Couche NDVI	Raster	10m	Spot 5	Annuel

TABLE 5.2 – Récapitulatif des données d'entrée du modèle de diffusion d'espèces invasives.

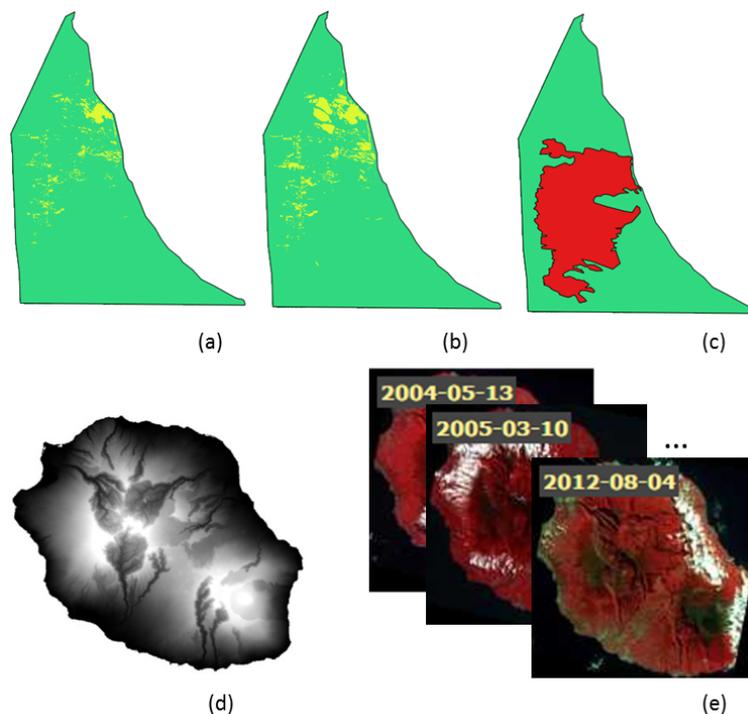


FIGURE 5.18 – Les différentes couches de données utilisées dans le modèle : la couverture de l'*Ulex europaeus* en 2004 (a) et en 2012 (b). La zone incendiée en rouge (c). Les données raster telles que le modèle numérique de terrain (d) et la série temporelle d'images satellite à partir desquelles sont attribuées les valeurs en NDVI (e). La zone d'étude est représentée par le polygone vert sur les figures (a), (b) et (c).

### 5.3.7 Résultats et discussion

Le modèle simule la dynamique de diffusion de l'année 2004 à 2012 avec un pas de temps annuel. A chaque pas de temps, le processus de diffusion et le processus de croissance sont lancés. Nous proposons deux types de résultats : le premier, visuel, montre l'évolution en surface de l'*Ulex europaeus* au format Google Earth KML (figure 5.19). Cette sortie a été présentée lors de la restitution du projet ANR Descartes dans une démarche participative avec les acteurs et les décideurs de la région du territoire de la côte Ouest de l'île de la Réunion (TCO).

Cette sortie présente la diffusion et la croissance de l'*Ulex europaeus* simulées avec comme état initial la surface en *Ulex europaeus* de 2004 présentée sur la figure 5.19 (a). La sortie présente visuellement la densité de l'espèce par niveaux de jaune ainsi que la diffusion au cours du temps comme le montre la figure 5.19 (b) et la figure

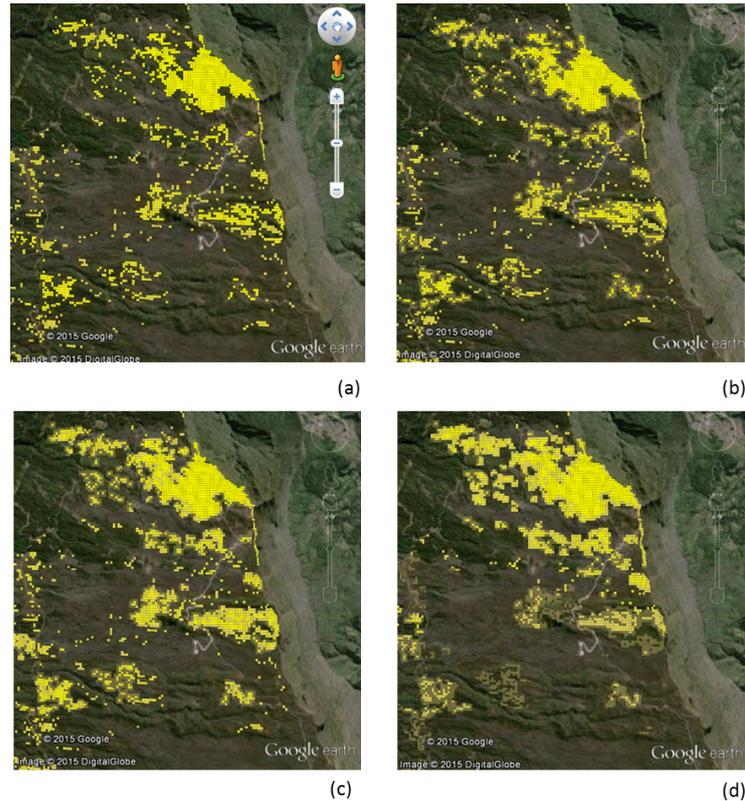


FIGURE 5.19 – Sorties de la simulation au format KML à différents pas de temps : 2004 (a), 2006 (b), 2008 (c) et 2012 (d).

5.19 (c). La baisse de densité observée dans le sud de la zone d'étude dans la figure 5.19 (d) est expliquée par le processus traduisant l'incendie de 2010. Nous pouvons cependant observer que les densités au nord n'ont pas subi de baisse caractéristique.

Le modèle développé dans ce cas d'étude sert essentiellement à mieux comprendre le système étudié. Il permet de vérifier si les processus inclus dans le modèle sont suffisants pour expliquer la diffusion de l'espèce constatée sur le terrain et sur les images satellite. Cela permet ensuite de revoir les processus, les affiner ou les simplifier, ou encore d'en inclure de nouveaux. C'est un travail qui est à réaliser par un expert du domaine avec l'aide, si besoin, d'un modélisateur. Ce travail peut s'appuyer, entre autres, sur des études de sensibilité sur certains paramètres clés du modèle. Pour illustrer cette possibilité, nous faisons varier le paramètre du nombre de voisins pris en compte pour l'interaction de diffusion. Dans cet exemple, nous proposons une sortie au format raster par distance de voisinage qui varie de 1 à 6 (Voisinage de Moore) comme présenté par la figure 5.20. Nous avons proposé ces sorties pour montrer d'une part qu'il était facile de faire varier le paramètre de distance de voisinage, modifié dans ce cas dans les paramètres de contrôles de la simulation. D'autre part nous montrons que nous pouvons sortir des images dont la valeur des

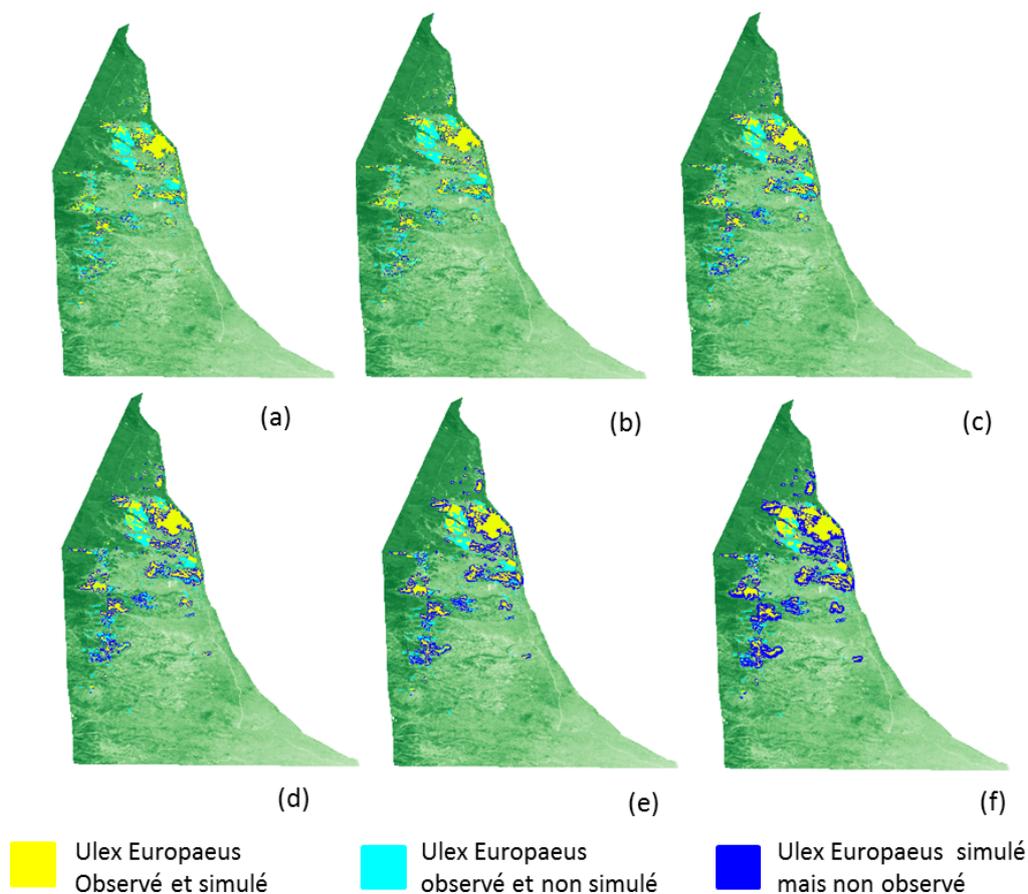


FIGURE 5.20 – Sorties de plusieurs simulations de diffusion de l’*Ulex europaeus* pour une distance de voisinage : de 1 (a), 2 (b), 3 (c), 4 (d), 5 (e) et 6 (f).

pixels est issue d’une ou plusieurs propriétés d’une entité du modèle. En terme de résultats du modèle, ceux-ci sont discutables sur le fait que la distance de voisinage n’est pas un facteur représentatif de la densité observé, plus la distance augmente et plus on a de densités observées et simulées sur une même cellule mais nous avons aussi bien plus de densités simulées alors que celles-ci ne sont pas attendues.

Nous avons proposé un modèle de diffusion d’une plante invasive qui correspond conceptuellement à la gestion d’un graphe de voisinage d’entités sur un même plan. Les processus de diffusion restent simples mais nous montrons que de nombreux types de données, raster, vecteur sont utilisables avec le formalisme d’Ocelet. Des processus plus complexes relevés dans l’étude de Commins (2015), tels que les facteurs de croissance plus élevés après un incendie, ou encore le rôle des routes et des chemins dans la dispersion de graines, peuvent être inclus dans le modèle pour vérifier si ces processus permettent de mieux expliquer les diffusions observées.

## 5.4 Système d'alerte précoce en sécurité alimentaire

En Afrique subsaharienne, plus de 70% de la population active travaille dans le secteur de l'agriculture selon FAO FIDA et PAM (2014). Les cultures pratiquées sont majoritairement céréalières, en pluvial stricte, et l'alimentation repose en grande partie sur leur production (Haile, 2005). Dans cette zone les climats sont variés, allant d'un climat humide vers des climats arides ou semi arides. Les rendements dépendent principalement de la variabilité des pluies dans une année. Toujours selon FAO FIDA et PAM (2014), avec 220 millions de personnes sous alimentées en 2015 sur l'ensemble de l'Afrique, le suivi des variations climatiques et leurs impacts sur la production agricole à une place prépondérante. Comment évaluer au mieux les dates de semis selon le climat au cours de l'année, quelles sont les variétés de céréales les mieux adaptées ? Comment prévoir la vulnérabilité des cultures face à un épisode de sécheresse ? Tels sont les enjeux des systèmes d'alerte précoces (SAP).

Le centre régional AGRHYMET (Niamey, Niger) (Traoré et al., 2014) est une institution spécialisée dans le suivi de l'impact du climat dans la zone du Sahel. Leurs travaux portent principalement sur le suivi de données climatiques ainsi que la surveillance des indicateurs concernant l'hydrologie, météorologie et suivi des cultures. Leurs activités se sont maintenant étendues au-delà de la zone de subsaharienne et touchent l'ensemble de l'Afrique de l'Ouest.

En partenariat avec cette institution nous nous basons sur un modèle de suivi des cultures développé au CIRAD, SARRA-H qui permet de simuler l'impact du climat sur les cultures céréalières. La croissance des cultures est simulée en intégrant trois grands processus : le bilan hydrique, l'évolution de la phénologie et le bilan carboné.

### 5.4.1 Objectifs du modèle

Le modèle SARRA-H simule l'évolution d'une culture sur une saison pour une parcelle ou en un point. L'enjeu est d'intégrer les processus de SARRA-H dans un modèle construit avec Ocelet afin de pouvoir spatialiser la simulation des rendements de cultures, et qui puisse utiliser des données satellites ou des séries temporelles d'images satellite. Les résultats des simulations sont sous la forme de cartes sur l'ensemble de l'Afrique de l'Ouest permettant de suivre l'état des cultures simulé selon divers types de cultures et de stratégies paysannes sous contraintes climatiques durant la saison des cultures. Dans un deuxième temps, un à deux mois avant les récoltes une simulation prédictive des rendements est effectuée en utilisant les séries historiques du climat pour compléter l'information climatique jusqu'à la date de récolte simulée. Cela permet d'obtenir la probabilité des rendements potentiels sous forme d'analyse de risques : par exemple proposer des cartes de rendement prévisionnel correspondant à une probabilité de 80% d'obtenir au moins ce rendement.

Des premiers travaux ont été réalisés dans le cadre de la thèse de C. Jahel (thèse en cours au Cirad, UMR TETIS) visant à reproduire les processus de SARRA-H en Ocelet prenant en compte la phénologie et les calculs de bilans carbonés et hydriques. Nous avons adapté ici ces processus ainsi que les implémentations menées dans cette thèse pour spatialiser les simulations du modèle SARRA-H. Il est à remarquer que pour ce modèle, en raison de son cas particulier d'utilisation/diffusion, les partenaires Africains d'AGRHYMET sont très fortement impliqués. Ils ont participé à définir les objectifs du modèle, et utilisent actuellement en test sur place les différentes versions du modèle que nous leur envoyons au cours de son processus d'élaboration. Il est aussi prévu, après cette période de test, que dès l'année prochaine il soit utilisable et diffusable vers un nombre important d'utilisateurs en Afrique de l'Ouest.

### 5.4.2 Entités, variables d'état

Nous distinguons trois types de catégories d'entités dans ce modèle, la première étant l'entité cellulaire "Crop" sur laquelle sont calculés les rendements. La deuxième catégorie étant l'ensemble des entités cellulaires permettant de mettre à jour les données climatiques ainsi que celles qui vont apporter l'information spatiale sur le type de sol, la température moyenne interannuelle ou encore les dates de débuts et de fin de saison. Enfin la dernière catégorie correspond aux entités vectorielles décrivant les pays avec des géométries.

**Crop** : l'ensemble de l'Afrique de l'Ouest est découpé régulièrement en entité "Crop" sous la forme de cellules. La taille des cellules est déterminée par la taille des pixels des couches de données en entrée. Dans le cas présent, il s'agit de données d'estimation de pluie par satellite avec un pixel de l'ordre de 3,5 km. Les processus sont lancés sur l'ensemble de ces cellules au pas de temps journalier. Un total de 383 091 entités de ce type est géré lors de la simulation. Les variables de cette entité sont en très grand nombre, nous proposons de présenter les principales :

Propriétés statiques :

- Type de sol
- Radiation globale
- Evapotranspiration
- Température minimum
- Température maximum
- Précipitation
- Date de fin de saison
- Date de début de saison

Propriétés d'état

- Stock d'eau du réservoir sol
- Phase phénologique
- Stress de la plante
- Rendement
- Biomasse
- Photopériode

**Country** : l'ensemble de l'espace étant régulier pour manipuler les entités "Crop", les processus sont lancés sur chaque cellule d'un pays en particulier évitant ainsi de lancer des calculs sur une zone négligeable (une partie de l'océan par exemple). Les pays ont une géométrie et nous lançons les processus à partir des pays suivants : Mauritanie, Sénégal, Guinée-Bissau, Guinée, Mali, Burkina-Faso, Gambie, Sierra Leone, Libéria, Côte d'Ivoire, Ghana, Togo, Bénin, Tchad, Niger et le Nigéria.

**RainCell, EVPTCell, TempMinCell, TempMaxCell, RadCell** : ces entités ont une seule propriété qui sert à mettre à jour les propriétés statiques des entités "Crop". Celles-ci seront mises à jour à chaque pas de temps en utilisant des séries temporelles d'images satellite. Pour chaque type de ces entités, un total de 15 792 entités est géré lors de la simulation.

**BeginDateCell, EndDateCell, GroundCell** : entités cellulaires avec une seule propriété servant à indiquer la date de début de saison et de fin des récoltes, ainsi que le type de sol. Pour chaque type de ces entités, un total de 383 091 entités est géré lors de la simulation.

### 5.4.3 Processus et scénarios

Dans le but de spatialiser les simulations du modèle de culture, nous avons choisi de lancer ces processus sur l'ensemble de l'Afrique de l'Ouest représenté par un espace régulier. Les bilans sont calculés pour différentes espèces céréalières de mil, sorgho et maïs ainsi que leurs variétés qui vont avoir des caractéristiques différentes en termes de longueur de cycle, potentiel de production ou encore de photopériodisme.

Pour chaque surface où les bilans sont lancés, nous attribuons un itinéraire technique qui correspond à la stratégie de semis par le cultivateur. Ces stratégies sont déterminées selon la moyenne pluviométrique interannuelle ainsi que le type de sol et indiquent quelles successions de cultures d'espèces sont attribuées à l'entité. Nous proposons de décrire le fonctionnement des trois bilans :

**Le bilan hydrique** estime l'indice de stress hydrique et tient compte des proprié-

tés du sol qui est constitué de trois réservoirs : de surface, profond et d'un réservoir racinaire dynamique. Le réservoir de surface permet de gérer le processus d'évaporation du sol. Le réservoir profond va simuler le front d'humectation permettant de borner la profondeur maximum des racines, de stocker l'eau non accessible par les racines et de bloquer l'enracinement dans le cas des sols indurés. Enfin le réservoir racinaire simule le stock d'eau accessible par les racines et sa profondeur va évoluer selon la vitesse d'enracinement et le front d'humectation. La consommation en eau du sol se calcule selon différentes fonctions. Dans un premier temps l'évaporation est retirée seulement sur le réservoir de surface. Ensuite la transpiration est extraite du réservoir racinaire.

**Le bilan carboné** repose sur le concept de la grande feuille : l'ensemble des feuilles de la culture dans le champ est symbolisé par une grande feuille dont le taux de recouvrement du sol et son épaisseur varient en fonction de la biomasse foliaire et l'évolution de la phénologie de la plante. La transformation de l'énergie correspond à la part photo synthétiquement active de l'énergie lumineuse interceptée par cette grande feuille. Cette énergie est alors transformée en assimilats selon l'activité photosynthétique, le stress hydrique intervenant alors en tant que frein à cette conversion. La répartition de ces assimilats entre les biomasses (racines, feuilles, tiges et grains) va alors dépendre des règles d'allocations variant en fonction des différentes phases de croissance. Une partie de ces assimilats servant à la maintenance des tissus vivants, nommée respiration de maintenance, est retranchée lors du calcul de la biomasse totale.

Les quatre **stades phénologiques** des espèces céréalières pris en compte sont les suivants :

- Phase végétative juvénile
- Phase sensible à la photopériode
- Phase reproductive
- Phase de maturation

Une fois la phase de maturation atteinte, le rendement n'évolue plus et on estime que la récolte peut avoir lieu.

#### 5.4.4 Principes de conception

Le modèle se construit sur une couche principale d'entités cellulaires sur lesquelles sont lancés les processus calculant l'évolution de la phénologie, et les bilans carbonés et hydriques. Ces entités que nous nommons "Crop" portent toutes les propriétés nécessaires aux calculs des bilans. Le calcul des bilans requiert cinq variables climatiques à savoir la pluie, l'évapotranspiration, la température minimum, la température maximale et le rayonnement global. Ces données viennent de séries temporelles d'images satellite. Pour chaque type de données, nous construisons une couche d'entités cellulaires leurs correspondant. Les propriétés de ces entités sont

misés à jour lorsqu'une image d'une série temporelle est disponible à un pas de temps donné. Par exemple nous avons des données pluviométriques journalières tandis que les autres données climatiques sont décadaires.

Les principales relations des modèles se divisent en trois catégories : la relation principale portant la dynamique de la simulation, les relations permettant de mettre à jour les données climatiques avec les séries temporelles d'images satellite et enfin les relations servant à attribuer aux entités "Crop" les propriétés statiques :

**Area** : la relation entre les pays et leurs entités cellulaires "Crop", c'est la relation principale permettant de lancer les bilans. La relation contient trois interactions, chacune correspondant à un processus en particulier (phénologie, bilan hydrique, bilan carboné).

**RainCrop, EtpCrop, RadCrop, TempMinCrop, TempMaxCrop** : ces relations servent à mettre à jour les données temporelles en précipitation, évapotranspiration, rayonnement, température minimum et température maximum des entités "Crop". Par exemple la relation permettant d'attribuer la valeur de pluie entre deux couches différentes est présentée en figure 5.21.

```

relation RainCrop<Crop crop, RainCell rCell> {
    interaction precipitation{
        crop.rain = rCell.rain
    }
}

```

FIGURE 5.21 – Exemple de définition de relation en Ocelet permettant de mettre à jour les données pluviométriques.

**BeginDateCrop, EndDateCrop, GroundCrop** : relations permettant d'attribuer les valeurs venant des entités "BeginDateCell", "EndDateCell" et "GroundCell" aux entités "Crop".

Une relation traduisant les mises à jour des données est définie pour chaque donnée météorologique à mettre à jour lors de la simulation. Chaque bilan correspond à une interaction entre les pays qui vont les appliquer à chaque entité "Crop" représentant leur espace régulier (figure 5.22).

Le but du modèle est de simuler les rendements pour chaque variété. Le calcul des bilans se fait au pas de temps journalier, concernant les données décadaires nous avons réparti simplement en divisant par dix les valeurs des données climatiques par rapport à leur cumul sur dix jours. Un exemple simplifié du scénario en Ocelet du modèle est présenté par la figure 5.23.

```

relation Area<Crop crop, Country country> {

    interaction phenoBalance() { ... }

    interaction carbonBalance() { ... }

    interaction waterBalance() { ... }

}

```

FIGURE 5.22 – Exemple de définition de la relation en Ocelet permettant de lancer les trois bilans traduits en interactions entre les entités représentées par un pays et leurs entités cellulaires "Crop" correspondantes.

```

scenario WestAfricaCropModel{

    //Phases d'initialisation, instanciation des datafacers, relations etc
    // Dynamique du modèle, pas de temps journalier :
    for(days : 1..365){
        // lancement des bilans par appel des interactions :

        area.phenoBalance
        area.carbonBalance
        area.waterBalance

        // mise à jour des données météorologiques
        rainSerieFile.update(rainCell.grid)
        ...
    }
}

```

FIGURE 5.23 – Figure synthétique de la description de la dynamique du modèle avec l'appel des interactions calculant les bilans lancés dans la boucle journalière.

### 5.4.5 Initialisation

La phase d'initialisation du modèle comprend l'instanciation de toutes les relations, ainsi que leurs connexions, par exemple entre les entités "Crop" et les entités Pays, ainsi que les connexions entre les entités "Crop" vers toutes les entités cellulaires servant à apporter les propriétés statiques (données climatiques). Dans un deuxième temps les fichiers de paramétrage CSV (données paramètres) sont lus et leurs paramètres stockés en mémoire pour y avoir accès lors des simulations.

Avant le début de la simulation un paramètre permet de lancer la simulation soit sur l'année en cours, soit en mode prévisionnel. Dans le cas de l'année en cours, la date d'arrêt de la simulation est spécifiée permettant d'effectuer des suivis en cours de saison. A cette date, différents indicateurs d'état du sol et de la plante des entités "Crop" sont exportés au format Raster (cartes thématiques), une bande du

raster correspondant à une propriété de l'entité. Cette exportation représente l'état de l'ensemble des entités "Crop" à la date donnée.

Dans le cas de la simulation prévisionnelle l'état initial des entités "Crop" est déterminé en chargeant les fichiers raster exportés lors de la simulation de l'année en cours. Ces fichiers sont rechargés pour chaque année devant être simulée.

#### 5.4.6 Données d'entrée

**Zone d'étude :** la zone d'étude est délimitée par les géométries du shapefile (figure 5.24) de chaque pays représentant l'ensemble de l'Afrique de l'Ouest. Ocelet va générer automatiquement les entités "Crop" qui ont une représentation cellulaire à partir de ces géométries.

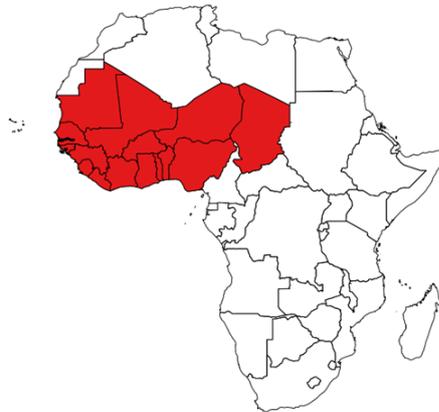


FIGURE 5.24 – La zone d'étude concerne les pays de l'Afrique de l'Ouest : Mauritanie, Sénégal, Guinée-Bissau, Guinée, Mali, Burkina-Faso, Sierra Leone, Gambie, Libéria, Côte d'Ivoire, Ghana, Togo, Bénin, Tchad, Niger et le Nigéria.

**Données raster statiques :** l'accès aux données raster statiques se fait avec des datafacers *RasterFile* permettant de générer les entités cellulaires décrites précédemment.

**Données raster temporelles :** l'accès aux données des séries temporelles d'image se fait avec un datafacer *TemporalSeriesFile* pour chaque série d'images. Les données d'évapotranspiration, rayonnement global, température minimum et température maximum sont décadaires et sont ramenées au pas de temps journalier. Celles-ci ont une résolution de 16 km. Les données des précipitations sont journalières avec une résolution de 3,5 km. Les cellules de cultures sont basées sur cette résolution.

**Données paramètres :** en plus des données spatiales, nous utilisons des données venant de fichiers CSV permettant d'obtenir la valeur des paramètres liés aux types

de sols, itinéraires techniques et des variétés.

Le tableau 5.1 présente un récapitulatif des données d'entrée.

Nom	Format	Résolution	Source	Pas de temps
Pays	Vecteur	NA	NA	NA
Typologie Sol	Raster	3,5km	FAO	NA
Températures moyennes	Raster	16km	ECMWF	NA
Début Saison	Raster	3,5km	AGRHYMET	NA
Fin Saison	Raster	3,5km	AGRHYMET	NA
Pluviométrie	Raster	3,5km	TAMSAT	Journalier
Température Minimum	Raster	16km	ECMWF	Décadaire
Température Maximum	Raster	16km	ECMWF	Décadaire
Evapotranspiration	Raster	16km	ECMWF	Décadaire
Radiation Globale	Raster	16km	ECMWF	Décadaire

TABLE 5.3 – Récapitulatif des données d'entrée du SAP.

### 5.4.7 Résultats et discussion

L'ensemble de ces résultats et de ces travaux a pour objectif de préparer la future génération du système d'alerte précoce d'AGRHYMET en proposant d'utiliser de façon intégrée les données d'observation de la Terre (MODIS, SMOS et Sentinel-2, Météosat) et le modèle de culture SARRA-H, intégré dans la plateforme de modélisation Ocelet. Cette intégration doit permettre de convertir l'information spatiale et temporelle en temps quasi-réel du suivi global de la biomasse fourni par les données satellitaires, en estimation de rendement pour les grandes cultures céréalières, par la prise en compte des itinéraires techniques et des indices de récoltes simulés par le modèle dans les conditions climatiques données.

#### Modèle de simulation sur l'année en cours

Les sorties du modèle se font au format raster, nous présentons ici des sorties de simulations sous la forme de cartes présentant le rendement de sorgho PP F2 (figure 5.25) de mil PP F2 (figure 5.26 (a)), de sorgho 90 F4 (figure 5.26 (c)) et Mais 90 F4 ( 5.26 (b)) en utilisant les données climatiques de 2014. Les cartes générées sont au format GeoTiff d'une résolution de 3,5km finalisées sous QGIS. Les termes PP ou 90 spécifient un caractère variétal de l'espèce : PP spécifie que la variété est photopériodique et aura la capacité d'adapter sa longueur de cycle en fonction de la date de semis de l'année en cours et de la date moyenne de fin de saison des pluies, alors que le terme 90 indique d'une part que la variété a une longueur de cycle

constante et d'autre part que sa durée en nombre de jours (on a ainsi des variétés de cycle court ou de cycle long). La seconde partie indique le niveau de fertilité du sol et des apports en engrais variant en fonction des situations. Quatre indices de fertilités ont été retenus : F1 très faible à F4 bon niveau de fertilité. Dans chaque cas l'instanciation des paramètres des cultures varie (fichier CSV).

Dans le cadre de la collaboration avec le centre régional AGRHYMET la plateforme

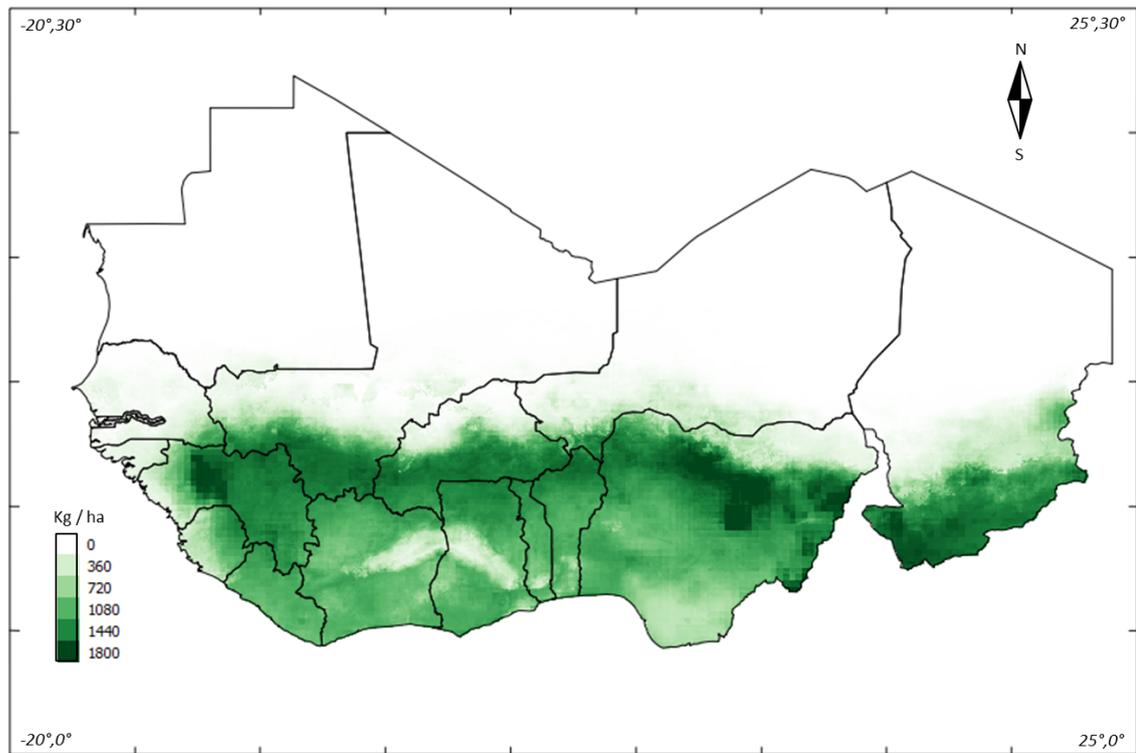


FIGURE 5.25 – Carte des rendements du Sorgho PP F2 pour l'année 2014.

Ocelet a été mise à disposition du groupe de recherche pour tester le suivi opérationnel de la campagne agricole, sur la base des données satellitaires (TAMSAT). La faisabilité est testée pour plusieurs paramètres de sorties du modèle, au pas de temps décadaire : constante de stress hydrique (CSTR)(figure 5.27 (b) et (d)), stock d'eau dans le sol, l'évapotranspiration maximale (ETM), l'évapotranspiration réelle (ETR) et le niveau de satisfaction des besoins en eau des cultures. Nous pouvons ainsi présenter des indicateurs spatialisés d'états de la plante et d'eau disponible tout au long de la saison des cultures.

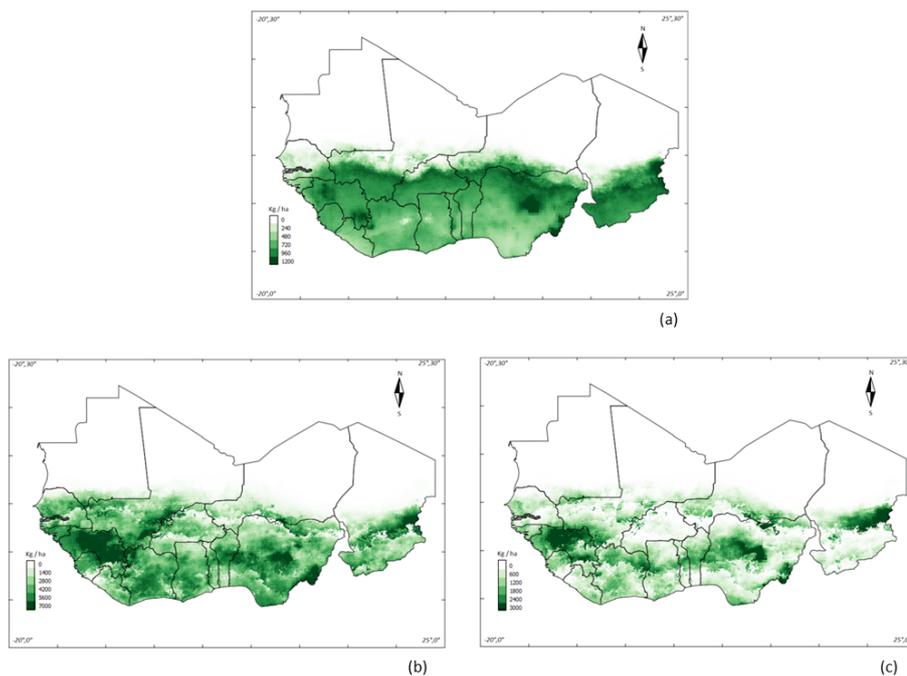


FIGURE 5.26 – Cartes des rendements pour le mil PP F2 (a), maïs 90 F4 (b) et sorgho 90 F4 (c) pour l'année 2014.

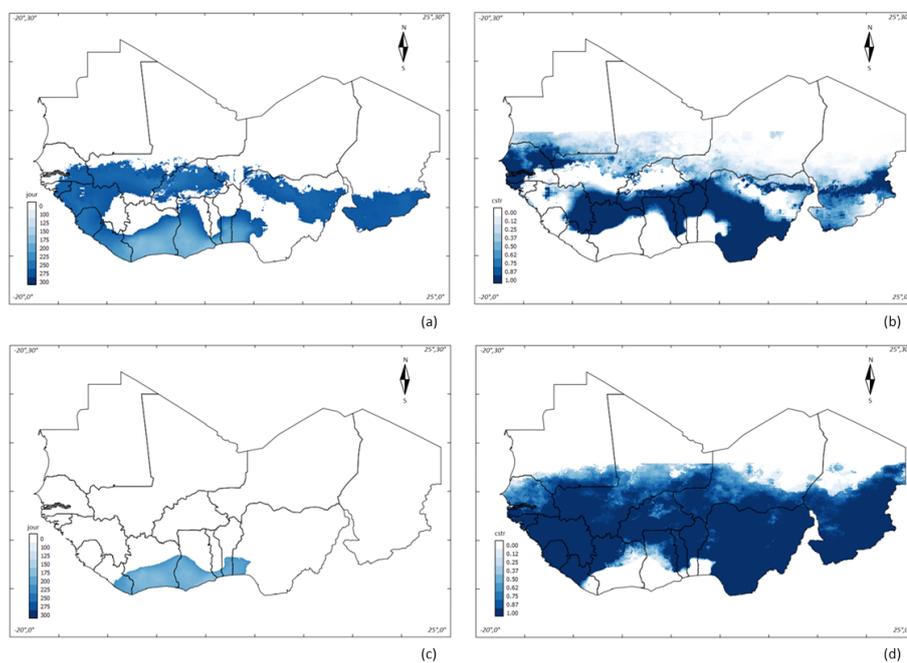


FIGURE 5.27 – Carte des récoltes au 10/08/2014 (a) et au 30/09/2014 (c). Carte du stress hydrique au 10/08/2014 (b) et au 30/09/2014 (d)

### Simulations prévisionnelles

Les résultats précédents montrent une série d'exemples d'indicateurs en cours de saison où à la récolte. Le deuxième intérêt du modèle est de pouvoir faire une estimation probabiliste et prévisionnelle des rendements sur l'année en court en finalisant la série climatique de l'année par les séries historiques en chaque lieu.

Pour réaliser cette modélisation nous avons gardé la même construction de modèle que précédemment mais nous sauvegardons toutes les propriétés des entités "Crop" sous format raster (une propriété par bande soit environ 90 bandes) à un temps donné et pour chaque scénario de simulation (variété, intensification. . .). Pour cet exercice nous nous arrêtons le 11 Août 2014. Nous allons ainsi compléter les informations climatiques à partir du 11 Août des années précédentes, finaliser la simulation jusqu'à la récolte et obtenir ainsi un rendement pour chacune des années de la série historique en commençant par l'année 1989, et en répétant cette action pour chaque année jusqu'à l'année 2014.

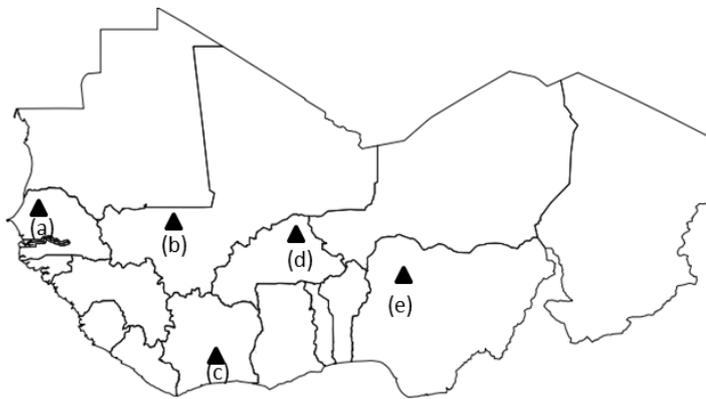


FIGURE 5.28 – Les points étudiés où les simulations ont été effectuées pour comparer les résultats de rendements avec les données climatiques des 24 dernières années : Nord du Sénégal (a), Mali (b), Sud Bénin (c), Nord Burkina (d) et Nord-Ouest Nigeria (e).

Nous avons pour cet exemple pris cinq points localisés dans différentes régions (figure 5.28) pour montrer les différences de rendements simulés en tenant compte de la variation des données climatiques. Nous présenterons ici les courbes de rendement pour chacun de ces points présentées par la figure 5.29 sur les vingt-quatre dernières années.

Ce modèle montre que l'on peut travailler à des échelles très larges comme un continent tout en décrivant des processus à l'échelle locale. Nous montrons aussi que l'approche permet de manipuler des séries temporelles d'images de différentes résolutions spatiales et de les intégrer dans un modèle pour mettre à jour les propriétés des entités.



FIGURE 5.29 – Simulation des rendements pour l'année 2014 au 11 août en simulant à partir des données climatiques de 1989 pour le Nord du Sénégal (a), Mali (b), Sud Bénin (c), Nord Burkina (d) et Nord-Ouest Nigeria (e).

## Autres résultats

Enfin nous présentons un résultat venant de l'utilisation de ce modèle par le centre régional de l'AGRHYMET qui l'a publié dans son bulletin d'information mensuel (02/06/2015) (Bouafou et al., 2015) présentant le niveau de satisfaction des besoins en eau des céréales au 30 Juin 2015 (figure 5.30).

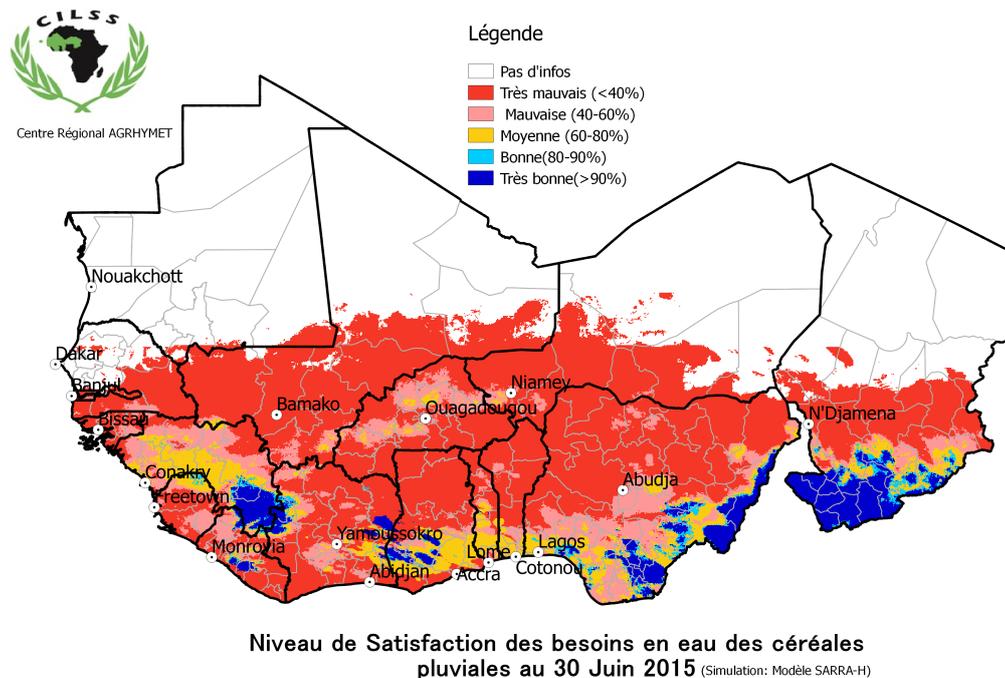


FIGURE 5.30 – Niveau de satisfaction des besoins en eau des céréales au 30 Juin 2015, bulletin du 02/06/2015 publié par AGRHYMET.

*Sur le plan Agricole, la saison des pluies 2015 a connu une installation tardive (avec un début tardif) dans presque toute la bande agricole de la zone soudanosa-hélienne de l'Afrique de l'Ouest. Ceci a engendré, çà et là dans la zone, un retard assez important (2 à 3 décades dans la bande soudanienne et 1 à 2 décades dans la zone sahélienne) dans les dates semis des céréales pluviales et dans la mise en place de la végétation naturelle. En effet, jusqu'au 30 juin, l'écrasante majorité des villages agricoles, notamment du Centre-Est Niger (Zinder et de Diffa), Sud-Est Burkina Faso, Nord-Ouest Mali et Sénégal, n'avait pas de l'ordre de 50 à 100 mm, pendant la troisième décade de juin. Aussi, le bilan hydrique, simulé à l'aide du modèle SARRA-H de simulation de la croissance des cultures et de prévision des rendements et de l'outil "Water requirement satisfaction index, (WRSI)" utilisé par Fews Net, indique une mauvaise satisfaction des besoins en eau des céréales pluviales (mil, sorgho) dans la majeure partie des aires de culture de l'Afrique de l'Ouest et du*

*Tchad, exceptées dans quelques localités (autour de Niamey au Niger, au centre du Burkina Faso, au Sud Mali, en Guinée et dans les parties Sud du Tchad, du Nigeria, du Benin, du Togo et du Ghana. L'analyse de cette situation permet de dire que le retard observée dans l'installation de la saison des pluies en Afrique de l'Ouest et au Tchad ne semble pas être assez inquiétante, en particulier dans la zone soudanienne où la longueur de la saison des pluies est assez longue pour permettre aux cultures pluviales de boucler leurs cycles avant l'arrêt des pluies. Par contre, dans la zone sahélienne, il y a lieu de s'interroger sur l'impact d'un retard de deux décades dans l'installation de la saison sur les rendements agricoles, surtout lorsqu'on sait que la mise à jour des prévisions saisonnières 2015 présage un arrêt précoce de la saison des pluies au Sahel. Toutefois, une bonne adoption (par les paysans) des conseils agrométéorologiques issus de ces prévisions saisonnières (choix de cultures/variétés et de dates de semis), combinée à une consultation régulière des agents de vulgarisation agricole, pourrait atténuer les impacts redoutés.*

## 5.5 Conclusion sur les cas d'application

Les cas d'application proposés ont permis de montrer et de tester la capacité d'Ocelet pour manipuler différentes représentations spatiales dans un même modèle. Ce travail a été l'occasion d'utiliser des sources de données raster différentes, sur leurs localisations, à des échelles différentes et à des fins différentes. Nous n'avons pas vu de limites particulières à la description de modèles utilisant les deux formes de représentations qui est maintenant possible en respectant la syntaxe du langage Ocelet. Ces modèles montrent ainsi la genericité de l'approche et la diversité des domaines dans lesquels la conception de modèles est possible :

- Les modèles de ruissellement de l'eau et de la diffusion d'espèces invasives mettent en avant la possibilité de décrire des processus affectant l'état d'entités n'appartenant pas au même espace de représentation : d'une part des processus sur un espace découpé régulièrement et d'autre part sur les objets vectoriels. Si le modèle de ruissellement montre des résultats simulés proches des données observées, le modèle de diffusion d'espèces invasives reste exploratoire et nous espérons qu'il donnera suite à une étude plus approfondie.
- Le modèle sur les rendements de cultures céréalières sur l'Afrique de l'Ouest met en avant la facilité de la gestion des données satellitaires et de leur intégration dans un modèle de culture. Ce modèle est aussi l'occasion de montrer que l'on peut travailler à des échelles très larges couvrant plusieurs pays tout en décrivant des processus à l'échelle locale : sur chaque élément d'un espace régulier.

Ces trois modèles ont été choisis d'une part parce qu'ils présentaient des situations de modélisation difficiles à traiter par d'autres approches. Il nous a alors semblé pertinent de les conceptualiser, simuler et tester avec notre approche. Ces modèles

---

ont aussi été choisis en fonction des divers partenariats que nous avons dans notre établissement : l'équipe du projet Descartes pour le modèle de ruissellement, le Parc National de l'île de la Réunion pour le modèle de diffusion et enfin l'AGRHYMET pour la simulation des rendements de cultures en Afrique de l'Ouest.

Enfin ces modèles ont tous été conceptualisés en travaillant sur des graphes et en décrivant les processus par des interactions entre les éléments. Nous mettons ainsi en avant que le formalisme des graphes d'interaction est suffisamment générique pour la description des interactions spatiales, quelle que soit la forme de représentation utilisée.



# Chapitre 6

## Conclusion générale

Le formalisme d'Ocelet proposé pour modéliser des dynamiques spatiales permet de concevoir le modèle d'un système à partir d'une structure qui porte la description des interactions entre les éléments de ce système. C'est le graphe d'interaction : les éléments du système (les entités) sont les sommets du graphe, et sont reliés par des arêtes lorsqu'ils peuvent interagir ou s'influencer. Des fonctions d'interaction sont alors définies sur les arêtes et sont appelées lors d'une simulation pour modifier l'état des entités connectées. Ces graphes sont dynamiques (les états des entités peuvent changer), à structure dynamique (les arêtes et les sommets peuvent aussi être ajoutés ou retirés du graphe). Ce formalisme est suffisamment générique pour définir différents types d'interactions : des interactions sociales, hiérarchiques, spatiales... Le modélisateur ne manipule ainsi que des graphes d'interaction pour traduire la dynamique d'un système.

La représentation spatiale des entités a cependant une influence sur leur manière d'interagir : les entités peuvent être connectées par voisinage, ou par d'autres relations qui dépendent de leur configuration respective dans l'espace. La structure du graphe en est alors influencée. Dans la première version d'Ocelet, la représentation spatiale des entités était vectorielle ; les entités étant pour cela assignées d'une propriété de type géométrie. Les entités cellulaires pouvaient déjà être utilisées, dans la mesure où chaque cellule pouvait être définie comme une entité avec une propriété géométrie de forme carrée, hexagonale ou triangulaire. Mais de tels modèles étaient rapidement limités par le nombre de cellules pouvant être gérées en mémoire, et surtout ne tiraient pas profit de l'arrangement de données propre au format raster, particulièrement efficace pour recevoir des données sous la forme de matrices de cellules. Il convenait alors d'étudier dans cette thèse la gestion de l'espace régulier induit par le format raster avec le formalisme d'Ocelet : sa gestion en tant que graphe d'interaction.

Cette thèse a d'abord montré que le concept de graphe d'interaction pouvait s'appliquer aux entités d'un modèle avec différentes formes de représentations spatiales,

et qu'il permettait de faire interagir dans un même modèle des entités avec une représentation vectorielle avec des entités représentées par un pavage régulier. Des solutions de mise en œuvre ont ensuite été proposées et implémentées pour rendre praticable ce concept étendu de graphe d'interaction dans la plateforme de modélisation Ocelet. Ces solutions s'appuient notamment sur la simulation de parcours de graphes, implicites dans le cas de pavages réguliers, qui utilise plus efficacement la mémoire car il n'est pas nécessaire d'y mettre toutes les entités cellulaires du graphe en même temps. Cela permet ainsi de prendre en compte un très grand nombre de cellules dans un modèle. Enfin, le développement de trois modèles de dynamiques spatiales dans des domaines thématiques assez éloignés a permis de vérifier la généralité des concepts et l'utilité des outils développés. Nous revenons sur ces trois parties dans la suite de cette conclusion.

## 6.1 Le formalisme de graphe d'interaction étendu aux pavages réguliers

Lors de la phase de conception, nous avons ajouté aux définitions sur le formalisme d'Ocelet (définition des entités, caractérisées par leurs propriétés, des datafactors qui facilitent la gestion et l'intégration de données, des relations qui permettent de connecter les entités et d'y appliquer les fonctions d'interactions et du scénario qui contient un enchaînement d'opérations pour traduire la dynamique du modèle), des définitions sur le format de données raster ainsi que sur la représentation d'un espace découpé régulièrement pour constituer un pavage régulier. Un raster est donc un format matriciel pouvant contenir de l'information dans chacun de ses éléments. Un élément de la matrice – une cellule, de par sa position dans la matrice, correspond à une portion de l'espace, positionnée dans un repère spatial. La matrice délimite une zone dans l'espace, et la cellule est représentée par une forme géométrique d'une taille donnée. Pour respecter la régularité du format, la représentation spatiale d'un élément est généralement le carré (pixel), mais deux autres formes supplémentaires conviennent aussi : les triangles équilatéraux et les hexagones réguliers. Ils permettent eux aussi de constituer un espace entièrement recouvert, régulièrement : un pavage régulier.

En tenant compte de ce format de données et de sa représentation spatiale, nous avons défini un nouveau type de propriété pouvant caractériser une entité en Ocelet : la cellule. Elle a la forme et la dimension permettant de constituer un pavage régulier. Les autres propriétés de l'entité sont quant à elles, contenues dans le format raster à la position de la cellule. Ainsi un pavage régulier est obtenu en Ocelet par un ensemble d'entités d'un même type qui ont dans leur définition une cellule comme propriété. Si cette seule définition d'entité "cellulaire" est suffisante pour représenter l'espace régulièrement, leur manipulation dans des graphes d'interaction a dû être abordée sous des angles différents :

- Le premier type de graphe proposé est celui qui est implicite entre des entités cellulaires d'un même type. Il permet de définir des fonctions d'interaction et de faire évoluer l'entité par son voisinage. Plusieurs types de voisinages peuvent être définis : pour tenir compte des entités qui ont seulement un côté en commun (voisinage de Von Neumann) ou les entités qui partagent un côté ou un sommet en commun (voisinage de Moore). En se basant sur ces types de voisinage, la notion de distance de voisinage est aussi introduite pour faire interagir les entités comprises dans un voisinage étendu.
- Le deuxième type de graphe d'interaction proposé permet de définir des fonctions d'interaction entre des entités cellulaires et des entités non cellulaires. Si celles-ci sont des entités vectorielles, le graphe est aussi implicite et est construit par correspondance spatiale. Par contre, si les entités n'ont pas de forme de représentation spatiale, il revient au modélisateur de spécifier les connexions avec les entités cellulaires.
- Le dernier type de graphe d'interaction proposé relie des entités cellulaires de différents types (pavages différents par la taille ou la forme des cellules). Il est ainsi possible de lier des entités de pavages réguliers différents qui sont en correspondance spatiale.

Lors de la construction d'un modèle avec Ocelet, les graphes doivent être définis explicitement. Dans le cas d'entités cellulaires, les graphes étant implicites, ils doivent être rendus disponibles déjà construits et directement manipulables dans le modèle. Après avoir déterminé conceptuellement comment gérer les pavages réguliers dans le formalisme de graphes d'interaction, l'étape suivante de la thèse a été d'en réaliser l'implémentation dans le langage et la plateforme Ocelet.

## 6.2 La plateforme de Ocelet enrichie de la capacité à incorporer de la donnée raster

Le moteur d'exécution d'Ocelet utilise un parcours de graphes conçu à l'origine pour appliquer des fonctions d'interaction et affecter le résultat simultanément sur toutes les entités reliées dans un graphe explicite. La simultanéité des opérations entraîne un risque d'affectation multiple d'une propriété lors de l'appel d'une interaction. C'est le cas lorsqu'une entité est connectée à plusieurs autres entités, avec plusieurs arêtes. Pour cela, l'ensemble des valeurs issues d'une fonction d'interaction sont gardées en mémoire et ne sont affectées que lorsque le parcours de graphe est terminé. Celles-ci peuvent être affectées de différentes manières, qu'il revient au modélisateur de spécifier par l'utilisation de fonctions d'agrégations.

Les modes de production de données spatiales et en particulier sous le format raster (comme les données de télédétection) sont en constante évolution. Les données sont de plus en plus nombreuses, de résolution spatiale plus fines, avec une répétitivité temporelle accrue, et finalement de plus en plus volumineuses. Les choix de conception pour gérer les données raster et leurs représentations en Ocelet implique d'avoir une entité pour chaque élément du pavage, et dans le cas d'une image, pour chaque pixel. Nous avons dans ce cas un nombre de sommets du graphe très élevé et un nombre d'arêtes encore plus élevé. Il n'était pas envisageable de garder en mémoire chaque nœud et arête, d'autant plus que plusieurs pavages doivent pouvoir être manipulés à l'intérieur d'un même modèle. C'est face à ce problème que nous avons proposé des méthodes de simulation de parcours de graphes implicites lors de l'application des fonctions d'interaction.

Pour un gain de mémoire, les entités cellulaires ne sont pas instanciées ; leurs propriétés sont contenues dans la matrice de données, au format raster. L'état d'une entité cellulaire est ainsi défini par l'ensemble des valeurs contenues sur un élément de la matrice. Pour pouvoir manipuler ces propriétés, la simulation des parcours de graphes doit alors pouvoir indiquer l'index d'une entité sur cette matrice.

Pour le parcours de graphes sur les entités d'un même pavage, les entités sont "connectées" à leurs entités voisines sur le pavage. Nous avons utilisé la régularité de la disposition spatiale de ces entités, et développé la simulation du parcours des voisins en nous appuyant sur la définition des graphes de Cayley (Cayley, 1878) dont il a été démontré qu'ils pouvaient simuler des automates cellulaires (Róka, 1999). Une arête est ainsi simulée en déterminant les index des entités cellulaires à ses extrémités. Les fonctions d'interaction sont alors applicables sur chaque arête simulée, sans que celles-ci soient gardées en mémoire.

Le parcours de graphe entre entités d'un pavage et des entités "autres" (géométriques ou sans représentation) est aussi simulé. Cependant, s'il est implicite avec des entités géométriques, par correspondance spatiale, il ne l'est plus avec des entités sans représentation spatiale. Le développement d'une structure d'indexation sur la position des entités cellulaires et dont les terminaisons sont les "autres" entités permet de simuler le parcours de graphe entre ces entités. Ce type d'indexation est nécessaire puisque les entités cellulaires ne sont pas gardées en mémoire mais seulement leurs propriétés sous la forme matricielle. L'indexation permet dans ce cas de retrouver une entité cellulaire sur la matrice de données.

Finalement le parcours de graphe restant à déterminer est celui entre des entités appartenant à différents pavages réguliers. Ce type de graphe étant implicite, chaque arête est simulée par correspondance spatiale entre une entité d'un pavage vers une entité d'un autre pavage.

Afin de garantir la simultanéité des fonctions d'interactions, des méthodes particulières ont été développées pour chaque cas de graphe, avec des structures de données permettant d'accueillir les valeurs issues des interactions et y appliquer par la suite les fonctions d'agrégation. Au final, les parcours de graphes sont simulés et les entités cellulaires non instanciées, mais ces implémentations permettent de manipuler les graphes d'interactions comme si ceux-ci étaient construits.

Les autres développements ont eu pour objectifs de faciliter la lecture de données au format raster comme des images satellite, ou séries temporelles d'images satellite en développant les datafacers appropriés. Le modélisateur obtient ainsi directement un graphe d'entités cellulaires par simple lecture d'images.

Ces implémentations vont étendre la capacité d'expression du modélisateur en lui permettant de gérer les données raster, représentées par des pavages réguliers, en tant que graphes d'interactions. Ces graphes sont utilisables immédiatement dans le modèle, leurs parcours étant déjà proposé, la création des entités cellulaires en est facilitée. Pour tester les nouvelles possibilités de notre approche nous avons choisi de développer plusieurs modèles relevant de domaines thématiques très différentes.

### 6.3 Des concepts et outils mis à l'épreuve dans différents cas d'application

L'accent était mis sur l'illustration des nouvelles possibilités offertes pour concevoir des modèles qui utilisent diverses sources de données spatiales raster et vectorielles. Bien que les modèles aient été développés en lien avec des spécialistes des différents domaines, nous ne sommes pas dans le cas où c'est le spécialiste lui-même qui modélise son point de vue sur le système qu'il étudie à l'aide d'un langage métier. Dans chacun des trois cas d'application, nous proposons une nouvelle façon de modéliser qui met en avant la flexibilité des graphes d'interaction qui s'appliquent à différentes représentations de l'espace – en cellules et vectorielle – dans un même modèle. Les processus modélisés traduisent seulement notre connaissance très imparfaite de non spécialiste des champs d'application, mais illustrent bien les nouvelles possibilités de gérer l'espace qui intéressent fortement, du moins pour ces trois cas, les spécialistes des domaines avec lesquels nous avons échangé. Ci-après, nous revisitons les trois modèles en faisant ressortir les potentialités des nouveaux concepts et outils.

#### **Modèle de ruissellement : application au bassin versant de Saint Gilles, île de La Réunion**

Ce modèle a été l'opportunité d'intégrer deux processus de ruissellement sur le bassin versant de Saint-Gilles, l'un sur un espace régulier, avec des phénomènes de transfert

d'eau de proche en proche sur les cellules, et l'autre agissant à une autre échelle, par des flux échangés entre entités routes et ravines représentées spatialement par des linéaires. Chaque cellule et chaque tronçon de linéaire possède des caractéristiques propres (pente et altitude pour les cellules, route principale ou chemin et ravine large ou encaissée pour les linéaires) qui peuvent intervenir dans les calculs de flux entre entités. Les entités sont renseignées aisément via les datafacers par des données spatialisées de sources différentes. Les simulations ont montré que le modèle se comportait correctement à la fois sur l'épisode exceptionnel (cyclone Dina, 2002) que pendant la période plus étendue d'une année, avec des événements d'intensité moindre. La comparaison des débits simulés avec les débits observés a montré des correspondances suffisamment encourageantes pour que ce modèle soit repris par la suite (Projet GIML, Gestion intégrée de la mer et du littoral ; <http://giml.re>).

### **Modèle de diffusion de plantes invasives**

Le cas de la diffusion de l'*Ulex europaeus* dans les hauts de la Réunion illustre bien la démarche d'acquisition de connaissances à l'aide d'un modèle. Dans un cadre exploratoire nous avons construit un modèle permettant de simuler la diffusion de cette espèce invasive à partir de seulement quelques connaissances sur les facteurs pouvant potentiellement influencer sa diffusion : progression de proche en proche pour coloniser l'espace, préférence d'un espace ouvert pour se développer, rôle des feux à la fois pour réduire les taches d'ajoncs mais aussi pour ouvrir l'espace favorisant le développement de celui-ci, rôle de vecteurs (humains, bovins) de dispersion. L'intégration de la gestion d'un espace régulier dans le modèle semblait convenir naturellement à ce type de diffusion, avec un voisinage régulier des entités. Mais l'inclusion du seul processus de diffusion de proche en proche, en prenant en compte l'ouverture du couvert végétal, ne permettait pas d'expliquer la situation de 2012 en partant de l'état initial de 2004. La prise en compte de l'extension de la zone de feu (incendie de 2010) a permis de se rapprocher de la situation observée en 2012. L'information spatiale concernant le feu est une couche SIG contenant des polygones. Son inclusion dans le modèle basé initialement sur des cellules s'est faite naturellement, sans que le format de la donnée ne soulève de problème éventuel de compatibilité pour le modélisateur. Cela lui permet de se focaliser davantage sur les processus à modéliser que sur les formats de données à mobiliser. Des études sur l'expansion de l'espèce sont en cours (Commins, 2015), et des processus supplémentaires, plus liés à la diffusion d'origine anthropique le long de voies de communication, seront inclus dans le modèle.

### **Système d'alerte précoce sur l'Afrique de l'Ouest**

Dans l'article faisant l'état de l'art sur la spatialisation de modèles de culture, Faivre et al. (2004) attire l'attention sur "le manque criant d'outils opérationnels et transférable qui soient adaptés à ce problème" (... crucial lack of operational and

transferable tools adapted to this problem'). Les nouvelles fonctionnalités d'Ocelet développées dans ce travail suggéraient qu'il était envisageable de spatialiser des modèles de culture. Nous avons donc entrepris de reproduire en Ocelet les processus présents dans un modèle de culture de céréales sèches de l'Afrique de l'Ouest, le modèle SARRA-H. Ce modèle permet de simuler l'évolution de la biomasse et les rendements d'une culture sur une saison de croissance en utilisant en entrée des données de stations météorologiques (pluie, température, rayonnement global). Le modèle simule une évolution dans le temps mais seulement sur un point ou une portion de l'espace supposée homogène. L'intégration de la gestion de l'espace régulier et l'utilisation de séries temporelles a permis de spatialiser le modèle sur l'ensemble de l'Afrique de l'ouest en simulant les rendements sur chaque cellule de cet espace, en utilisant en entrée des données issues de satellites météorologiques. Le modèle spatialisé de simulation de rendements de cultures intéresse fortement nos partenaires du Centre AGRHYMET (Niger) qui est responsable d'un système d'alerte précoce en sécurité alimentaire. En se basant sur les données de suivi de culture recueillies lors de la saison passée et la saison en cours, ils évaluent le modèle en conditions quasi-réelles et notamment sa capacité à fournir des résultats de simulation pertinents pour leur système d'alerte précoce.

## 6.4 Perspectives et futurs développements

Depuis sa création, le langage Ocelet a connu de nombreuses modifications et améliorations. L'intégration de la gestion des deux formes de représentation de l'espace en respectant le formalisme des graphes d'interaction s'inscrit dans la continuité de son développement. Les utilisateurs peuvent désormais concevoir des modèles spatialisés avec plus de liberté en termes de représentation spatiale. De nombreuses questions tant sur la conception que sur l'implémentation de nos travaux demandent néanmoins à être traitées.

Conceptuellement, nous n'avons pas abordé la gestion de graphes correspondant à des structures particulières comme montré dans les travaux de Cantwell and Forman (1993), avec des formes de graphes remarquables telles que des représentations en toiles, en étoiles, circulaires etc. Peut-on définir des parcours spécifiques pour ces types de graphes, des fonctions d'interactions particulières? Les pavages semi-réguliers avec des formes différentes mais répétitives sur le pavage peuvent avoir un intérêt pour simuler les dynamiques spatiales et méritent d'être étudiés. Les interactions spatiales peuvent être très complexes. Nous verrons sûrement émerger de nouveaux formats de données et notre approche devra s'adapter en conséquence. La gestion de l'espace a été une priorité jusque là, mais la gestion du temps dans le formalisme reste à être approfondie. Pourrait-on gérer le temps en graphes, des graphes d'interaction temporels? Qu'impliquerait la définition de fonctions d'interaction sur ce type de graphe. Au même titre que la représentation spatiale d'une

entité pouvant être attribuée à une propriété, faut-il caractériser une entité par une propriété temporelle ?

L'implémentation même de la gestion des graphes sur l'espace régulier est améliorable. Par exemple, nous n'avons pas abordé le parallélisme des calculs sur le format raster. La syntaxe proposée nous a semblée cohérente avec le langage lui-même mais fera sûrement l'objet d'évolution avec les futurs développements du langage. Si le format raster était au centre de ces travaux, qu'en est-il des autres ? Peut-on proposer des graphes implicites sur des bases de données spatiales par exemple ? Même si le formalisme d'Ocelet peut supporter les hypergraphes, ceux-ci n'ont pas encore été traités en termes d'implémentation et soulèvent des questions d'optimisation, d'applications, principalement pour garantir l'application simultanée des fonctions d'interaction.

Au sujet des cas d'applications, nous avons vu que les trois modèles développés lors de cette thèse sont ou seront pris en main par des spécialistes des différents domaines, et donneront lieu soit à des applications opérationnelles, soit à des études scientifiques. L'illustration faite par ces modèles des nouvelles possibilités de modélisation pourrait susciter d'autres applications similaires ou la conception d'autres modèles sur des thématiques proches. Enfin, nous espérons que le parti pris de donner plus de liberté au modélisateur sur la manière de traiter l'espace sera générateur d'applications futures dans des domaines proches ou inattendus, et qu'en retour ces cas d'études feront émerger de nouvelles pistes conceptuelles à explorer, à développer, à évaluer et à appliquer.

# Bibliographie

- Agarwal, C., Green, G., Grove, J., Evans, T., and Schweik, C. (2002). A review and assessment of land-use change models : dynamics of space, time, and human choice. *Newtown Square, PA : U.S. Department of Agriculture, Forest Service, Northeastern Research Station. 61 p.*
- Anh Dao, H. (2015). Etude et caractérisation hydrologique du bassin versant de la ravine de st-gilles pour le projet giml (gestion intégrée de la mer et du littoral). Master's thesis, Ense3 Grenoble.
- Atlan, A. and Tarayre, M. (2012). Etude de la variabilité génétique et de la phénologie de l'ajonc d'europe *ulex europaeus* (fabacée) sur l'île de la réunion. (*Rapport d'étude*). *ECOBIO/ONF Réunion.*
- Baker, W. (1989). A review of models of landscape change. *Landscape Ecology*, 2(2) :111–133.
- Barcelo, A. (1996). *Analyse des mécanismes hydrologiques en domaine volcanique insulaire tropical à relief jeune ; Apports à la connaissance du bilan hydrique. Massif du Piton de la Fournaise (île de la Réunion).* PhD thesis, Université Montpellier II.
- Berge, C. (1970). *Graphes et Hypergraphes.* Dunod.
- Bettini, L. (2013). *Implementing Domain-Specific Languages with Xtext and Xtend.* Packt Publishing Ltd.
- Borshchev, A. and Filippov, A. (2004). From system dynamics and discrete event to practical agent based modeling : Reasons, techniques, tools. In *The 22nd International Conference of the System Dynamics Society.*
- Bouafou, K., Ba Diao, M., Garba, I., Samba, A., Djaby, B., Traoré, S., Ali, A., Alhassane, A., Tinni Hakidou, S., Mohamed, H., Maigary, I., Minounguo, B., and Namodji, L. (2015). Bulletin de suivi de la campagne agropastorale en afrique de l'ouest.
- Bousquet, F., Bakam, I., Proton, H., and Le Page, C. (1998). Cormas : Common-pool resources and multi-agent systems. In Pasqual del Pobil, A., Mira, J., and Ali,

- M., editors, *Tasks and Methods in Applied Artificial Intelligence*, volume 1416 of *Lecture Notes in Computer Science*, pages 826–837. Springer Berlin / Heidelberg.
- Bousquet, F. and Le Page, C. (2004). Multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176(3-4) :313–332.
- Boussard, H., Martel, G., and Vasseur, C. (2010). Spatial links specifications in the apiland simulation approach : an application to the coupling of a farm model and a carabid population model. *Communication présentée à Colloque LandMod 2010, Montpellier, FRA ( 2010-02-03 - 2010-02-05 ) Versailles : Editions Quae.*
- Bowers, P. and Stephenson, K. (1997). A “regular” pentagonal tiling of the plane. *Conformal. Geometry. Dynamics.*, 1 :58–86.
- Bresenham, J. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1) :25–30.
- Cantrell, C. (2000). Modern mathematical methods for physicists and engineers. *Cambridge University Press.*
- Cantwell, M. and Forman, R. (1993). Landscape graphs : Ecological modeling with graph theory to detect configurations common to diverse landscapes. *Landscape Ecology*, 8 :239–255. 10.1007/BF00125131.
- Cayley, A. (1878). The theory of groups ; graphical representations. *Amer. J. Math.*, 1 :174–176.
- Cervelle, J., Forax, R., and Roussel, G. (2006). Tatoo : An innovative parser generator. In : *Proc. of the 4th Int. Conf. PPPJ'06. ACM International Conference Proceedings, Mannheim, Germany*, pages 13–20.
- Clarke, K., Hoppen, S., and Gaydos, L. (1997). A self-modifying cellular automaton model of historical urbanization in the san francisco bay area. *Environment and Planning B : Planning and Design*, 24 :247–261.
- Commins, J. (2015). Modélisation et scénarisation de la gestion d’une espèce envahissante : cas de l’ajonc d’europe, *ulex europaeus*, à la réunion (océan indien)., mémoire de master 2 " télédétection et risques naturels". Master’s thesis, Université de la Réunion.
- Costanza, R. (1987). Simulation modeling on the macintosh using stella. *BioScience*, 37 :129–132.
- Costanza, R. and Gottlieb, S. (1998). Modeling ecological and economic systems with stella :part ii. *Ecological Modeling*, 112 :81–247.
- Costanza, R. and Voinov, A. (2004). *Landscape Simulation Modeling. A spatially explicit, dynamic approach.* Springer-Verlag New York.

- Couclelis, H. (1992). People manipulate objects (but cultivate fields) : Beyond the raster-vector debate in gis. In *Proceedings of the International Conference GIS - From Space to Territory : Theories and Methods of Spatio-Temporal Reasoning on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 65–77, London, UK. Springer-Verlag.
- David, D., Gangat, Y., Payet, D., and Courdier, R. (2011). Reification of emergent urban areas in a land-use simulation model in reunion island. *Journées Francophones des Systèmes Multi-Agents*.
- Degenne, P. (2012). *Une approche générique de la modélisation spatiale et temporelle : application à la modélisation des paysages*. PhD thesis, Université Paris-Est.
- Degenne, P., Lo Seen, D., Parigot, D., Forax, R., Tran, A., Ait Lahcen, A., Curé, O., and Jeansoulin, R. (2009). Design of a domain specific language for modelling processes in landscapes. *Ecological Modelling*, 220(24) :3527–3535.
- Degenne, P., Parigot, D., Curé, O., Ait Lahcen, A., Forax, R., and Lo Seen, D. (2010). Modelling the environment using graphs with behaviour : do you speak ocelet? In *International Congress on Environmental Modelling and Software*, Ottawa, Canada.
- Delbrück, T. (1993). Silicon retina with correlation-based, velocity-tuned pixels. *Neural Networks, IEEE Transactions*.
- Develay, M. (1989). *Sur la méthode expérimentale*. Aster, 8, INRP.
- Diestel, R. (2005). *Graph Theory*. Springer-Verlag, New York.
- Drogoul, A. (1993). *De la Simulation Multi-Agents à la Résolution Collective de Problèmes*. PhD thesis, Université de Paris VI.
- Durand, D. (2013). *La systémique*.
- Eberlein, R. and Peterson, D. (1992). Understanding models with vensim tm. *European Journal of Operational Research*, 59 :216–219.
- Engelen, G., White, R., Uljee, I., and Drazan, P. (1995). Using cellular automata for integrated modelling of socio-environmental systems. *Environmental monitoring and Assessment*, 30 :203–214.
- Fabre, J., Louchart, X., Moussa, R., Dagès, C., Colin, F., Rabotin, M., Raclot, D., Lagacherie, P., and Voltz, M. (2010). Openfluid : a software environment for modelling fluxes in landscapes. *LANDMOD2010, Montpellier, France*.
- Faivre, R., Leenhardt, D., Voltz, M., Benoît, M., Papy, F., Dedieu, G., and Wallach, D. (2004). Spatialising crop models. *agronomie*. 24 :205–217.

- Fall, A. and Fall, J. (2001). A domain-specific language for models of landscape dynamics. *Ecological Modelling*, 141(1-3) :1–18.
- Fall, A., Fortin, M., Manseau, M., and OBrien, D. (2007). Spatial graphs : Principles and applications for habitat connectivity. *Ecosystems*, 10 :448–461.
- FAO FIDA et PAM (2014). *L'état de l'insécurité alimentaire dans le monde 2014. Créer un environnement plus propice à la sécurité alimentaire et à la nutrition.*
- Feder, F. and Bourgeon, G. (2009). Mise à jour de la carte des sols de l'île de la réunion. *Etude et Gestion des Sols*.
- Ferber, J. (1995). *Les systèmes multi-agents, vers une intelligence collective*. Inter-Editions.
- Ferguson, J. (1963). Multivariable curve interpolation,. *The Boeing Company, Document D2-22504*.
- Ford, A. (1999). *Modeling the environment : an introduction to system dynamics models of environmental systems*. Island Press, Washington DC.
- Forrester, J. (1968). *Principles of Systems*. Wright-Allen Press Cambridge, Mass.
- Gardner, M. (1970). The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223 :120–123.
- Gasser, L., Braganza, C., and Herman, N. (1987). *Distributed Artificial Intelligence*, chapter MACE A Flexible Testbed for Distributed AI Research. Pitman.
- Gaucherel, C., Boudon, F., Houet, T., Castets, M., and Godin, C. (2012). Understanding patchy landscape dynamics : Towards a landscape language. *PLoS ONE* 7(9).
- Gaucherel, C., Fleury, D., and Auclair, D. (2006a). Neutral models for patchy landscapes. *Ecol Modell*, 197 :159–170.
- Gaucherel, C., Giboire, N., Viaud, V., Houet, T., Baudry, J., and Burel, F. (2006b). A domain-specific language for patchy landscape modelling : The brittany agricultural mosaic as a case study. *Ecological Modelling*, 194(1-3) :233–243.
- Gaucherel, C., Houllier, F., Auclair, D., and Houet, T. (2014). Dynamic landscape modelling : The quest for a unifying theory. *Living Reviews in Landscape Research*, 8.
- Girard, M. and Girard, C. (2010). *Traitement des données de télédétection - 2ème édition - Environnement et ressources naturelles : Environnement et ressources naturelles*. Dunod.

- Grelley, P. (2012). Contrepoint — la méthode expérimentale. . *Informations sociales*, 174 :23–23.
- Grignard, A., Taillandier, P., Gaudou, B., Vo, D., Huynh, N., and Drogoul, A. (2013). Gama 1.6 : Advancing the art of complex agent-based modeling and simulation. In *'PRIMA 2013 : Principles and Practice of Multi-Agent Systems', Lecture Notes in Computer Science, Springer*, 8291 :117–131.
- Grimm, V., Berger, U., DeAngelis, D., Gary Polhill, J., Giske, J., and S.F., R. (2010). The odd protocol : A review and first update. *Ecological Modelling*, 221(23) :2760–2768.
- Grünbaum, B. and Shephard, G. (1987). *Tilings and patterns*. Spektrum Akademischer Verlag.
- Gutknecht, O. and Ferber, J. (2000). Madkit : a generic multi-agent platform. In *Proceedings of the fourth international conference on Autonomous agents, AGENTS '00*, pages 78–79, New York, NY, USA. ACM.
- Haile, M. (2005). Weather patterns, food security and humanitarian response in sub-saharan africa. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 360 :2169–2182.
- Hewitt, C. (1976). Viewing control structures as patterns of passing messages. Technical report, Massachussets Institute of Technology.
- HMS (2000). Us army corps of engineers ; hydrological modeling system, technical reference manual ; 158 pages.
- HMS (2010). Us army corps of engineers ; hydrological modeling system, users 's manual ; v 3.5 ; 318 pages.
- Houet, T., Schaller, N., Castets, M., and Gaucherel, C. (2014). Improving the simulation of fine-resolution landscape changes by coupling top-down and bottom-up land use and cover changes rules. *International Journal of Geographical Information Science*, 28 :1–29.
- IPCC (2014). Climate change 2014 : Synthesis report. Contribution of working groups i, ii and iii to the fifth assessment report of the intergovernmental panel on climate change [core writing team : R.K. Pachaurin, and L.A. Meyer, (eds.)]. IPCC, Geneva, Switzerland, 151 pp.
- Jiménez, J., Feito, F., and Segura, R. (2009). A new hierarchical triangle-based point-in-polygon data structure. *Computers & Geosciences*, 35(9) :1843–1853.
- Kelly, R., Jakeman, A., Barreteau, O., Borsuk, M., Elsawah, S., Hamilton, S., Henriksen, H., Kuikka, S., Maier, H., Rizzoli, A., Van Delden, H., and Voinov, A.

- (2013). Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental Modelling and Software*, 47 :159–181.
- Kuipers, B. (1994). *Qualitative Reasoning*. The MIT Press.
- Langlois, A. and Phipps, M. (1997). Automates cellulaires : application à la simulation urbaine. *Hermès, Paris*.
- Langlois, P. (2005). Les automates cellulaires pour la modélisation des systèmes spatiaux. in modélisation en géographie : déterminismes et complexités. *Sous la direction de Yves Guermond. Hermès – Lavoisier, Paris*.
- Laniak, G., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., and Hughes, A. (2013). Integrated environmental modeling : a vision and roadmap for the future. *Environmental Modelling and Software*, 39 :3–23.
- Le Moigne, J. (2006). *La Théorie du Système Général, Théorie de la Modélisation*.
- Lindenmayer, A. and Rozenberg, G. (1972). Developmental systems and languages. *STOC '72 Proceedings of the fourth annual ACM symposium on Theory of computing*, pages 214–221.
- Longley, P., Goodchild, M., Maguire, D., and Rhind, D. (2015). *Geographic Information Science and Systems, 4th Edition*.
- Lowe, S., Browne, M., Boudjelas, S., and De Poorter, M. (2000). 100 of the world's worst invasive alien species : a selection from the global invasive species database. *Invasive Species Specialist Group Auckland,, New Zealand*.
- Magliocca, N., Rudel, T., Verburg, P., McConnell, W., Mertz, O., Gerstner, K., Heinimann, A., and Ellis, E. (2015). Synthesis in land change science : methodological patterns, challenges, and guidelines. *Regional Environmental Change*, 15(2) :211–226.
- Maxwell, T. and Costanza, R. (1997). A language for modular spatio-temporal simulation. *Ecological Modelling*, 103(2-3) :105–113. To Read.
- Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system : A toolkit for building multi-agent simulations. *Simulation*, (96-06-042) :1–11.
- Minsky, M. (1988). *La société de l'esprit*. InterEditions, Paris.
- Mitchell, M., Crutchfield, J., and Das, R. (1996). Evolving cellular automata with genetic algorithms : A review of recent work. in *Proceedings of the First International Conference on Evolutionary Computation and its Applications (EvCA '96)*.

- Mladenoff, D. (2004). Landis and forest landscape models. *Ecological Modelling*, 180 :7–19.
- Müller, J. (2010). Mimosa modelling and simulation platform - <http://mimosa.sourceforge.net/index.html>.
- Morin, E. (1994). *Sur l'interdisciplinarité*.
- Muetzelfeldt, R. and Massheder, J. (2003). The simile visual modelling environment. *European Journal of Agronomy*, 18 :345–358.
- North, M., Howe, T., Collier, N., and Vos, R. (2005). The repast symphony development environment. In *Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*, Argonne National Laboratory, Argonne, IL USA.
- Pavé, A. (2005). *La modélisation et la simulation des objets et processus complexes. Questions scientifiques, méthodologiques et éthiques*.
- Payet, G. (2012). Impact d'un incendie sur un habitat naturel en coeur de parc national : le cas du maïdo, île de la réunion (océan indien) – état des lieux et suggestions pour une gestion conservatoire à court, moyen et long terme. mémoire de stage pour l'obtention du diplôme de mastère spécialité forêt, nature et société. agroparistech-engref, centre de montpellier.
- Popper, J. (1973). *La Dynamique des systèmes, principes et applications*,.
- Raunet, M. (1991). *Le milieu physique et les sols de l'île de La Réunion*.
- Reaser, J., Meyerson, L., Cronk, G., De Poorter, M., Eldrege, L., Green, E., Kairo, M., Latasi, P., Mack, R., Mauremootoo, J., O'dowd, D., Orapa, W., Sastroutomo, S., Saunders, A., Shine, C., Thrainsson, S., and Vaiutu, L. (2007). Ecological and socioeconomic impacts of invasive alien species in island ecosystems. *Environ. Conserv.*, 34 :98–111.
- Rindfuss, R., Walsh, S., Turner, B., Fox, J., and Mishra, V. (2004). Developing a science of land change : challenges and methodological issues. *Proc Natl Acad Sci USA*, 101 :13976–13981.
- Risser, P., Karr, J., and Forman, R. (1984). Landscape ecology : Directions and approaches. *Special Publ. No.2. I11. Nat. Hist. Surv., Champaign, I11*.
- Róka, Z. (1994). One-way cellular automata on cayley graphs. *Theoretical Computer Science*, 132 :259–290.
- Róka, Z. (1999). Simulations between cellular automata on cayley graphs. *Theoret. Comput. Sci.*, 225 :81–111.

- Roques, J. (2013). Modélisation prospective et rétrospective du ruissellement sur deux bassins versants de l'ouest réunionnais. apport de la modélisation hydrologique à la connaissance des hydrosystèmes insulaires volcaniques. Master's thesis, Université d'Avignon.
- Ruas, A. and Sanders, L. (2015). Regards croisés sur la modélisation des dynamiques spatiales. *Revue Internationale de Géomatique*, 25 :275–300.
- Rubel, D. (2006). The heart of eclipse. *Queue - System Evolution*, 4 :36–44.
- SCDB (2014). *Secrétariat de la Convention sur la diversité biologique. 4ème édition des Perspectives mondiales de la diversité biologique. Montréal 155 pages.*
- Schmidt-Lainé, C. and Pavé, A. (2002). Environnement : modélisation et modèles pour comprendre, agir ou décider dans un contexte interdisciplinaire. *Natures Sciences Sociétés « Sciences pour l'ingénierie de l'environnement »*, 10 :5–25.
- Soares-Filho, B., Coutinho, G., and Pennachin, C. (2002). Dinamica—a stochastic cellular automata model designed to simulate the landscape dynamics in an amazonian colonization frontier. *Ecological Modelling*, 154(3) :217–235.
- Taillandier, P., Grignard, A., Gaudou, B., and Drogoul, A. (2014). Des données géographiques à la simulation à base d'agents : application de la plate-forme gama. *Cybergeo : European Journal of Geography [En ligne], Systèmes, Modélisation, Géostatistiques.*
- Touzet, C. (1992). *Les réseaux de neurones : Introduction connexionnisme.*
- Traoré, S., Ali, A., Tinni, S., Samake, M., Garba, I., Maigari, I., Alhassane, A., Samba, A., Diao, M., Atta, S., Dieye, P., Nacro, H., and Bouafou, K. (2014). Agrhymet : A drought monitoring and capacity building center in the west africa region. *Weather Clim. Extrem.*, 3 :22–30.
- Treml, E., Halpin, P., Urban, D., and Pratson, L. (2007). Modeling population connectivity by ocean currents, a graph-theoretic approach for marine conservation. *Landscape Ecology*, 23(S1) :19–36.
- Triolo, J. and Hoff, M. (2006). Stratégie de lutte contre l'ajonc d'europe (ulex europaeus) dans les hauts de l'ouest. office national des forêts, st andré (réunion).
- Urban, D. and Keitt, T. (2001). Landscape connectivity : A graph-theoretic perspective. *Ecology*, 82(5) :1205–218.
- Van Ittersum, M., Ewert, F., Heckelei, T., Wery, J., Alkan Olsson, J., Andersen, E., Bezlepina, I., Brouwer, F., Donatelli, M., Flichman, G., Olsson, L., Rizzoli, A., Van der Wal, T., Wien, J., and Wolf, J. (2008). Integrated assessment of agricultural systems – a component-based framework for the european union (seamless). *Agricultural Systems*, 96 :150–165.

- Veldkamp, A. and Fresco, L. (1996). Clue-cr : an integrated multi-scale model to simulate land use change scenarios in costa rica. *Ecol Model*, 91 :231–248.
- Verburg, P., Soepboer, W., Limpiada, R., Espaldon, M., Sharifa, M., and Veldkamp, A. (2002). Modelling the spatial dynamics of regional land use : The clue-s model. *Environmental Management*, 30 :391–405.
- Vivid Solutions, JTS. (2003). Java topology suite. <http://www.vividsolutions.com/jts/JTSHome.htm>.
- Von Bertalanffy, L. (1950). An outline of general system theory. *The British Journal for the Philosophy of Science*, 1(2) :134–165.
- Von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.
- Warmerdam, F. (2008). The geospatial data abstraction library. *Open Source Approaches in Spatial Data Handling. Vol. 2 of Advances in Geographic Information Science. Springer Berlin Heidelberg*, . 87-104, pages 87–104.
- White, A. (1973). Graphs, groups und surfaces. *North-Holland Mathematics Studies (North-Holland, Amsterdam)*.
- Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (<http://ccl.northwestern.edu/netlogo/>).
- Yang, S., Yong, J., Sun, J., Gu, H., and Paul, J. (2010). A point-in-polygon method based on a quasi-closest point. *Computers & Geosciences*, 36(2) :205–213.



# Table des figures

2.1	Structure composée d'éléments permettant l'échange de matière par flux avec Stella (Costanza and Gottlieb, 1998). . . . .	26
2.2	Les différents motifs du jeu de la vie. . . . .	27
2.3	Les différents modes de communication des agents, entre eux et leur environnement (Ferber, 1995). . . . .	28
2.4	Les différents motifs de graphes de paysages identifiés par Cantwell and Forman (1993) . . . . .	32
2.5	Différents types d'interaction (a et b : spatiale ; c : sociale ; d : fonctionnelle) entre les éléments d'un paysage (Degenne, 2012). . . . .	33
3.1	Un graphe biparti entre les sommets appartenant aux ensembles $X_1$ et $X_2$ (a). Un graphe triparti entre les éléments appartenant aux ensembles $X_1$ , $X_2$ et $X_3$ en (b). Un graphe complet (c). . . . .	38
3.2	Un graphe planaire en (a), il est possible d'avoir une représentation de ce graphe sans intersection entre les arêtes. Un graphe planaire topologique (b). . . . .	38
3.3	Un modèle complet représenté par un graphe. Différents types d'éléments sont représentés par les sommets : communes, parcelles cadastrales, exploitations. . . . .	39
3.4	Décomposition du graphe du modèle complet en plusieurs sous-graphes : un sous-graphe entre les communes et les parcelles qui en dépendent (a), un sous-graphe entre les exploitations et leurs parcelles (b) et le sous-graphe de voisinage entre les parcelles (c). . . . .	40
3.5	Exemple de graphes sur lesquels appliquer des fonctions d'interaction. . . . .	43
3.6	Les différentes opérations booléennes applicables sur deux géométries $A$ et $B$ (Vivid Solutions, 2003) . . . . .	45
3.7	Représentation de l'espace par objets vecteurs. Un diagramme de Voronoï (a). Des polygones caractérisant un cadastre (b). . . . .	46
3.8	Exemple de représentation d'une image par une matrice 3x3 avec trois bandes qui contiennent les composantes des couleurs rouge, verte et bleue. . . . .	47

3.9	Représentation d'une série temporelle contenant les données en précipitations et la possibilité de mettre à jour l'information spatiale à chaque décade lors de la simulation. . . . .	48
3.10	Les différents pavages réguliers avec des formes géométriques régulières : carrées (a), hexagonales (b) et triangulaires (c). . . . .	49
3.11	Un pavage semi-régulier de plusieurs types de pentagones (a). Un pavage semi-régulier composé de pentagones, triangles et quadrilatères (b) (Bowers and Stephenson, 1997). . . . .	49
3.12	Le voisinage de Von Neumann pour des cellules carrées (a), triangulaires (b) et hexagonales (c). Le voisinage de Moore pour des cellules carrées (d) et triangulaires (e). . . . .	53
3.13	Représentations des voisins pour une cellule centrale en rouge : en (a) un voisinage de Von Neumann de distance 2, en (b) un voisinage de Moore de distance 2 et en (c) un voisinage selon un cercle de rayon 2. . . . .	54
3.14	En (a) la matrice de dimension $6 \times 6$ contenant les valeurs d'altitude correspondant à l'échelle en (d). En (b) l'ensemble des arêtes sur cette matrice avec un voisinage de Von Neumann. En (c) l'orientation des arêtes selon l'altitude des cellules. . . . .	55
3.15	En (a) l'état initial $t = 1$ des entités cellulaires avec une valeur pour les propriétés "eau" de 4. En (b) le résultat au temps $t = 2$ . . . . .	56
3.16	Schéma de la représentation d'un espace régulier en (a) et la représentation en objets, ou géométries en (b). Le graphe de voisinage sur les géométries est présenté en (c). . . . .	56
3.17	En (a) les couches superposées spatialement d'entités géométriques avec des entités cellulaires. En (b) le graphe permettant à ces entités d'interagir. En (c) le graphe orienté d'entités géométriques après avoir obtenu l'information sur l'altitude venant des entités cellulaires. . . . .	57
3.18	En (a) des flux qui ont un chemin différent qui n'est pas traduit sur les entités géométriques (b). Le calcul de flux au niveau des entités cellulaires permet de palier ce problème et de rapporter le résultat sur les entités géométriques correspondantes en (c) et (d). . . . .	58
3.19	En (a) la matrice représentant les entités cellulaires et leurs valeurs en eau. En (b) un graphe entre les entités cellulaires et une entité sans représentation spatiale. Une interaction entre la cellule sans représentation spatiale et les entités cellulaires permet d'affecter une valeur en eau pour toutes les entités cellulaires connectées (c). . . . .	59
3.20	En (a) deux couches d'entités cellulaires de résolutions différentes. En (b) le graphe permettant de lancer des interactions sur ces entités. En (c) l'affectation des propriétés "mnt" des cellules de résolution la plus fine sur les entités cellulaires de la deuxième couche par interaction. . . . .	60
4.1	Schéma des phases d'analyses lexicales et syntaxiques. . . . .	65

4.2	Définition d'une entité cellulaire en Ocelet, caractérisée par une propriété "cell" (a). Cette déclaration correspond à la création de deux bandes de matrice pour deux propriétés de l'entité (b). . . . .	66
4.3	Définition d'un datafacier permettant de lire des données raster en Ocelet. . . . .	67
4.4	Un modèle numérique de terrain en (a) dont les données sont extraites pour créer une matrice de $3 \times 3$ , la zone de la matrice doit être spécifiée dans le modèle. En (c) le stockage des valeurs correspondantes pour les propriétés "mnt" des entités cellulaires. . . . .	68
4.5	Définition d'un datafacier permettant de lire des séries temporelles d'images. . . . .	68
4.6	Exemple d'utilisation du datafacier de série temporelle d'images et de la mise à jour des propriétés "mnt" des entités cellulaires contenues dans la relation "cells". . . . .	69
4.7	(a) Relation entre des entités cellulaires de même type. (b) Relation entre entités géométriques et cellulaires. (c) Relation entre entités cellulaires de types différents. . . . .	69
4.8	Diagramme UML des classes de graphes dans le moteur d'exécution d'Ocelet. . . . .	70
4.9	Représentation du voisinage d'un automate cellulaire. Pour des cellules carrées avec en (a) le voisinage de Von Neumann, et en (b), le voisinage de Moore. Le voisinage pour des cellules triangulaires en (c) et hexagonales en (d) (Róka, 1999) . . . . .	72
4.10	Diagramme de classes des générateurs . . . . .	73
4.11	En (a) les voisins qui sont simulés par les générateurs pour un voisinage de Moore de distance 1. En (b) la liste chaînée de générateurs permettant de simuler les arêtes d'un graphe. En (c) les valeurs des positions des générateurs pour mettre à jour la position des entités cellulaires. . . . .	73
4.12	(a) Parcours des générateurs sans leurs inverses pour l'entité cellulaire située au milieu. (b) Parcours des arêtes simulées par les générateurs lors du passage sur les entités précédentes. (c) Parcours total simulé. . . . .	75
4.13	Méthode de stockage des valeurs temporaires sur la structure <i>St</i> en (b), (d) et (f) et leurs synchronisations à la fin du parcours de la première ligne en (c) et de la deuxième ligne en (g). Le parcours des voisins est simulé pour la première ligne en (a) et en (e) sur la deuxième ligne. . . . .	77
4.14	Différents pavages constitués d'entités cellulaires hexagonales en (b) ou triangulaires en (c) qui peuvent se rapporter à une matrice carrée en (a) par décalage. . . . .	77

- 4.15 Les arêtes en pointillées simulent le parcours effectué par les générateurs sur les lignes précédentes. Les figures (a) et (b) présentent les générateurs d'une cellule hexagonale d'une colonne paire. Les figures (c) et (d) présentent les générateurs d'une cellule hexagonale de colonnes impaires. . . . . 78
- 4.16 La figure (a) présente les voisins d'une cellule hexagonale sur une colonne paire et impaire et leurs positions sur la matrice. La figure (b) présente la liste chaînée de générateurs. En (c) la valeur des positions des générateurs qui serviront à mettre à jour la position des cellules voisines. . . . . 78
- 4.17 Les arêtes en pointillées simulent le parcours effectué par les générateurs sur les lignes précédentes. Les figures (a) et (b) présentent les générateurs d'une cellule triangulaire d'une colonne paire. Les figures (c) et (d) présentent les générateurs d'une cellule triangulaire de colonnes impaires. . . . . 79
- 4.18 Définition d'une relation entre des entités cellulaires de même type en Ocelet. La fonction d'interaction traduit un échange d'eau selon le voisinage de chaque cellule. . . . . 79
- 4.19 La structure *CellIndex* construite pour stocker les positions des cellules extraites sur des géométries. Un couple de sommets  $(x, y)$  avec  $x \in X$  et  $y \in Y$  correspond à la position de l'entité cellulaire sur la matrice. Les entités géométriques correspondant spatialement à cette entité sont reliées à un sommet appartenant à  $Y$ . . . . . 82
- 4.20 Exemple de définition d'une relation entre entités géométriques "Plot" et entités cellulaires "CellEntity". La fonction d'interaction "setMnt" permet d'attribuer les valeurs des propriétés "mnt" des entités cellulaires et de les attribuer aux propriétés "mnt" des entités "Plot". La fonction d'agrégation calcule la moyenne sur les valeurs venant des entités cellulaires. . . . . 85
- 4.21 (a) Les entités géométriques (polygones). (b) Les entités cellulaires. (c) Les connexions entre l'entité géométrique  $P2$  et les entités cellulaires correspondantes. (d) Résultat de l'interaction. . . . . 85
- 4.22 Présentation d'un code en Ocelet pour connecter explicitement une entité avec des entités cellulaires et la représentation schématique du graphe obtenu. . . . . 86
- 4.23 Définition d'une relation en Ocelet entre des entités cellulaires de types différents : "CellEntity" et "CellEntity2". La fonction d'interaction entre ces entités permet de donner une valeur à la propriété "mnt" des entités "CellEntity2" à partir des entités "CellEntity" correspondantes. . . . . 87

4.24	En (a) la première couche d'entités cellulaires correspondant au rôle "CellEntity". En (b) la deuxième couche correspondant au rôle "CellEntity2". En (c) Le graphe final simulé entre les deux couches. En (d) le résultat de l'application de l'interaction. . . . .	87
4.25	Deux couches d'entités cellulaires de résolutions différentes en (a) et (b). La superposition des couches peut inclure un décalage selon la différence de résolution. Le graphe obtenu se base sur le plus proche voisin (d). . . . .	89
5.1	Représentation de diverses couches d'information telles que le modèle numérique de terrain, une couche de NDVI ainsi que le parcellaire. Les flux sont représentés en tenant compte de l'altitude. . . . .	98
5.2	Définition de la relation et de la fonction d'interaction traduisant les flux d'eau entre les entités cellulaires. . . . .	98
5.3	Représentation du graphe de flux entre les entités linéaires et leurs nœuds d'intersection (a). Représentation du graphe entre les entités linéaires et les entités cellulaires (b). . . . .	99
5.4	Définition de la relation en Ocelet et de l'interaction traduisant les échanges d'eau entre les entités cellulaires et les entités linéaires. . . .	99
5.5	Définition de la relation entre les entités linéaires et leurs nœuds à leurs intersections. La fonction d'interaction traduit les échanges de quantités d'eau entre les linéaires en amont ou en aval d'un nœud. . .	100
5.6	Définition en Ocelet de la relation permettant d'attribuer la valeur des précipitations vers les entités "CellEntity". . . . .	100
5.7	Définition des paramètres de contrôles des taux d'infiltrations selon la valeur en NDVI ou encore des types de surface des entités. . . . .	101
5.8	La zone d'étude se situe à l'Ouest de l'île de la Réunion, la limite (polygone rouge) du bassin versant de Saint Gilles présente des variations d'altitude allant de 150 à 2000 mètres. . . . .	101
5.9	Les différentes couches de données du modèle : les polygones de Thiessen (a) ; la couche du parcellaire (b) ; le réseau de routes et des cours d'eau (c) et la carte pédologique(d). Les différentes couches raster : avec le MNT (e) et les couches NDVI de l'image de 2002 pour l'épisode cyclonique et celle de 2009 pour la simulation sur l'année 2009 (f). Les différents types de sols (g) correspondant à la carte (d). . . .	102
5.10	En (a) les données des précipitations horaires venant de la station du Tan Rouge, utilisées pour simuler les débits. Comparaison des débits simulés et observés à l'exutoire lors de l'épisode cyclonique Dina de 2002 (b). . . . .	103
5.11	En (a) les données des précipitations horaires venant de la station du Tan Rouge. Comparaison des débits simulés et observés à l'exutoire le 05/02/2009 (b). . . . .	104
5.12	Photo de l' <i>Ulex europaeus</i> à l'île de la Réunion. . . . .	105

5.13	Relation entre les entités "NDVICELL" et les entités "UlexCell" pour attribuer une valeur en NDVI aux entités "UlexCell". . . . .	108
5.14	Schémas de la diffusion de proche en proche des graines (a) et (c) pour un voisinage de distance de 1. Le processus de croissance des plantes est appliqué lorsque le milieu est ouvert et toléré par l'espèce (b). . . . .	109
5.15	Relation dont l'interaction permet de définir une aire de couverture aux entités "UlexCell" par intersection géométrique entre la cellule des entités et le polygone des entités "Coverage". . . . .	109
5.16	Les paramètres de contrôles définis dans le scénario en Ocelet. . . . .	110
5.17	Zone d'étude délimitée par un polygone, en rouge. . . . .	111
5.18	Les différentes couches de données utilisées dans le modèle : la couverture de l'Ulex europaeus en 2004 (a) et en 2012 (b). La zone incendiée en rouge (c). Les données raster telles que le modèle numérique de terrain (d) et la série temporelle d'images satellite à partir desquelles sont attribuées les valeurs en NDVI (e). La zone d'étude est représentée par le polygone vert sur les figures (a), (b) et (c). . . . .	112
5.19	Sorties de la simulation au format KML à différents pas de temps : 2004 (a), 2006 (b), 2008 (c) et 2012 (d). . . . .	113
5.20	Sorties de plusieurs simulations de diffusion de l'Ulex europaeus pour une distance de voisinage : de 1 (a), 2 (b), 3 (c), 4 (d), 5 (e) et 6 (f). . . . .	114
5.21	Exemple de définition de relation en Ocelet permettant de mettre à jour les données pluviométriques. . . . .	119
5.22	Exemple de définition de la relation en Ocelet permettant de lancer les trois bilans traduits en interactions entre les entités représentées par un pays et leurs entités cellulaires "Crop" correspondantes. . . . .	120
5.23	Figure synthétique de la description de la dynamique du modèle avec l'appel des interactions calculant les bilans lancés dans la boucle journalière. . . . .	120
5.24	La zone d'étude concerne les pays de l'Afrique de l'Ouest : Mauritanie, Sénégal, Guinée-Bissau, Guinée, Mali, Burkina-Faso, Sierra Leone, Gambie, Libéria, Côte d'Ivoire, Ghana , Togo, Bénin, Tchad, Niger et le Nigéria. . . . .	121
5.25	Carte des rendements du Sorgho PP F2 pour l'année 2014. . . . .	123
5.26	Cartes des rendements pour le mil PP F2 (a), maïs 90 F4 (b) et sorgho 90 F4 (c) pour l'année 2014. . . . .	124
5.27	Carte des récoltes au 10/08/2014 (a) et au 30/09/2014 (c). Carte du stress hydrique au 10/08/2014 (b) et au 30/09/2014 (d) . . . . .	124
5.28	Les points étudiés où les simulations ont été effectuées pour comparer les résultats de rendements avec les données climatiques des 24 dernières années : Nord du Sénégal (a), Mali (b), Sud Bénin (c), Nord Burkina (d) et Nord-Ouest Nigeria (e). . . . .	125

---

5.29	Simulation des rendements pour l'année 2014 au 11 août en simulant à partir des données climatiques de 1989 pour le Nord du Sénégal (a), Mali (b), Sud Bénin (c), Nord Burkina (d) et Nord-Ouest Nigeria (e).	126
5.30	Niveau de satisfaction des besoins en eau des céréales au 30 Juin 2015, bulletin du 02/06/2015 publié par AGRHYMET. . . . .	127
A.1	Environnement de développement d'Ocelet (OMP). . . . .	161



# Liste des tableaux

5.1	Récapitulatif des données d'entrée du modèle du ruissellement. . . . .	103
5.2	Récapitulatif des données d'entrée du modèle de diffusion d'espèces invasives. . . . .	111
5.3	Récapitulatif des données d'entrée du SAP. . . . .	122



# Appendices



# Annexe A

## Environnement de développement d'Ocelet (OMP)

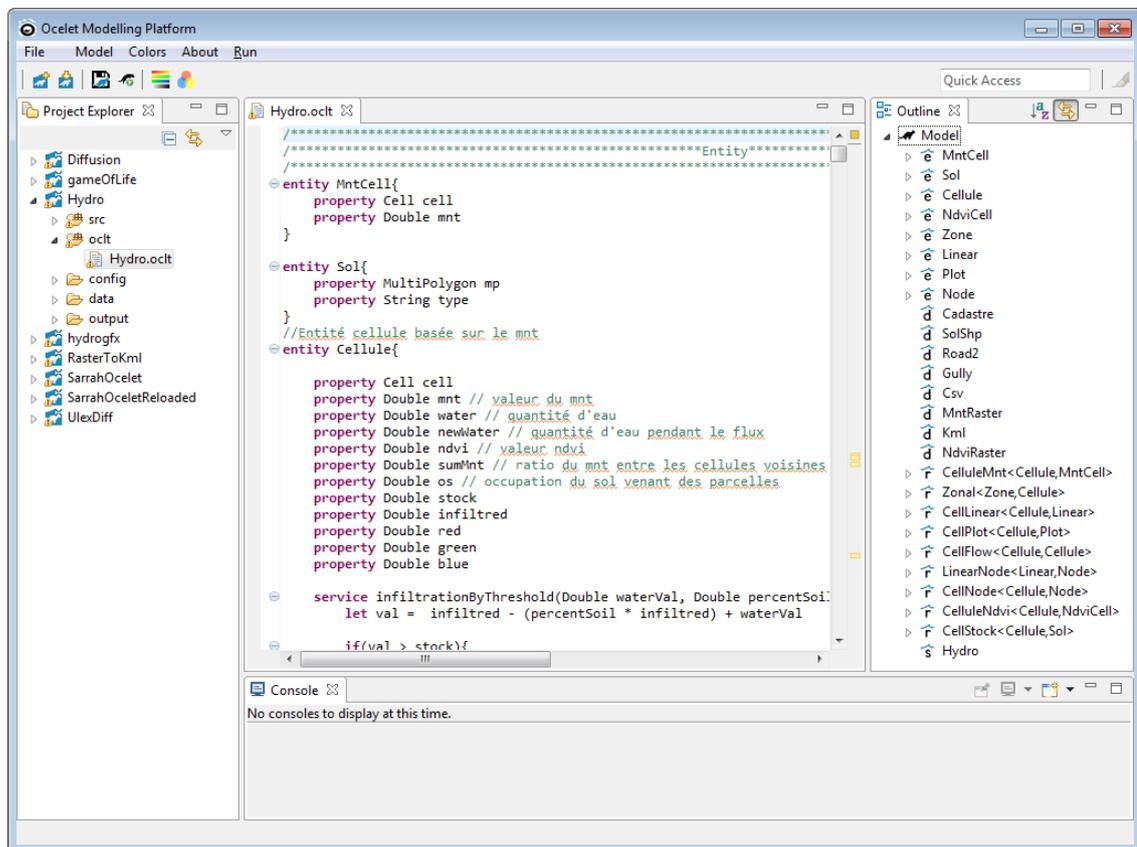


FIGURE A.1 – Environnement de développement d'Ocelet (OMP).



# Annexe B

## Principales Méthodes

### B.1 Méthodes de la propriété "Cell"

1. **getRes()** : retourne la résolution de la cellule.
2. **getType()** : retourne le type de la forme de la cellule (carrée, triangle, hexagone).
3. **getX()** : retourne le numéro de colonne de la cellule sur la matrice.
4. **getY()** : retourne le numéro de la ligne de la cellule sur la matrice.
5. **toPolygon()** : retourne la forme géométrique de la cellule en coordonnées projetées.
6. **distance(Cell cell)** : retourne la distance euclidienne par rapport à une autre cellule.
7. **distance(Geometry geometry)** : retourne la distance euclidienne par rapport à une géométrie.
8. **intersection(Geometry geometry)** : retourne l'intersection géométrique de la cellule avec une géométrie.
9. **intersect(Geometry geometry)** : fonction de test d'intersection avec une géométrie.
10. **symDifference(Geometry geometry)** : retourne la différence symétrique par rapport à une géométrie.
11. **difference(Geometry geometry)** : retourne la différence par rapport à une géométrie.
12. **union(Geometry geometry)** : retourne l'union entre la cellule et une géométrie.
13. **buffer(Double distance)** : retourne un polygone de la cellule sur lequel à été appliqué l'opération géométrique "buffer".
14. **centroid()** : retourne un point de type géométrie qui est le centre de la cellule.

## B.2 Méthodes des graphes d'interaction

### Graphes entre entités cellulaires de même type

1. **connect**(*Grid grid*) : prend en compte une matrice de données et en fait un graphe d'interaction d'entités cellulaires.
2. **createSquares**(*Shapefile shapefile, Double resolution*) : créer une matrice basée sur l'enveloppe du shapefile passé en paramètre dont les longueurs des lignes et des colonnes sont déduites de la résolution passée en paramètre. Le résultat est un pavage régulier composé d'entités cellulaires du type spécifié sur le graphe dont la forme de la cellule est carrée.
3. **createHexagons**(*Double[] boundaries, Double resolution*) créer une matrice basée sur l'enveloppe du shapefile passé en paramètre dont les longueurs des lignes et des colonnes sont déduites de la résolution passée en paramètre. Le résultat est un pavage régulier composé d'entités cellulaires du type spécifié sur le graphe dont la forme de la cellule est hexagonale.
4. **createTriangles**(*Double[] boundaries, Double resolution*) créer une matrice basée sur l'enveloppe du shapefile passé en paramètre dont les longueurs des lignes et des colonnes sont déduites de la résolution passée en paramètre. Le résultat est un pavage régulier composé d'entités cellulaires du type spécifié sur le graphe dont la forme de la cellule est triangulaire.
5. **neighbourMoore**(*Double distance*) : spécifie un voisinage de Moore entre les entités cellulaires avec une distance de voisinage.
6. **neighbourVonNeumann**(*Double distance*) : spécifie un voisinage de Von Neumann entre les entités cellulaires avec une distance de voisinage.
7. **neighbourCircle**(*Double distance*) : spécifie un voisinage circulaire entre les entités cellulaires avec une distance de voisinage.

### Graphes entre entités cellulaires et entités géométriques ou sans représentation spatiale

1. **connect**(*Grid grid, List Eg*) : connecte toutes les entités géométriques *Eg* avec les entités cellulaires spatialement correspondantes de la grille "grid".
2. **connect**(*Ec ec, E e*) : connexion entre une entité cellulaire *ec* et une entité *e* dans le graphe d'interaction.
3. **disconnect**(*E e*) : déconnecte toutes les arêtes entre une entité non cellulaire *e* et les toutes entités cellulaires correspondantes dans le graphe d'interaction.
4. **disconnect**(*Ec ec, E e*) : déconnecte les arêtes entre une entité non cellulaire *e* et une entité cellulaire *ec*.

### Graphes entre entités cellulaires de types différents

1. **connect**(*Grid grid1, Grid grid2*) : prend en compte deux matrices de données et en fait un graphe d'interaction d'entités cellulaires de types différents.

## B.3 Méthodes des datafacers "RasterFile" et "TemporalSeriesFile"

### RasterFile

1. **readAllGeneratedEntity()** : retourne une grille d'entités cellulaires d'une taille égale à la taille de l'image passée en définition du datafacer.
2. **readAllGeneratedEntity(Shapefile shapefile)** : retourne une grille d'entités cellulaires correspondant spatialement à la zone du shapefile passé en paramètre.
3. **export(Grid grid, String fileName, String EPSG, String[] properties)** : exporte un fichier raster avec la grille de données correspondant à un type d'entité cellulaire passée en paramètre "grid". Le nom de fichier en paramètre "fileName" et le géoréférencement en paramètre "EPSG". Une bande du fichier raster exporté correspond à une propriété des entités passée en paramètre "properties".
4. **exportAll()** : exporte un fichier raster avec la grille de données correspondant à un type d'entité cellulaire passée en paramètre "grid". Le nom de fichier en paramètre "fileName" et le géoréférencement en paramètre "EPSG". Une bande est créée et contient les valeurs pour chaque propriété du type d'entité cellulaire de la grille.

### TemporalSeriesFile

1. **update()** : met à jour l'ensemble de la bande "property" d'une matrice passée en paramètre.
2. **next()** : donne accès à la prochaine image raster contenue dans la série temporelle d'images.
3. **getIndex()** : retourne l'index du fichier de la série.
4. **setIndex()** : met à jour l'index de la série.
5. **setFileName()** : met à jour l'index de la série par rapport à un nom d'image de la série.
6. **getFileName()** : retourne le nom du fichier raster actuel de la série.



## Annexe C

# Intégration de plusieurs formes de représentations spatiales dans un modèle de simulation

## Intégration de plusieurs formes de représentations spatiales dans un modèle de simulation

Mathieu Castets\*, Pascal Degenne\*, Danny Lo Seen\*, Pascal Poncelet\*\*

\* CIRAD, UMR TETIS : Territoire, Environnement, Télédétection et Information Spatiale  
{Mathieu.Castets,Pascal.Degenne,Danny.Lo-Seen}@teledetection.fr

\*\* LIRMM : Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier  
Pascal.Poncelet@lirimm.fr

### 1 Introduction

L'étude de l'évolution d'un système complexe spatialisé devient critique dans de nombreux domaines. Par exemple, pour comprendre l'influence de l'homme sur le ruissellement dans les entités spatiales et selon leurs types d'occupation du sol (culture, urbanisation, habitation...) de manière à permettre aux décideurs d'anticiper les politiques d'urbanisme et d'agriculture. Dans cette démonstration, nous nous intéressons plus particulièrement à la modélisation des compétences de l'expert pour simuler les dynamiques spatiales de systèmes paysagers complexes. La représentation de l'espace nécessite de pouvoir manipuler différentes entités géométriques ainsi que les interactions qui peuvent exister entre ces entités. Actuellement la plupart des approches nécessite de sélectionner au préalable le format de discrétisation de l'espace : (i) le format vectoriel où les entités sont représentées par leur forme géométrique, ou (ii) le découpage de l'espace en éléments réguliers (raster). Nous proposons dans cette démonstration une extension d'un langage de modélisation appelé Ocelet, qui utilise des graphes d'interaction afin de décrire les dynamiques spatiales. Nous montrons dans la section 2 qu'il est possible via Ocelet de combiner différentes représentations spatiales dans les processus de modélisation. Nous décrivons en section 3 un cas réel d'application sur un modèle de ruissellement en milieu insulaire tropical.

### 2 Présentation d'Ocelet

Ocelet est un langage de programmation « métier » pour la modélisation de dynamiques spatiales et paysagères (Degenne (2012)). Les concepts clés du langage ont été définis pour permettre aux modélisateurs de décrire un paysage à l'aide d'entités spatiales en interaction

dynamique. Les relations entre entités sont exprimées à l'aide du concept de graphe d'interaction où les nœuds sont les entités, et les arcs entre entités portent les fonctions traduisant la nature des interactions entre ces entités. Le système évolue dans le temps suite aux appels à des fonctions d'interaction à partir du scénario. Un modèle Ocelet contient ainsi plusieurs graphes d'interaction que l'on peut comparer à un réseau sémantique qui représente la partie statique des connaissances sur un système (Curé et al. (2010)). Mais avec Ocelet on fait aussi porter pour chaque type d'arc une sémantique opérationnelle décrivant les changements d'état des entités résultant de leur interaction. Cette sémantique opérationnelle décrit les aspects dynamiques du système. Une génération de code à partir de ces descriptions permet d'obtenir un programme exécutable pour simuler l'évolution du système (Degenne et al. (2010)). Ocelet permet déjà de simuler les dynamiques paysagères en utilisant des géométries (format vectoriel). Cependant, les géométries ne sont pas appropriées pour représenter un champ continu de valeurs, telle qu'une hétérogénéité à l'intérieur de la géométrie qui peut être utile pour modéliser certains processus. Ce constat nous a amené à doter Ocelet de la capacité à manipuler dans un même modèle des entités ayant des représentations duales : vecteur et raster. Cette dualité permet d'utiliser la forme raster soit en tant que grille de cellules, soit en tant que réseau d'entités en interaction les unes avec les autres. On peut ainsi s'appuyer sur les concepts de base du langage pour exprimer les dynamiques à différentes échelles en choisissant la représentation la plus appropriée à chaque niveau.

### 3 Cas d'application

Nous avons expérimenté l'usage d'Ocelet ainsi étendu avec la construction d'un modèle de ruissellement sur une zone située sur le territoire de la côte ouest de l'île de La Réunion (TCO). Le modèle doit permettre de simuler le ruissellement en tenant compte de l'artificialisation des sols. Ce cas d'expérimentation est en lien avec un modèle hydrologique développé dans le cadre de l'ANR Descartes (Augusseau et al. (2013)). Le modèle intègre ici la connaissance du processus de ruissellement à deux échelles : l'une mettant en avant les flux d'eau entre entités parcelles, et l'autre à une échelle plus fine prenant en compte l'hétérogénéité à l'intérieur d'une parcelle. Nous avons défini pour ce modèle plusieurs types d'entités : l'habitation, la parcelle, la ravine et la route. Nous construisons un graphe basé sur le voisinage spatial entre entités de façon à faire porter par ce graphe les écoulements qui peuvent avoir lieu entre elles. Les arcs de ce graphe indiquent qu'il peut y avoir un flux d'eau d'une entité vers une autre. Chacun de ces arcs porte une pondération relative à la proportion du flux sortant d'une entité dans cette direction. Pour calculer ces pondérations nous avons procédé de la façon suivante : le flux se propage entre entités voisines en suivant le sens de la pente. Cette pente est calculée avec la valeur moyenne des données raster du modèle numérique de terrain sur la zone couverte par chaque entité. Nous obtenons ainsi le graphe d'interaction (Figure 1a). Pour traduire le fait que l'artificialisation impacte les flux d'eau entre les entités, nous avons calculé un flux à l'intérieur de chaque entité, en se basant sur le parcours des pixels des indices d'artificialisation pour chaque entité. L'intensité du flux entre les entités est ainsi pondérée par les flux calculés à l'intérieur de celles-ci (Figure 1b). La dynamique du modèle est exprimée selon des pas de temps discrets. A chaque pas de temps, les entités vont échanger les quantités d'eau selon la méthode décrite précédemment. Pour mettre en avant le changement de l'artificialisation, nous intégrons une nouvelle image contenant les indices d'artificialisation (par exemple une

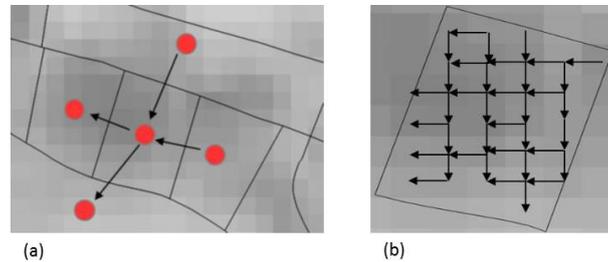


FIG. 1 – *Graphe d'entités spatiales, les nœuds en rouge sont les entités, les arêtes en noir sont connectées entre entités voisines et orientées selon la pente moyenne de ces entités. A droite : Parcours de pixels à l'intérieur d'une géométrie pour une entité, le flux entre pixels voisins est dirigé selon la pente des pixels et pondéré selon l'indice d'artificialisation.*

image tous les  $n$  pas de temps). A chaque intégration d'une nouvelle image, nous recalculons les pondérations des flux entre entités, mais seulement pour celles des entités dont on a pu détecter un changement significatif du niveau de l'artificialisation. La sortie du modèle peut être visualisée sous Google Earth. Les parcelles sont reliées entre elles par des arêtes orientées suivant les flux d'eau détectés lors de la simulation. Pour mettre en avant la quantité d'eau passant à travers ces flux, la quantité d'eau est représentée par un volume qui varie au cours du temps. La Figure 2 illustre les sorties du modèle avec un pas de temps journalier sur trois mois en simulant aléatoirement des épisodes pluvieux.

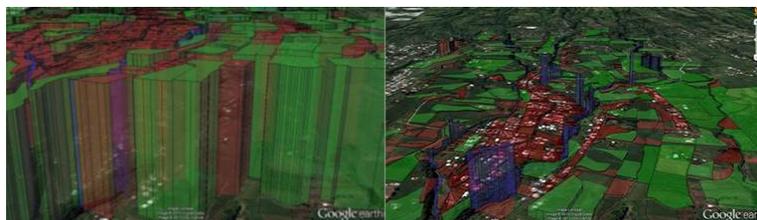


FIG. 2 – *Sorties du modèle au milieu d'un épisode pluvieux et trois jours après l'épisode pluvieux.*

## 4 Conclusion

Nous avons proposé une extension du langage de modélisation Ocelet qui rend possible la manipulation conjointe de représentations vecteur ou raster des données géographiques dans les modèles de dynamiques spatiales et paysagères. Ces développements ont impliqué l'intégration d'un nouveau concept de représentation duale (vecteur ou raster) d'une même entité géographique. Nous avons expérimenté cette extension du langage sur un modèle de ruissellement, en travaillant en mode vecteur à une échelle parcellaire et en mode raster à une échelle plus fine. Ces travaux ouvrent la voie à une meilleure intégration vecteur / raster dans les tra-

vaux de modélisation de dynamiques spatiales et paysagères. Nous envisageons d'expérimenter plus avant les applications de forçage de modèle à l'aide de séries temporelles d'images de télédétection.

## 5 Remerciements

Cette recherche est en partie financée par l'Agence Nationale de la Recherche à travers le projet DESCARTES (ANR 11-AGRO-002-01 Descartes <http://www.projet-descartes.fr/>). Ce travail a aussi bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'avenir pour le projet GEOSUD portant la référence ANR-10-EQPX-20.

## Références

- Augusseau, X., F. Aulanier, A. Bégué, J. Bourgoïn, D. David, G. David, P. Degenne, M. Fargette, E. Lagabrielle, G. Lajoie, J. Laurent, V. Lebourgeois, G. Lestrelin, D. Lo Seen, M. Loireau, E. Tillard, P. Todoroff, J. Tonneau, et F. Vitry (2013). Simulation cartographique au service de l'innovation collective - géoprospective territoriale à l'île de la réunion. In *Spatial Analysis and GEomatics conference (SAGEO), Brest (FR)*.
- Curé, O., R. Forax, P. Degenne, D. Parigot, et A. Ait Lahcen (2010). Ocelet : An ontology-based domain specific language to model complex domains. In *The First International Conference on Models and Ontology-based Design of Protocols, Architectures and Services - MOPAS, June 13-19, Athens, Greece*.
- Degenne, P. (2012). *Une approche générique de la modélisation spatiale et temporelle : application à la modélisation des paysages*. Ph. D. thesis, Université Paris-Est.
- Degenne, P., A. Ait Lahcen, O. Curé, R. Forax, D. Parigot, et D. Lo Seen (2010). Modelling the environment using graphs with behaviour : do you speak ocelet ? In *International Congress on Environmental Modelling and Software, July 5-8, Ottawa, Ontario, Canada*.

## Summary

In this paper, we focus on modeling expert knowledge for simulating complex landscape spatial dynamics. One modeling tool to do that is the Ocelet modeling language that uses interaction graphs to describe spatial dynamics. Most present approaches impose an a priori choice of spatial format between: (i) a vector format representing the shapes of the entities, or (ii) a gridding of space into regular elements (raster). In this paper we show how Ocelet was extended to support the interaction semantics between these two spatial formats (vector and raster). As case study, we present a runoff model in a tropical insular environment.



## Annexe D

# Integrating raster and vector spatial representations with interaction graphs for multi-scale environmental simulations

# Integrating raster and vector spatial representations with interaction graphs for multi-scale environmental simulations

**Mathieu Castets<sup>a</sup>, Pascal Degenne<sup>a</sup>, Pascal Poncelet<sup>b</sup>, and Danny Lo Seen<sup>a</sup>**

<sup>a</sup>*CIRAD - UMR TETIS, Montpellier, France (mathieu.castets@cirad.fr, pascal.degenne@cirad.fr, danny.lo\_seen@cirad.fr)*

<sup>b</sup>*LIRMM, Montpellier, France (pascal.poncelet@lirmm.fr)*

**Abstract:** Nowadays, developing tools for modelling complex system is a mandatory issue in environmental research. Environmental modelling implies taking into account processes at different scales of space and time. Ocelet is a domain specific environmental modelling language that uses graphs to describe how spatial entities in the system interact, with functions attached to the arcs of the graphs that are activated within scenarios to represent the dynamics of the system. The language has a strong focus on spatial interactions where the vector format is used for attributing geometries to entities, thus allowing a precise description of their shapes. However, the vector format is not appropriate for representing spatially continuous information for which the grid format is more efficient. In this paper, we describe how the grid format was formalized in Ocelet to allow a joint use with the vector format and interaction graphs, together with the key concepts and constraints for the combination of the two spatial formats. In this formalization, entities can be cells of different predefined shapes (square, beehive) and sizes. Processes can then be described with relations between cells that are handled as proper entities with their own properties as defined by the modeller. An example is given to illustrate this combination with a case study on water runoff modelling in a tropical insular environment subject to urban sprawl and soil sealing.

**Keywords:** Spatial dynamics, Domain specific language, Interaction graph, Raster

## 1 INTRODUCTION

The importance of a knowledge-driven approach to modelling in environmental studies has been highlighted in Villa et al. [2009]. Processes at different scales of space and time has to be taken into account in an integrated way including social, economic and environmental considerations (Laniak et al. [2013]). Most spatial modelling and simulation tools are however designed around one preferred internal representation of space: they either discretize continuous spatial information into a grid of regular cells, as in Netlogo, Tissue and Wilensky [2004] or they work with a set of vector geometries representing the shape of well delimited objects as shown in Gaucherel et al. [2006]. In many environmental modelling situations though, one could benefit from being capable of manipulating both forms in a seamlessly integrated modelling design.

The advantages of using a Domain Specific Language (DSL) when expressing environmental knowledge in a model are described in Athanasiadis and Villa [2013]. Ocelet is a DSL dedicated to spatio-temporal modelling and simulation that was designed independently of one specific internal representation of space (Degenne et al. [2009]). When modelling with Ocelet one manipulates explicit interaction graphs to represent relationships between the entities of a model. The different kinds of interactions present in a model are all described with one same formalism (interaction graphs), whether

they express spatial, functional, hierarchical or social relationships. To be able to relate such interaction graphs with geographical information, the Ocelet language already provide appropriate features to import and export vector data formats, to store vector geometry attributes in entities and to build spatial relationship graphs from that information.

In this paper, we describe how the grid format was formalized and integrated into Ocelet to allow a joint use with the vector format and interaction graphs, together with the key concepts and constraints for the combination of the two spatial formats. The key concepts at the heart of the Ocelet language are presented first. Then we show how these concepts were used to integrate raster based geographical information into the Ocelet runtime. Finally, the new feature is experimented on a case application of water runoff modelling in a tropical insular environment that requires both vector and raster spatial representations of multi-scale processes.

## **2 OCELET CONCEPTS AND RASTER INTEGRATION METHODS**

### **2.1 Ocelet**

Ocelet is a DSL designed to write spatially explicit models and to perform spatial dynamics simulations as shown in Degenne et al. [2010]. The language is meant to ease working with geographical data and information, and is kept generic enough to offer a wide range of possible applications when studying urban sprawl, land use change, coastal dynamics or species distribution in space and time.

In a rather classical approach, Ocelet is based on concepts such as (i) Entities that contain Properties reflecting their state, (ii) Relations that allow to describe what types of entities are expected to interact with each other and what exactly happen when they do interact and (iii) Scenarios where the sequence of operations and interactions are specified in order to simulate the evolution of the system. A fourth concept named Datafacer is provided to help reading and writing geographical or non-geographical data from different storage formats (such as Shapefile, PostGis database, CSV or KML files).

More original is the way spatial and non-spatial relationships are formalized in Ocelet. Explicit interaction graphs are used in all cases. An interaction graph is an instance of a Relation. It means for example that one can define a Relation between a Parcel entity and a Farmer entity, and the Relation can be instantiated as a bipartite graph where the vertices are either individual parcels or individual farmers, and the edges would link some parcels with some farmers. Another Relation could for example be defined between Parcel entities to create a spatial neighbourhood graph between parcels.

Within one relation definition, one can specify a series of interaction functions describing how the entities at both ends of an edge have their properties affected when those entities interact with each other. When one such interaction function is called during a simulation, it is applied simultaneously on all the edges of a graph. It is however possible to reduce the number of edges involved by defining and using graph filters.

Two more details must be given to enlighten the explanations of the next section:

- The entities can have properties of different common types like Integer, Boolean or String but Ocelet provides also vector geometry types such as Point, Line, Polygon, MultiPoint etc.
- When a Datafacer is used to read data from a file or a database, a series of individual entities are obtained with their properties already initialized from the data source. Those entities can then easily be used in interaction graphs when needed.

## 2.2 Raster integration

Studies on duality in spatial representation from grid or discrete objects are not recent, Couclelis [1992]. Previous studies present a conceptual approach leading to link these two formats, Goodchild et al. [2007]. These studies are more about the representation of space and time with geographical information system than for using them in simulations. However, these studies have suggested that the choice of a format often depends on the spatial scale. A related work concerned the reduction of scale sensitivity using vector data and a cellular automata as shown in Moreno et al. [2009]. Conceptually, two main aspects were taken into consideration when integrating raster (or regular grids) into the Ocelet language:

1. We want to be able to access every grid cell (or pixel of a raster image) and to relate these cells with properties inside entities of a model. We also need to be able to extract all cells within the boundaries of a Polygon or following a linear geometry so that we can link one geometry to a group of cells located in the same area.
2. A regular grid of triangular, square or hexagonal cells can be seen as a neighbourhood graph where the edges are implicitly deduced from the relative position of the cells. We then need to be able to define formal Relations between cell entities and write interaction functions completely consistent with the already existing interaction graphs.

More concretely that integration was realized in several steps:

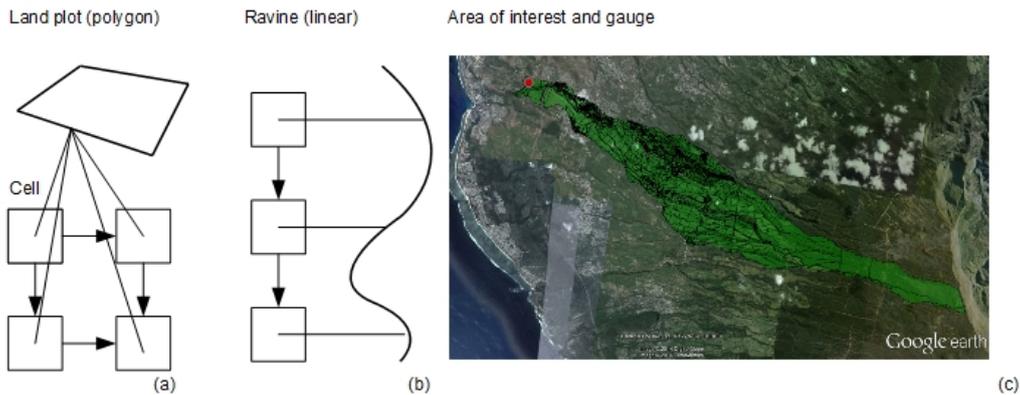
- A new Cell type was added to the language that allows definition of property of type Cell in entities. A Cell is characterized by its location (x and y), a shape (square, hexagonal, triangular) and a size.
- A new type of Datafacer was developed to allow access to raster image data (like remote sensing images). Modellers have to define an entity with a Cell property as defined before. This is only a spatial representation; the values from different bands of the image are attributed to entity properties via the datafacer by matching the property to a band value. The datafacer automatically creates entities from the user definition and assign the corresponding property values. The entities are then ready to be used in the model and come associated in an interaction graph.
- New functions were added to the runtime to give access to all cells located in the same area of a point, line,(using the Bresenham [1965] algorithm) or surface geometry (using the scan line algorithm).
- The construction of two types of interaction graphs was provided. The first type relates entities with a vector geometry to entities with Cell geometries. Ocelet can then automatically create the edges of such graphs. The second type of graphs relates entities with Cell geometries with each other. In that case, the edges of the graph are not explicitly created because they can be deduced on demand due to the regular topology of the grid. Internally an iterator on the edges is provided that can be filtered like any other graph made with Ocelet. That second type of graph would for example allow to easily model cellular automata using Ocelet.

## 3 RESULTS AND DISCUSSIONS

### 3.1 Case application

We test the new feature on a test case that requires both vector and raster spatial representations of multi-scale processes. In a wider context of a project where a cartographic simulation tool is being developed for the agro-ecological evaluation of land use scenarios in a space constrained insular environment, we focus on the effect of soil sealing resulting from urban sprawl on runoff during cyclonic

events. Reunion Island is located in south-western part of the Indian Ocean. It has a surface area of around 2500 km<sup>2</sup> and lies about 800 km to the east of Madagascar. The Saint Gilles catchment area is located in the drier north-western part of the island with an annual rainfall of 2 m. Slopes are high, with more than 2000 m altitude difference over a distance of about 18 km. Deep gorges called ravines lie along the catchment to collect water during rainfall events, but remain dry most of the time. Planned and unplanned urbanization (sprawl) results in soil sealing that accelerate water into the ravines, reducing concentration time and thus increasing risks of floods and landslides downstream. This process is often ignored in runoff models such as USACE-HEC [2008] whereas it is quite important in the present setup.



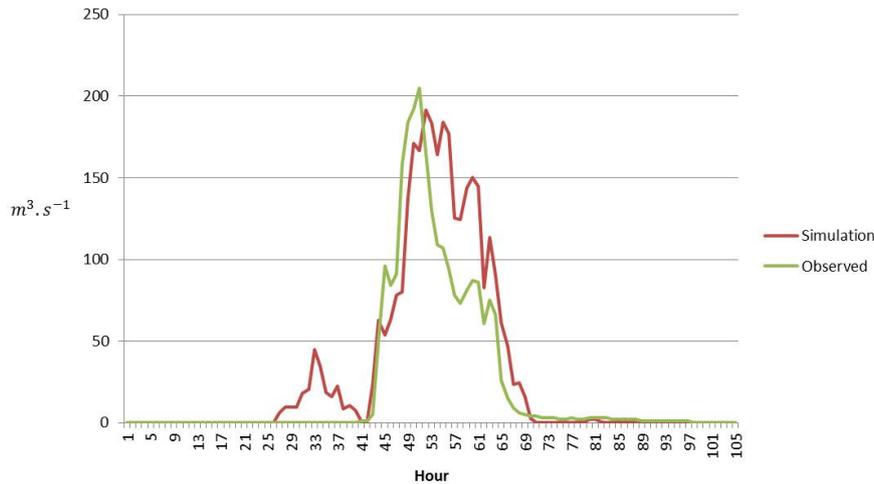
**Figure 1.** (a) Relation in cells for water flows and access to entity property (land cover) with a geometry. (b) Linear entity affected by water quantities contained in attached cells. (c) Geometrical view of a landscape and the gauge (red) where water flows are measured and computed.

We use Ocelet to model runoff in the Saint Gilles catchment where urban areas account to about 10 % of the area. In particular, we try to reproduce how water is collected around houses and transferred quickly to the ravines through the road network. We use the vector format to represent land parcels (polygons), the road network (line segments) and the ravine / river network (line segments). A 10 m resolution raster DEM gives the slope and the direction of flow within parcels and along the networks. We also use another raster (Spot 5 06/07/2002) to better estimate vegetated areas where infiltration is higher. The model describes how water is transferred between adjacent cells within parcels and into line segments of road and ravines networks. The basic raster spatial unit is the DEM cell. A simple water budget is computed for each cell with rainfall and output of adjacent cells as input, and infiltration and flows to adjacent cells as output. Infiltration depends on land cover, and is nil for artificial surfaces. Lateral flows depend on altitude differences. Cells and parcels in contact with the road and ravine will output their water into the network instead of nearby cells. The time step is 1 min and the simulation period of the Dina (2002) cyclonic event is of 4 days.

### 3.2 Discussion

Figure 2 shows that comparison between simulated water flows at Saint Gilles with measurements made in this area during the Dina cyclone in 2002. The order of magnitude of water flow is well reproduced, however, a first peak appear in the first steps of the simulation suggesting that the infiltration function is too simple, and could have been replaced by one with a threshold on water input (when water input is low, infiltration remains high until a saturation threshold is reached, after which runoff is predominant). The same effect may explain why the decrease after the peak value is not more rapid.

This modelling experiment shows that the interaction graph formalism is capable of combining raster and vector spatial representations, not only for display purposes but more so for modelling processes



**Figure 2.** Restitution of the water flood and comparison between the observed and simulated values

at different spatial scales. Vector and raster model entities have been made to interact (here by exchanging water) when modelling a process which would otherwise be too difficult or impractical to model with either vector or raster representations alone, hence combining multiple processes in a system. This spatial data integration follows the semantics of the Ocelet's concepts in order to describe natural processes without being limited by spatial representation.

The simulated water flow at the outlet of the catchment does not fit well with the hourly observations, although the correct order of magnitude is obtained. Many reasons can be put forward to explain the discrepancies, including the spatial resolution of input data (one rain gauge for the whole catchment), the lack of appropriate parameterization and above all the simplicity of the transfer and infiltration functions used. However, the objective of the study was more to test a formalism combining raster and vector spatial representations than to build a new runoff model for cyclone events in steep catchments as found in Reunion Island. Having said that, we find that the results are encouraging enough either i) to use the multi-scale modelling, which is more relevant mainly on urbanized areas, to better parameterize existing models to better take into account soil sealing on runoff or ii) to develop a complete runoff model with the new formalism by integrating necessary processes and related functions appropriate for these types of catchments and rainfall regimes.

#### 4 CONCLUSION

The Ocelet DSL was designed along the idea that it may be possible to provide a spatio-temporal modelling and simulation tool that would not impose a preferred form of representation of space. We showed how it was possible to improve the first version of Ocelet by seamlessly integrating raster and grid data forms into the language and its runtime, allowing mixing vector and raster in one same set of coherent concepts. The experimental case application showed encouraging results especially in demonstrating that this raster integration into Ocelet allows to create models that use both vector and raster data representing spatial dynamics at two different scales. Perspectives on this work concern the optimisation in case of big data raster images and graph synchronization for multiple processes. Moreover we planned to integrate time series with such formalism in order to force or help model processes.

## ACKNOWLEDGEMENTS

This work was funded by the French National Research agency (ANR) through the DESCARTES project (ANR 11-AGRO-002-01 Descartes <http://www.projet-descartes.fr/>) and GEOSUD, a project (ANR-10-EQPX-20) of the program 'Investissements d'Avenir' managed by the French National Research. Interesting discussions with J.L. Join, E. Lagabrielle, J. Roques, D. David, X. Augusseau and G. Lestrelin are gratefully acknowledged.

## REFERENCES

- Athanasiadis, I. N. and Villa, F. (2013). A roadmap to domain specific programming languages for environmental modeling: key requirements and concepts. *In Proceedings of the 2013 ACM workshop on Domain-specific modeling*, pages 27–32.
- Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4:25–30.
- Couclelis, H. (1992). People manipulate objects (but cultivate fields): Beyond the raster-vector debate in gis. in frank, a., u., campari, i. and formentini, u. (eds.). *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 639:65–77.
- Degenne, P., Ait Lahcen, A., Cure, O., Forax, R., Parigot, D., and Lo Seen, D. (2010). Modelling the environment using graphs with behaviour : do you speak ocelet ? *In International Congress on Environmental Modelling and Software (IEMSS)*.
- Degenne, P., Lo Seen, D., Parigot, D., Forax, R., Tran, A., Ait Lahcen, A., Cure, O., and Jeansoulin, R. (2009). Design of a domain specific language for modelling processes in landscapes. *Ecological Modelling*, 220:3527–3535.
- Gaucherel, C., Giboire, N., Viaud, V., Houet, T., Baudry, J., and Burel, F. (2006). A domain-specific language for patchy landscape modelling: The brittany agricultural mosaic as a case study. *A domain-specific language for patchy landscape modelling: The brittany agricultural mosaic as a case study*, 194:233–243.
- Goodchild, M. F., Yuan, M., and Cova, T. J. (2007). Towards a general theory of geographic representation in gis. *International Journal of Geographical Information Science*, 21:239–260.
- Laniak, G. F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., and Hughes, A. (2013). Integrated environmental modeling: A vision and roadmap for the future. *Environmental Modelling and Software*, 39:3–23.
- Moreno, N., Wang, F., and Marceau, J. D. (2009). Implementation of a dynamic neighborhood in a land-use vector-based cellular automata model. *Computers, Environment and Urban Systems*, 33:44–54.
- Tisue, D. and Wilensky, U. (2004). Netlogo : A simple environment for modeling complexity. *In International Conference on Complex Systems (ICCS)*.
- USACE-HEC (2008). *Hydrologic modeling system HEC-HMS v3.2 users manual*. US Army Corps of Engineers, Hydrologic Engineering Center (HEC). Davis.
- Villa, F., Athanasiadis, I. N., and Rizzoli, A. E. (2009). Modelling with knowledge: A review of emerging semantic approaches to environmental modelling. *Environmental Modelling & Software*, 24 (5):577–587.





## Résumé

La modélisation et la simulation de dynamiques spatiales, en particulier pour l'étude de l'évolution de paysages ou de problématiques environnementales pose la question de l'intégration des différentes formes de représentation de l'espace au sein d'un même modèle. Ocelet est une approche de modélisation de dynamiques spatiales basée sur le concept original de graphe d'interaction. Le graphe porte à la fois la structure d'une relation entre entités d'un modèle et la sémantique décrivant son évolution. Les relations entre entités spatiales sont ici traduites en graphes d'interactions et ce sont ces graphes que l'on fait évoluer lors d'une simulation. Les concepts à la base d'Ocelet peuvent potentiellement manipuler les deux formes de représentation spatiale connues, celle aux contours définis (format vecteur) ou la discrétisation en grille régulière (format raster). Le format vecteur est déjà intégré dans la première version d'Ocelet. L'intégration du format raster et la combinaison des deux restaient à étudier et à réaliser. L'objectif de la thèse est d'abord étudier les problématiques liées à l'intégration des champs continus et leur représentation discrétisée en pavage régulier, à la fois dans le langage Ocelet et dans les concepts sur lesquels il repose. Il a fallu notamment prendre en compte les aspects dynamiques de cette intégration, et d'étudier les transitions entre données géographiques de différentes formes et graphe d'interactions à l'aide de concepts formalisés. Il s'est agi ensuite de réaliser l'implémentation de ces concepts dans la plateforme de modélisation Ocelet, en adaptant à la fois son compilateur et son moteur d'exécution. Enfin, ces nouveaux concepts et outils ont été mis à l'épreuve dans trois cas d'application très différents : deux modèles sur l'île de la Réunion, le premier simulant le ruissellement dans le bassin versant de la Ravine Saint Gilles s'écoulant vers la Côte Ouest de l'île, l'autre simulant la diffusion de plantes invasives dans les plaines des hauts à l'intérieur du Parc National de La Réunion. Le dernier cas décrit la spatialisation d'un modèle de culture et est appliqué ici pour simuler les rendements de cultures céréalières sur l'ensemble de l'Afrique de l'Ouest, dans le contexte d'un système d'alerte précoce de suivi des cultures à l'échelle régionale.

## Summary

The modelling and simulation of spatial dynamics, particularly for studying landscape changes or environmental issues, raises the question of integrating different forms of spatial representation within the same model. Ocelet is an approach for modelling spatial dynamics based on the original concept of interaction graph. Such a graph holds both the structure of a relation between entities of a model and the semantics describing its evolution. The relationships between spatial entities are here translated into interaction graphs and these graphs are made to evolve during a simulation. The concepts on which Ocelet is based can potentially handle two known forms of spatial representation : shapes with contours (vector format) or regular grid cells (raster). The vector format is already integrated in the first version of Ocelet. The integration of raster and the combination of the two remained to be studied and carried out. The aim of the thesis is to first study the issues related to the integration of continuous fields and their representation by regular tiling, both in the Ocelet language and the concepts on which it is based. The dynamic aspects of this integration had to be taken into account and transitions between different forms of geographic data and interaction graphs had to be studied in the light of the concepts formalized. The concepts were then implemented in the Ocelet modelling platform, with the adaptation of both its compiler and runtime. Finally, these new concepts and tools were tested in three very different cases : two models on Reunion Island, the first simulating runoff in Ravine Saint Gilles watershed in the West Coast of the island, the other simulating the spread of invasive plants in the high plains inside the Reunion National Park. The last case describes the spatialisation of a crop model and is applied here to simulate the cereal crop yields in West Africa, in the context of an early warning system for regional crop monitoring.

