



**HAL**  
open science

# Interactions entre niveaux dans un modèle orienté agent de généralisation cartographique: Le modèle DIOGEN

Adrien Maudet

## ► To cite this version:

Adrien Maudet. Interactions entre niveaux dans un modèle orienté agent de généralisation cartographique: Le modèle DIOGEN. Informatique et langage [cs.CL]. Université Paris-Est, 2016. Français. NNT: 2016PESC1055 . tel-01539595

**HAL Id: tel-01539595**

**<https://theses.hal.science/tel-01539595v1>**

Submitted on 15 Jun 2017

**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 de l'Université Paris-Est

Spécialité Informatique

Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information et  
de la Communication

Adrien Maudet

# Interactions multi-niveaux dans un modèle de généralisation cartographique

## Le modèle DIOGEN

Soutenue publiquement le 10 novembre 2016 devant le jury composé de :

Pr. Vincent Chevrier	Rapporteur
Dr. Cécile Duchêne	Directrice
Pr. Bernard Moulin	Examineur
Dr. Sébastien Picault	Directeur
Dr. Anne Ruas	Examinatrice
Pr. Nicolas Sabouret	Rapporteur
Dr. Guillaume Touya	Encadrant



# Remerciements

---

Je remercie Guillaume Touya, Cécile Duchêne et Sébastien Picault pour leur encadrement.

Je remercie également Jules Teulade-Denantes, Olivia Gautrais et Alban Kraus.

Je remercie Vincent Chevrier et Nicolas Sabouret d'avoir accepté de rapporter mon mémoire, et Bernard Moulin et Anne Ruas d'avoir accepté de rejoindre le jury de thèse.

Je remercie les membres de l'équipe COGIT pour m'avoir accueilli, pour leurs conseils et leur aide.

Je remercie les personnes de l'IGN pour leurs conseils, aides et retours quant aux travaux réalisés au cours de ma thèse, notamment Anne Bérard, Didier Blaquière, Bruno Bordin, Arnaud Braun, Frédéric Cantat, Régine Cochet, Blandine Delavault, Xavier Halbecq, Sylvain Lafay, François Lecordix, Emmanuel Maugeais, Michael Michaud, Nicolas Papparoditis et Christophe Pécot.

Je remercie les membres de l'équipe SMAC de l'UMR CRISTAL et de l'IUT A de Lille de m'avoir accueilli pendant ma dernière année.

Je remercie aussi Didier Laronce, Véronique Rousselle et Sylvie Ponnou-Delaffon pour l'impression du mémoire.

Je remercie également Shawn Larouche de l'Université Laval, ainsi que François Algot, Odile Dieda, Olivier Gueguen et Sylvie Lige de l'ENSG pour l'organisation de la soutenance.

Je remercie Sylvie Cach de l'École Doctorale de Paris-Est pour la gestion des aspects administratifs de la thèse.

Je remercie enfin ma maman Claudine pour ses relectures orthographiques plus que nécessaires.



# Table des matières

---

Remerciements .....	3
Table des matières .....	5
Introduction.....	9
Genèse de la thèse .....	9
Plan du mémoire .....	9
Chapitre I :    La généralisation cartographique.....	11
I.1 La généralisation cartographique.....	11
I.1.1 La généralisation cartographique, définition générale.....	11
I.1.2 L'automatisation de la généralisation.....	13
I.1.3 Séquencement des opérateurs à appliquer .....	15
I.1.4 Problème de l'orchestration des algorithmes.....	16
I.2 Les systèmes multi-agents pour la généralisation .....	17
I.2.1 Généralités sur les Systèmes Multi-Agents.....	17
I.2.2 Les SMA pour la généralisation .....	18
I.3 Présentation des différents modèles .....	19
I.3.1 Le modèle SIGMA .....	19
I.3.2 Le modèle AGENT .....	20
I.3.3 Généralisation d'isobathes.....	22
I.3.4 Le modèle CartACom.....	23
I.3.5 Le modèle GAEL.....	24
I.3.6 Structures réactionnelles .....	26
I.3.7 Conflits persistants .....	27
I.3.8 Utilisation de SMA pour la généralisation de cartes à la volée .....	28
I.4 Problématique.....	29
I.4.1 Utilisation combinée de modèles.....	29
I.4.2 Expression des connaissances selon un paradigme commun .....	30
I.4.3 Approche proposée .....	31
Chapitre II :    Étude des interactions multi-niveaux.....	33
II.1 Définitions .....	33

---

II.2 La problématique multi-niveaux en généralisation .....	34
II.2.1 L'organisation multi-niveaux de l'information géographique .....	34
II.2.2 Généralisation et multi-niveaux.....	35
II.2.3 Les aspects multi-niveaux dans les SMA de généralisation existants.....	39
II.2.4 Les aspects multi-niveaux non gérés .....	40
II.2.4 Synthèse .....	42
II.3 : Les SMA multi-niveaux : vue d'ensemble .....	44
II.3.1 Modèles multi-agents multi-niveaux .....	44
II.3.2 Modèles formalisés pour les SMA multi-niveaux .....	48
II.3.3 Synthèses des caractéristiques des modèles multi-niveaux.....	50
II.4 Caractéristiques de PADAWAN et adéquation pour nos modèles .....	51
II.4.1 Le modèle PADAWAN .....	51
II.4.2 Différences entre simulation et résolution de problèmes.....	53
II.4.3 Adaptations nécessaires.....	54
II.5 Synthèse du chapitre.....	55
Chapitre III : DIOGEN, un modèle multi-agents multi-niveaux pour la généralisation.....	57
III.1 Description du modèle DIOGEN .....	58
III.1.1 Traduction dans le formalisme PADAWAN : interaction, agents, primitives .....	58
III.1.2 Expression des contraintes, inversion de la perspective .....	62
III.1.3 Adaptation du cycle de vie .....	65
III.1.4 Ordonnancement de l'activation des agents.....	66
III.2 Implémentation.....	68
III.3 Reproduction de l'existant .....	68
III.3.1 Reproduction du comportement d'AGENT .....	68
III.3.2 Reproduction du comportement de CartACom.....	70
III.3.3 Intégration d'éléments du modèle GAEL .....	72
III.4 Résolutions de cas nouveaux .....	73
III.4.1 Intégration de comportements empruntés à la fois à AGENT et CartACom .....	74
III.4.2 Définition d'un espace embarqué pour le voisinage .....	75
III.4.3 Référencement multiple d'un agent dans plusieurs environnements .....	78
III.5 Synthèse du chapitre.....	80
Chapitre IV : Cas d'étude : cartes d'itinéraires de randonnées.....	81
IV.1 Les cartes d'itinéraires .....	81

---

IV.1.1 Définition des cartes d'itinéraires.....	82
IV.1.2 Les problématiques du positionnement automatique d'itinéraires .....	85
IV.1.3 Contexte de travail : les données utilisées .....	87
IV.1.4 Identification des zones impactées par la présence d'itinéraires .....	89
IV.2 Optimisation du positionnement d'itinéraires sur un réseau support .....	90
IV.2.1 Approches issues des cartes schématiques.....	90
IV.2.2 Le concept de « Stroke » d'itinéraire .....	91
IV.2.3 Placement des itinéraires par optimisation de contraintes .....	93
IV.3 Interactions entre agents pour la réalisation de cartes de randonnées .....	97
IV.3.1 Interactions d'un itinéraire dans un environnement de route support.....	98
IV.3.2 Interaction des itinéraires avec des objets compacts .....	98
IV.3.3 Interaction des itinéraires avec un objet déformable .....	101
IV.4 Résultats et analyse .....	105
IV.4.1 Déplacement de compacts .....	105
IV.4.2 Déplacement et déformation de déformables .....	106
IV.5 Synthèse du chapitre .....	106
Chapitre V : Vers des <i>patterns</i> multi-niveaux.....	109
V.1 Motivations à l'identification de <i>patterns</i> .....	109
V.2 Cadre formel .....	111
V.3 Présentation des patterns identifiés.....	112
V.3.1 <i>Pattern</i> Agrégation.....	113
V.3.2 <i>Pattern</i> Décomposition .....	114
V.3.3 <i>Pattern</i> Hiérarchisation.....	115
V.3.4 <i>Pattern</i> Frontière.....	116
V.3.5 <i>Pattern</i> Porte.....	117
V.3.6 <i>Pattern</i> Embarquement .....	118
V.3.7 <i>Patterns</i> Meso .....	119
V.3.8 <i>Pattern</i> Ordonnanceur .....	120
V.4 Discussion autour des <i>patterns</i> .....	120
V.5 Synthèse du chapitre .....	122
Conclusion et perspectives.....	123
Rappel des objectifs .....	123
Résumé des contributions.....	123



## Table des matières

---

Perspectives .....	125
Perspectives immédiates .....	125
Perspectives cartographiques .....	126
Perspectives en résolution de problèmes contraints .....	126
Perspectives dans le domaine des SMA.....	127
Bibliographie .....	129

# Introduction

---

## Genèse de la thèse

Le sujet de cette thèse de doctorat est issu de la rencontre entre deux domaines de recherche : d'une part, celui de l'automatisation de la généralisation cartographique, s'appuyant sur des modélisations multi-agents, et d'autre part celui des simulations multi-agents multi-niveaux. La généralisation cartographique est un processus qui par simplification et caricature permet d'obtenir une carte lisible au bon niveau de détail pour une échelle et un besoin donnés. Son automatisation est un des domaines de recherche historiques du laboratoire COGIT (Cartographie et Géomatique) de l'IGN (Institut National de l'Information Géographique et Forestière), où plusieurs modèles multi-agents ont été proposés. La conception d'une carte par généralisation est toujours un problème de recherche ouvert, en raison, notamment, de la démocratisation de la cartographie et de la saisie de données. Un des questionnements récurrents que soulève la conception de ces modèles de généralisation automatique est la structuration en niveaux de l'espace cartographique (par exemple, une ville qui peut être divisée en îlots urbains, eux-mêmes composés de bâtiments).

L'étude de l'aspect multi-niveaux des systèmes multi-agents est un des domaines de recherche de l'équipe SMAC (Systèmes Multi-Agents et Comportements) de l'UMR CRISTAL (Lille). Les travaux de recherche dans ce domaine ont notamment conduit cette équipe à proposer un modèle générique pour concevoir des simulations multi-agents multi-niveaux, PADAWAN. Les questionnements mis en avant dans l'élaboration d'un tel modèle font écho à ceux de l'automatisation de la généralisation par système multi-agents.

Lors d'une rencontre aux Journées Francophones sur les Systèmes Multi-Agents 2010 (JFSMA 2010), l'intuition que ces deux axes de recherche s'enrichiraient l'un et l'autre a été émise : le formalisme d'un modèle comme PADAWAN pourrait permettre d'unifier les modèles existants pour la généralisation et d'aller plus loin dans les possibilités proposées par ces modèles. Cette intuition a conduit à deux stages de recherche. Les premiers résultats étant encourageants, il a été décidé de continuer l'étude des interactions des agents de niveaux différents pour la généralisation dans le cadre d'un travail de thèse de doctorat qui fait l'objet du présent mémoire.

## Plan du mémoire

Dans le premier chapitre, nous présentons dans un premier temps le contexte dans lequel s'inscrit le travail de thèse, à savoir la généralisation cartographique et son automatisation. Ensuite, nous mettons plus particulièrement l'accent sur les solutions d'automatisation précédemment proposées impliquant l'utilisation de modèles multi-agents. Enfin nous expo-

sons la problématique que nous avons essayé de résoudre, à savoir étudier comment la résolution de problèmes de généralisation automatisée et l'unification des modèles existants est facilitée par la prise en compte des différents niveaux d'objets géographiques de façon formalisée dans le cadre d'une solution multi-agents.

Dans le chapitre II, nous nous intéressons aux aspects multi-niveaux dans le cadre des systèmes-multi-agents. Dans un premier temps, nous expliquons en quoi le problème de la généralisation cartographique peut être vu comme un problème multi-niveaux. Ensuite, nous discutons de la problématique multi-niveaux dans d'autres contextes. Si cette problématique est rarement abordée dans le cadre de la résolution de problèmes contraints, elle se pose de plus en plus souvent dans le domaine de la simulation multi-agents, où diverses solutions et approches ont été proposées. Ensuite, nous nous intéressons au modèle PADAWAN, en mettant l'accent sur les caractéristiques de ce modèle qui nous intéressent et qui sont en adéquation avec nos besoins pour la généralisation.

Le chapitre III expose le modèle DIOGEN que nous proposons pour considérer plusieurs niveaux dans un processus de généralisation automatisé. Ce modèle s'appuie sur le paradigme PADAWAN et plusieurs adaptations et ajouts ont été nécessaires afin de répondre aux besoins spécifiques de la généralisation cartographique, et de façon plus générale à un problème contraint. D'abord nous présentons le socle de notre modèle que constituent l'ensemble de ces adaptations. Ensuite, nous exposons comment les caractéristiques de certains modèles pour la généralisation ont été intégrées dans notre proposition de modèle. Enfin, nous décrivons les expérimentations qui ont été effectuées afin de valider notre modèle, d'une part en reproduisant les résultats des modèles adaptés, et d'autre part en proposant de nouveaux mécanismes pour résoudre certains cas de généralisation impliquant des interactions multi-niveaux.

Dans le chapitre IV, nous présentons une étude de cas mettant en œuvre notre modèle. La génération de cartes d'itinéraires de randonnées, avec représentation des différents itinéraires avec des symboles différents (à la façon des plans de bus) est un cas d'application pour lequel la généralisation est particulièrement difficile à automatiser. Nous exposons l'ensemble des relations multi-niveaux à modéliser pour dériver automatiquement ce type de carte, ainsi que la spécialisation pour ce cas, et sa mise en œuvre.

Nos recherches ont conduit à l'identification de plusieurs structures-types propres aux relations entre agents de différents niveaux. Dans le chapitre V, nous proposons de mettre en évidence certaines relations propres à la généralisation cartographique, et de les formaliser sous forme de *patterns* d'analyse afin d'ouvrir leur utilisation à d'autres situations, notamment dans le cadre de la simulation multi-niveaux.

# Chapitre I : La généralisation cartographique

---

Dans ce chapitre, nous présentons la généralisation cartographique et les enjeux soulevés par son automatisation depuis l'avènement de l'information géographique. L'automatisation de la généralisation, que nous présentons après la généralisation en elle-même, constitue une problématique impliquant l'élaboration d'algorithmes et de modélisations complexes. Certaines de ces modélisations s'appuient sur des systèmes multi-agents. Après avoir présenté le principe des modèles multi-agents de façon générale, nous proposons un état de l'art des solutions multi-agents pour la généralisation. Nous expliquons enfin la problématique que nous explorerons dans le reste du mémoire.

## I.1 La généralisation cartographique

### I.1.1 La généralisation cartographique, définition générale

La création de cartes est une activité complexe. Naïvement, nous pourrions décrire l'objectif de la création d'une carte comme celui de représenter sur une surface restreinte un ensemble d'informations portant sur un espace géographique donné. Comme toute représentation de la réalité, la cartographie est l'application de différents processus d'analyse intellectuel. Ces processus sont la **symbolisation**, qui consiste à représenter l'information en respectant un ensemble de codes préalablement définis, la **généralisation** et le **placement de toponymes**. Ce processus de généralisation a pour objectif d'une part de **simplifier** l'information, étape nécessaire à partir du moment où il est souhaité de mettre l'information d'une certaine taille (celle de la réalité) sur un support de taille moindre (celle de la carte), d'**abstraire** cette information, afin de la communiquer selon un langage compréhensible par l'utilisateur et de **rendre lisible** la carte.

Le fait de devoir exprimer le contenu d'un espace sur une carte de plus petite taille renvoie à la notion d'**échelle**, qui exprime le ratio entre la réalité et la représentation cartographique. Par exemple, si une distance de 1 km doit être représentée par 1 cm, l'échelle sera de 1 : 100 000. Plus la taille de représentation sera petite, plus l'échelle sera elle aussi petite. Plus l'échelle sera petite, moins il sera possible de représenter de détails, d'où la nécessaire opération de simplification de la généralisation. Cette simplification de l'information nécessite une abstraction, notamment en caricaturant certaines caractéristiques de l'espace géographique. Par ailleurs, la symbolisation, qui consiste à associer aux objets des symboles se-

Une certaine sémantique, implique que l'objet, sur la carte, peut être représenté par une forme différente de celle qu'il avait initialement. Cela entraîne qu'un même objet géographique du monde réel pourra être représenté différemment selon l'échelle choisie. Par exemple, la Figure 1 montre deux extraits de deux différentes cartes d'une même zone mais d'échelle différente. On remarque plusieurs différences, par exemple :

- l'épaisseur des symboles des routes et leur tracé moins détaillé dans la carte au 1 : 100 000 ;
- l'absence des bâtiments dans la carte au 1 : 100 000, remplacé par un aplat orange pour désigner la zone urbaine, ou par un symbole pour les bâtiments importants comme le château ou l'église ;
- la suppression de certaines routes.



Figure 1 Deux extraits de carte IGN : à gauche, carte conçue pour l'échelle 1 : 25000 ; au milieu carte conçue pour l'échelle 1 : 100 000, avec une taille agrandie pour que la même zone soit représentée sur le papier sur une surface de même taille ; à droite, le même extrait aux proportions réelles par rapport à l'extrait de gauche.

Une partie du travail du cartographe a donc, depuis longtemps, consisté à résumer l'information géographique et l'adapter à des besoins utilisateurs spécifiques. Le choix des thèmes à représenter (par exemple le bâti, l'altimétrie, l'hydrographie), le choix des symboles, la définition des règles quant à l'information à conserver et à simplifier sont autant d'aspects du rôle du cartographe où les besoins des utilisateurs de la carte doivent être pris en compte. Au cours de l'histoire les cartographes ont développé un ensemble de techniques et de connaissances quant à l'élaboration des cartes. Ces techniques et connaissances n'ont que rarement fait l'objet de théorisation, et ce plutôt récemment au vu de l'histoire de la cartographie, par exemple par Bertin (1967) [7] ou Kolácny (1969) [53].

Afin de mieux saisir la notion de généralisation, nous allons caractériser la notion d'**information géographique** de façon plus précise, non pas pour définir un vocabulaire, mais pour cerner cette notion. Cette notion peut être perçue comme un ensemble d'éléments géographiques. Par élément géographique, nous entendons une entité ayant un sens dans la logique de la perception de l'espace des utilisateurs de la carte. Ainsi, un bâtiment, un tron-

çon de route, une ville sont des éléments géographiques. Une altitude donnée pour un point de l'espace aussi. De même l'ensemble des sections de routes empruntées par une ligne de bus forme aussi un élément géographique.

Une des premières actions d'un cartographe est d'identifier les éléments qui seront représentés sur la carte. Les objets géographiques sont regroupés par thème (par exemple les bâtiments dans le thème bâti, les tronçons de routes dans le thème routier). Le choix des thèmes à représenter est guidé par les besoins de l'utilisateur final de la carte, mais aussi par l'échelle. Les bâtiments, par exemple, sont plus rarement représentés sur les cartes à petite échelle (par exemple sur la Figure 1 au 1 : 100 000), car leur taille serait trop petite pour qu'ils soient lisibles par l'utilisateur de la carte.

La symbolisation attribuée aux objets géographiques des symboles afin d'informer l'utilisateur de leur nature. Dans la Figure 1, les routes sont représentées par des rubans de largeur et de couleurs différentes selon la catégorie de la route (par exemples autoroute, départementale, sentier). L'ajout de symboles implique une modification de l'occupation de l'espace par les objets. En fonction de la géométrie de l'objet, cet ajout de symboles peut le rendre illisible. Il est donc nécessaire de modifier la géométrie de l'objet pour le rendre lisible. De même certains objets, dont la présence sur la carte est requise par les spécifications, peuvent être trop petits pour être convenablement visibles. Le cartographe pourra choisir d'augmenter la taille des géométries des objets pour en faciliter la lisibilité (par exemple, l'église dans la Figure 1).

Enfin, certains détails concernant les géométries des objets (par exemple des détails sur la forme d'un bâtiment) sont illisibles à certaines échelles. Le cartographe préférera supprimer ces détails en simplifiant la géométrie des objets par exemple.

La cartographie couvre aussi d'autres questions, comme le positionnement de toponymes, qui est soumis aux mêmes contraintes d'exploitation de l'espace que la généralisation.

### **I.1.2 L'automatisation de la généralisation**

Comme expliqué dans la partie précédente, la généralisation consiste à simplifier et rendre lisible l'information géographique symbolisée représentée sur une carte. Cette opération était du ressort du cartographe qui effectuait ses choix de généralisation selon un savoir-faire et des connaissances propres à son métier. Comme pour de nombreux processus, l'avènement de l'informatique amène à s'interroger sur la façon dont il serait possible de l'automatiser, afin d'accélérer le rythme de production des cartes, ou de proposer la génération de cartes à la demande. La généralisation étant un processus prenant en entrée de l'information géographique, nous présentons en premier temps la façon dont l'information géographique est stockée sous forme de bases de données. Les bases de données géographiques permettent de stocker cette information. Parmi les formes que peuvent prendre ces bases de données, nous nous intéressons aux bases de données vectorielles. Les éléments géographiques peuvent être représentés dans une base de données géographiques, comme la combinaison d'attributs géométriques représentant l'objet dans un référentiel donné (ces attributs peuvent être des points, des polygones, des polygones, exprimés dans des coor-

données cartésiennes), et d'attributs de types plus classique (nom de lieu, valeur numérique comme une altitude).

*A priori*, et de façon simplifiée un processus de généralisation automatisé prend en entrée ces données, et fournit en sortie une carte lisible. Ce processus automatisé s'avère complexe à définir. Mackaness (2007) [61] identifie plusieurs caractéristiques qui expliquent cette complexité:

- Plusieurs solutions sont possibles à partir d'une même source de données, et pour un même besoin.
- Ces solutions relèvent souvent d'un compromis, dans la mesure où des choix sont nécessaires quant aux caractéristiques à respecter.
- Les informations portées par la carte sont complexes et multiples. Elles peuvent impliquer un ou plusieurs objets. La perception d'éléments d'une carte par son utilisateur suit les propriétés du Gestalt (DeLucia and Blacke (1987) [17]).
- La différence de l'information donnée pour une zone donnée diffère selon l'échelle et les besoins utilisateurs, ce qui implique que les paramètres de simplification appliqués lors du processus automatisé sont différents.
- La carte exprime finalement une information essentiellement abstraite, alors que les données initiales sont concrètes. De nombreux outils d'analyses sont nécessaires pour abstraire l'information et manipuler les abstractions résultantes.

Des **opérateurs** pour adapter la géométrie d'un ou plusieurs objets ont été proposés, et l'élaboration d'**algorithmes** pour effectuer ces opérations a constitué une première étape pour l'automatisation de la généralisation, ces algorithmes s'appliquant sur les géométries des objets individuels de base (bâtiments, tronçons de routes), ou des groupes d'objets (ensemble de bâtiments, réseaux routiers). Il existe plusieurs inventaires et classifications pour ces algorithmes (Lecordix *et al.* (1997), MacMaster (1987), MacMaster et Shea (1992), Gaffuri (2008b) [35], Galanda et Weibel (2002) [38], Regnaud et McMaster (2007) [86], Stanislawski *et al.* (2014) [107]). Le Tableau 1 expose des exemples d'opérateurs avec des algorithmes possibles.

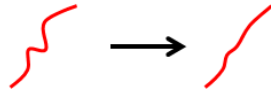
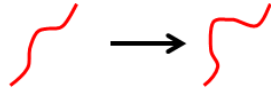
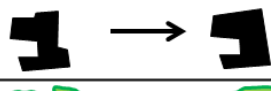
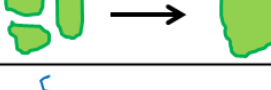

Opérateurs	Exemples illustrés	Exemples d'algorithmes
Simplification d'une polyligne		Douglas-Peucker (MacMaster, 1987 ; MacMaster et Shea, 1992) Lang (Lang, 1969) Visvalingam (Visvalingam et Whyatt, 1993)
Caricature de virages		Accordéon (Plazanet, 1996) Faille Min (Mustière, 1998) Faille Max (Mustière, 1998) Suppression de virages (Lecordix et al. 1997)
Simplification d'une surface		Modification et suppression d'arêtes (Lichtner, 1979 ; Sester, 2005 ; Sester et Brenner, 2004) Remplacement par un rectangle
Agrégation		Fusion de ligne Amalgame de points Amalgame de surfaces
Simplification de réseaux		Simplification de réseaux hydrographiques (Richardson, 1994) Simplification de réseaux routiers (Reynes, 1997)

Tableau 1 : Exemples d'opérateurs, avec des algorithmes possibles, d'après la classification de Stanislawski *et al.* (2014)[107].

### I.1.3 Séquencement des opérateurs à appliquer

Tout comme les modifications de la carte sont guidées par la volonté de simplifier et rendre lisible l'information, l'application des algorithmes peut être guidé par un ensemble de règles exprimant cette volonté de simplification. La définition de telles règles est difficile, dans la mesure où il y a une part de subjectivité dans les choix à faire lors de la généralisation.

Les premières approches ont consisté à proposer une association entre une condition et une action à effectuer. La condition porte sur des mesures concernant des objets géographiques (par exemple, la surface d'un bâtiment qui ne doit pas dépasser une certaine surface). Une action est l'application d'un algorithme sur un objet, avec un paramétrage particulier. Les conditions non respectées constituent des conflits. Pour identifier ces conflits, plusieurs algorithmes ont été proposés pour analyser l'espace géographique. Ces algorithmes sont appelés **algorithmes d'analyse spatiale**. Le paramétrage de ces algorithmes est définie par un ensemble de connaissances établit à partir des pratiques des cartographes et de connaissances acquises lors de la généralisation. Cette approche est néanmoins problématique, car l'acquisition et la formalisation de telles connaissances sont difficiles. Un des aspects les plus complexes est l'ordre dans lequel les algorithmes de généralisation doit être appliqués. Il s'avère que l'utilisation de règles strictes n'est pas une méthode suffisamment souple pour exprimer ces connaissances, et mesurer l'impact de l'application d'algorithme.

Beard (1991) [6] propose d'utiliser des **contraintes**, qui s'avèrent plus efficaces que l'utilisation de règles pour formaliser les spécifications d'une carte. D'après Ruas et Plazanet (1996) [96], les contraintes permettent de mesurer à différents niveaux (au niveau d'un objet, d'une partie d'un objet, d'un groupe d'objets) dans quelle mesure les spécifications de la carte sont respectées : par exemples, deux objets symbolisés ne doivent pas se superposer ou un objet doit avoir une taille minimale sur la carte. Plusieurs classifications de contraintes ont été proposées. De ces classifications, deux catégories de contraintes sont dis-



cernables : les **contraintes de lisibilité**, portant sur des caractères visuels de l'information, et les **contraintes de préservation** qui tendent à maintenir la véracité de l'information en la préservant. Les contraintes de lisibilité peuvent être portées par des objets, et évaluées en tenant compte de ces objets uniquement, un même objet pouvant avoir plusieurs contraintes partagées avec différents autres objets. Les contraintes de préservation doivent être paramétrées pour maintenir une information. Ces informations peuvent décrire une abstraction, qui s'exprime de selon des relations spatiales (par exemple, si un point d'intérêt important se trouve au bout d'une impasse, cette information doit apparaître sur la carte, une contrainte de préservation du positionnement relatif doit donc être prise en compte).

### I.1.4 Problème de l'orchestration des algorithmes

La façon dont il est souhaitable d'appliquer ces algorithmes est complexe. D'une part, ils nécessitent d'être paramétrés selon le type de carte, l'échelle, les besoins utilisateurs. D'autre part, les différents algorithmes n'agissent pas de concert naturellement pour modifier de façon satisfaisante les informations de la carte. L'application d'un algorithme peut défaire ce qu'un autre algorithme a pu faire par exemple. Enfin, les algorithmes agissent à plusieurs niveaux, et un même objet pourra être modifié par un algorithme agissant sur un groupe auquel il appartient, sur lui uniquement, ou sur une de ses parties. L'application des algorithmes, leur critère d'utilisation, leur paramétrage posent un problème complexe tant les combinaisons possibles sont nombreuses, et retournent des résultats variables. Pour Shea et McMaster (1989) [104], ces problématiques sont résumées par les questions où (sur quel espace de la carte agir ?), quand (quelle opération faire avant telle autre ?) et comment (quelle action faire ?). La proposition d'exprimer les besoins de la carte sous forme de contraintes évaluables permet d'envisager des pistes pour automatiser l'application de ces algorithmes, en considérant le problème de la généralisation comme un problème de résolution de contraintes.

Plusieurs approches ont été proposées pour résoudre ce problème. Des méthodes, appelées **méthodes globales**, consistent à proposer une solution optimale en tenant compte d'un bloc de toutes les contraintes impliquant tous les objets de la carte (méthode d'estimation par moindres carrés (Harrie et Sarjakoski (2002), [45] ; Sester (2005) [102]), basées sur une exploration par recuit simulé (Ware *et al.* 2003) [118] ou encore inspirées de modèles physiques comme les *beams* (Bader *et al.* 2005) [2] ou les *snakes* (Burghardt et Meier, 1997) [10] ; Gallanda et Weibel, 2003) [39].

Néanmoins ces approches présentent des limites, à la fois dans la façon dont les contraintes peuvent être exprimées, et dans les choix d'opérations à effectuer (par exemple, la suppression d'objets étant difficile à intégrer). Nous avons aussi mentionné que les contraintes pouvaient être associées à des objets. Appliquer des algorithmes aux objets concernés sur la base de connaissances sur l'effet qu'ils peuvent avoir sur les contraintes, tout en mesurant l'impact qu'ils peuvent avoir, semble être une méthode intuitivement intéressante pour résoudre ce problème. Ces méthodes, appelées **méthodes pas-à-pas locales**, ont prouvé leur efficacité, par exemple le Collaborative Filtering (Burghardt et Neun, 2006) [11] ou encore

les méthodes multi-agents qui sont celles qui nous intéressent dans le cadre de notre travail de thèse, qui sont présentées dans la section suivante.

## I.2 Les systèmes multi-agents pour la généralisation

La généralisation automatique d'une carte peut être considérée comme un problème contraint et un problème distribué. Les approches pas-à-pas locales, et en particulier les méthodes à base de Systèmes Multi-Agents (SMA) constituent un domaine de recherche couvrant ce second aspect. Dans cette partie, après avoir exposé succinctement les principes et enjeux de la recherche sur les systèmes multi-agents, nous présentons les principes sur lesquels reposent les modèles de généralisation automatisée s'appuyant sur de tels systèmes.

### I.2.1 Généralités sur les Systèmes Multi-Agents

L'étude des Systèmes Multi-Agents (SMA) est une discipline particulière du champ de l'intelligence artificielle. Elle implique aussi le domaine des systèmes distribués et celui du génie logiciel (Ferber, 1999) [29]. Dans un SMA, les **agents** sont des entités autonomes, qui cherchent à réaliser des buts qui leur sont propres dans un cadre souvent contraint, ne serait-ce que par la présence d'autres agents ayant des buts différents et possiblement contradictoire. Les agents évoluent dans un environnement, duquel ils ont une perception partielle.

Les SMA sont couramment utilisés pour effectuer de la simulation de phénomènes divers où il est possible d'identifier des entités ayant une autonomie dans leur comportement et leur prise de décision. Par exemple, un écosystème animalier, où chaque individu chercherait à survivre, à se nourrir et à se reproduire peut être simulé selon les principes d'un SMA. Un autre cas d'application des SMA est le cadre des problèmes dont la résolution peut être distribuée à diverses entités. L'éco-résolution (Drogoul et Dubreuil, 1995) [19] propose une approche d'utilisation de SMA pour résoudre des problèmes de façon distribuée. La généralisation est un de ces cas d'application, nous le verrons dans la section suivante.

La nature de ces SMA soulève un ensemble de questions et de problématiques très nombreuses, parmi lesquelles :

- La modélisation de l'architecture d'un agent, à savoir les processus de prises de décisions et la gestion de ses connaissances.
- La modélisation de l'environnement dans lequel les agents évoluent (étudiée par exemple par Weyns et Holvoet, 2006 [122] ; Weyns *et al.* 2004 [120]).
- La modélisation des interactions. Les interactions impliquent d'une part les interactions entre les agents et l'environnement, d'autre part les interactions des agents entre eux. La simultanéité possible des interactions soulève des questions complexes (par exemple, si deux agents ramassent un objet en même temps) (étudiée par exemple par Michel *et al.* 2003 [72]).
- La modélisation de la communication entre agents, qui est un type d'interaction particulier qui recoupe plusieurs problèmes complexes, comme la mémorisation de la

conversation, l'exploitation des informations issues d'un échange, la communication d'informations fausses...

- La modélisation de structures et de groupes entre agents. Des agents peuvent se regrouper (par exemple un troupeau ou une meute d'animaux) ou se hiérarchiser. L'expression des comportements au niveau du groupe influe sur chacune des parties du niveau des individus, ce qui pose des questionnements variables selon les situations (étudiée par exemple par Minar *et al.* 1996 [73] ; Ferber *et al.* 2005 [31]).

Cette liste non exhaustive du domaine de recherche des modèles multi-agents amène à faire le parallèle avec les questionnements que soulève une approche distribuée de l'automatisation de la généralisation.

### **I.2.2 Les SMA pour la généralisation**

Le principe de la généralisation, comme expliqué précédemment, amène à vouloir généraliser les différentes entités constituant l'information géographique que nous souhaitons présenter sur la carte une à une. Néanmoins, la généralisation de ces objets géographiques est soumise à des contraintes impliquant au moins une partie de la carte et de ses éléments allant au-delà de l'objet à généraliser. De plus, nous l'avons vu, les algorithmes permettant la généralisation peuvent porter sur plusieurs objets, tout comme l'évaluation des contraintes à satisfaire. L'idée de modéliser les objets géographiques comme des agents évoluant dans un SMA a donc été proposée.

Dans les SMA pour la généralisation, chaque objet géographique est modélisé comme un agent autonome. Cet agent a connaissance des contraintes qui le concernent. Par exemple, un agent bâtiment peut évaluer si sa contrainte de respect de taille minimale est satisfaite, tout comme un agent route avec sa contrainte de non-empâtement. Des contraintes peuvent être partagées par des agents, et ces agents vont chacun tenter de résoudre ces contraintes via des algorithmes différents. Cette approche est complexe, car elle soulève les points évoqués précédemment du domaine de recherche des SMA. Il s'agit notamment d'identifier les processus individuels qui permettront à chaque agent de choisir tel ou tel algorithme, avec un paramétrage particulier. Ce choix est influencé par les contraintes notamment par leur degré de satisfaction, mais aussi leur importance (c'est-à-dire la nécessité relative à être satisfaite, permettant ainsi de trancher si deux contraintes ne peuvent être toutes deux satisfaites) et des connaissances sur l'impact qu'ont les différents algorithmes sur la satisfaction des contraintes.

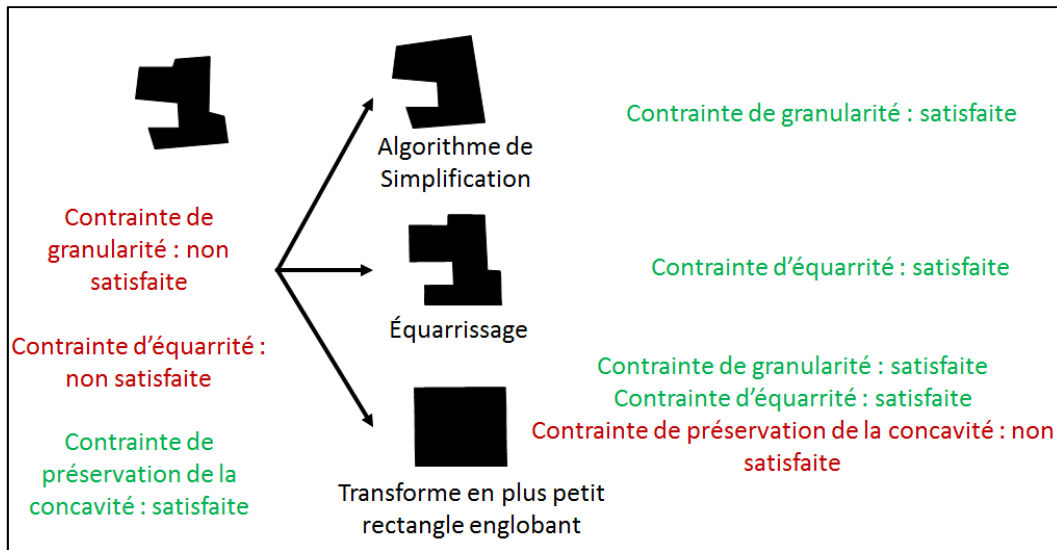


Figure 2 Comment différentes opérations peuvent impacter différentes contraintes.

L'évaluation systématique des contraintes permet de s'assurer que l'application des algorithmes n'a pas détérioré une situation. Le cas échéant, il est possible de revenir sur certains choix, et de tenir compte d'échecs pour en effectuer de nouveaux.

L'espace dans lequel évoluent les agents géographiques est celui de la carte en construction. Ils peuvent en avoir une perception locale, dont les limites sont définies par les besoins propres à l'évaluation des contraintes et l'application des algorithmes.

## I.3 Présentation des différents modèles

Dans cette partie, nous présentons différents modèles existants pour la généralisation impliquant une utilisation de système multi-agents dans le cadre d'interactions entre objets géographiques.

### I.3.1 Le modèle SIGMA

SIGMA ( Baeijs, 1998 [3] ; Baeijs *et al.* 1996 [4]) propose de modéliser les points des géométries comme des agents, et de le regrouper dans des espaces de perception appelés champs (par exemple, un champ pour l'objet auquel appartiennent les points). Les agents points subissent des forces qui sont la conséquence de situations prédéterminées. Par exemple une force de répulsion est appliquée sur deux points considérés comme trop proches. Les points se déplacent en cherchant une position d'équilibre.

#### Résumé des propriétés multi-niveaux du modèle :

- relations statiques transversales entre agents basées sur des critères d'appartenance à de mêmes ensembles.
- relations dynamiques transversales exprimées sous forme de forces que l'agent subit.

### I.3.2 Le modèle AGENT

#### I.3.2.1 Les bases du modèle AGENT

Le modèle AGENT (Ruas, 1998 [91], 1999 [92] ; Barrault *et al.* 2001 [5]) est un modèle proposé pour la généralisation s'appuyant sur une modélisation multi-agents. C'est le modèle qui est à l'origine des principes de base de la généralisation orientée agents décrits dans la partie précédente. En plus de ces principes de base, il pose aussi une modélisation des différents niveaux d'analyse. Ainsi, les objets géographiques fournis directement par la base de données géographiques (par exemple les bâtiments, les sections de routes ou encore les sections de cours d'eau), sont, dans le modèle AGENT appelés objets *micro*. Certaines connaissances permettant l'évaluation de contraintes et l'application d'algorithmes, portent sur des objets issus de l'agrégation d'autres objets. Ces objets agrégés sont appelés objets *meso*, et les objets qui les composent *composants*. Notons qu'un composant peut être un objet micro ou meso. Par exemple, les îlots urbains, sont composés des tronçons de routes qui les délimitent, des tronçons des impasses et des bâtiments à l'intérieur, dans tous les cas des objets micro. Autre exemple, les villes sont des objets meso composés des îlots urbains, eux-mêmes objets meso. Le modèle propose aussi un niveau d'objets *macro* qui désigne l'intégralité des objets d'un type donné (par exemple, l'ensemble des bâtiments).

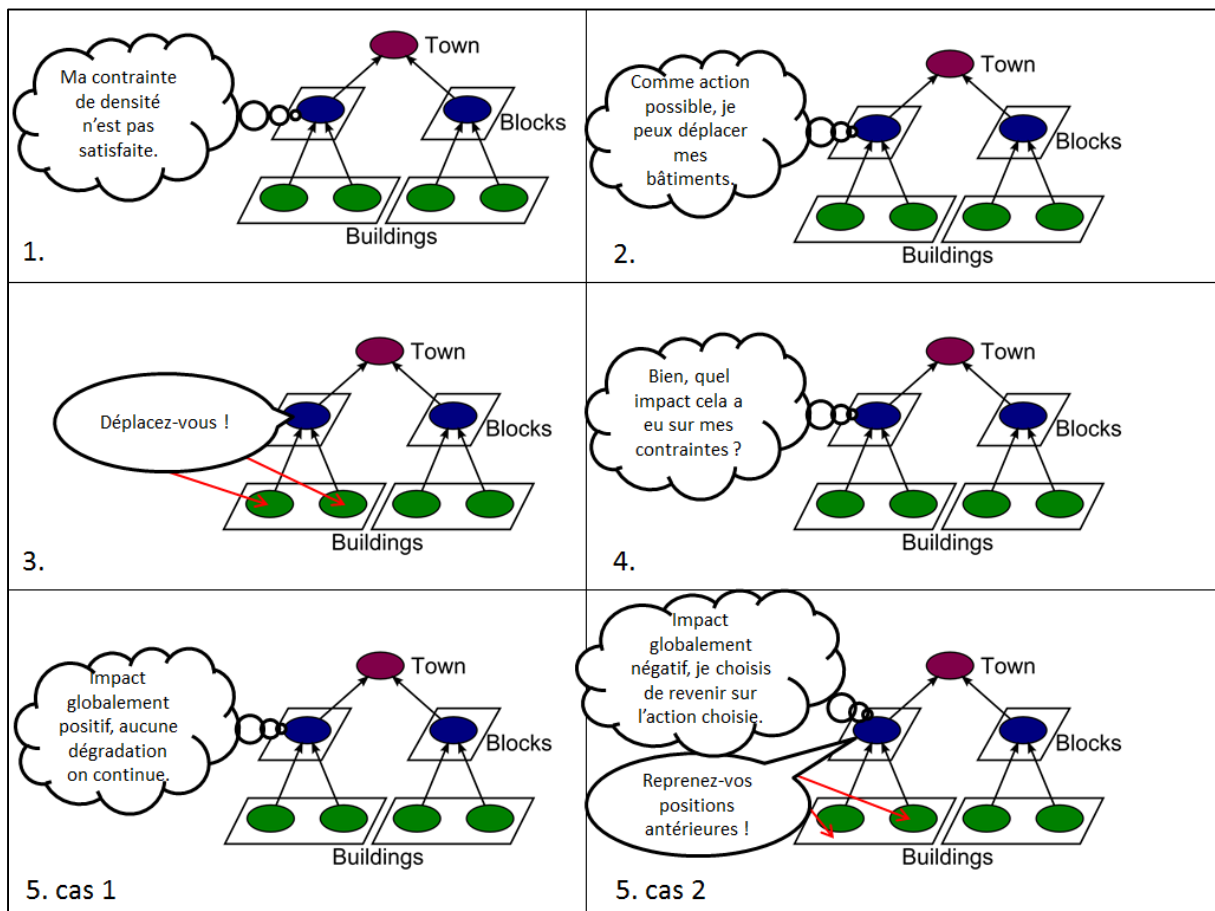


Figure 3 Exemple de scénario d'interactions entre agents dans AGENT, avec deux conclusions possibles.

Tous ces objets sont modélisés comme des agents dans le modèle. AGENT, en tant que SMA, propose des interactions entre agents. Les agents micro ont leurs propres contraintes à résoudre, et peuvent réaliser comme action individuelle, l'application d'algorithmes sur eux-mêmes, algorithmes affectant principalement leur géométrie. Les agents meso ont aussi des contraintes à satisfaire. Lorsqu'ils agissent pour les satisfaire, ils appliquent des algorithmes ayant des répercussions sur les agents composant le meso. De plus, le meso peut aussi, comme action, activer ces composants. Ruas (1999) [93] propose une liste de différents rôles d'un agent meso :

- le rôle de législateur qui modifie les contraintes de ses composants ;
- le rôle de coordinateur qui propose un ordre, trouve un ordre d'activation pour les agents composants, et les active selon cet ordre ;
- le rôle de contrôleur qui vérifie que ses contraintes sont respectées, revient sur certaines actions et propose des solutions pour des conflits non résolus.

La Figure 3 illustre un scénario d'interactions dans le modèle AGENT.

#### Résumé des propriétés multi-niveaux du modèle :

- relations statiques hiérarchiques en arborescence construites sur la base d'un lien de composition entre objets.
- relations dynamiques au sein d'une hiérarchie de façon descendante.

#### ***1.3.2 Applications dans le cadre du projet AGENT***

Le projet AGENT est un projet européen qui a mis en évidence l'efficacité du modèle AGENT sur les zones urbaines et sur les thèmes routiers. Pour les zones urbaines, les interactions concernent les bâtiments et les îlots d'une part (Barrault *et al.* 2001 [5]) (voir Figure 4), et les routes et le réseau routier d'autre part (Duchêne *et al.* 2001 [22]). Dans le cadre de la généralisation des zones urbaines, les îlots urbains cherchent à se généraliser via des actions agissant sur les bâtiments les composant.



Figure 4 Exemple d'application d'AGENT sur un îlot urbain.

Dans le cas des routes, ce sont les tronçons de routes qui peuvent être amenés à se diviser pour permettre à leurs sous-parties de résoudre des conflits localement. Un agent meso correspondant à la section divisée se charge d'orchestrer ses sous-parties modélisées comme des agents micro, et de maintenir la cohérence du tout.

### ***1.3.2.3 Applications dans le cadre de la généralisation de mosaïques de polygones***

Galanda (2003) [37] s'intéresse à la généralisation de cartes composée de mosaïques de polygones en utilisant le modèle AGENT. Contrairement aux situations du projet AGENT, il permet une structure hiérarchique non limitée à l'arborescence, et inclut les éléments des géométries comme agents dans le système. Ainsi les lignes de polygones sont des agents au même titre que les surfaces qu'elles délimitent. Deux surfaces délimitées par une même polyligne en seront toutes deux un meso. En termes de processus, les interactions se font de haut en bas, comme dans le modèle de base. Ainsi, un agent polyligne pourra être activé à plusieurs reprises si plusieurs agents polygones lui donnent la main. Il ne synthétisera pas les actions qu'il pourra faire lors de sa généralisation.

#### **Résumé des propriétés multi-niveaux du modèle :**

- relations statiques hiérarchiques non arborescentes construites sur la base d'un lien de composition entre objets.
- relations dynamiques au sein d'une hiérarchie de façon descendante.

### **1.3.3 Généralisation d'isobathes**

La généralisation de cartes marines, utilisées pour les déplacements de bateaux, impliquent de s'intéresser au relief des fonds marins. Des contraintes particulières liées à la sécurité de la navigation sont identifiées : il est indispensable que pour un point de l'espace donné, le relief sous-marin ne soit pas sous-évalué. Cette information est donnée par les isobathes, des lignes dont les points partagent la même profondeur. Afin de généraliser des données géographiques de ce type, Guilbert (2013) [44] propose de structurer les caractéristiques du relief sous-marin selon un arbre de contours. Cet arbre relie une isobathe aux isobathes immédiatement situées à l'intérieur du polygone délimité par cette isobathe, sans être elle-même située dans le polygone d'une autre de ces isobathes (voir Figure 5). Il garde aussi, à chacun de ses nœuds, l'information topologique du terrain.

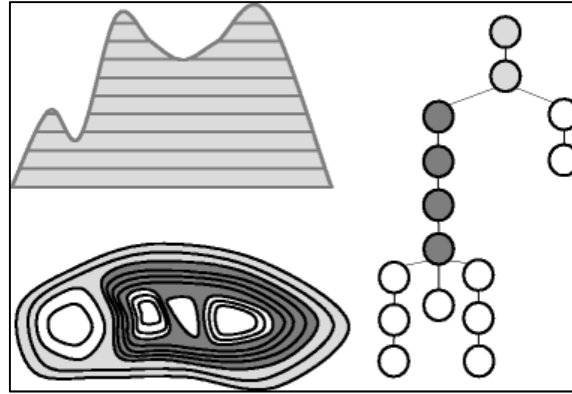


Figure 5 Exemple d'organisation de données isobathes pour la généralisation, d'après [17].

Zhang et Guilbert (2011) [123] propose une utilisation du modèle AGENT pour le processus de généralisation des isobathes, s'appuyant sur les arbres de contours. Les agents considérés sont les isobathes et des caractéristiques de terrain décrites par ces isobathes (les pics et les gouffres), ces deux catégories ayant leur propre ensemble d'actions possibles et de contraintes. Un lien hiérarchique associe un agent caractéristique aux agents isobathes situés à l'intérieur. Le modèle permet que, lorsqu'un agent caractéristique est généralisé, les agents isobathes qui lui sont dépendant le soient aussi.

#### Résumé des propriétés multi-niveaux du modèle :

- relations statiques hiérarchiques arborescentes construites sur la base d'un lien de situation entre objets.
- relations dynamiques au sein d'une hiérarchie de façon descendante.

#### I.3.4 Le modèle CartACom

Les modèles précédents présentent plusieurs limites. Leur utilisation s'avère efficace dans les situations où une hiérarchie peut être identifiée entre les différents objets, dans la mesure où les échanges ne se font que dans ce genre de cas. Par contre, des objets de mêmes niveaux n'interagissent pas entre eux, et d'éventuels contraintes les impliquant ne peuvent être résolues qu'à un niveau supérieur commun. Cette limitation s'avère problématique lorsque des objets géographiques ayant des contraintes les impliquant tous les deux, n'ont pas de supérieur hiérarchique évident qui pourrait se charger de satisfaire cette contrainte.

Le modèle CartACom (Duchêne, 2004 [25] ; Duchêne et al. 2012 [26]) propose un système élaboré de communication entre agents d'un même niveau. Deux agents en relation, c'est-à-dire partageant au moins une contrainte commune, appelée contrainte relationnelle (par exemple, une contrainte de maintien entre deux bâtiments proches l'un de l'autre) peuvent satisfaire cette contrainte en agissant individuellement. Lorsqu'un agent décide d'agir, il informe l'autre de sa nouvelle situation. S'il ne trouve pas d'action permettant de satisfaire la contrainte, il peut proposer à l'autre agent de faire une action précise. Ce dernier devra informer l'agent à l'initiative de la demande de l'action effectuée ou refusée. La Figure 6 illustre un scénario d'interaction entre agents dans le modèle CartACom.



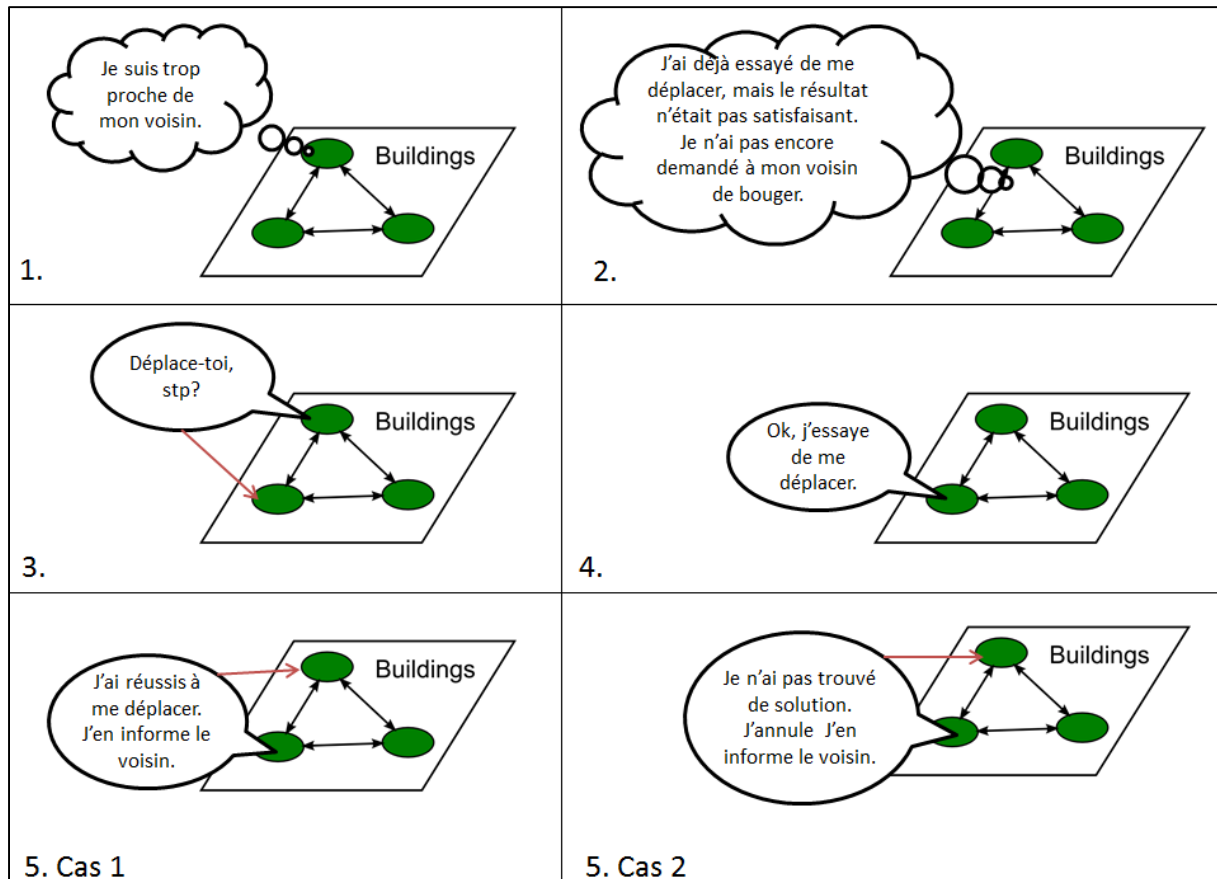


Figure 6 Exemple de scénario d'interactions entre agents dans CartACom, avec deux conclusions possibles.

Le modèle a été testé et validé dans le cadre de la généralisation de zones rurales. Les bâtiments ne pouvant pas appartenir à des îlots urbains, doivent par eux-mêmes se généraliser en communiquant avec les autres bâtiments, les sections de routes, et les faces réseaux (des polygones délimités par les routes). Les actions de généralisation individuelles (qui n'impliquent qu'un objet) sont aussi déclenchées en suivant les principes du modèle AGENT.

**Résumé des propriétés multi-niveaux du modèle :**

- relations statiques transversales entre agents basées sur des critères topographiques (distances, objets entre).
- relations dynamiques transversales au sein d'un même niveau, impliquant communication entre agents.

**I.3.5 Le modèle GAEL**

Le modèle CartACom propose des interactions objets à objets. Néanmoins, il est possible qu'une contrainte implique seulement une partie d'un objet géographique. En effet, certains objets qui peuvent être déformés dans le cadre de la généralisation, peuvent l'être localement. S'il existe des algorithmes pour contrôler la déformation des objets, il peut s'avérer nécessaire de prendre en compte les relations avec d'autres objets pouvant n'influer que sur une partie de l'objet déformable. Le modèle GAEL (Gaffuri, 2008 [34] ; Gaffuri, 2007 [33]) propose une approche multi-agents pour résoudre ces déformations locales.

Le principe de base du modèle est d'inclure un niveau supplémentaire par rapport aux niveaux déjà définis et exploités dans les autres modèles (meso, micro et macro). Ce niveau, appelé submicro, implique des sous-parties d'un objet déformable. Ces sous-parties, les objets submicro, sont obtenues par décomposition des objets déformables : ce sont des points, des segments, des triangles et des angles. Ces objets portent leurs propres contraintes submicro : la préservation de position pour le point, la conservation d'une surface pour un triangle.

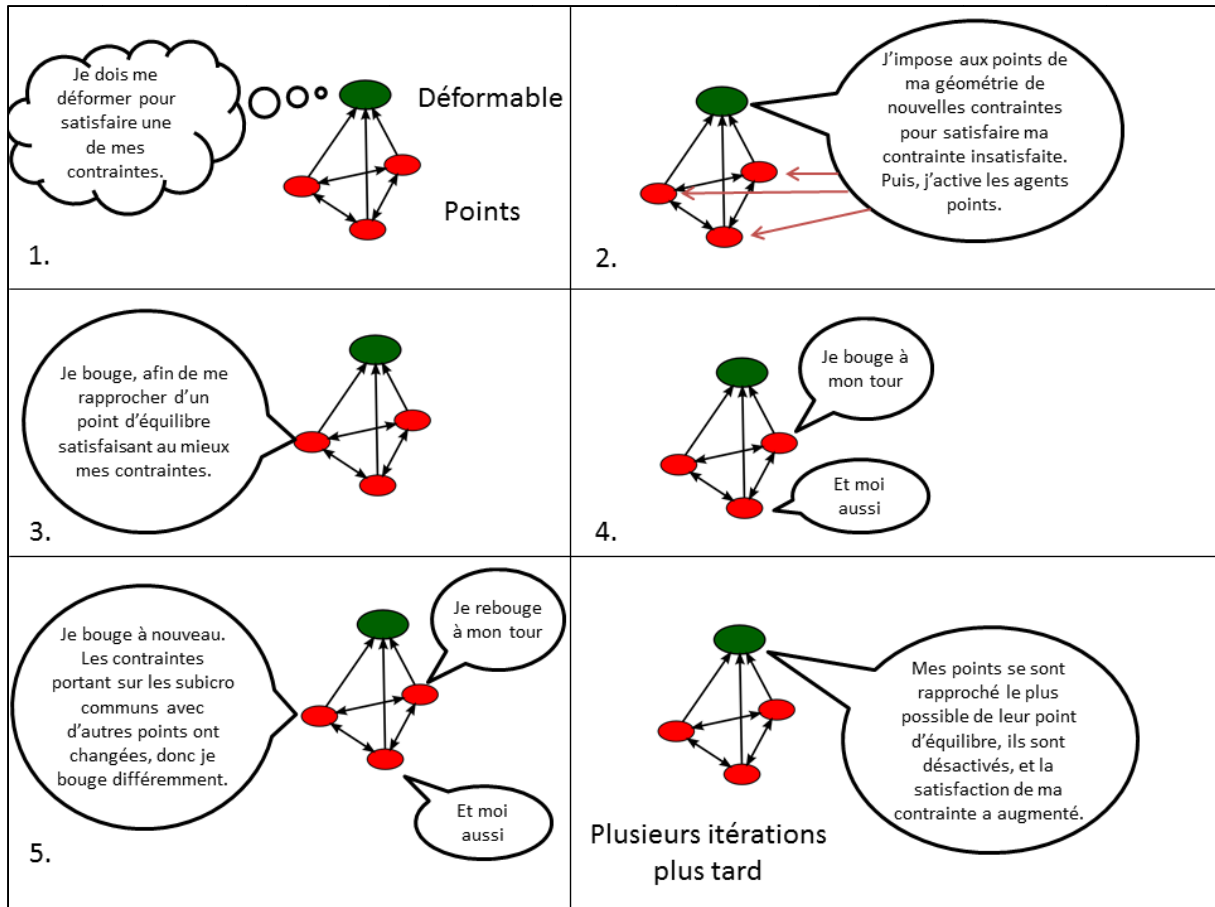


Figure 7 Exemple de scénario d'interactions d'agents dans GAEL.

Les seuls objets de ce niveau modélisés comme des agents sont les points, qui peuvent se déplacer afin de satisfaire les contraintes portant sur les submicro dont ils sont une partie. Ils peuvent aussi appeler l'activation de leurs voisins. La Figure 7 expose un exemple de scénario type d'une application de GAEL.

Ce modèle permet la généralisation de thème « champs » en cohésion avec les autres objets. Un champ est présenté comme l'ensemble des valeurs d'une caractéristique portant sur tous les points de l'espace (par exemple le relief).

Le modèle est aussi proposé pour résoudre d'autres cas de généralisation impliquant non plus des objets considérés comme déformables, mais sur des objets de type compact (Gaffuri, 2009 [36]).

**Résumé des propriétés multi-niveaux du modèle :**

- relations statiques hiérarchiques non arborescences construites sur la base d'un lien de composition entre objets.
- relations dynamiques transversales exprimées sous forme de forces que l'agent subit.

**I.3.6 Structures réactionnelles**

Nombres de situations de généralisation s'appuient sur une structuration hiérarchique de l'espace. Néanmoins, les interactions sont toujours descendantes lorsqu'une hiérarchie existe entre deux objets. Cela peut poser problème lorsqu'un objet s'inscrit dans deux relations hiérarchiques différentes, les demandes d'interaction d'un « supérieur » hiérarchique pouvant avoir des répercussions négatives sur les contraintes de l'autre « supérieur » hiérarchique. Par exemple, un bâtiment appartenant à un îlot, peut aussi appartenir à un (ou plusieurs) alignement de bâtiments. Les déplacements que peut lui imposer l'îlot pour satisfaire certaines de ses contraintes (par exemple une contrainte de non superposition), peuvent détruire l'alignement, causant ainsi une perte d'information sur l'organisation urbaine, potentiellement utile pour l'utilisateur de la carte. Renard (2012) [87] propose une modélisation enrichissant le modèle AGENT qui permet à un agent d'activer un autre agent en position « supérieure », agent qui tentera de résoudre le conflit qu'aurait dû tenter de résoudre l'agent initialement activé (voir Figure 8 pour un exemple de scénario spécifique à ce cas).

L'introduction d'interactions ascendantes dans un modèle hiérarchique descendant permet la généralisation de zones urbaines en tenant compte des alignements.

**Résumé des propriétés multi-niveaux du modèle :**

- relations statiques hiérarchiques non arborescences construites sur la base d'un lien de composition entre objets.
- relations dynamiques au sein d'une hiérarchie de façon descendante et ascendante.

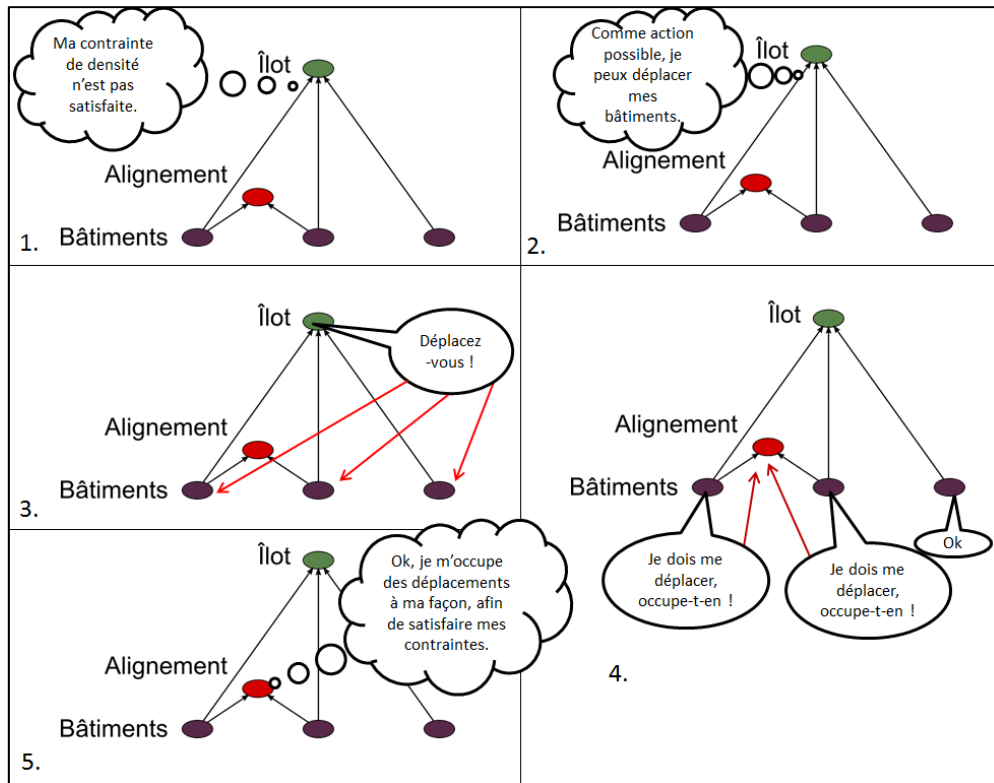


Figure 8 Exemple de scénario d'interactions entre agents avec structures réactionnelles, les trois premières étapes étant similaires à l'exemple de la Figure 3.

### I.3.7 Conflits persistants

Les algorithmes ont été conçus pour tenter de satisfaire les contraintes d'objets géographiques. Néanmoins, ces algorithmes peuvent se montrer inefficaces sur certaines situations. De même, durant le cycle d'activation de plusieurs agents, il est possible qu'ils ne trouvent pas de solutions à un conflit. Plusieurs approches ont été proposées pour identifier ces conflits et y apporter une solution.

Duchêne et Touya (2010) [27] propose un modèle de détection de conflits et de création d'entités pouvant tenter de les résoudre. Une proposition d'intégration de cette modélisation dans CartACom s'appuie sur l'introduction d'un *superviseur*. Lorsqu'un agent a tenté toutes les alternatives pour satisfaire une contrainte sans succès, il en informe le superviseur. Régulièrement, le superviseur établit des groupes candidats d'agents en situation de conflit non résolu sur la base des informations données par les agents ne réussissant pas à résoudre leurs contraintes, et d'autres informations portant sur les échanges que les agents ont pu faire entre eux, ou sur des critères spatiaux. Ces groupes sont ensuite amenés à subir un traitement collectif particulier (dont la nature impacte le procédé de constitution du groupe).

#### Résumé des propriétés multi-niveaux du modèle :

- relations statiques hiérarchiques construites sur la base d'identification de structures émergentes.

- relations dynamiques au sein d'une hiérarchie de façon descendante.

### **I.3.8 Utilisation de SMA pour la généralisation de cartes à la volée**

Jabeur (2006) [52] s'intéresse à la généralisation de carte **à la volée**, c'est-à-dire à la modification en temps réel des objets géographiques pour proposer une carte lisible à l'utilisateur. Aux enjeux de la généralisation classique s'ajoutent les enjeux propres aux applications à exécution en temps réel. Le but est de proposer un résultat correct dans une plage de temps limitée. Jabeur (2006) [51] propose un modèle multi-agents pour la généralisation à la volée. Ce modèle introduit plusieurs types d'agents, notamment des agents *Instances*, équivalent des agents micro de AGENT, et des agents *Conteneurs (Container)*, qui agrègent plusieurs agents. Si le contenant a à sa charge des rôles qui sont attribués aux agents meso du modèle AGENT, ils ont en plus un rôle à jouer propre aux enjeux de la généralisation à la volée. En effet, comme certains agents Instances peuvent être amenés à ne pas se généraliser efficacement, des situations conflictuelles demeurent, et il est à la charge de l'agent Conteneur de les résoudre. Un agent Instance doit donc, s'il est confronté à une situation qu'il n'arrive pas à résoudre efficacement, demander de l'aide à son Conteneur. La difficulté d'une telle approche consiste à identifier les Conteneurs (qui ont chacun à charge plusieurs Instances) les plus pertinents pour optimiser la résolution de conflit.

#### **Résumé des propriétés multi-niveaux du modèle :**

- relations statiques hiérarchiques construites sur la base d'identification de regroupements.
- relations dynamiques au sein d'une hiérarchie de façon descendante.

Sabo (2007) [98] propose la notion d'Objet Auto-généralisant (SGO : Self Generalising Object) pour faciliter la généralisation de cartes à la volée. Ces objets auto-généralisants sont des entités qui associent à un patron un ensemble de contraintes et d'algorithmes. Un patron est une forme type à laquelle sont rattachés différents objets de la carte, par exemple un patron « bâtiment en forme de L ». Un objet peut être rattaché à différents patrons. Un ensemble de patrons a été identifié par Sabo et al. (2005) [99] et Sabo et Bernier (2005) [97]. L'utilisation des algorithmes sur les patrons permet d'optimiser le temps de la généralisation, ce qui est un des buts lors de la généralisation à la volée. L'application de ces algorithmes s'appuie sur une logique orientée agent, où chaque objet auto généralisant est modélisé sous forme d'agent. Les agents simples sont distingués des agents complexes (agents composés d'autres agents). Trois types de relations sont gérés : des relations verticales entre un agent complexe et un de ses composants, des relations horizontales entre deux agents au sein d'un même niveau, sous un agent complexe, et des relations obliques. Ces dernières interviennent quand deux agents de niveaux différents interagissent (par exemple, un bâtiment avec un groupe de bâtiments auquel il n'appartient pas). Ce type de relation est en fait déduit du cas où un agent (par exemple un bâtiment) est agrégé à d'autres agents pour constituer un groupe : dans ce genre de situation, l'agrégat hérite des relations des objets agrégés).

---

**Résumé des propriétés multi-niveaux du modèle :**

- relations statiques hiérarchiques construites sur la base d'identification de regroupements.
- relations dynamiques au sein d'une hiérarchie de façon descendante, transversale, ou diagonale.

## I.4 Problématique

Nous avons décrit, dans la partie précédente, différents modèles multi-agents pour la généralisation cartographique. Bien que ces modèles soient basés sur un même ensemble de principes, ils introduisent chacun des concepts nouveaux propres à des situations problématiques particulières. L'implication, dans une carte, de plusieurs situations problématiques conjointes amène à s'interroger sur les **possibilités d'utilisation combinée** de différents modèles. De plus l'existence de problématiques de généralisation dont l'automatisation n'est pas encore aboutie soulève la question de savoir qui, parmi les concepts existants, existe déjà pour automatiser la généralisation de tels types de cartes. Or, répondre à cette question n'est pas évident.

Dans cette section, nous décrivons plus précisément ces deux motivations, avant de présenter l'approche que nous proposons et les questions qu'elle soulève.

### I.4.1 Utilisation combinée de modèles

La volonté d'utiliser, de façon combinée, plusieurs modèles est motivée avant tout par le fait que les cartes ont tendance à couvrir plusieurs situations variées. Pour Duchêne et Gaffuri (2008) [23], l'intérêt est porté sur la possibilité d'utiliser conjointement les modèles AGENT, CartACom et GAEL ont chacun été testés avec succès dans des cadres applicatifs particuliers, à savoir les zones structurées à forte densité pour AGENT, les zones à faible densité d'objets répartis de façon hétérogènes pour CartACom, et la gestion des thèmes d'arrière-plan, comme le relief pour GAEL. Bien entendu, une même carte peut combiner l'ensemble de ces caractéristiques, et sa généralisation sera plus efficace si elle bénéficie des apports des trois modèles.

Toujours dans Duchêne et Gaffuri (2008) [23], trois scénarios sont proposés pour permettre l'utilisation commune des trois modèles.

Le premier consiste à utiliser les modèles sur des partitions des données. Ces partitions peuvent être spatiales (par exemple les zones urbaines et les zones rurales), ou thématiques (l'ensemble des courbes de niveaux). S'ensuit alors l'application disjointe des processus sur chacune de ces zones. Ce scénario a conduit à l'élaboration du modèle CollaGen (Touya, 2011 [113], Touya et Duchêne, 2011 [114]). Ce modèle permet la collaboration de processus automatiques différents (parmi lesquels les modèles multi-agents susmentionnés, mais aussi des algorithmes différents, n'impliquant pas de SMA). CollaGen permet l'utilisation de plusieurs processus sur la même partition. Par contre, les agents géographiques utilisés par un

processus multi-agents ne peuvent être utilisés dans un autre processus tout comme leurs connaissances construites lors de l'exécution du processus, (parmi lesquels leurs échecs et leur conversations).

Le deuxième scénario propose d'entrelacer l'exécution des trois modèles multi-agents. Ainsi, les différents objets sont modélisés simultanément sous forme d'agents différents dans plusieurs modèles. Un modèle laisserait alors la main à un autre modèle pour diverses raisons (par exemple lorsqu'un conflit n'est pas résolu). Un tel procédé implique d'actualiser les connaissances des différents modèles lors du passage de l'un à l'autre (par exemple la mise à jour des contraintes ou la suppression des contraintes relationnelles lors de la suppression d'un objet).

Le troisième scénario propose d'utiliser des agents des trois modèles initiaux dans le même modèle. Cette solution implique une factorisation des caractéristiques des agents et une reconstruction de ces derniers lorsqu'ils portent sur le même type d'objet, en combinant à la fois les opérations effectives d'un modèle avec celles de l'autre. Néanmoins, plusieurs questions restent en suspens, notamment l'évaluation et la comparaison de contraintes, certaines étant individuelles, d'autres partagées, ou encore l'activation d'un agent par un autre. La diversité des caractéristiques des agents des différents modèles est conséquente, et les exploiter dans un même modèle soulève des questions complexes qui vont au-delà d'une simple reconception d'un modèle.

### **I.4.2 Expression des connaissances selon un paradigme commun**

L'organisation des agents géographiques implique des procédés particuliers à chaque situation : la généralisation est un procédé hautement contextuel. La diversité des données, échelles et paysages engendre continuellement de nouveaux problèmes à traiter. Chaque fois qu'un nouveau niveau est proposé, de nouveaux principes sont intégrés de façon ad-hoc dans un modèle afin de couvrir de nouvelles situations et enrichir les connaissances du domaine de l'automatisation de la généralisation. Néanmoins, ces connaissances sont finalement propres au modèle et difficilement exploitables en dehors de celui-ci, sans passer par un travail de reconception systématique. Vouloir combiner des modèles présente un attrait majeur. Mais proposer une approche qui permet d'exploiter l'existant de façon fluide et modulaire est d'autant plus intéressant que cela répond aux questionnements quant à la prise en compte des besoins utilisateurs. L'expression des connaissances de façon générique est néanmoins un objectif difficile, tant la variété des situations est grande. Un des aspects les plus compliqués de cet objectif est le fait que des actions puissent être effectuées à plusieurs niveaux de la carte.

Il est aussi à noter que des cas non encore gérés de façon automatisée sont régulièrement soulevés dans le domaine de la généralisation. Par exemple, comme évoqué dans [21], les cartes avec des informations thématiques dont la géométrie s'appuie sur des objets supports, comme les cartes d'itinéraires, les cartes touristiques ou les cartes de transport non-schématiques, révèlent de nouveaux défis avec des relations entre objets différentes, comme un objet positionné sur un autre (par exemple un itinéraire sur une route, ou un ar-

---

rêt de bus sur un itinéraire de bus). L'idée d'une modélisation générique s'étend alors au-delà d'une formalisation commune de l'existant, mais s'étend aussi à la facilitation de l'introduction de nouveaux concepts.

### I.4.3 Approche proposée

Les deux aspects présentés dans les précédentes sous-sections peuvent résumer la problématique à laquelle ce projet de thèse tente de répondre :

*Comment exprimer les connaissances et les processus SMA propres à la généralisation automatisée dans un modèle commun, et tout particulièrement les aspects multi-niveaux du problème ?*

Nous pouvons identifier plusieurs verrous scientifiques à cette problématique :

- **l'organisation des objets** : si l'existant propose des organisations des objets géographiques pour plusieurs situations différentes (par exemple les bâtiments qui composent un îlot ou les points qui décrivent une géométrie), de nouveaux cas suggèrent l'introduction de nouveaux types d'organisations (par exemple, un itinéraire porté par un ensemble de tronçons de routes), et la gestion concurrente de hiérarchies différentes.
- **l'orchestration** : l'ajout possible d'actions entre niveaux différents entraîne une démultiplication de ces interactions. Les modèles existant pour l'orchestration peuvent ne pas être assez robustes pour ces nouvelles situations.
- **le choix des actions à effectuer** : de même, lorsqu'un agent est activé, il doit choisir une action. L'augmentation du nombre d'interactions possibles et de leur diversité (notamment pour ce qui est des agents avec qui effectuer ces interactions), supposent là aussi des critères de sélections plus complexes.
- **l'articulation des deux précédents points** : l'hypothèse des modèles agents est de positionner une partie de la prise de décision au niveau local de l'agent. Mais une prise de décision d'action en amont du choix d'activation de l'agent ne doit pas être écartée.
- **les répercussions des modifications d'un contexte à un autre** : une modification dans un contexte donné peut impliquer des modifications à un autre niveau. Quand et comment ces modifications doivent-elles être répercutées ?
- la définition de nouvelles structures pour résoudre de nouveaux cas implique la création de **nouvelles actions**, avec des répercussions particulières. La conception de ces nouvelles actions est en soi un problème.
- **les procédés de validation des résultats** : dans la mesure où les cartes résultant d'une opération de généralisation peuvent apporter des résultats inédits, c'est-à-dire qu'ils n'auraient pas été produits par un cartographe, tout en étant valable, il est impossible d'établir une base de résultats desquels nous chercherions à nous approcher. Pour pallier cette difficulté, il s'avère nécessaire d'évaluer nos résultats sur la base de critères à identifier.



Rendre générique un ensemble de connaissances de conception et de processus est un enjeu fréquent dans le cadre des sciences informatiques, et le domaine des SMA ne déroge pas à la règle. La prise en compte des aspects multi-niveaux est aussi particulièrement étudiée et a conduit à l'élaboration de méta-modèles. Nous souhaitons nous appuyer sur l'existant de ce domaine pour proposer une modélisation générique de l'aspect multi-niveaux de la généralisation s'appuyant sur des SMA. Dans le prochain chapitre nous présenterons une étude sur l'exploitation des aspects multi-niveaux dans le cadre des modèles multi-agents, étude qui conduira à exposer sur quelles bases nous nous appuyons pour développer notre proposition.

# Chapitre II :

# Étude des interactions

# multi-niveaux

---

Dans ce chapitre, nous nous intéressons aux aspects multi-niveaux dans les systèmes multi-agents, et tout particulièrement dans le cas des modèles pour la généralisation. Le domaine du multi-niveaux est un domaine en plein essor de la recherche en SMA.

Après avoir défini les éléments de vocabulaire que nous utiliserons dans notre discussion, nous revenons sur les interactions multi-niveaux dans le cadre de la généralisation cartographique. Ensuite, nous nous intéressons aux SMA multi-niveaux dans le domaine de la simulation. Enfin, nous terminons sur le modèle PADAWAN, modèle choisi *a priori* de la thèse pour supporter notre proposition.

## II.1 Définitions

Nous donnons ici les définitions de quelques termes importants liés aux aspects multi-niveaux utilisés dans la suite de notre mémoire.

Pour décrire la variété d'objets géographiques pouvant être en interaction, nous nous appuyons sur les notions de **niveau** et de **multi-niveaux**. Ces notions sont évoquées dans la littérature SMA. Ainsi, Morvan (2013) [75] propose un ensemble de définitions pour ces concepts. Par exemple, il définit la notion de niveau comme « un point de vue sur un système, intégré dans un modèle comme une abstraction spécifique ». Gil-Quijano et al. (2009) [83] et Gil-Quijano et al. (2010) [84] mettent en avant le fait qu'historiquement, les niveaux sont associés à un vocabulaire particulier pour les caractériser : macro, meso et micro. Néanmoins, ils insistent aussi sur le fait que ces caractérisations sont différentes selon les cas d'application, et qu'une vision plus souple et abstraite des niveaux est pertinente.

Nous proposons de définir la notion de niveau, en réutilisant la notion de point de vue utilisée par Morvan (2013) [75] :

**Définition** : Un *niveau* est un point de vue au sein d'un système pouvant présenter des aspects spécifiques (fonctionnements, relations, connaissances...).

Cette notion de niveau amène la définition de multi-niveaux comme portant sur une variété de point de vue dans un même modèle. Gil-Quijano et al. (2010) [84] soulignent que « l'apport d'un modèle multi-niveaux se situe plus dans l'articulation de différents niveaux

d'organisation que dans leur seule juxtaposition. » Morvan (2013) [75] définit un modèle multi-agents multi-niveaux comme un « modèle **intégrant** des modélisations **hétérogènes**, représentant des **points de vue complémentaires**, appelés niveaux, d'un même système. » Cette définition s'appuie sur la définition d'autres termes (Morvan, 2013) [75] :

- « L'*intégration* signifie que les modélisations orientées agents à l'intérieur de la modélisation multi-agents multi-niveaux peuvent interagir et partager des entités comme les environnements et les agents. »
- « L'*hétérogénéité* signifie que les modélisations orientées agents intégrées dans la modélisation multi-agents multi-niveaux peut se baser sur différents paradigmes de modélisation (par exemple des équations différentielles, des automates cellulaires), utiliser des représentations du temps différentes (événements discrets, pas-à-pas) et représenter des processus à différentes échelles spatio-temporelles. »
- « Des points de vue sont dits *complémentaires* pour un problème donné quand ils ne peuvent pas être utilisés isolément des autres pour le résoudre. »

La notion d'hétérogénéité est importante dans l'aspect multi-niveaux que nous étudions. Aussi, l'importance des mécanismes d'articulation des niveaux entre eux est aussi un point essentiel. Nous proposons une définition mettant en avant ces notions.

**Définition** : Un *modèle multi-niveaux* est un modèle où plusieurs niveaux aux modélisations potentiellement hétérogènes cohabitent et partagent des informations, et qui propose des mécanismes mettant en œuvre l'articulation de ces niveaux entre eux.

Nous voyons dans la section suivante comment notre problématique s'inscrit dans ces définitions.

## II.2 La problématique multi-niveaux en généralisation

Dans cette partie, nous explorons en quoi la recherche en automatisation de la généralisation amène à se poser des questions quant aux relations multi-niveaux entre objets géographiques, puis nous revenons sur la manière dont elles ont été traitées dans les modèles de généralisation SMA existants et enfin nous mettons en avant de nouvelles relations que nous souhaiterions savoir traiter.

### II.2.1 L'organisation multi-niveaux de l'information géographique

L'information géographique et sa représentation apparaissent comme étant liées à des questions de hiérarchie. Chaudhry et Mackaness (2007) [14] mettent en avant l'importance du concept de paronomie pour établir des relations hiérarchiques entre objets visibles ou non selon l'échelle de la carte. Mackaness et Gould (2014) [60] avancent la multiplicité des relations que peuvent avoir des objets géographiques entre eux, selon la façon dont ils sont abordés, et quel type de carte doit être créé. Par exemple, sur une carte mettant en évidence des voies de transports fluviaux, on peut considérer des façons différentes de modéliser les relations impliquant ces voies fluviales (par exemple, comme des propriétés des tron-

çons de fleuves, ou du réseau de tronçons, ou d'un réseau de transport, lui-même propriété du réseau hydrographique). La plupart des approches de création de relations entre les objets géographiques s'inscrivent dans un cadre procédural, plutôt que descriptif.

### II.2.2 Généralisation et multi-niveaux

Dans le cadre de la généralisation automatisée, il est parfois nécessaire de considérer plusieurs objets comme un groupe, sur lequel des opérations seront effectuées, que ce soit pour identifier et mesurer des contraintes, ou pour appliquer des algorithmes de transformation qui contribuent à la généralisation de la carte (Shea et McMaster, 1989 [104], Ruas et Plazanet, 1996 [96]). La création de ces entités groupées s'effectue par **enrichissement des données** (Ruas et Lagrange 1994 [95]) : des données initiales, que ce soit à partir de leur géométrie ou de leurs autres attributs, il est possible de déduire de nouveaux objets sur lesquels il est pertinent de raisonner. Ce premier constat est la base d'un raisonnement sur plusieurs niveaux dans le cadre de la généralisation. Nous revenons plus en détail sur ces notions, telles qu'elles ont été identifiées dans le cadre des recherches en généralisation.

**Les réseaux linéaires**

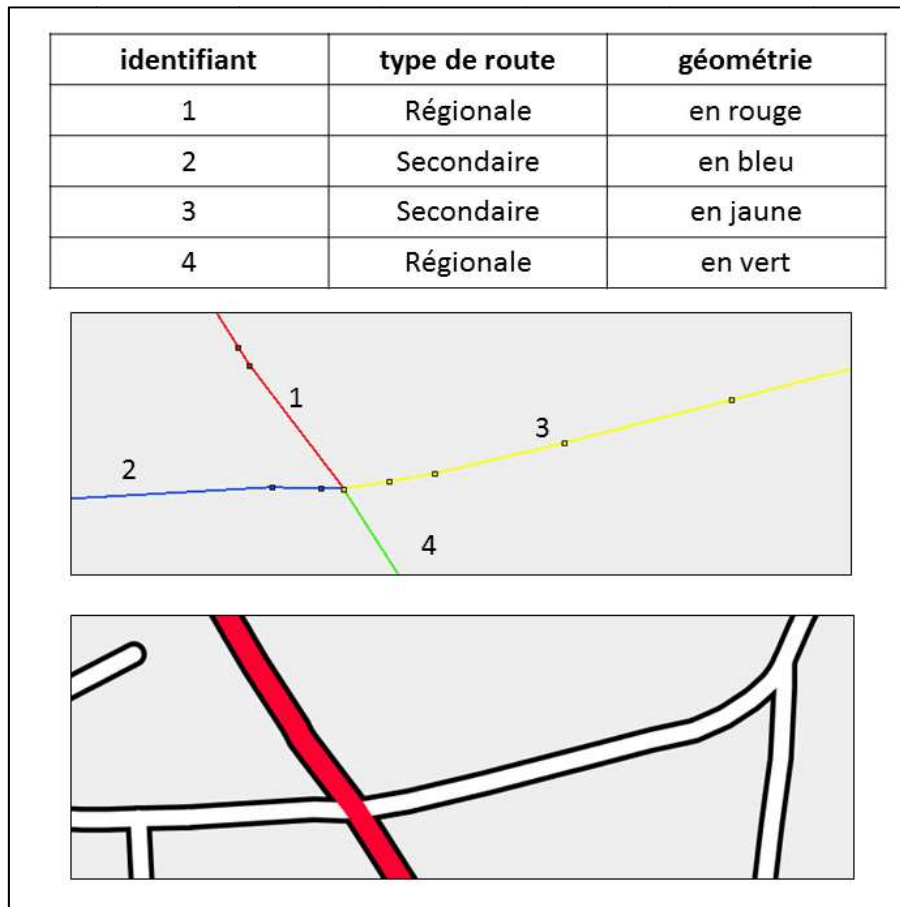


Figure 9 Exemple d'extrait d'une table de tronçons de routes linéaires. En haut, extrait simplifié de la table ; au milieu représentation des géométries ; en bas extrait de carte avec les tronçons symbolisés.

Certains objets géographiques ont des géométries linéaires, c'est-à-dire des objets dont la géométrie est une polyligne. Parmi ces objets, une partie d'entre eux s'inscrivent dans des réseaux cohérents. C'est le cas des réseaux routiers, hydrographiques et ferrés. Les données de ce type sont décrites de façon à ce que chaque tronçon du réseau soit présenté comme une entité dans les bases de données. Un tronçon est décrit comme une polyligne entre deux intersections (exemple Figure 9).

Le réseau est une entité constituée d'un ensemble de tronçons et des nœuds reliant les tronçons entre eux. La structuration des réseaux s'appuie sur la théorie des graphes, où les arêtes représentent les tronçons, et les nœuds les jonctions entre ces tronçons ou bien les culs-de-sac. La modélisation de ces réseaux géographiques sous forme de réseaux au sens de la théorie des graphes est motivée par plusieurs applications, comme la recherche d'itinéraire, qui peut alors s'appuyer sur des algorithmes de recherche de plus court chemin. Cette modélisation sous forme de graphe est aussi importante en généralisation, car le réseau géographique est perçu comme une entité à part entière par l'utilisateur de la carte, entité porteuse d'information, comme la densité des tronçons qui la composent. Elle permet aussi de faire une sélection intelligente des tronçons à représenter sur la carte, en évitant d'avoir une discontinuité dans le réseau. Nous identifions ici un exemple de relation entre

niveaux, celui de l'entité tronçon de base, et celui du réseau contenant l'ensemble des tronçons. Les nœuds peuvent aussi être modélisés. Ils ont alors une relation avec le(s) tronçon(s) auquel(s) ils sont connectés. La Figure 10 montre un diagramme de classe de ces relations.

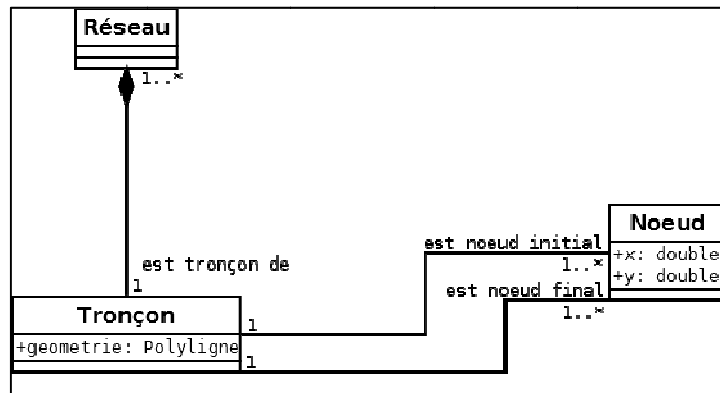


Figure 10 Diagramme de classe simplifié montrant les relations de tronçons dans un réseau.

Une autre relation entre les éléments d'un réseau est la perception de la façon de grouper un ensemble de tronçons, pouvant être perçus comme issus d'un même « coup de crayon ». Le terme **stroke** (signifiant « coup de crayon ») est utilisé pour désigner ces objets. Thomson et Richardson (1999) [112] proposent une méthode de détection de ces strokes (Figure 11). Une fois que ces strokes sont identifiés, il est pertinent de raisonner à leur niveau plutôt qu'au niveau des tronçons pour décider quels tronçons garder sur la carte généralisée (Thomson et Richardson, 1999) [112].

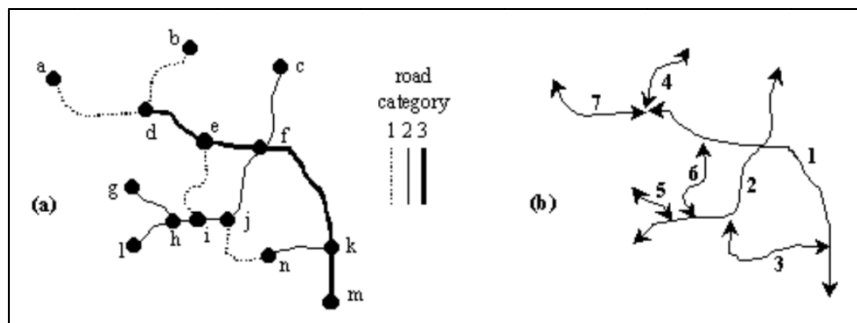


Figure 11 Définition de strokes dans un réseau routier, d'après Thomson et Richardson (1999) [112]. Par exemple, le stroke « 1 » est composé des routes délimitées par les couples de nœuds d-e, e-f, f-k et k-m.

### L'espace urbain

L'espace urbain est un cas d'étude important de la généralisation automatique. L'importante densité des objets géographiques au sein de ce type de zone encourage un traitement de ces derniers de façon groupée. Une structuration hiérarchique de ces objets peut s'effectuer via plusieurs procédés d'analyse spatiale. Boffet (2002) [8] propose un découpage de l'espace urbain qui est utilisé dans plusieurs situations, et notamment en généralisation. Ce découpage (illustré Figure 12), s'effectue sur la base d'analyse géométrique.

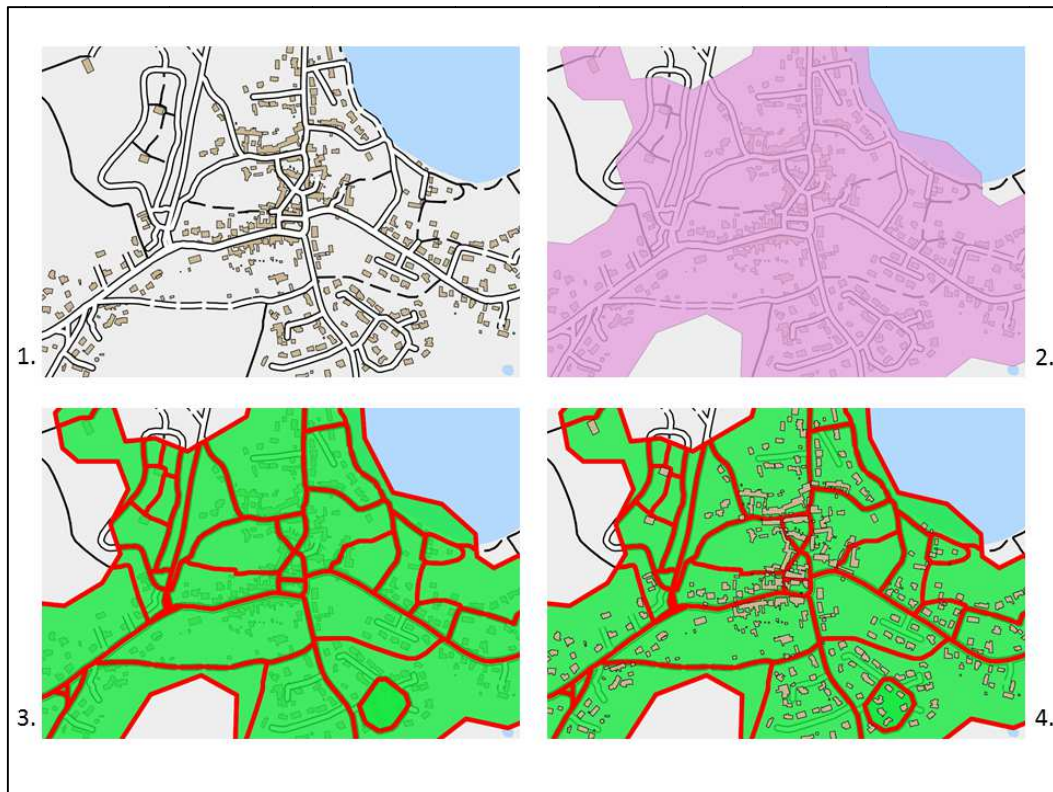


Figure 12 Enrichissement d'une zone urbaine, d'après Boffet (2002) [8] : 1. Données initiales ; 2. Création de la ville ; 3. Identification des îlots ; 4. Ajout des bâtiments dans les îlots.

Dans un premier temps, les bâtiments permettent d'identifier une surface représentant la ville, sur la base d'une extension de leur géométrie par dilatation morphologique. Puis, le réseau routier permet le découpage de cette surface en îlots urbains. Les îlots urbains, sur la base d'informations sur les bâtiments qui le composent (par exemple, leur taille ou leur forme), peuvent être caractérisés (par exemple îlot résidentiel ou îlot industriel). De fait, la création de ces entités amène une hiérarchisation entre elles. Ainsi, une ville est composée d'îlots, eux-mêmes composés de bâtiments. Les tronçons routiers, dans cette organisation, peuvent être considérés comme éléments appartenant à la ville, et éléments appartenant aux îlots urbains en tant que frontières. D'autres objets peuvent être considérés dans ce genre d'analyse. Par exemple, les espaces vides au sein d'une zone de bâtiments, ou encore les alignements de bâtiments qui sont des structures importantes jouant un rôle lors de la perception visuelle d'une carte (Boffet, 2002) [8].

### **La géométrie**

Les objets géographiques sont porteurs d'une géométrie, pouvant être définie de façon vectorielle (points, polygones, polygones). Les premiers algorithmes visant à automatiser la généralisation agissent directement sur cette géométrie. Ils peuvent agir directement sur les points de ces géométries, ou intervenir en tenant compte des segments par exemple. Ces algorithmes interviennent avec une logique propre à chacun et au type d'objet à généraliser. Parfois, ces algorithmes agissent seulement sur des parties de ces géométries, ou de façon

graduée. Les approches utilisant les moindres carrés (Harrie et Sarjakoski, 2002) [45] ; Sester, 2005) [102], ou encore des modèles comme GAEL peuvent intervenir de façon localisée.

### ***Bilan***

L'analyse de l'espace géographique met en avant plusieurs niveaux d'objets géographiques, qu'ils soient liés à la décomposition d'objets existants, ou à la composition à partir de ces objets existants. La généralisation amène donc à réfléchir sur plusieurs niveaux selon le cas étudié. Dans la partie suivante, nous nous intéressons à la façon dont la prise en compte de plusieurs niveaux peut être effectuée au sein d'un même processus.

### **II.2.3 Les aspects multi-niveaux dans les SMA de généralisation existants**

Dans la section I.2 **Les systèmes multi-agents pour la généralisation** nous avons décrit les différents modèles SMA pour la généralisation, en mettant en avant leurs propriétés liées aux aspects multi-niveaux, tant pour ce qui est des relations statiques (liées à l'organisation des agents entre eux), que des relations dynamiques (liées à la façon dont ces agents fonctionnent). Nous revenons ici sur ces caractéristiques de façon plus précise.

#### ***Relations hiérarchiques en arborescence***

Les modèles SMA proposent souvent des relations hiérarchiques arborescentes. Ce choix de structuration facilite l'orchestration des agents, l'activation des agents pouvant être systématiquement allouée à un supérieur hiérarchique. L'unicité de ce dernier et l'absence de négociation d'un point de vue transversal évitent de se confronter aux problèmes de convergence que soulèverait la gestion de plusieurs activateurs.

#### ***Relations transversales***

La modélisation de relations transversales avec des objets ayant un même niveau a été introduite par le modèle CartACom. Ces interactions transversales influent sur l'ordonnement des agents géographiques pour permettre à ces derniers de dialoguer efficacement entre eux. Un agent avec lequel une conversation est commencée doit être activé pour lui permettre de donner suite à cette conversation. L'existence de ce type d'interactions est important lors des considérations multi-niveaux, car elle soulève des questions d'ordonnement importantes, lorsqu'il s'agit de prendre en compte dans un même cadre des relations hiérarchiques.

#### ***Relations non arborescentes***

Nous l'avons constaté, un objet géographique peut appartenir à plusieurs organisations hiérarchiques. La limitation des modèles hiérarchiques purs implique une arborescence entre agents, avec un unique parent. Si nous voulons permettre la coexistence de hiérarchies multiples, nous devons identifier une structure plus souple. Si nous voulons permettre la cohabitation de plusieurs modèles, décrivant des relations hiérarchiques distinctes, nous devons nous intéresser aux relations non arborescentes.



## **II.2.4 Les aspects multi-niveaux non gérés**

Un des objectifs de la thèse présentée est de résoudre de nouveaux cas de généralisation induits par l'importance croissante de données thématiques dans le cadre de l'information géographique, et par la construction de carte à la demande. L'ajout de données thématiques, en parallèle de données topographiques, amène à considérer des relations inédites. De plus, des cas de généralisation impliquent de prendre en considération, au sein d'un même modèle, plusieurs type de relations, qu'elles soient hiérarchiques ou transversales. Nous présentons ici un ensemble de ces situations.

### ***Relation de position sur un objet support***

Dans les données géographiques, il existe des situations où un objet voit son positionnement défini par un autre objet ou un ensemble d'autres objets (nous renvoyons à Jaara (2015) [50] pour plus de précisions). C'est le cas par exemple des itinéraires par rapport à un réseau le supportant. Un itinéraire (que ce soit un itinéraire de randonnée pédestre, équestre, cycliste, etc... ou le parcours d'une ligne de bus), peut être défini géographiquement une liste des tronçons que l'utilisateur parcourant cet itinéraire doit emprunter. Cela amène à considérer la relation d'un objet géographique (ou un ensemble d'objets) avec d'autres objets qu'il supporte. Nous voyons dans ce genre de situation une relation hiérarchique, dans la mesure où un objet en porte un autre. Néanmoins ce genre de relation hiérarchique est différent des relations de composition présentées jusqu'à maintenant.

### ***Relation de positionnement dans un voisinage***

Un autre cas est le positionnement d'objet dans un voisinage d'un autre. Par exemple, les bâtiments longeant une rue doivent rester dans le voisinage de celle-ci, de même, certains points d'intérêt (par exemples des arrêts de bus dans la Figure 13) doivent garder leur positionnement relatif à un réseau de routes ou de chemins afin d'exprimer des relations importantes pour l'accès à ce point. Lors des différents déplacements des objets durant le processus de généralisation, ces relations peuvent être perdues. La conception de cette relation de voisinage doit donc être intégrée dans un processus de généralisation automatisée afin d'être préservée.

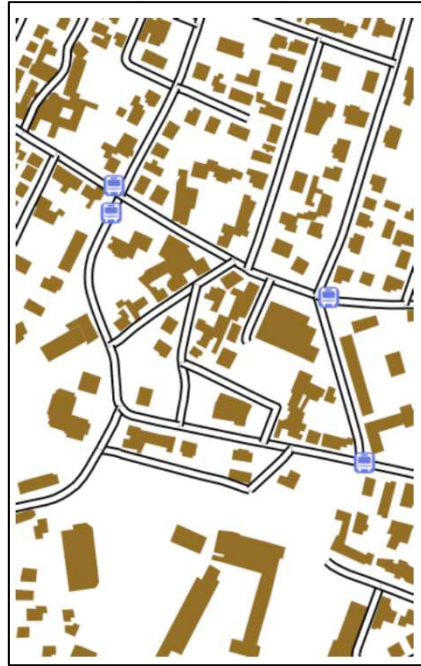


Figure 13 Exemple de carte avec arrêts de bus indiqués. La position des arrêts de bus au sein du réseau routier est importante, ce qui invite à voir une relation hiérarchique de situation entre le voisinage du réseau routier et les arrêts des bus.

### **Mélange des relations multi-niveaux**



Figure 14 Exemples de hiérarchies multiples : à gauche un bâtiment composant plusieurs alignements ; à droite un pont situé sur une route et sur une rivière.

Les modèles existant proposent des modélisations différentes pour des situations qui leur sont propres. Or, le processus de généralisation peut être confronté à des situations impliquant des situations hybrides de modèles différents. Par exemple, nous avons vu que les îlots urbains et les bâtiments avaient une relation hiérarchique, mais des bâtiments d'îlots

différents peuvent avoir des relations entre eux. Autre cas, des bâtiments accolés pourront être considérés individuellement, ou comme un tout selon les situations (Figure 15). Ces situations nous amènent à considérer deux caractéristiques importantes souhaitées pour l'évolution des modèles pour la généralisation automatique : la possibilité pour un objet de s'inscrire dans plusieurs hiérarchies (Figure 14) et la possibilité pour un objet d'interagir conjointement avec des objets d'un niveau donné et d'un niveau supérieur, situation que nous appelons relations diagonales.

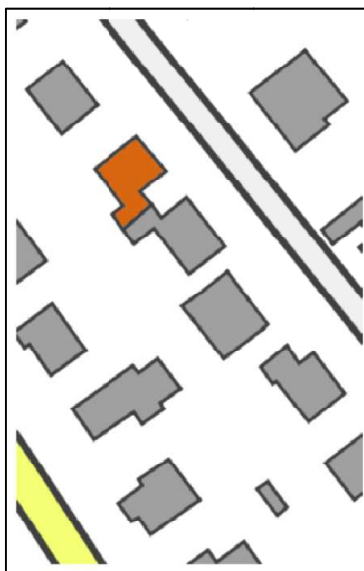


Figure 15 Exemple de bâtiments adjacents ayant une symbolisation différente. Les deux bâtiments peuvent être amenés à être considérés comme deux bâtiments indépendants, ou comme un unique agrégat selon les situations.

### II.2.4 Synthèse

Nous avons vu dans ce chapitre plusieurs types de relations qu'il est nécessaire de prendre en compte dans le cadre de la généralisation cartographique. Nous pouvons discerner trois types de relations hiérarchiques (relation de situation, relation de composition et relation de composition géométrique). La notion de hiérarchie a notamment été étudiée par Lane (2006) [55], et par Gibson *et al.* (2000) [42]. Lane (2006) [55] discerne quatre types de hiérarchies mettant en avant différentes caractéristiques associées à ce terme polysémique :

- les *hiérarchies d'ordonnement* : la hiérarchie se résume d'une façon abstraite et minimaliste comme une relation d'ordre entre entités ;
- les *hiérarchies d'inclusion* : la hiérarchie se décrit d'une façon plus précise qu'une notion d'ordre, en s'appuyant sur une image de boîtes emboîtées, une boîte contenant un ensemble de boîtes pouvant elles-mêmes contenir d'autres boîtes ;
- les *hiérarchies de contrôle* : la hiérarchie est décrite comme une relation de contrôle d'une entité sur une autre ;
- les *hiérarchies de niveau* : la hiérarchie est décrite comme la relation de plusieurs niveaux en relation. Ces relations impliquent des *causalités ascendantes*, lorsque les éléments d'un niveau inférieur impliquent une création du niveau supérieur, et des

*causalités descendantes* lorsque l'inclusion d'un niveau dans un autre modifie le niveau le plus bas.

La hiérarchie de niveau est la plus souple, que ce soit en termes de description de relations spatiales, ou en termes de relation comportementale (avec une influence ascendante, absente de la hiérarchie de contrôle). Toutes les relations hiérarchiques que nous avons observées précédemment peuvent s'inscrire dans cette définition. Gibson *et al.* (2000) [42] définissent eux trois types de hiérarchies :

- les *hiérarchies exclusives* : le lien est uniquement défini en termes d'interactions, et ne considère pas de notion d'inclusion. Ce type de hiérarchie caractérise les relations de commandement par exemple ;
- les *hiérarchies inclusives* : le lien est un lien d'inclusion : l'élément en position supérieure inclut l'élément inférieur ;
- les *hiérarchies constitutives* : le lien implique une définition du niveau supérieur par les éléments sur lesquels il est en position hiérarchique.

Pour notre analyse, ils permettent de mettre en évidence les différents types de relation qui sont nécessaires pour la généralisation automatisée. Nous en proposons une liste dans les paragraphes ci-dessous.

### ***Relation de composition***

Lorsque des objets géographiques peuvent être regroupés pour constituer d'autres objets géographiques, nous proposons de parler d'un lien de composition. La dimension constitutive (Gibson *et al.*, 2000) [42] de cette relation hiérarchique est mise en évidence par la dépendance de l'objet composé vis-à-vis de ces composants dans leur ensemble ; tandis que la dimension exclusive apparaît dans la prise de décision unilatérale du niveau supérieur.

### ***Relation de composition géométrique***

Ce type de relation peut être considéré comme un lien de composition normal, où l'organisation des éléments inclus est particulière. Néanmoins, la vision des points de la géométrie comme éléments autonomes de celle-ci n'est pas pertinente (contrairement, par exemple, à un bâtiment d'un îlot qui existe indépendamment de l'îlot). Cela nous amène à voir ce type de relation comme un cas particulier de relation de composition, avec la contrainte supplémentaire que les éléments inférieurs n'existent que pour définir l'élément supérieur.

### ***Relation de situation***

Ce type de relation, restant à gérer dans les modèles de généralisation automatisée, s'inscrit dans le cadre d'une hiérarchie inclusive. Elle implique qu'une relation soit identifiée entre les éléments, et que la géométrie de l'élément inférieur dépende de l'élément supérieur. Bien que l'existence de l'élément supérieur ne dépende pas de l'élément inférieur, ce dernier peut amener l'élément supérieur à se comporter différemment.

### ***Relations hiérarchiques multiples***

La complexité des relations entre objets géographiques amène à voir les organisations arborescentes comme trop limitatives. L'évolution des modèles va dans le sens de la prise en compte de relations hiérarchiques multiples.

### ***Relations diagonales***

Les relations diagonales sont une conséquence particulière des relations hiérarchiques multiples, si nous ne mettons aucune restriction sur les relations transversales. Cette situation correspond au cas où un objet va être en relation à la fois avec un autre objet, et un objet en relation hiérarchique avec cet autre objet.

## **II.3 : Les SMA multi-niveaux : vue d'ensemble**

Les problématiques multi-niveaux sont actuellement en plein essor dans la littérature scientifique portant sur les SMA. Cette littérature porte pour beaucoup sur des modèles de simulation, à l'exception de l'un d'entre eux portant directement sur la résolution de problèmes. Plusieurs analyses transversales des modélisations multi-niveaux ont été effectuées :

- Morvan (2013) [75] liste un ensemble de domaines de simulation variés utilisant des relations multi-niveaux: le biomédical, la biologie, l'écologie, les simulations sociales, ou encore la modélisation de flux.
- Gil-Quijano *et al.* (2009) [84] proposent de comparer les fonctionnements multi-niveaux dans plusieurs cas de simulations.
- Ratzé *et al.* (2007) [85] analysent les différentes façons d'organiser des éléments dans le cadre de hiérarchies, avec des modèles variables, parmi lesquels des modèles orientés agent.

Dans cette section, nous revenons sur ces modèles multi-niveaux, en mettant en perspective leurs spécificités avec celles issues de la généralisation décrites précédemment.

### **II.3.1 Modèles multi-agents multi-niveaux**

Des modèles multi-niveaux ont été proposés pour plusieurs simulations dans des domaines différents. L'aspect multi-niveaux est à chaque fois motivé par l'existence de connaissances sur plusieurs niveaux. Dans cet état de l'art, nous avons organisé les différents cas étudiés selon la motivation qui amène à considérer plusieurs niveaux : l'existence de connaissances variées et hétéroclites dans le domaine étudié, la volonté de regrouper des agents en groupes devant être réifiés sous forme d'agent avec un comportement propre, et les cas hybrides s'appuyant sur les deux précédentes motivations. Ce regroupement, dans un premier temps arbitraire, permet de mettre en avant des structures multi-niveaux communes indépendantes dépassant ces cadres.

---

### **Modèles multi-niveaux avec sources de connaissances multiples**

Dans de nombreux domaines d'application, la volonté d'exprimer différents niveaux au sein d'un SMA est motivée par la multiplicité des domaines de connaissance. Le cas des simulations de modèles biologiques illustrent parfaitement cette situation. Des connaissances sur la modélisation de phénomènes ont été établies à plusieurs niveaux : cellules, organes, individus, etc. L'expression du comportement d'un niveau se fait selon des modèles différents. Dans certains cas, il apparaît comme non souhaitable de présenter le comportement d'un niveau donné comme l'agrégation des éléments du niveau inférieur (par exemple, le comportement d'un organe ne peut être décrit comme le résultat de l'agrégation des comportements des cellules). Dans ce genre de situation, la question du transfert d'informations d'un niveau à un autre se pose.

La simulation du comportement d'une cellule cancéreuse au sein d'un organisme est l'objet du modèle proposé par Zhang *et al.* (2011) [124]. Dans ce modèle, plusieurs niveaux de connaissances, avec des modélisations qui leur sont propres, sont utilisés : au niveau intracellulaire, au niveau cellulaire et au niveau des tissus. Les différents niveaux communiquent du bas vers le haut, les conséquences de la simulation au niveau intracellulaire influençant le niveau extracellulaire, produisant lui-même des résultats exploités par le niveau du tissu. Une communication du haut vers le bas est aussi intégrée : les résultats au niveau du tissu influencent la simulation au niveau intracellulaire. Néanmoins, seul le niveau extracellulaire est modélisé par un modèle orienté agent, les cellules y étant modélisées sous-forme d'agents choisissant entre migrer ou proliférer. Les interactions entre niveaux s'inscrivent seulement dans un cadre hiérarchique, et les agents du niveau extracellulaire ne communiquent pas avec les autres niveaux. La situation décrite concerne les relations hiérarchiques montantes et descendantes.

La modélisation de flux est un exemple de domaine mettant en œuvre à la fois des connaissances à un niveau individuel et à un niveau collectif. Le comportement des entités individuelles (par exemple, des piétons ou des véhicules), se déplaçant sur un espace métrique peut être décrit par des SMA (par exemple, Treiber *et al.* (2000) [115]). Ce niveau est appelé niveau *micro* dans la littérature du domaine. Le flux en tant qu'entité a un comportement exprimé par des modèles d'équations (par exemple LWR de Lighthill et Whitham (1955) [58] et Richards (1956) [90], ou le modèle de Payne (1971) [81]). Ce niveau est appelé niveau *macro*. Ces modèles micro et macro ont pour la plupart été développés de façon indépendante. L'intérêt de modéliser les deux niveaux dans une même simulation est motivé par le fait que ces deux modèles sont efficaces sur deux cadres différents : les modèles micro dans les milieux urbains avec une forte densité de route, et les modèles macro sur les autoroutes.

El hnam *et al.* (2008) [28] proposent un modèle où les deux types de modèles sont appliqués sur des zones de jonctions entre les deux types de milieu. Sur ces zones hybrides, des fonctions de conversions sont proposées pour adapter les données d'un modèle à l'autre.

Sewall *et al.* (2011) [103] proposent un modèle intégrant aussi les deux types de régions, mais ajoutent la possibilité de définir dynamiquement les zones qui font l'objet d'une simu-

lation macro ou micro. Ainsi, en plus de proposer des méthodes d'adaptation aux zones limites, le modèle propose des méthodes de conversion lorsqu'une zone passe d'un modèle à un autre.

Vo *et al.* (2012) [117] proposent une simulation d'évacuation de populations en zone urbaine suite à une alerte au tsunami. Pour des raisons de performances, il est difficile de gérer les 350 000 personnes de la zone étudiée. Le modèle propose d'utiliser une modélisation de flux de piétons (Hughes, 2002) [48] pour simuler le déplacement des individus sur les grands axes. Une zone où le déplacement est modélisé par le modèle macro est modélisée sous forme d'agent, et l'agent gérant cette zone *capturera* et *relâchera* les agents entrant et sortant, c'est-à-dire qu'il transforme les agents en agents d'un autre type pour les soumettre au comportement macro ou micro, selon qu'ils sont capturés ou relâchés.

Abouaissa *et al.* (2013) [46] proposent le modèle JAM-FREE (pour *Java, Agent and Multi-level based Framework for Road-traffic Examination and Enhancement*), basé sur le méta-modèle SIMILAR présenté dans la sous-partie suivante. Le but est d'analyser le trafic routier et autoroutier, et d'étudier des mesures de régulation du trafic. Tout comme dans les modèles précédents, les différents niveaux identifiés concernent le type d'algorithme qui est utilisé pour modéliser le flux, à savoir une modélisation sous forme d'équation (permettant de simuler le trafic sur de grandes zones) et sous forme de modèles agents afin d'identifier des phénomènes de perturbations de la circulation.

Dans leur modélisation d'un écosystème, Duboz *et al.* (2002) [20] proposent de modéliser un même niveau de deux points de vue différents : une modélisation SMA où des animaux interagissent (les animaux sont des petits crustacés, des copépodes, et le phytoplancton), et une modélisation à base de système d'équations permettant de simuler l'évolution de la taille de populations de proies et de prédateurs (modèle Holling-Tanner (Holling, 1966 [47] ; Tanner, 1975 [110])). Le passage de la modélisation SMA à la modélisation à base d'équations se fait par l'intermédiaire de fonction d'émergences au sein du SMA pour déduire les paramètres du système d'équations.

### **Modèles multi-niveaux avec constitution dynamique de groupes**

Le regroupement d'agents en structure devant être réifiée apparaît pour plusieurs raisons, par exemple pour des questions de performances.

Le regroupement peut se faire en s'appuyant sur l'apparition de phénomènes émergents. L'émergence est un cadre de recherche important au sein des SMA. La détection des phénomènes émergents est un domaine vaste, notamment lorsqu'il s'agit de détecter un tel phénomène à des fins de mise en évidence et d'observation. Dans le cadre de l'étude des systèmes multi-niveaux, ce phénomène est particulièrement intéressant lorsqu'il conduit à la réification du phénomène émergent sous forme d'agent avec un comportement propre défini autrement que par le cumul du comportement des agents constituant le phénomène émergent.

Le phénomène de nuées d'oiseaux, ou *flocking*, est un phénomène régulièrement évoqué lorsqu'il est question de phénomènes émergents. Le comportement des oiseaux en tant qu'individus peut être modélisé selon un modèle orienté agent (Reynolds, 1987) [89]. L'identification de nuées s'effectue selon des relations de positionnement et de direction. Camus *et al.* (2012) [13] proposent un modèle de comportement de la nuée et une modélisation des échanges entre la nuée et les oiseaux qui la composent. Cette situation amène à une modélisation formelle décrite dans la sous-partie suivante.

RIVAGE (Servat *et al.*, 1998) [101] est un projet de simulation d'écoulement de fluides se basant sur une approche multi-agent proposant plusieurs niveaux d'exécution. Des agents boules d'eau se regroupent pour former des agents marres et ravines dont le comportement permet la reproduction du phénomène identique à une simulation où les agents n'auraient pas été regroupés. Il n'existe néanmoins pas de relation entre niveaux au sein d'une même modélisation.

David *et al.* (2011) [16] proposent de réifier les phénomènes émergents dans leur modélisation de l'évolution du territoire de la Réunion. Les agents y représentent différentes entités surfaciques (région, microrégion, commune, parcelle) qui interagissent pour simuler l'évolution de l'urbanisation de l'île. L'identification de zones urbaines, et leur réification sous forme d'agents y est proposée. Un *élément d'interposition* est créé lorsque qu'un phénomène émergent est réifié. Les objets constituant le phénomène agent réifié doivent alors interagir en se référant à cet élément d'interposition, qui modifiera alors le comportement des agents induits par l'existence du phénomène réifié.

Le modèle SIMULBOGOTA (Gil-Quijano *et al.* (2007) [43], analysé et comparé par Gil-Quijano *et al.* (2009) [83]) cherche à simuler les déplacements des populations d'une ville, Bogota, au cours du temps. La simulation du comportement de déplacements au niveau individuel (personnes, ménages, logements) est présenté comme impossible à simuler, en raison de la complexité des comportements et l'importance de la population. Le modèle propose donc de modéliser le comportement de groupes de ménages et de groupes de logements. Ces groupes sont constitués dynamiquement sur la base d'un mécanisme construit à partir de données réelles. Les comportements de ces groupes sont gérés en s'appuyant sur des hypothèses sociologiques. Un mécanisme agissant au niveau global, tenant compte de l'ensemble des individus des populations (ménages et logements), permet de simuler l'évolution de la population dans le temps en créant et supprimant des entités individuelles.

### ***Modèles multi-niveaux avec constitution dynamique de groupes et sources de connaissances multiples***

Lepagnot et Hutzler (2009) [57] proposent une simulation orientée agent permettant d'étudier l'évolution de cellules cancéreuses selon l'influence d'une protéine particulière (la molécule PAI-1). Le modèle a été analysé par Gil-Quijano *et al.* (2009) [83], qui ont mis en avant ses caractéristiques multi-niveaux. Le modèle simule le comportement des cellules et des molécules, et cherche à identifier la mécanique des cellules susceptibles d'échapper au tissu cellulaire cancéreux, la tumeur, et d'être à l'origine de métastases, c'est-à-dire des tu-

---



meurs ailleurs dans l'organisme. Les molécules et les cellules interagissent à un même niveau, et la tumeur est un objet émergent qui est seulement observé. Néanmoins, pour des soucis de performance, le modèle intègre un niveau intermédiaire, le cœur de la tumeur, où des cellules susceptibles de migrer ont peu de chance d'être observées. Le comportement des cellules et molécules situées dans ce nouvel objet, dont l'émergence est identifiée dynamiquement, est simulé par un système d'équations différentielles. Néanmoins, les interactions entre le cœur de la tumeur et les objets situés à l'extérieur sont aussi gérées, pour permettre l'entrée et la sortie des molécules et des cellules. Nous observons ici un exemple d'interactions diagonales.

Le modèle SIMPOP3, également analysé par Gil-Quijano *et al.* (2009) [83], s'inscrit dans la lignée des modèles Simpop cherchant à simuler la croissance urbaine sur plusieurs siècles. SIMPOP3 couple les comportements des modèles SIMPOP2 (Pumain *et al.*, 2009) [82] et SIMPOPANANO (Louail, 2009) [59]. Ces modèles s'appuient sur les interactions entre, d'une part des entités spatialisées (villes, quartiers), et d'autre part des entités fonctionnelles, pouvant être associées à plusieurs entités spatialisées (elles-mêmes pouvant être associées à plusieurs entités spatialisées). SIMPOP2 simule le comportement des villes déjà identifiées, et vise à faire émerger les fonctions de celles-ci. SIMPOPANANO décrit l'évolution de fonctions au sein d'un espace urbain, et permet l'identification d'une ville. SIMPOP3 intègre la ville produite par SIMPOPANANO pour l'intégrer dans le modèle SIMPOP2. Cette situation exhibe une situation d'un niveau émergent réifié sous forme d'agent. La problématique consiste à identifier comment les interactions entre villes au niveau de SIMPOP2 sont intégrées au sein d'une ville au niveau SIMPOPANANO. Du fonctionnement interne de SIMPOPANANO, des informations utiles pour SIMPOP2 sont déduites. Ainsi, la relation liant la ville de SIMPOP2 et la ville émergente de SIMPOPANANO est le seul pont entre les deux modèles.

### II.3.2 Modèles formalisés pour les SMA multi-niveaux

La multiplicité des situations multi-niveaux a conduit à la volonté d'établir des modèles formalisés permettant de concevoir des simulations sur la base de connaissances génériques. Plusieurs propositions ont été faites, dont certaines sont encore en cours de développement.

La plateforme SWARM (Minar *et al.*, 1996) [73] propose une approche multi-niveaux avec agents imbriqués. Les agents appartiennent à un essaim (*swarm* en anglais), qui joue le rôle de planificateur et qui peut lui-même être un agent dans un autre essaim. Néanmoins la structure en arborescence du modèle ne permet pas à un agent d'appartenir à plusieurs essaims. De plus, le comportement d'un essaim est régi exclusivement par les phénomènes émergents des interactions entre agents le composant.

Le modèle AA4MM (pour *Agents and Artefacts for Multiple Models*), proposé par Siebert *et al.* (2011) [105], propose d'introduire la notion d'artefact pour gérer la communication entre deux modèles simulant un même phénomène. Cette idée d'objet servant d'intermédiaire se retrouve par exemple avec le concept de « pont » proposé par Seck et Honig (2012) [100] dans une approche multi-perspective, pouvant être associée à une approche multi-modèle.

Néanmoins, AA4MM propose aussi des artefacts de couplage reliant un agent d'un modèle avec un agent de l'autre modèle. La fonction de cet artefact de couplage est de transmettre des informations spécifiques à l'agent d'un modèle à un autre. Le modèle de flocking (Reynolds, 1987) [89] présenté précédemment est un exemple d'utilisation de ces artefacts de couplage, pour le cas particulier des phénomènes émergents. Camus *et al.* (2012) [13] proposent un début de formalisation pour les problèmes impliquant l'utilisation parallèle de deux modélisations à deux niveaux différents, notamment en proposant de définir des fonctions d'émergence et d'immersion notées respectivement  $e$  et  $i$  permettant de maintenir la cohérence entre une entité et l'ensemble des entités situées à l'intérieur, et l'intégration de ces fonctions dans les artefacts de AA4MM.

GEAMAS-NG (David *et al.* 2011) [16] est un modèle s'appuyant sur GEAMAS (Marcenac *et al.* 1998 [62] ; Marcenac et Giroux, 1998 [63]), généralisant la proposition d'agent réifié couplé à un élément d'interposition décrit précédemment, lors de la présentation du modèle de David *et al.* (2011) [16].

AGRE (pour *Agents-Groupes-Rôles + Environnements*) (Ferber *et al.* 2005) [31] est un modèle proposant de structurer un système multi-agents autour de différents concepts disjoints les uns des autres. Dans le modèle antérieur AGR (Ferber et Gutknecht, 1998) [30], un agent est une entité pouvant appartenir à un ou plusieurs groupes dans chacun desquels il joue un ou plusieurs rôles. Dans AGRE, la notion d'environnement a été ajoutée au modèle pour permettre des interactions dépendantes du contexte spatial commun aux agents au lieu de leur rôle au sein du groupe. La partie structurelle de ce modèle n'est pas explicitée et ne propose pas de gestion de l'imbrication des environnements et il n'est pas prévu de permettre à des agents d'encapsuler des environnements.

La plateforme GAMA (Drogoul *et al.*, 2013 [18] ; Taillandier *et al.*, 2010 [109]) est une plateforme générique proposant une modélisation multi-agents facilitée avec un langage dédié à la simulation. Une extension, GAML, permet de gérer des aspects multi-niveaux. Le modèle de Vo *et al.* (2012) [117], dont un cas d'application a été présenté précédemment, propose de changer le type d'agent lorsqu'un agent change d'environnement, par les phénomènes de capture et de relâchement.

SIMILAR (pour *Simulations with Multi-Level Agents and Reactions*) (Morvan et Jolly, 2012 [76] ; Morvan *et al.* 2011 [77] ; Soyez *et al.* 2012 [106]) est un méta-modèle multi-agents multi-niveaux s'appuyant sur le modèle IRM4S (Michel, 2007) [71]. Il propose une description par niveau, avec la possibilité de lier un agent à aucun, un ou plusieurs niveaux. Les niveaux sont reliés entre eux selon des relations d'influence, si d'un niveau, un agent peut interagir sur un autre niveau, et des relations de perception, si d'un niveau, un agent peut percevoir un autre niveau. Le comportement des agents est formalisé sous forme de fonctions composites pour chaque couple agent/niveau. L'application de l'ensemble de ces fonctions permet d'évaluer l'état du système au prochain pas de temps.

Le modèle SIMLAB (Huraux *et al.*, 2014) [49] est un modèle permettant des relations entre niveaux d'agents. Les agents ont une relation avec l'ensemble de leurs super-agents et une

---

autre relation avec l'ensemble de leurs sous-agents. Afin d'identifier la nature du lien hiérarchique qui réunit plusieurs agents différents, la notion d'axe de modélisation est proposée. L'existence d'un axe pour chaque type de hiérarchie permet la coexistence de plusieurs hiérarchies. Le modèle propose aussi une création dynamique de groupes.

Par ailleurs, le modèle PADAWAN est un modèle multi-niveaux que nous présentons dans la section précédente.

### II.3.3 Synthèses des caractéristiques des modèles multi-niveaux

Les modèles étudiés présentent des caractéristiques multi-niveaux variées. Nous synthétisons dans le Tableau 2 les caractéristiques observées. Nous soulignons que le fait que ce tableau met en évidence les caractéristiques observées dans des cas d'étude d'applications pour lesquels ils ont été mis en œuvre et non les capacités potentielles des modèles sous-jacents.

Modèles	Interactions ascendantes	Interactions descendantes	Relation de composition	Relation de situation	Cohabitation de hiérarchies	Identification dynamique de groupes
Zhang et al.	Transfert d'informations	Transfert d'informations	Oui	Non	Non	Non
El hnam et al.	Transfert d'informations	Transfert d'informations	Non	Oui	Non	Non
Sewall et al.	Transfert d'informations	Transfert d'informations	Non	Oui	Non	Non
Vo et al.	Transfert d'informations	Transfert d'informations	Non	Oui	Oui	Non
JAM FREE	Transfert d'informations	Transfert d'informations	Oui	Oui	Non	Non
Duboz et al.	Non	Transfert d'informations	Oui	Non	Non	Non
Flocking avec AA4MM	Transfert d'informations	Transfert d'informations	Oui	Non	Non	Oui
RIVAGE	Non	Non	Oui	Non	Non	Non
David et al.	Transfert d'informations	Transfert d'informations	Oui	Non	Oui	Oui
SIMULBOGOTA	Échanges dynamiques	Non	Oui	Non	Non	Oui
Lepagnot et al.	Non	Non	Non	Non	Non	Non
SIMPOP3	Échanges dynamiques	Échanges dynamiques	Oui	Non	Non	Oui

Tableau 2 Propriétés multi-niveaux observées dans différents modèles.

Nous observons dans ce tableau comparatif que certaines caractéristiques sont plutôt rares (comme la cohabitation de hiérarchies). Enfin, aucune caractéristique n'est partagée par tous les modèles proposés. Ainsi, la cohabitation de toutes ces caractéristiques reste une question ouverte.

## II.4 Caractéristiques de PADAWAN et adéquation pour nos modèles

Notre problématique s'inscrit dans une logique multi-niveaux, dans la mesure où un processus de généralisation doit pouvoir tenir compte de plusieurs points de vue complémentaires et de la notion d'abstraction pour agir efficacement. Prendre en compte, de façon formelle et explicite ces différents niveaux est l'intuition à l'origine de cette thèse. PADAWAN (Mathieu et Picault, 2011) [64] est un modèle choisi comme support possible d'une modélisation multi-niveaux pour la généralisation. Ce modèle, de part sa gestion de hiérarchies multiples ne s'appuyant pas sur une structure arborescente peut en effet permettre d'aider à l'expression des différents niveaux d'abstraction et leurs relations.

Dans cette section nous exposons dans un premier temps les caractéristiques du modèle PADAWAN, en mettant l'accent sur le vocabulaire propre au modèle. Ce modèle étant conçu et validé pour la simulation, nous nous interrogeons ensuite sur les similitudes et différences entre le domaine de la simulation et celui de la résolution de problème spatialisé contraint dans lequel s'inscrit la généralisation. Enfin, nous terminons par une identification des adaptations nécessaires à PADAWAN pour répondre aux problématiques spécifiques de la généralisation automatisée.

### II.4.1 Le modèle PADAWAN

PADAWAN (acronyme pour *Pattern for Accurate Design of Agent Worlds in Agent Nests*) est un modèle multi-agents multi-niveaux conçu pour la simulation. Ce modèle étend le modèle de simulation IODA de Kubera *et al.* (2008) [54].

IODA (pour *Interaction-Oriented Agent Simulation*) est un modèle de simulation multi-agents proposant une expression du comportement des agents s'appuyant sur la notion d'**interaction**. Une interaction décrit comment et sous quelles conditions un agent peut interagir dans son environnement avec les autres agents qui s'y trouvent. La réalisation d'une interaction va impliquer plusieurs agents, en se référant à des **primitives** propres aux agents impliqués. Les agents impliqués dans une interaction ont deux rôles différents : les **sources**, qui réalisent l'interaction, et les **cibles** qui la subissent. Lorsqu'une interaction implique l'agent source comme unique cible, il est question d'**interactions réflexives**. Les conditions à la réalisation d'une interaction sont décrites d'une part par ses **préconditions**, c'est-à-dire l'ensemble des conditions devant être satisfaites pour qu'une interaction se produise, et d'autre part par les **déclencheurs**, c'est-à-dire les motivations des agents à l'exécution de l'interaction. L'expression d'une interaction est donc constituée de l'expression de son déclencheur, de ses préconditions et de l'action à effectuer.

Les agents sont regroupés par **familles**, qui sont des « ensembles abstraits d'agents, au sein desquels les agents partagent tout ou une partie de leurs propriétés, de leurs primitives d'action ou d'interaction, ou de leurs comportements. » (Kubera et al. 2008) [54]).

Afin de décrire le comportement des agents d'une famille donnée, IODA propose d'utiliser des **matrices d'interactions**. Les matrices d'interactions sont des tableaux à double entrée, où chaque ligne correspond à une famille d'agents-sources et où chaque colonne correspond à une famille d'agents-cibles, avec en plus une colonne correspondant aux interactions réflexives.

À chaque case de la matrice est attribuée un ensemble d'**assignments**. Une assignation est composée d'une interaction, associée à une **priorité** et à une **distance maximum**. La priorité permet de sélectionner une interaction prioritaire lorsque plusieurs peuvent être réalisées. La distance maximum séparant la source de ses cibles potentielles permet d'indiquer une condition d'exécution de l'interaction propre à l'environnement et au positionnement des agents au sein de celui-ci.

Le modèle PADAWAN propose d'étendre le modèle IODA pour y inclure la possibilité de multiples **environnements**. La notion d'environnement, largement exploitée dans le domaine des SMA, a longtemps couvert des réalités variées : objet pour la description des connaissances d'un écosystème, support pour une infrastructure de communication, espace pour échange entre agents (Weyns et al., 2004) [121]. Mathieu et al. (2014) [65] propose une définition minimaliste essayant de couvrir cette diversité derrière la notion d'environnement. Nous nous appuyons sur cette définition :

**Définition** (d'après Mathieu et al., 2014) [65] : « Au sein d'un SMA, un *environnement* est une entité ayant deux fonctions :

1. placer des agents les uns par rapport aux autres, selon des relations de voisinage et d'accessibilité particulières ;
2. porter des informations correspondant à des niveaux n'ayant pas été agentifiés. »

PADAWAN propose d'abord une architecture permettant l'imbrication de plusieurs environnements, par l'intermédiaire des agents. Un environnement, dans PADAWAN, est **encapsulé** par un agent. Les agents qui évoluent dans un environnement sont dits **situés** dans celui-ci, et l'agent encapsulant l'environnement **héberge** les agents situés dans cet environnement. Si un agent encapsule un environnement, alors cet environnement est **inclus** dans l'environnement où l'agent est situé. Un même agent peut être situé dans plusieurs environnements différents, et, par conséquent, un environnement peut être inclus dans plusieurs environnements différents.

Une matrice d'interactions est attribuée à chaque environnement. Une ligne et une colonne supplémentaires sont ajoutées à ces matrices, afin de décrire les interactions impliquant l'agent encapsulant l'environnement.

Afin de gérer l'activation d'agents présents dans plusieurs environnements, le modèle propose de considérer un agent de façon multiple : le **noyau** de l'agent, décrivant les aspects non situés de l'agent, et les **faces** de l'agent, chaque face décrivant la partie de l'agent située dans un environnement. Chaque face d'un agent peut être considérée comme un agent pour ce qui concerne le cycle d'activation. Ainsi, au lieu d'activer un agent dans l'ensemble de la simulation, qui doit tenir compte de l'intégralité des possibilités d'interactions offertes dans chaque environnement où il est situé, la simulation active la face de l'agent dans chaque environnement.

La gestion du temps dans PADAWAN permet une gestion pas à pas. Pour chaque agent (ou face d'agent), il est possible de définir la régularité de son activation.

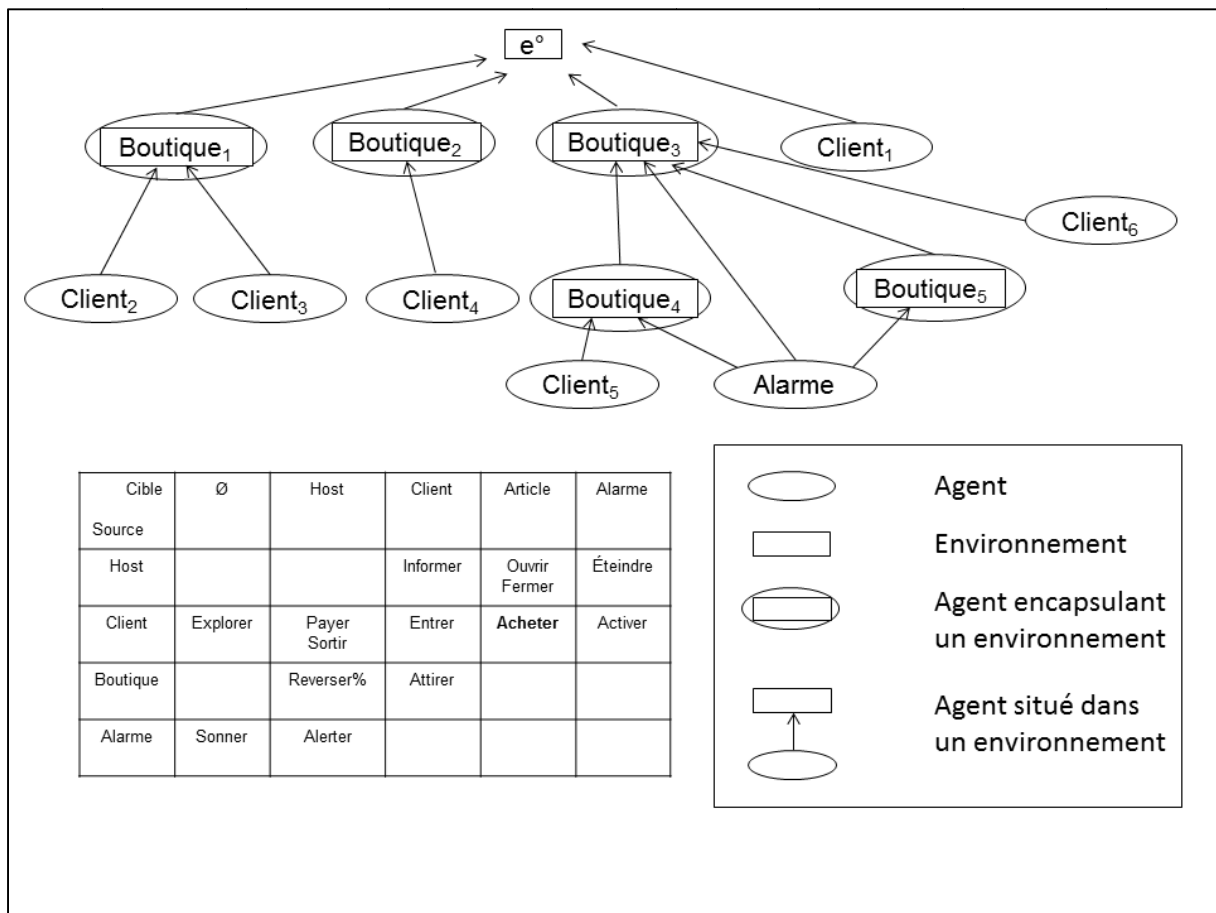


Figure 16 Exemple de modélisation de centre commercial avec boutiques intégrées les unes aux autres : en haut, l'organisation statique du modèle ; en bas à droite, la légende pour l'organisation statique du modèle ; en bas à gauche, un exemple de matrice d'interaction pour les environnements encapsulés par un agent de type **Boutique** : par exemple, un agent de type **Client** situé dans la boutique, peut dans cette boutique effectuer l'interaction **Acheter** avec un agent de type **Article**. (d'après Mathieu et Picault, 2011) [64].

## II.4.2 Différences entre simulation et résolution de problèmes

PADAWAN a été conçu pour de la simulation orientée agent. Le paradigme agent est pourtant utilisé pour la résolution de problèmes. Afin de permettre une sélection des caractéristiques de PADAWAN intéressantes pour notre problématique, nous identifions dans cette partie les caractéristiques propres à la résolution de problèmes.

### ***Méthode de conception et objectifs***

En simulation, la conception d'un modèle est motivée par une volonté de coller à la réalité du phénomène simulé. Ainsi, l'identification de biais de simulation est une étape importante de la conception d'une simulation multi-agents, et implique une observation d'un déroulement d'interactions. En résolution de problèmes, l'objectif n'est pas de reproduire un phénomène particulier, mais d'obtenir un résultat satisfaisant. La finalité de l'exécution en résolution de problèmes est donc l'objet d'une évaluation explicite, alors qu'en simulation, le résultat n'est évalué que durant l'évaluation et la calibration, et observé et considéré comme un résultat de recherche après.

### ***Cycle de vie***

La façon dont le cycle de vie d'un agent va être conçu en simulation est motivée par des notions de réalisme quant au déroulé d'une instance de la simulation. En résolution de problèmes, l'objectif est de résoudre le problème d'une façon efficace et efficiente, c'est-à-dire d'avoir le meilleur résultat, dans les meilleurs délais. Le déroulement du cycle de vie d'un agent n'est donc pas limité par des contraintes de réalisme, mais reste contraint par d'autres questions, comme savoir identifier quand il est pertinent de laisser la main à un autre agent pour résoudre des conflits, ou comment tester plusieurs possibilités, tout en anticipant suffisamment pour augmenter la probabilité d'atteindre une solution satisfaisante rapidement.

### ***Expression du temps***

D'une façon générale, en SMA, les questions liées au temps tournent en partie autour de la synchronisation des actions des agents. Dans le cadre des simulations multi-niveaux, cette synchronisation doit tenir compte de la variabilité des aspects temporels des actions effectuées selon les niveaux. Ces problématiques, régulièrement évoquées dans les différents modèles pour la simulation, ne se retrouvent pas dans les modèles de résolution de problèmes. L'important demeure l'ordre d'activation des agents, et des actions effectuées, les agents des différents niveaux devant collaborer pour résoudre le problème.

### **II.4.3 Adaptations nécessaires**

Les premières études menées sur l'application de PADAWAN à la généralisation automatisée (Atrash, 2011) [1] ont mis en évidence son adéquation à ces problématiques. La modélisation de l'inclusion des environnements les uns dans les autres offre la souplesse souhaitée pour modéliser la diversité des relations hiérarchiques qui nous intéressent, tandis que l'ergonomie de la définition des interactions permet d'exprimer tant les actions transversales (entre agents appartenant à un même environnement), que les influences hiérarchiques, qu'elles soient ascendantes ou descendantes. Néanmoins, le méta-modèle, conçu pour la simulation, nécessite des ajouts pour permettre d'exprimer ce que les modèles pour la généralisation automatique proposent.

### ***Conditions et contraintes***

Les contraintes sont des éléments importants des modèles de généralisation automatique. Dans tous les modèles orientés agents pour la généralisation, les contraintes sont explicitées. Leur rôle, en plus d'indiquer une évaluation locale de leur satisfaction, est de permettre de choisir les actions en fonction de leur état, grâce à des connaissances sur les liens et actions entre contraintes dont elles permettent a priori d'améliorer la satisfaction. La satisfaction des contraintes est considérée dans ces modèles comme la principale motivation pour les agents d'agir. Les connaissances sur l'impact des algorithmes sur les contraintes constituent aussi un critère influençant le choix de l'algorithme. Nous pouvons donc faire le rapprochement entre les relations liant contraintes à algorithmes et celles liant conditions (préconditions et déclencheurs) d'exécution d'interactions dans le modèle PADAWAN.

### ***Prise de décision et cycle de vie***

Dans les modèles pour la généralisation, plusieurs éléments en termes de décision sont définis dans le cycle de vie : choix d'action, annulation d'action, agrégation d'action. Ces caractéristiques doivent être prises en compte lors de la conception d'un cycle de vie utilisant les matrices d'interaction.

### ***Activation des agents***

Dans les modèles pour la généralisation, l'activation d'un ou plusieurs agents peut être motivée par la prise de décision d'un agent. Cela intervient notamment dans les relations hiérarchiques, quand un agent meso décide d'activer tous ses composants. Une telle situation peut aussi se produire dans le cadre de la simulation, notamment lors des interactions entre niveaux. Dans le cadre de la généralisation automatique, une modélisation plus souple de cet aspect serait plus indiquée.

## **II.5 Synthèse du chapitre**

Nous avons mis en évidence dans ce chapitre le fait que la généralisation est un problème multi-niveaux, type de problème d'importance croissante dans le cadre des SMA. L'état de l'art en modélisation multi-niveaux montre une grande diversité de relations multi-niveaux et d'interactions entre agents de niveaux différents. Mais aucun de ces modèles ne semble avoir la flexibilité nécessaire pour résoudre tous les problèmes posés par la généralisation cartographique. Nous avons présenté le modèle PADAWAN sur lequel nous nous appuyerons pour proposer un modèle multi-niveaux pour la généralisation automatisée, puis nous avons identifié les différences de conception entre un modèle conçu pour la simulation et un modèle conçu pour la résolution de problèmes, différences qu'il faudra prendre en compte pour l'adaptation de PADAWAN à la généralisation.





# Chapitre III :

## DIOGEN, un modèle multi-agents multi-niveaux pour la généralisation

---

Nous présentons dans ce chapitre un modèle permettant l'expression générique de relations multi-niveaux issues des modèles multi-agents pour la généralisation. Nous souhaitons proposer un modèle pour la généralisation tenant compte des aspects multi-niveaux entre agents et de les utiliser d'une façon souple et efficace. PADAWAN couvre ces aspects. Néanmoins, la généralisation automatisée présente des caractéristiques propres, qui impliquent une extension du modèle. Notre modèle s'appuie sur les principes de PADAWAN et intègre de nouvelles capacités et de nouveaux mécanismes propres à la problématique de la généralisation. Permettre d'exprimer les connaissances exploitées dans ces modèles est un objectif intermédiaire de notre approche. Pour cela, nous nous focalisons sur l'intégration des éléments des modèles AGENT, CartACom et GAEL, présentés dans la première partie du mémoire. Un autre objectif est la validation de nouveaux mécanismes par la vérification sur des cas d'étude de généralisation particuliers. L'ensemble de ces travaux nous amène à proposer un nouveau modèle, que nous appelons **DIOGEN** pour **Distributed Interaction Oriented GENeralisation**. Les principes de ce modèle ont fait l'objet d'une publication et d'une démonstration à la conférence PAAMS 2014 (Maudet et al. 2014a [70], Maudet et al. 2014b [69]).

Nous présentons dans une première section les bases de notre modèle, puis, dans une deuxième partie, nous décrivons comment nous reproduisons les modèles AGENT, CartACom et GAEL. Enfin, dans la troisième partie, nous explorons les nouvelles possibilités de notre modèle sur des cas d'applications concrets, avant de conclure dans une quatrième partie.

## III.1 Description du modèle DIOGEN

### III.1.1 Traduction dans le formalisme PADAWAN : interaction, agents, primitives

Plusieurs corrélations peuvent être facilement établies entre des concepts des modèles orientés agents pour la généralisation, et le modèle PADAWAN, au-delà des aspects multi-niveaux déjà présentés dans le deuxième chapitre.

#### *Actions et interactions*

Dans le cadre multi-agents, une action est un concept permettant de retranscrire l'ensemble des modifications effectuées par un agent qui se modifie lui-même et/ou son environnement. Les modèles pour la généralisation s'appuient sur des actions basées sur des algorithmes s'appliquant sur les agents géographiques à l'origine de leur activation. Les actions sont paramétrées par des éléments propres aux agents concernés, et par les contraintes responsables de l'activation des algorithmes. La Figure 17 illustre cette relation.

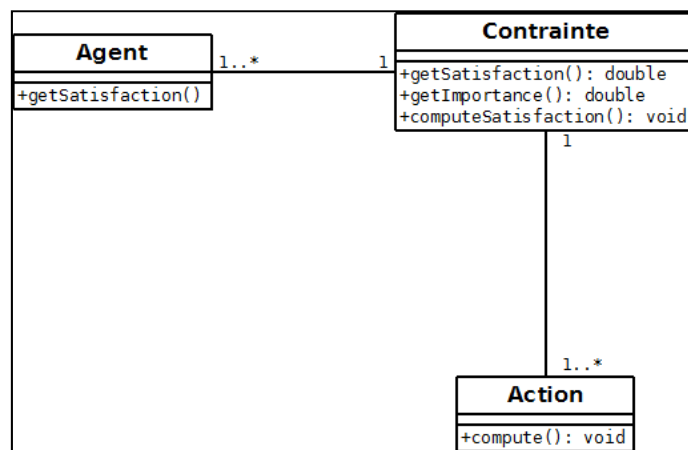


Figure 17 Diagramme de classes illustrant les relations entre agents, actions et contraintes.

Lorsqu'un agent est activé, il est susceptible de réaliser une ou plusieurs actions, avec des paramètres appropriés. Ces actions peuvent être assujetties à l'exécution des interactions du modèle PADAWAN. Dans le modèle PADAWAN, une interaction est exécutée par un agent particulier (la source) et peut impliquer d'autres agents (les cibles). Dans les modèles pour la généralisation, plusieurs situations peuvent être observées.

- Les actions impliquant un seul objet sont nombreuses (par exemples : algorithme de simplification de la géométrie (Lang, 1969 [56] ; Visalingam-Whyatt, 1993 [116]), remplacement par un rectangle ; voir chapitre I) et peuvent être décrites sous forme d'interactions réflexives dans PADAWAN.
- Des actions impliquant un deuxième agent se présentent dans CartACom, avec des agents communiquant et échangeant des messages deux à deux (par exemple, un algorithme de déplacement implique l'agent qui se déplace, mais prend aussi en

compte les agents de son environnement et en particulier le cas échéant celui qui lui a demandé de se déplacer).

- Des actions impliquant un ensemble d'agents. Dans le modèle AGENT, par exemple, cette situation se produit quand un agent d'un niveau donné intervient sur les agents qui le composent (par exemple, un îlot urbain qui sélectionne et supprime des bâtiments parmi l'ensemble qui le compose). PADAWAN permet de définir une interaction ayant un ensemble d'agents comme cible.
- L'activation d'agents par d'autres agents. Ce cas particulier se produit lorsqu'un agent déclenche la généralisation d'un ensemble d'autres agents qui le constituent. Ce genre de situation n'est pas explicitement décrit dans le modèle PADAWAN. Elles peuvent néanmoins l'être si nous définissons une interaction dont la fonction est l'activation d'un ensemble d'agents.

Ces différentes situations peuvent être représentées dans le modèle PADAWAN. Chaque algorithme peut être décrit sous forme d'interaction et lié aux agents par l'intermédiaire de matrices d'interactions. Nous rappelons que dans le modèle PADAWAN, une interaction est définie par trois éléments : l'action à effectuer, le déclencheur et les préconditions. Nous venons d'associer la première de ces notions aux actions des modèles pour la généralisation. Nous voyons dans la suite comment exploiter les deux derniers éléments.

### ***Contraintes, déclencheurs et préconditions***

En généralisation, les contraintes sont un élément important pour la description des attentes quant à la carte. Comme vu dans le chapitre I, il existe plusieurs typologies de contraintes. Dans les modèles multi-agents pour la généralisation, les contraintes permettent de mesurer à quel point une caractéristique importante pour un objet est satisfaite (par exemple, la densité d'un îlot urbain ou la taille minimum d'un bâtiment) : la notion de satisfaction permet d'exprimer cet aspect mesurable de la contrainte. Si une de ces contraintes n'est pas satisfaite, une action doit être appliquée. La non-satisfaction d'une contrainte peut donc être associée à la notion de déclencheur d'interaction de PADAWAN. Cette notion exprime la motivation de l'agent à effectuer une action (par exemple, dans le cadre d'une simulation, la faim déclenche l'action **manger** pour un animal). De même, dans les modèles pour la généralisation, à un type de contrainte non satisfait sont associées une ou plusieurs actions à essayer. Le déclencheur dans PADAWAN retourne une valeur booléenne. La mesure de la satisfaction d'une contrainte ayant une granularité variable, mais généralement non binaire, une discussion doit néanmoins se faire pour rapprocher les deux notions.

De plus, la présence d'une contrainte à respecter pour un agent peut être un indicateur sur la pertinence ou non d'utiliser un algorithme pour résoudre une autre contrainte (par exemple, la contrainte de maintien de la concavité peut s'opposer à l'utilisation d'un algorithme de « remplacement par un rectangle » sur un bâtiment). Ces connaissances peuvent être associées à la notion de précondition à respecter de PADAWAN, les préconditions exprimant un ensemble de critères devant être remplis pour que l'interaction puisse être exécutée.

L'importance des contraintes pour décider de l'exécution, ou non, d'interactions nous encourage à rendre explicites les contraintes dans notre modèle, ainsi que leur relation avec les interactions. Nous détaillons notre proposition dans la sous-section **III.1.2 Expression des contraintes, inversion de la perspective**.

### ***Cohabitation de niveaux***

Comme nous l'avons vu précédemment, l'existence de différents niveaux dans le domaine de la généralisation est un des principaux arguments à l'utilisation de PADAWAN pour évoquer une notion de situation d'un agent dans un environnement. Plusieurs contextes au sein desquels les agents sont amenés à interagir peuvent être décrits comme environnements du modèle PADAWAN. Dans AGENT, le cadre défini comme un meso peut définir un environnement, encapsulé par un agent meso. La relation meso/composant peut correspondre à une relation géographique entre le meso et ses composants (cas des bâtiments dans leur îlot urbain). Elle peut aussi coïncider avec une situation où les composants sont des sous-structures géométriques de leur meso (cas des routes subdivisées en sous-sections). Une relation géométrique est aussi identifiable dans la relation entre les agents points et les objets submicro avec leur agents déformables. La Figure 18 montre plusieurs types de relations hiérarchiques, assimilables à des relations entre environnements et agents de PADAWAN.

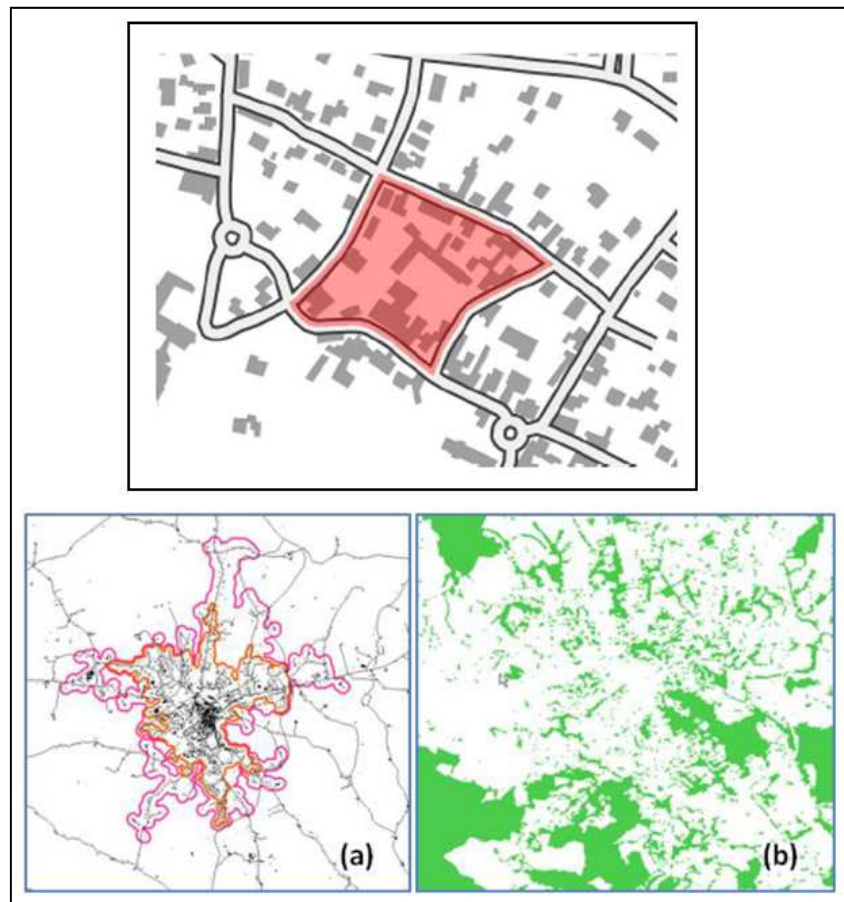


Figure 18 En haut : exemple de meso/îlot urbain (en rouge), et sa relation avec les bâtiments. En bas, exemples d'espaces géographiques, d'après Touya (2011) [113] : (a) espace urbain/périurbain/rural ; (b) espace thématique « végétation ».

### Un processus de décision

Le modèle PADAWAN propose une modélisation de cycle de vie ayant des caractéristiques propres à la simulation, permettant à un agent d'effectuer un choix durant son activation. Le principe de ces cycles de vie est de s'appuyer sur une modélisation discrète du temps, les agents exécutant leur cycle de vie chacun leur tour. Au cours d'un tel cycle de vie, l'agent va choisir parmi les interactions qu'il lui est possible de réaliser. Il peut être convenu qu'il exécute seulement une action, ou plusieurs. La Figure 19 illustre le cycle de vie d'un agent dans les modèles pour la généralisation. Dans les modèles pour la généralisation, un agent peut, dans son cycle de vie, réaliser plusieurs interactions. De plus, dans CartACom, il est possible de fusionner des actions réalisables. Cela concerne des interactions de même type (par exemple, lorsqu'une translation doit être effectuée, d'autres actions de déplacements possibles vont être considérées pour calculer le vecteur de déplacement). Nous proposons un cycle de vie exploitant le modèle de PADAWAN, mais reproduisant les possibilités des modèles pour la généralisation. Ce dernier sera exposé dans la section **III.3.1 Reproduction du comportement d'AGENT**. Un cycle de vie gérant les communications entre agents est aussi proposé pour intégrer la gestion des communications propres à CARTACOM (voir section **III.3.2 Reproduction du comportement de CartACom**).

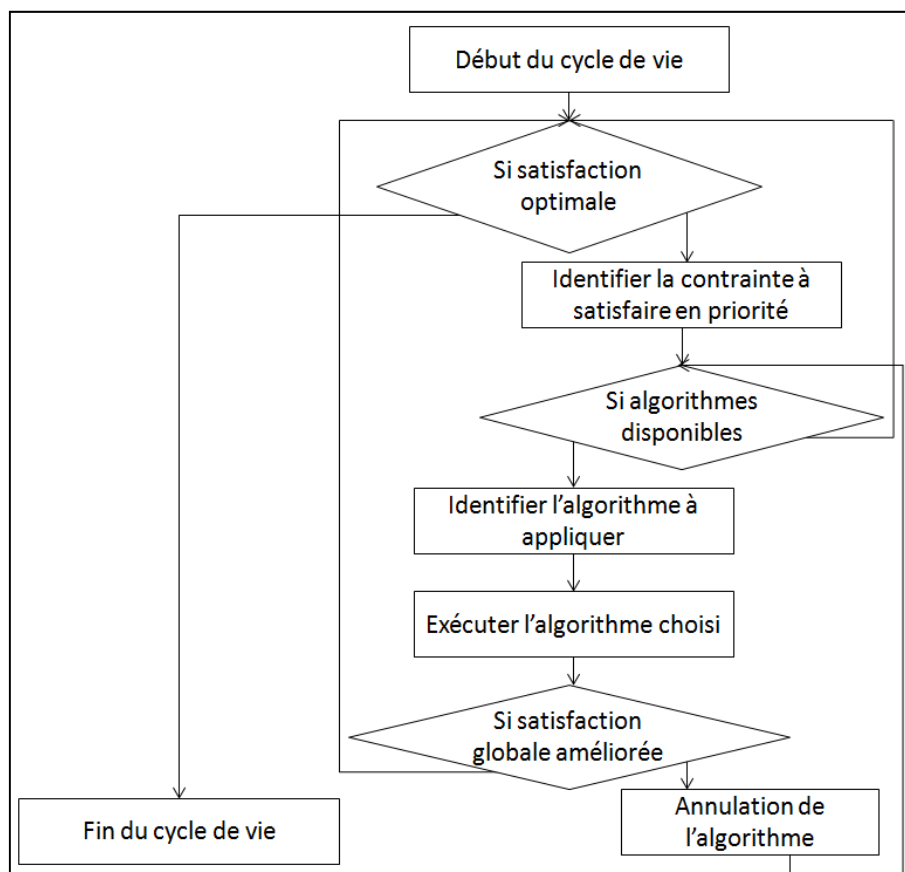


Figure 19 Cycle de vie simplifié du modèle AGENT. (Lors d'un test, la branche **vrai** est à gauche de la branche **faux**).

### III.1.2 Expression des contraintes, inversion de la perspective

Les modèles pour la généralisation s'appuient sur les contraintes pour choisir les interactions à effectuer. C'est d'abord la contrainte identifiée comme la plus urgente à satisfaire qui est perçue, puis c'est sur la base de connaissances établies sur cette contrainte que l'action à effectuer par l'agent est choisie. Nous souhaitons proposer une nouvelle façon d'identifier l'action à effectuer en prenant en compte toutes les interactions, et leur lien avec les contraintes. Un des buts est de permettre l'optimisation du choix des interactions en tenant compte de toutes les contraintes, ce qui a été mis en évidence par Taillandier (2008) [108]. Par cette démarche, nous nous approchons du modèle PADAWAN mettant l'interaction au cœur du modèle. Une telle adaptation consiste à inverser la perspective de fonctionnement du système. La Figure 20 montre le fonctionnement commun aux trois modèles agents pour la généralisation considérés (AGENT, CartACom, GAEL). Un agent effectue l'évaluation de la satisfaction de ses contraintes sur la base de sa situation actuelle. Il choisit ensuite la contrainte à satisfaire, sur la base de sa priorité (ou urgence) de traitement, de son importance (à être satisfaite) et de sa satisfaction. Sur la base de cette contrainte et de connaissances de l'impact des actions vis-à-vis de la contrainte, l'agent choisit une action à effectuer. Ce processus de sélection n'est pas explicite, propre à chaque contrainte et codé en dur à chaque situation

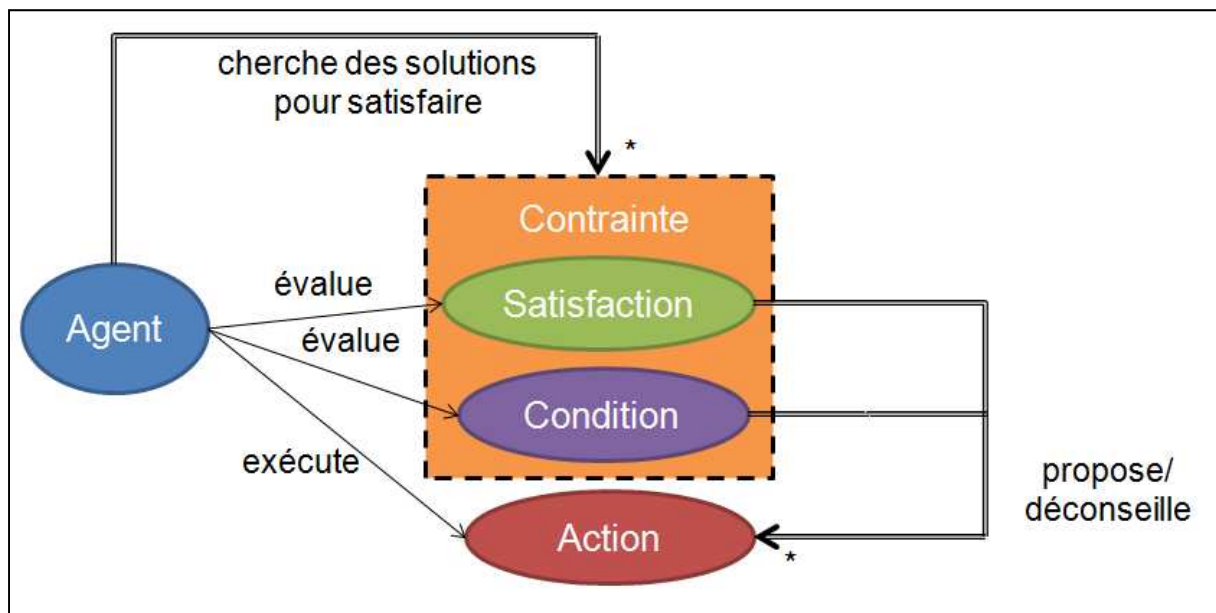


Figure 20 Description du fonctionnement des modèles pour la généralisation. Les agents cherchent à satisfaire des contraintes. Ces contraintes proposent ou déconseillent des actions pouvant être exécutées par les agents.

La Figure 21 montre le processus de choix d'interaction dans PADAWAN. Lorsqu'un agent est activé dans un environnement particulier, il identifie, sur la base de la matrice d'interactions de l'environnement en question, l'interaction à effectuer. Pour cela, il identifie toutes les interactions réalisables avec son voisinage, en évaluant les déclencheurs et préconditions de ces interactions. Par exemple, dans une simulation d'un écosystème, un animal a la possibilité d'effectuer des interactions différentes : se reposer, se nourrir d'une proie (a), se nourrir d'une proie (b), ou encore se déplacer. Il en choisit une selon ses motivations, exprimées par

ses déclencheurs (être fatigué, avoir faim, n'avoir aucun besoin à satisfaire dans l'immédiat), et des possibilités offertes par l'environnement

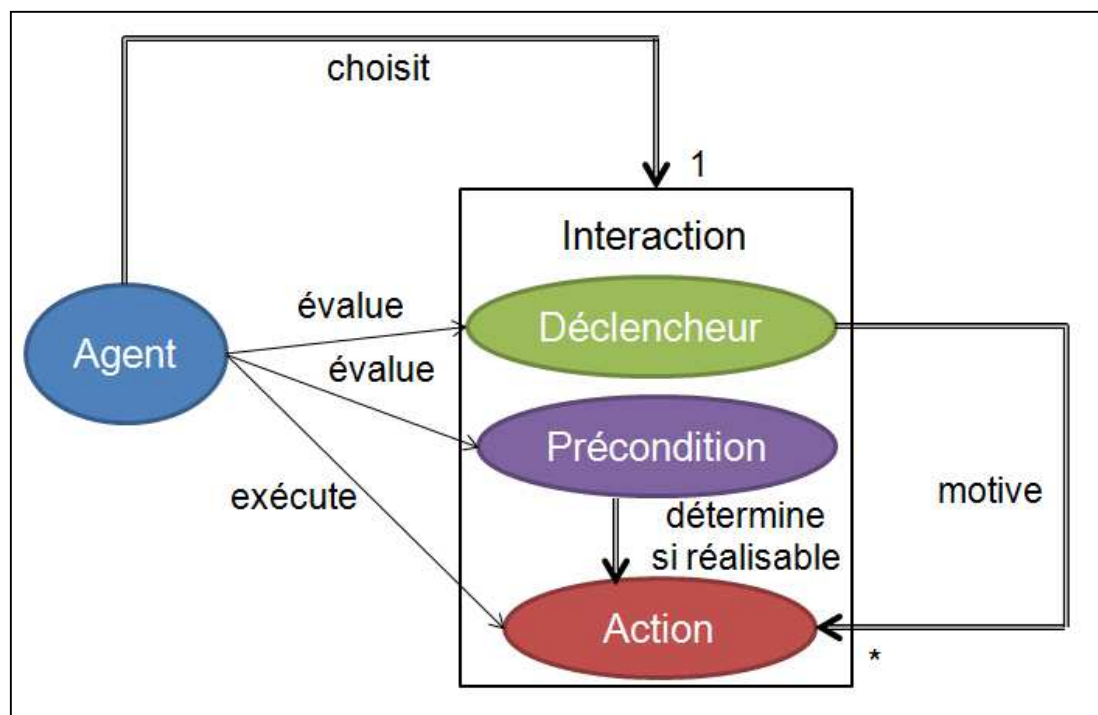


Figure 21 Fonctionnement du modèle PADAWAN. Contrairement à la Figure 20, le fonctionnement de l'activation des interactions n'est pas explicité de la même façon. Des déclencheurs et préconditions sont liés à chaque action pouvant être effectué par l'agent, et conditionnent sa sélection.

Nous proposons de faire cohabiter les notions de contrainte et d'interaction. Pour lier les deux, nous introduisons la notion d'**opinion** de contraintes. Cette notion d'opinion permet d'exprimer la position *a priori* d'une contrainte vis-à-vis de l'exécution des interactions. Pour cela, l'avis des contraintes influe sur les préconditions et le déclencheur. La Figure 22 expose le lien de cette nouvelle notion avec les notions de contraintes et d'interactions. Un agent, au cours de son cycle de vie, commence par consulter l'ensemble de ses contraintes qui exprimeront leur opinion sur chaque interaction possible. L'ensemble de ces opinions pour une interaction possible permet de définir si cette interaction est réalisable. Ensuite, la priorité et l'insatisfaction des contraintes permettent d'établir une hiérarchisation des interactions possibles. Cette hiérarchie permet d'effectuer une sélection de la prochaine interaction à effectuer.

Une contrainte peut exprimer une opinion parmi ces quatre types :

- **indifférente** : l'opinion par défaut, utilisée quand il n'existe pas de connaissance particulière quant à l'influence de l'exécution de l'action sur la satisfaction de la contrainte ;
- **favorable** : seulement lorsqu'une contrainte est insatisfaite, et que les connaissances en généralisation permettent de supposer que l'exécution de l'interaction aura un impact favorable ;



- **non favorable** : contraire de la précédente, lorsque l'exécution de l'interaction aura un impact *a priori* potentiellement négatif sur la satisfaction de la contrainte, cette opinion n'empêche pas l'exécution de l'interaction, mais ne la permet qu'en dernier ressort ;
- **opposée** : quand les connaissances permettent de dire que la contrainte verra sa satisfaction certainement baissée, avec des modifications possiblement irréversibles.

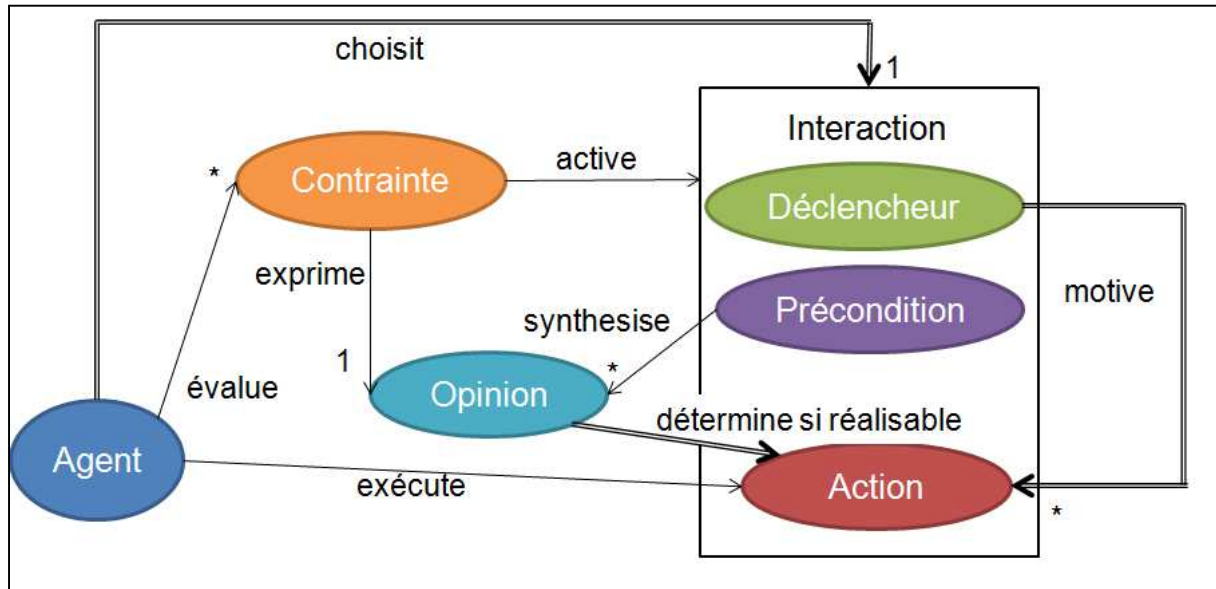


Figure 22 Fonctionnement du modèle DIOGEN. Les contraintes (issues des modèles pour la généralisation, voir Figure 20) sont ici explicitées et jouent un rôle dans le processus de sélection d'interaction à exécuter. Elles sont reliées au concept d'interaction issu de PADAWAN (Figure 21), grâce à la notion d'Opinion, introduite pour DIOGEN.

Par exemple, pour un bâtiment, une contrainte insatisfaite d'**équarrité** (qui exprime la volonté de caricaturer les angles presque droits en angles droits) se montre favorable à l'interaction **Simplifier le bâtiment en rectangle**, mais la contrainte de **préservation de la concavité** (qui s'applique aux objets à la forme initialement concave) se montre opposée à l'exécution de cette interaction. La contrainte de taille minimum est indifférente à l'exécution de cette interaction.

La synthèse de toutes les opinions des contraintes peut mener à trois conclusions possibles quant à l'interaction qui est, selon les situations :

- **faisable** : si au moins une contrainte est favorable, et que les autres sont au pire indifférentes ;
- **faisable mais risquée** : si au moins une contrainte est favorable, qu'il existe au moins une autre non favorable, et qu'aucune n'est opposée ;
- **non faisable** : si aucune contrainte n'est favorable, ou qu'au moins une est opposée.

Nous utilisons la notion de précondition de PADAWAN au sein de DIOGEN pour exploiter cette synthèse d'opinions et effectuer une première sélection d'interactions réalisables, avec un premier ensemble de choix faisables, et un second faisables mais risqués (ce qui constitue une modification de la précondition au sein de PADAWAN, qui retourne une valeur boo-

léenne). Néanmoins, ce n'est pas suffisant pour identifier l'interaction la plus pertinente à réaliser. Dans les modèles pour la généralisation, la contrainte à exécuter en premier lieu est sélectionnée selon les facteurs suivants, par ordre de priorité descendante : l'importance de la contrainte (qui indique qu'une contrainte ne sera pas abandonnée au détriment d'une contrainte d'importance moindre), l'insatisfaction de la contrainte, et un poids donné à l'action par la contrainte. Pour notre proposition, nous nous appuyons sur la notion de déclencheur de PADAWAN pour classer les interactions réalisables.

### III.1.3 Adaptation du cycle de vie

Les applications du modèle PADAWAN ont pour le cadre d'application la simulation. Ceci implique que l'agent choisit une interaction (ou plusieurs) selon un processus logique par rapport au phénomène à simuler. Le cycle de vie est un algorithme explicitant le comportement d'un agent pendant son activation. Généralement, un agent choisit une seule interaction à réaliser au cours de son activation. Nous souhaitons autoriser la réalisation de plusieurs interactions au cours du cycle de vie, voire d'agréger des interactions (en retranscrivant les opérations d'agréments issues de CartACom). Nous avons donc adapté le cycle de vie de PADAWAN pour ces besoins. Lorsqu'un agent a identifié les interactions réalisables, il essaye d'agréger les interactions réalisables à agréger (éventuellement en ajoutant d'autres interactions possibles). Puis, il choisit l'interaction avec le plus fort taux de déclenchement, basé sur une heuristique prenant en compte l'importance des contraintes favorables, leur priorité et leur insatisfaction.

Ce calcul peut être donné par la fonction suivante :

$$\text{déclencheur}(c, i) = w_p * \text{priorité} + w_i * \text{importance} + w_s * (100 - \text{satisfaction}) + w_w * \text{poids}$$

Avec :

- $c$ , la contrainte qui exprime son opinion sur une interaction particulière.
- $i$ , l'interaction sur laquelle une opinion est exprimée.
- $\text{priorité}$ , la priorité de la contrainte à être satisfaite.
- $\text{importance}$ , l'importance de la contrainte.
- $\text{satisfaction}$ , la satisfaction de la contrainte, sur une échelle de 0 à 100.
- $\text{poids}$ , un poids afin de favoriser certaines interactions par rapport à d'autres.
- un ensemble de constante ( $w_p, w_i, w_s, w_w$ ), permettant de favoriser certains critères par rapport à d'autres, avec  $w_p \gg w_i \gg w_s \gg w_w$ .

La Figure 23 montre le cycle de vie d'un agent.

Dans le modèle AGENT, il est aussi possible de revenir en arrière en annulant l'exécution d'une interaction lorsque cette dernière s'avère avoir un impact négatif sur les contraintes de l'agent l'exécutant. De plus, le cycle de vie d'un agent explore toutes les configurations possibles, du moment que la satisfaction s'améliore, tant que l'ensemble des contraintes de l'agent ne sont pas pleinement satisfaites. Ce cycle de vie, appelé cycle de vie d'exploration

---

en arborescence, est particulier au modèle AGENT et à sa structure hiérarchique, et son exploitation dans des modèles non arborescents, comme CartACom et GAEL, ou l'adaptation que nous proposons, pose un problème en terme de dimensions des étapes à sauvegarder pour pouvoir garder le tout cohérent : en effet les possibilités d'interactions sont beaucoup trop nombreuses. Nous avons donc conçu un cycle de vie avec retour arrière seulement pour les situations sans interactions entre objet de même niveau, utilisé uniquement pour la transposition du modèle AGENT, et un autre cycle de vie, sans retour arrière, utilisé dans tous les autres cas.

### **III.1.4 Ordonnancement de l'activation des agents**

L'organisation de l'activation des agents est un point important de l'exécution d'un système multi-agents. Dans le cadre des modèles pour la généralisation, il est important de pouvoir modifier l'ordonnancement des agents durant le processus d'exécution du modèle. En effet, un agent avec qui une interaction à été amorcé doit pouvoir être ajouté. De plus, lors d'interactions entre un meso et ses composants, le meso peut vouloir trier l'ordre d'activation des agents. Dans AGENT, cet ordonnancement est descendant : les agents meso décident, durant leur processus de généralisation d'activer les agents qui les composent. Dans CartACom, ce processus est horizontal. Il faut s'assurer que les agents qui ont un dialogue en cours soient à un moment donné activés afin de les résoudre. Dans le modèle DIOGEN, la file d'activation est accessible aux agents. Ainsi, il est possible pour un agent d'y ajouter d'autres agents hébergés, d'un même niveau ou encore lui-même. Ainsi, pour la résolution d'un problème donné de généralisation, il est nécessaire d'établir une liste initiale des agents à activer (par exemple, l'ensemble des agents îlots urbains pour la généralisation d'un espace urbain), et d'établir des interactions qui peuvent éventuellement modifier la file d'exécution.

Cet ordonnancement est séquentiel, dans la mesure où les agents sont activés les uns après les autres. Ils ont par contre une connaissance exacte et à jour de l'environnement qu'ils peuvent observer. L'activation des agents ne respecte par contre pas de règles d'équité. Un agent pourra être activé plusieurs fois, ou aucune fois si aucun autre agent n'appelle son activation, et qu'il ne fait pas partie du lot initial d'agents ajoutés à la file d'exécution. Nous sommes donc dans un système séquentiel et non équitable, d'après la classification proposée par Mathieu et Secq (2012) [66]. Ceci se justifie par le fait que ce choix est conforme à ceux des modèles AGENT, CartACom et GAEL, et justifié par les connaissances du domaine de la généralisation (Duchêne, 2016) [24].

Dans le cadre du travail de reproduction des comportements issus des modèles existants, nous avons reproduit le fonctionnement des files d'attente initiales. Nous avons proposé des ordonnancements particuliers pour chaque expérimentation. Nous revenons dans chaque section sur le choix d'ordonnancement proposé.

Un travail de formalisation plus systémique des fonctions d'ordonnancement de l'activation des agents est envisagé dans les perspectives de ce mémoire.

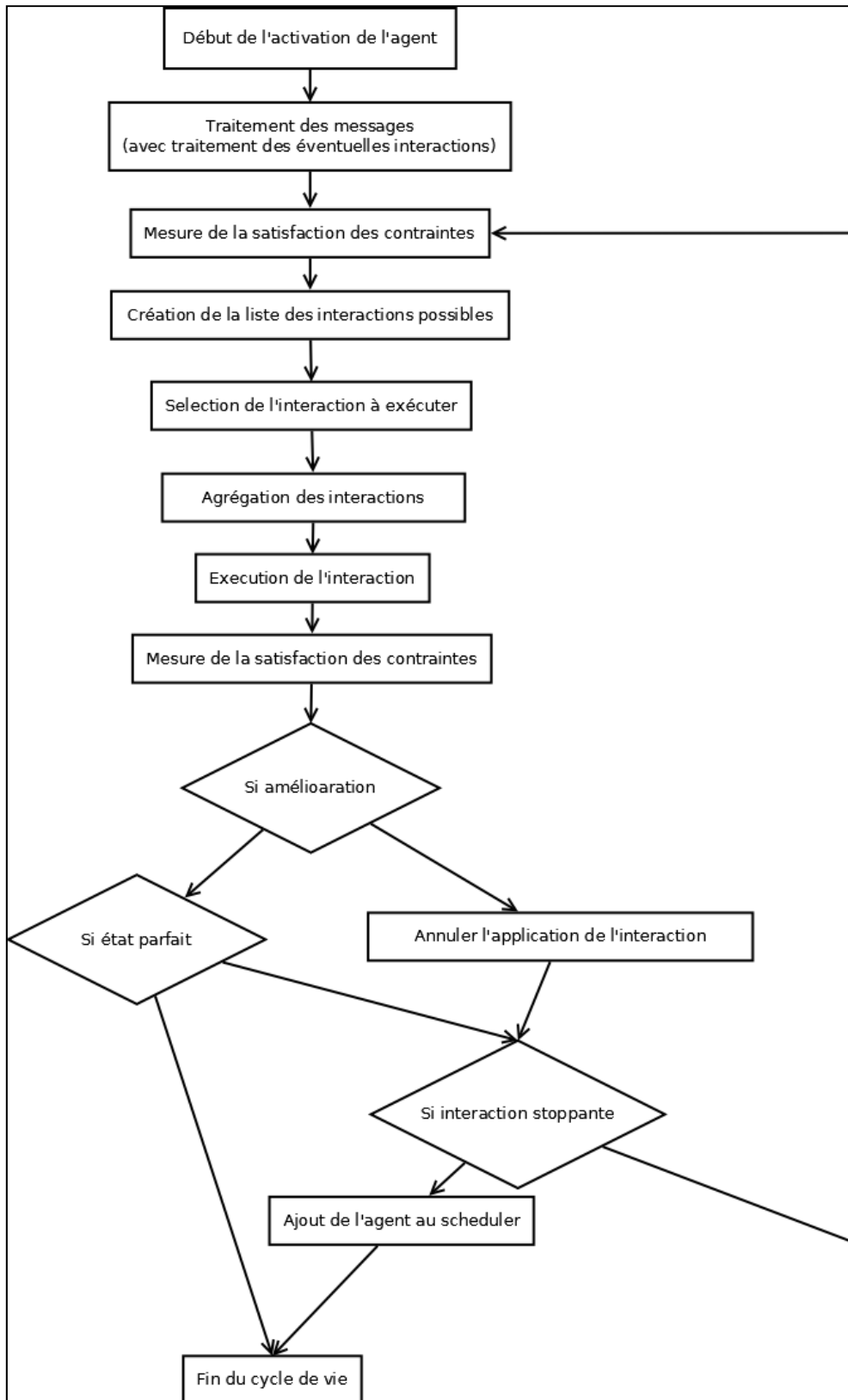


Figure 23 Cycle de vie simplifié d'un agent dans le modèle DIOGEN. (Lors d'un test, la branche **vrai** est à gauche de la branche **faux**).

## III.2 Implémentation

Nous avons implémenté le modèle DIOGEN dans le cadre de la plateforme Géoxygène, et la librairie Cartagen (Renard et al. 2010) [88]. Géoxygène est un Système d'Information Géographique (SIG) open source développé en Java conçu pour expérimenter les travaux de recherche du laboratoire COGIT.

Cartagen est une librairie de Géoxygène qui capitalise les recherches de l'équipe en généralisation automatisée. Une partie de cette librairie est open source, ce qui n'est pas le cas des développements multi-agents pour l'instant. Elle n'est pas complètement fonctionnelle en termes d'implémentation, les modèles de généralisation successifs conçus avant sa création (2010) ayant été implémentés de façon hétérogène dans d'autres plateformes. Ceci a impliqué un travail de reprise et de débogage, notamment de CartACom et GAEL pour intégrer leurs fonctionnalités à DIOGEN.

Une implémentation des éléments de PADAWAN (environnements, matrices, interactions) a d'abord été effectuée, puis, pour chaque situation décrite dans la suite de notre mémoire, des interactions, agents, contraintes et environnements spécifiques ont été implémentés.

Pour les travaux comparatifs décrits dans la section suivante, un travail de reprise du code existant a été effectué, afin de permettre non seulement une comparaison, mais aussi une réutilisation de l'existant. Tous les résultats présentés sont issus de cette implémentation.

## III.3 Reproduction de l'existant

Nous avons conçu notre proposition pour tenir compte des éléments propres à la généralisation. L'objectif suivant est de confronter DIOGEN à des cas réels de généralisation à résoudre. Dans cette section, nous essayons de reproduire les comportements des modèles AGENT, CartACom et GAEL. Cette démarche présente trois intérêts : d'abord donner une première validation de notre modèle, en vérifiant s'il peut remplacer ces modèles avec des résultats identiques, ensuite commencer à instancier DIOGEN avec des interactions construites sur la base d'algorithmes déjà éprouvés, enfin assurer l'inversion de perspective et le bon fonctionnement des préconditions et déclencheurs.

### III.3.1 Reproduction du comportement d'AGENT

Dans un premier temps, nous avons essayé de reproduire le comportement du modèle AGENT (cf. I.3.2 Le modèle AGENT) en utilisant notre modélisation. L'expérimentation porte sur la reproduction de la généralisation de zones urbaines. Pour rappel, AGENT fonctionne de manière hiérarchique, les niveaux meso dirigeant la généralisation des niveaux micro.

La première étape consiste en la définition des agents et des environnements. Les agents sont les mêmes que pour la modélisation AGENT initiale : des agents bâtiments et des agents îlots urbains. Un environnement est créé pour l'ensemble de la carte, dans lequel

vont se situer tous les agents créés. Des environnements sont aussi créés pour chaque îlot urbain. Ces environnements sont tous du même type, et hébergent les agents bâtiments.

La seconde étape consiste à décrire les algorithmes possibles sous forme d'interactions à réaliser, ainsi que leur lien avec les contraintes. Ces interactions sont de deux types :

- des actions concernant la transformation d'un bâtiment ;
  - o transformation de la géométrie en rectangle,
  - o équarrissage,
  - o agrandissement,
  - o simplification de la géométrie.
- des actions concernant un îlot urbain dans sa globalité impliquant les bâtiments le composant ;
  - o suppression de composants,
  - o déplacement de composants,
  - o activation de composants.
- des actions concernant un îlot urbain et modifiant sa représentation.
  - o transformation en îlot urbain de centre-ville (les bâtiments sont supprimés et l'îlot est symbolisé rempli d'un aplat de couleur, figurant une forte densité, comme s'il était intégralement couvert de bâtiments).

Les actions concernant les bâtiments (par exemple, **simplifier la géométrie, équarrissage**) sont décrites comme des interactions réflexives. Les bâtiments agissent sur eux-mêmes de façon à satisfaire leurs contraintes propres. Les actions concernant les îlots urbains (par exemple, **suppressions de bâtiments**) impliquent les bâtiments contenus, elles sont donc décrites comme des interactions agissant sur plusieurs cibles. Cas particulier, l'interaction **d'activation des agents composants** permet d'ajouter en tête de file d'activation les agents composant du type cible un îlot urbain. Les agents y sont ajoutés dans un ordre aléatoire, dans la mesure où, dans nos expérimentations cherchant à reproduire le modèle AGENT, il n'y a pas d'interactions transversale, ce qui implique que l'ordre importe peu, les modifications d'un composants n'influent pas la situation d'une autre. Enfin, l'interaction **transformation en centre-ville** est décrite comme une interaction réflexive de l'îlot sur lui-même.

La troisième étape consiste à décrire les relations des interactions avec les contraintes des agents. Pour chaque interaction, les contraintes pouvant être influencées par leur exécution sont identifiées et mises en relation avec l'interaction.

La quatrième étape consiste à intégrer les interactions dans une matrice d'interactions. Une matrice type est créée pour les îlots urbains. Elle intègre à la fois les interactions réflexives utilisées par les bâtiments, les interactions d'hôte sur un ensemble d'agents-cibles bâtiments et l'interaction de l'hôte sur lui-même. La Figure 24 présente une matrice d'interactions permettant de reproduire le comportement du modèle AGENT.

L'ajout des agents à l'ordonnanceur initial concerne seulement les agents îlots urbains et se fait de façon aléatoire, là encore, l'ordre d'activation n'important peu, dans la mesure où les modifications d'un agent n'a pas de répercussion sur ceux à un même niveau

Target Source	∅	Host	Building
Host	Transformation in town centre		Suppression of components Displacement of Components Activation of Components
Building	Suppression, Growing, Simplification, Simplification into Rectangle		

Figure 24 Matrice d'interactions pour l'environnement « îlot urbain » permettant la reproduction du comportement d'AGENT appliqué aux zones urbaines.

La reproduction du modèle a été implémentée et testée dans Cartagen (Renard et al. 2010) [88]. La Figure 25 montre qu'on obtient la même séquence.

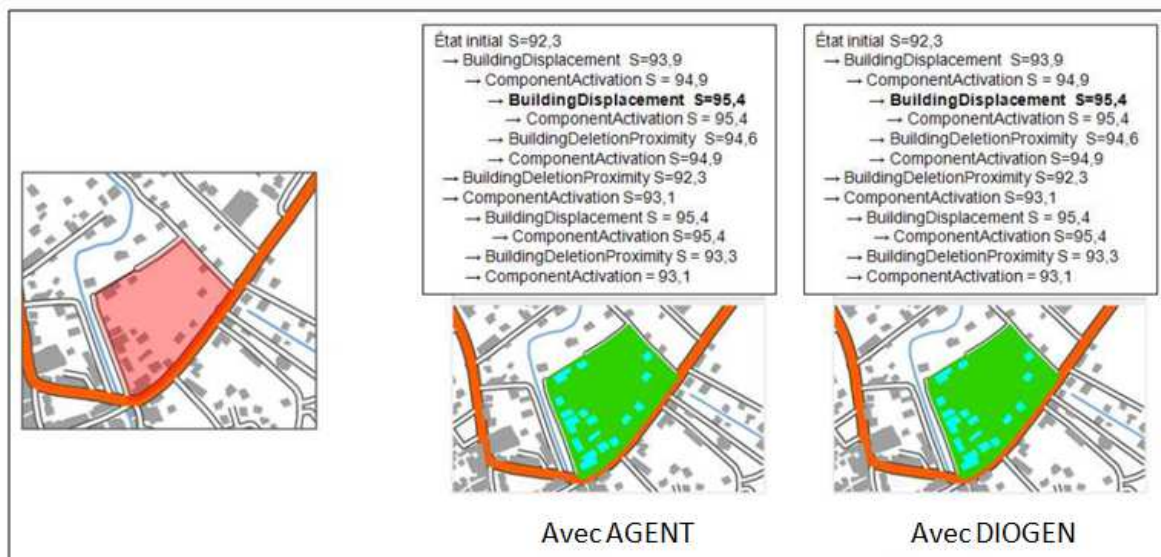


Figure 25 Reproduction du comportement d'AGENT adapté à DIOGEN. Dans les deux cas, la zone généralisée est identique, mais le processus est différent, comme le montre l'arborescence des interactions réalisées.

### III.3.2 Reproduction du comportement de CartACom

La reproduction du comportement de CartACom met avant tout en avant les possibilités d'interaction transversale de notre modèle. Le modèle CartACom propose des interactions entre agents sur un même niveau (cf. 1.3.4 Le modèle CartACom). Les agents étant tous sur un même niveau, un seul environnement, celui de la carte, est représenté, dans lequel les agents réalisent leurs actions. Néanmoins, ceci est extensible à d'autres types d'environnements plus petits (e.g. une partie d'une zone rurale de la carte).

Dans CartACom, ces actions impliquent l'agent qui les exécute, en prenant en compte son entourage, ainsi qu'une cible, qui est l'agent avec lequel la contrainte est partagée. En effet,

une contrainte peut être partagée entre deux agents, on parle alors de contraintes relationnelles. De plus, il est possible qu'un agent communique avec un autre agent pour lui demander d'agir lui-même pour résoudre la contrainte. CartACom propose une modélisation de dialogues permettant un échange entre agents. Ainsi, le déplacement d'un agent peut venir à l'initiative d'un autre agent. L'envoi d'une demande est le fait d'un agent, la réception et le traitement des messages reçus se faisant au début de l'exécution du cycle de vie de l'autre agent, avant l'évaluation des contraintes et du choix d'une interaction à effectuer. Dans DIOGEN, ces comportements sont décrits par des interactions d'agent à agent.

Les agents ajoutés dans l'ordonnanceur au début de chaque généralisation automatisée le sont dans un ordre similaire à celui du modèle initial CartACom. À cela s'ajoute le fait pour un agent d'ajouter les agents avec lesquels il est en relation à l'ordonnanceur.

Target	∅	Host	Building	Road
Source				
Host				
Building			Displacement Ask to displace	Rotation
Road			Ask for Rotation	

Figure 26 Matrice d'interactions permettant la reproduction des résultats de CartACom.

Ces interactions sont liées aux contraintes relationnelles de la même façon que les autres contraintes. Une contrainte reste liée à un agent. Lorsqu'un agent doit choisir une interaction, il tient compte de toutes ses contraintes, y compris les contraintes relationnelles. Les actions de CartACom peuvent être fusionnées, afin de tenir compte de toutes les cibles potentielles. Dans notre adaptation, nous avons ajouté, par rapport au cycle de vie illustré Figure 23, la possibilité d'agréger deux interactions, après avoir choisi une interaction. La Figure 26 montre la matrice d'interaction de l'environnement global qui héberge tous les agents. La Figure 27 montre un cas d'utilisation pour la mise en œuvre de test de DIOGEN utilisant la matrice reproduisant le comportement de CartACom.

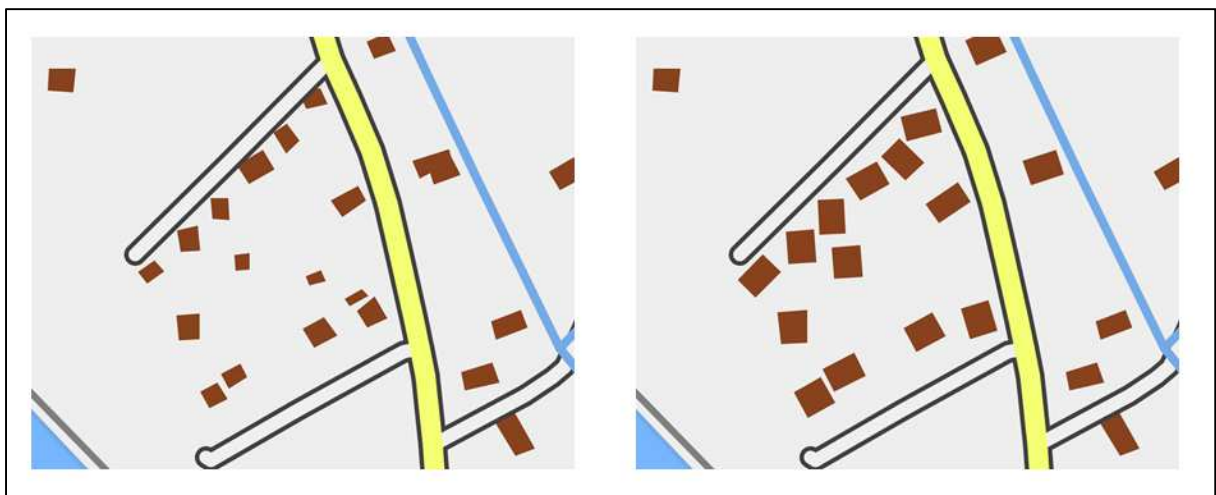




Figure 27 Exemples de tests pour la mise en œuvre de DIOGEN utilisant des interactions issues de CartACom. A gauche, la situation non généralisée, à droite la version généralisée.

### III.3.3 Intégration d'éléments du modèle GAEL

Le modèle GAEL (cf. I.3.5 Le modèle GAEL) propose l'introduction des points comme agents, et décrit les interactions entre eux, avec des objets dits submicro composés de points (points, segments, angles, triangles) et avec les agents déformables dont ils composent la géométrie (par exemple, l'ensemble des courbes de niveaux triangulées ou les contours d'un lac). L'intégration de ces éléments dans DIOGEN implique de prendre en compte la différence du fonctionnement d'interactions entre les agents par rapport à celui utilisé pour l'adaptation d'AGENT et CartACom. Les agents points sont au cœur du modèle GAEL. Nous proposons de garder cette modélisation des points comme agents dans notre modèle.

Les objets submicro n'ont pas été modélisés comme agents dans le modèle GAEL. Dans DIOGEN, nous choisissons de modéliser les objets submicro par des agents. Nous avons plusieurs raisons à ce choix :

- les submicro sont des objets porteurs de contraintes, or nous avons proposé d'ajouter les contraintes comme nouveaux type d'éléments dans le modèle DIOGEN, ainsi que leurs relations avec les agents. Ajouter une nouvelle notion d'objets non agents, se posant comme intermédiaire avec les contraintes, complexifierait le modèle que nous proposons ;
- les objets submicro décrivent des géométries dans l'espace (points, segments, angles, triangles), en cela, ils rejoignent les agents d'autres types vus jusqu'à maintenant ;
- dans le paradigme PADAWAN, toute entité est agent, or nous souhaitons conserver le plus possible ce caractère homogène dans DIOGEN ;
- dans la mesure où les objets submicro n'ont pas d'interactions à effectuer (ou à subir), leur modélisation comme agent ne pose pas de problèmes en terme d'interactions.

Une contrainte portée par un submicro dans le modèle GAEL, est dédoublée pour être utilisée à la fois par l'agent submicro et par l'agent point dans notre modèle. Par exemple, la contrainte de maintien de la longueur d'un submicro « segment » donne naissance à deux nouvelles contraintes : une pour chaque point extrémité du segment. Dans GAEL, les contraintes submicro sont construites différemment des contraintes des modèles AGENT et CartACom. Elles ne proposent pas d'évaluations de satisfaction, mais plutôt une force à appliquer sur l'agent point. Nous avons modifié ces contraintes pour qu'elles présentent un calcul de satisfaction utilisable par notre proposition de modèle, afin de nous rapprocher de la généralité des interactions entre agents.

La principale interaction du modèle GAEL est celle effectuée par un agent point lorsque celui-ci doit s'approcher de son point d'équilibre sur la base des agents submicro auquel il appartient. Une telle interaction est provoquée par l'insatisfaction d'une contrainte d'un sub-

micro dont l'agent point fait partie. Dans DIOGEN, les contraintes des submicro sont décrites comme partagées avec les agents points.

Un type d'environnement spécifique est défini pour mettre en œuvre le modèle GAEL. Il héberge tous les agents points et submicro. La matrice d'interaction propre à ce type d'environnement présente l'interaction de déplacement de point. Lors du cycle de vie d'un agent point, ce dernier sélectionne une interaction possible afin de satisfaire une contrainte partagée avec un submicro, puis l'agrège avec les autres interactions possibles du même type afin de permettre un déplacement tenant compte de toutes les contraintes. Nous réutilisons le mécanisme introduit pour l'adaptation de CartACom pour ce faire. La Figure 28 montre le résultat d'une utilisation des éléments issus de GAEL dans le cadre de l'utilisation de DIOGEN présentée dans le chapitre 4.

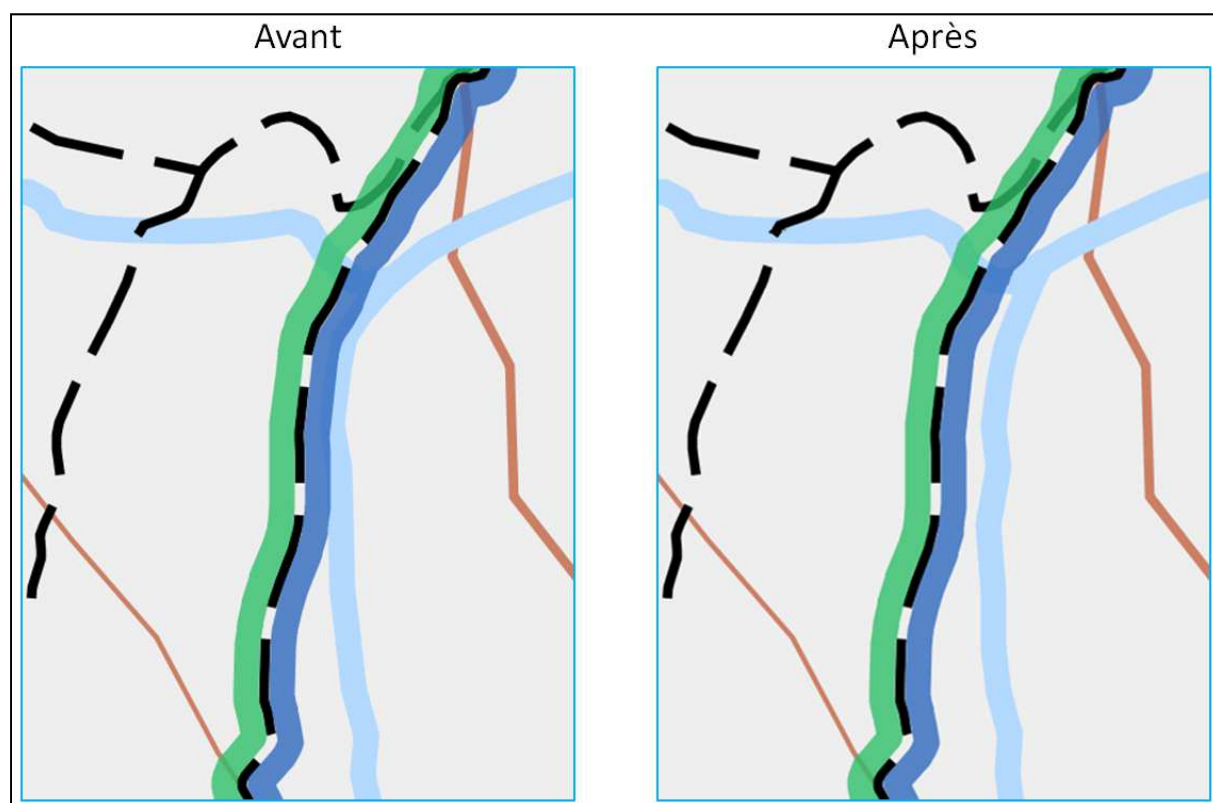


Figure 28 Exemple d'application de DIOGEN exploitant des interactions et contraintes construites issues de GAEL. La rivière a été déformée pour ne plus être recouverte par les itinéraires. Voir le chapitre 4 pour le contexte d'application.

### III.4 Résolutions de cas nouveaux

Les expérimentations quant à la reproduction des comportements existants se sont avérées positives. Nous avons gagné en souplesse pour ce qui est de la représentation des connaissances propres à la résolution d'un problème de généralisation donné (quelles interactions appliquer, avec quels agents, sur la base de quelles contraintes). Nous allons maintenant présenter une mise en œuvre de notre modèle sur des cas nouveaux, où nous cherchons à obtenir des comportements qui ne pouvaient pas être obtenus avec les modèles agents initiaux utilisés indépendamment les uns des autres.

### III.4.1 Intégration de comportements empruntés à la fois à AGENT et CartA-Com

Dans la section précédente, nous avons présenté des modélisations permettant de reproduire les comportements et résultats des modèles AGENT et CartACom séparément. Pour cela, il a fallu décrire des interactions sur la base des actions utilisées dans les modèles initiaux, puis les intégrer dans une matrice d'interaction. Afin de vérifier que les concepts issus de ces deux modèles ont été traduits d'une façon générique et commune, nous nous intéressons à une étude de cas permettant de mettre en œuvre conjointement des comportements des deux modèles.

Target Source	∅	Host	Building	Road
Host			Suppression of components Displacement of Components Activation of Components	
Building	Suppression Growing Simplification Simplification into Rectangle		Displacement Ask to displace	Rotation
Road			Ask for Rotation	

Figure 29 Matrice d'interaction intégrant des interactions issues de AGENT et de CartACom pour l'inclusion de la parallélisation des bâtiments aux routes dans un environnement urbain.

L'interaction de **rotation** issue du modèle CartACom a pour objectif de caricaturer le parallélisme d'un bâtiment avec une route. Un bâtiment ayant une orientation presque parallèle à une route à proximité doit être représenté comme parallèle à la route, l'orientation des bâtiments presque parallèles doit donc être modifiée pour devenir strictement parallèle (Mustière et Moulin, 2002) [78]. Cette caricature du parallélisme s'effectue dans le modèle CartACom, donc dans les cadres ruraux, domaines d'application du modèle. Dans le cadre urbain, l'introduction d'un tel comportement s'avère plus compliquée, car il s'appuie sur une relation entre deux objets : un tronçon de route et un bâtiment. Cette interaction doit être prise en compte parallèlement aux interactions réflexives et surtout aux interactions de l'hôte îlot avec ses bâtiments. Dans le cadre de l'ordonnement des agents bâtiments, ceux qui sont en relation avec une section de route pour caricaturer le parallélisme sont ajoutés en fin de file par l'interaction d'**activation des agents composants**. Nous pouvons maintenant gérer des aspects de AGENT et CartACom dans un même modèle. La Figure 29 montre une matrice d'interaction intégrant l'interaction **rotation** en plus des interactions issues de AGENT. La Figure 30 montre un îlot généralisé sans et avec la prise en compte de la parallélisation.

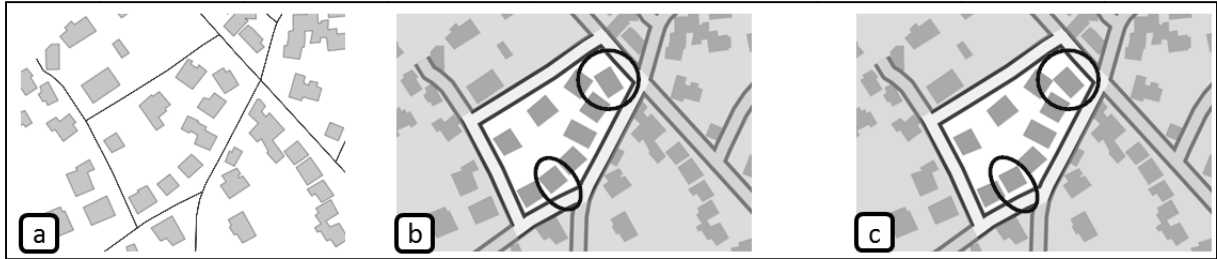


Figure 30 (a) Données non généralisées avec symbolisation minimale ; (b) données généralisées sans parallélisation des bâtiments avec une matrice d'interaction sans interactions entre routes et bâtiments ; (c) données généralisées avec parallélisation, voir la matrice d'interaction Figure 29.

L'interaction de rotation se produit après toutes les autres, dans la mesure où sa réalisation a un impact limité par les autres contraintes et dans la réalisation des interactions. La parallélisation peut influencer sur la contrainte de non superposition, mais d'une manière trop limitée pour que cela puisse avoir un impact significatif. Nous calibrons néanmoins la contrainte de caricature du parallélisme avec une priorité moindre afin que l'interaction appropriée soit exécutée après les autres. Mais plus que le résultat, c'est la possibilité d'utiliser conjointement des comportements de deux modèles conçus différemment que valorise cette expérience.

Cette expérimentation et ces résultats illustrent l'article de conférence soumis à la conférence PAAMS 2014 (Maudet et al. 2014 [70]).

### III.4.2 Définition d'un espace embarqué pour le voisinage

Sur une carte, un objet géographique peut constituer un référentiel pour d'autres objets, c'est-à-dire que la position de ces objets par rapport à l'objet constituant le référentiel est plus importante que leur position absolue dans le système de coordonnées de la carte (en termes d'interprétation par le lecteur de la carte) par leur relation à l'objet constituant le référentiel. Par exemple, au sein d'un îlot urbain, certains bâtiments peuvent avoir un positionnement défini selon les impasses. L'impasse fait office de référentiel pour les bâtiments. Si l'impasse est amenée à bouger, notamment pour faire de la place à d'autres bâtiments, les bâtiments ayant une position relative par rapport à l'impasse doivent la maintenir. Nous sommes dans une situation où les bâtiments, pour maintenir une position dans un référentiel donné, doivent se déplacer dans un autre référentiel.



Figure 31 Exemple d'impasse avec le réseau et les bâtiments environnant. En rouge le tracé du buffer permettant de délimiter le voisinage.

Nous proposons d'introduire la notion d'environnement embarqué définissant un cadre de référence pour les agents bâtiments. Chaque bâtiment est encapsulé au sein de PADAWAN dans un agent définissant le voisinage de l'impasse. L'impasse et les bâtiments de son voisinage sont positionnés dans cet environnement. Cette structure est présentée Figure 32.

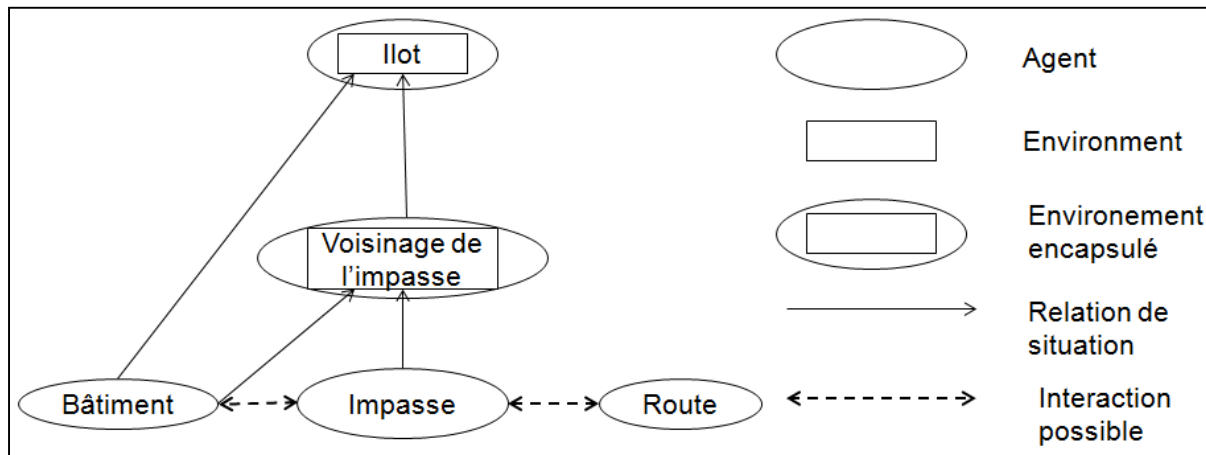


Figure 32 Relations entre îlots, impasse, bâtiments et voisinage.

Les bâtiments appartenant au voisinage d'une impasse sont identifiés à l'aide d'un buffer, comme montré dans la Figure 31. Lorsqu'une impasse doit se déplacer, elle signale son déplacement à l'agent voisinage qui répercute le déplacement aux bâtiments du voisinage. Cette interaction de déplacement est incluse dans une matrice d'interaction (celle qui définit le comportement des agents au sein d'un environnement urbains) décrite Figure 33, et est exécutée préalablement aux autres interactions.

Cible \ Source	∅	Ilot en tant qu'Hôte	Bâtiment	Voisinage de l'impasse
Ilot en tant qu'Hôte	Transformer en centre ville		Supprimer Déplacer Activer	Activer
Bâtiment	Agrandir Tourner Simplifier Remplacer par un rectangle			
Voisinage de l'impasse	Glisser à gauche Glisser à droite			

Figure 33 Matrice d'interaction d'un îlot urbain, avec voisinage d'impasse.

L'activation des agents est ordonnée de façon à ce que les voisinages des impasses s'exécutent en premier lieu, sur la base de la taille que doivent faire les bâtiments à terme de la généralisation. Les voisinages des impasses contenant les bâtiments ayant en moyenne la plus grande taille finale attendue sont activés en premier.

La Figure 34 montre l'application de notre proposition sur un îlot.

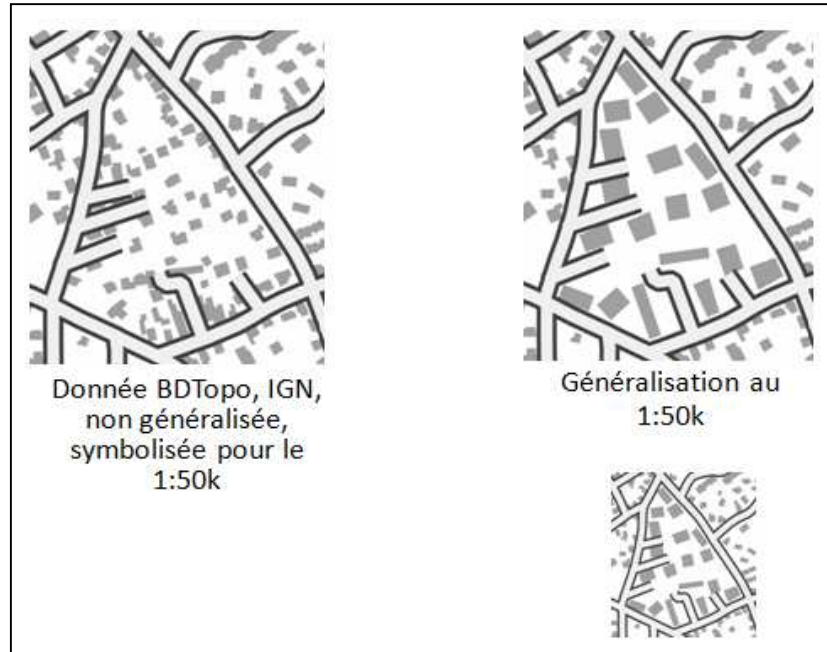


Figure 34 Généralisation d'un îlot urbain en tenant compte des impasses. À gauche, les données initiales, à droite la généralisation en tenant compte de des impasses, avec la même zone en échelle proportionnelle en dessous ?

La Figure 35 met en avant d'autres résultats. Dans le cas du milieu, nous constatons une limitation de la modélisation proposée. Pour résoudre ce genre de situation, identifier les impasses dans la continuité d'une route comme un élément d'un ensemble contenant cette route en plus de l'impasse, et qui aura pour objectif de se maintenir. Par exemple, la notion de « stroke » routier, ou groupe de tronçons adjacents formant un groupe perceptuel sur le critère de bonne continuité de la théorie du Gestalt (Wertheimer, 1923 [119], Thomson et Richardson, 1999 [112]) ; nous reviendrons sur la question des strokes pour une autre problématique, cf. **IV.2.2 Le concept de « Stroke » d'itinéraire**).

Cette expérimentation est illustrée dans une publication soumise à l'édition 2013 du workshop sur la généralisation et les représentations multiples de l'ICA (Maudet et al. 2013[68]).





Figure 35 Exemples d'applications du glissement des impasses. En haut, un cas attendu et positif. Au milieu, un problème est observé : la cohérence de l'impasse s'inscrivant dans la continuité de la route. En bas, un autre cas satisfaisant.

### III.4.3 Référencement multiple d'un agent dans plusieurs environnements

L'existence de plusieurs environnements, avec déplacement des agents dans un environnement pour maintenir sa position dans un autre, peut être raffinée par la définition de référentiel propre à un type d'environnement particulier.

Nous pouvons illustrer cette situation avec le cas de bâtiments adjacents au sein d'un même îlot. Lorsque deux bâtiments sont de types différents, ils peuvent potentiellement être symbolisés différemment, par exemple par deux couleurs différentes (par exemple l'un est un bâtiment public symbolisé en rose, quand l'autre est un bâtiment indifférencié symbolisé en gris). Il est donc impossible de fusionner les deux bâtiments dans la carte finale, car il est important de maintenir l'existence des deux bâtiments. Il est aussi impossible de généraliser les deux bâtiments indépendamment l'un de l'autre, car la relation d'adjacence serait perdue. Une solution que nous proposons consiste à fusionner temporairement les bâtiments, à les généraliser comme un tout, puis à les séparer à nouveau. Pour réaliser une telle opération, il est nécessaire de maintenir une position relative de la polyligne séparant les deux bâtiments, par rapport à l'agrégation, et de la reconstruire une fois la généralisation de l'objet agrégé terminée. Etant donné que la géométrie du bâtiment change, il n'est pas évident de repositionner cette polyligne. Il est donc nécessaire d'identifier où se positionnent

les points extrêmes de cette polyligne sur la géométrie agrégée. Les différentes opérations influant sur la géométrie au cours de la généralisation conduisent à la suppression des points (sommets) de la géométrie. Nous proposons de maintenir ces points non pas comme des sommets, mais comme des points se positionnant sur les arêtes de la géométrie résultante (Figure 36 haut). Le positionnement du point se fait sur un environnement segment qui est créé à l'occasion, en tenant compte d'une abscisse curviligne proportionnelle. Le segment est donc un environnement (encapsulé par un agent), qui héberge les agents points. Lorsque des segments sont fusionnés (avec suppression d'un sommet), les agents points hébergés dans les deux segments sont ajoutés au nouvel environnement segment, tout comme les points qui étaient à l'extrémité des segments (Figure 36 bas). Ainsi, les points faisant office d'extrémités de la polyligne séparant les deux polygones adjacents existent toujours dans la géométrie agrégée et peuvent servir de points de repère pour l'insertion de la nouvelle polyligne.

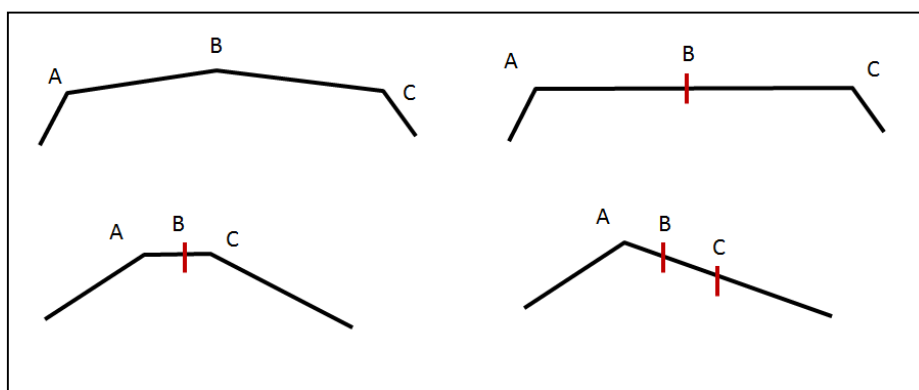


Figure 36 En haut : le point B, sommet à gauche, est supprimé à droite en tant que sommet, mais existe en tant qu'agent point, situé dans l'environnement que constitue le segment AC. En bas : le point C est supprimé en tant que sommet, le point B alors situé sur l'arête AC est ajouté au nouveau segment avec le point C.

Nous avons implémenté et testé notre proposition pour l'application d'une opération de simplification du contour (Ruas, 1988) [91]. Certains résultats, présentés dans la Figure 37, sont convaincants. D'autres le sont moins, et appellent des améliorations, par exemple :

- la prise en compte de nouvelles contraintes, agissant à d'autres niveaux, par exemple le maintien d'une proportion de volumes de chaque bâtiment ;
- un positionnement des points basé sur des critères de caractéristiques de la géométrie (par exemple, la distance relative à un angle particulier, cf. notion de relation à un objet support introduite par Jaara (2015) [50]).



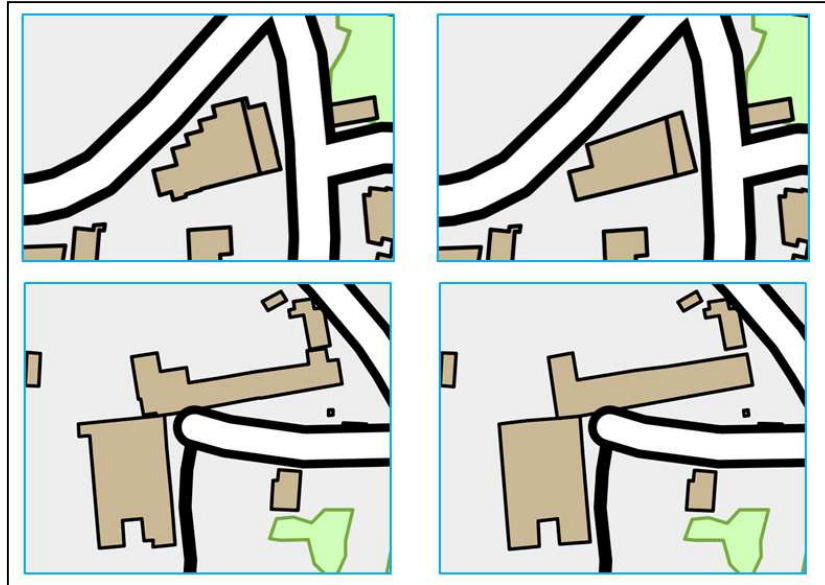


Figure 37 Résultat de l'utilisation de DIOGEN pour le problème des bâtiments adjacents. Les deux cas montrent une généralisation satisfaisante : les contours ont été simplifiés et l'adjacence a été préservée.

### III.5 Synthèse du chapitre

Nous avons proposé un modèle multi-agents multi-niveaux pour la généralisation, DIOGEN, basé sur PADAWAN. Ce modèle permet l'utilisation d'algorithmes sous forme d'interactions utilisées par les matrices d'interactions pour proposer une solution à un problème de généralisation donné. Chacune de ces interactions décrit une opération sur l'agent source susceptible de la réaliser, ainsi que sa ou ses cible(s) ; et sur le positionnement, sous forme d'influence, des différentes contraintes portées par l'agent source (et la ou les éventuelles cible(s)). Ce modèle permet, pour concevoir une solution pour un problème de généralisation donné, d'écrire la solution au moyen d'une matrice d'interaction entre agents de différents niveaux, et d'un ensemble d'interactions faisant appel à des algorithmes géométriques existants.

Par ailleurs, l'imbrication d'environnements propre au modèle PADAWAN permet de décrire des environnements imbriqués, ce qui permet de définir des cadres dans lesquels les agents peuvent interagir de façon plus large que dans les modèles antérieurs.

Dans un premier temps, nous avons proposé des interactions permettant de reproduire le comportement des modèles AGENT, CartACom et partiellement GAEL, et dans un second temps, nous nous sommes confrontés à des cas nouveaux. Nous avons ainsi montré (1) que notre modèle permet de mettre en œuvre conjointement des interactions hiérarchiques comme dans AGENT et transversale comme dans CARTACOM, donc de traiter conjointement des contraintes cartographiques portant sur un objet, un groupe d'objet ou une relation entre objets, et (2) que notre modèle permet d'étendre à moindre coût (i.e. sans modification du modèle) les modèles existants, par exemple en ajoutant la notion de référentiel interne à un agent qui en héberge d'autres. Dans le prochain chapitre, nous nous intéressons à une étude de cas particulière, mettant en œuvre des aspects multi-niveaux de notre modèle.

# Chapitre IV :

## Cas d'étude : cartes d'itinéraires de randonnées

---

Les services de cartographie à la demande adaptés aux besoins d'utilisateurs grand public ou professionnels sont des objectifs importants de l'IGN. Alors que les opérations de généralisations sur des cartes topographiques standard n'impliquent une nouvelle généralisation seulement lors de mise à jour (c'est-à-dire lorsque les données initiales changent), la possibilité de cartes à la demande implique des processus de généralisation plus spécifiques. Pour parvenir à fournir ce type de service, il faut pouvoir intégrer les données de l'utilisateur et construire les cartes le plus automatiquement possible. La réalisation des cartes représentant les itinéraires de randonnées, à la manière des plans de bus sont un cas particulier de cette problématique de cartographie à la demande automatique.

Dans une première section, nous présenteront les cartes d'itinéraires auxquelles nous nous intéressons et les problématiques associées. Ensuite, dans une deuxième section, nous nous intéresserons à la problématique du placement des itinéraires sur un réseau support, indépendamment des autres problématiques propres à ce type de carte. Nous décrirons enfin dans la troisième section les autres interactions nécessaires pour permettre de généraliser une carte d'itinéraires de randonnées.

### IV.1 Les cartes d'itinéraires

Les travaux de recherche en généralisation portent essentiellement sur les éléments topographiques (par exemples, routes, bâtis, altitudes) des cartes. L'ajout de données thématiques (par exemple : tourisme, santé, transport) parfois représentées sous forme schématique implique de les prendre en compte dans le cadre de la généralisation. Dans le cadre des cartes représentant des itinéraires de randonnées, les considérations spécifiques concernent la symbolisation des itinéraires et leurs relations avec le réseau les supportant, les autres itinéraires et les autres éléments de la carte. Ce type de carte peut aussi faire l'objet de production à la demande. La principale problématique de ce genre de carte est la possibilité pour les utilisateurs d'ajouter des données. Dans les cartes d'itinéraires, les possibles ajouts des utilisateurs portent principalement sur le tracé des parcours et l'identification de points d'intérêt. La représentation de ces données, notamment celle du tracer des itinéraires

peut se faire de façons variées. Une de ces représentations, la représentation des itinéraires à la façon de certains plans de transport en commun (un ruban coloré pour chaque itinéraire), implique des spécifications de généralisation particulières. Nous avons décidé de travailler dessus parce que les services de cartographie de l'IGN s'est montré intéressé par ce type de représentation et parce qu'elle nous a paru pouvoir constituer un cas de mise en œuvre de notre modèle.

Dans cette section, nous allons définir dans un premier temps la notion d'itinéraires, et identifier différentes façons de les représenter. Ensuite, nous identifierons les similitudes de notre problématique avec celle des cartes schématiques et nous continuerons avec l'identification de la problématique du positionnement d'itinéraire, problème au cœur de la génération automatique de ces cartes. Nous terminerons par la présentation du cadre de travail et d'expérimentations utilisé pour nos expériences (données utilisées, échelles choisies, etc.).

#### **IV.1.1 Définition des cartes d'itinéraires**

Un itinéraire est un parcours précis dans l'espace qui permet à celui qui le suit de joindre plusieurs points géographiques. Il peut correspondre à plusieurs types de déplacements : pédestres, équestres, cyclistes, motorisés, etc... Un itinéraire peut aussi décrire le parcours d'une ligne de transport en commun au sein d'un espace géographique donné.

D'une façon générale, la représentation d'un itinéraire sur une carte répond au besoin de décrire un trajet particulier par rapport à un réseau représentant des voies parcourables. Il s'agit donc de mettre en avant sur la carte la succession d'arêtes du réseau qui représentera l'itinéraire en question. Plusieurs procédés existent et sont exploités sur les cartes d'itinéraires existantes. Néanmoins, tous impliquent d'indiquer sur un arc du réseau son appartenance à l'itinéraire représenté. Parmi les différentes manières de symboliser un itinéraire, nous observons, parmi les cartes produites par divers organismes, trois approches :

- Le marquage linéaire, où l'ensemble de tronçons de réseau appartenant à l'itinéraire est surligné par un symbole linéaire (Figure 38a) : trait coloré, pointillé ... Les trois cartes de la Figure 39 montrent différents marquages linéaires.
- Le marquage ponctuel, où un symbole ponctuel est apposé sur les tronçons appartenant à l'itinéraire (Figure 38b). La Figure 39b montre un extrait de carte avec des marquages ponctuels pour distinguer les itinéraires.
- Le marquage labélisé, où le nom de l'itinéraire est labélisé le long des tronçons appartenant à l'itinéraire (Figure 38c). La Figure 39a montre un exemple de labellisation des itinéraires.

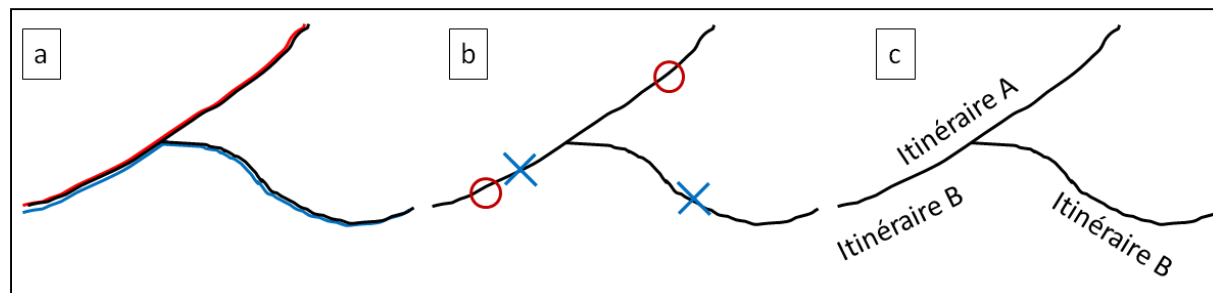


Figure 38 Schématisation de trois types de symboles possibles pour représenter des itinéraires (ponctuel, linéaire et étiquette).

Ces différentes approches sont souvent utilisées conjointement, et c'est dans le cadre d'une carte multi-itinéraires qu'elles peuvent paraître plus ou moins pertinentes selon les circonstances. En effet, lorsque plusieurs itinéraires doivent être représentés, il est nécessaire de pouvoir les distinguer, sachant qu'un même tronçon peut être utilisé par plusieurs itinéraires différents. Ainsi, la variation des symboles (couleurs, pointillés, formes ...) permet de distinguer les itinéraires. Néanmoins, l'ajout de trop nombreux symboles amènent de nombreuses problématiques de cartographie :

- Comment ne pas alourdir la carte avec trop d'informations ?
- Comment gérer la superposition de plusieurs itinéraires sur un même tronçon ?
- Quel impact ont ces ajouts de symboles sur les autres éléments de la carte ?

Dans le cadre d'un projet étudiant que j'ai encadré (Gautrais et Kraus, 2015) [41], les pratiques existantes en matière de cartographie des itinéraires de randonnées ont été étudiées. Ainsi la Figure 39a et la Figure 39b montrent une symbolisation des itinéraires commune (chaque tronçon de route parcouru par un itinéraire se voit apposer un symbole linéaire commun, un trait plein ou pointillé magenta, seul les toponymes ou un indicateur ponctuel permettent de discerner les parcours des différents itinéraires ; GR5, GR533, GR534 pour la Figure 39a ; les itinéraires présentés par leur symbole, triangle rouge, rond bleu ..., dans la Figure 39b). La Figure 39c, avec symbolisation multiple des itinéraires associe à chaque itinéraire une symbolisation particulière, et lorsque plusieurs itinéraires parcourent le même tronçon, ils sont apposés les uns après les autres.

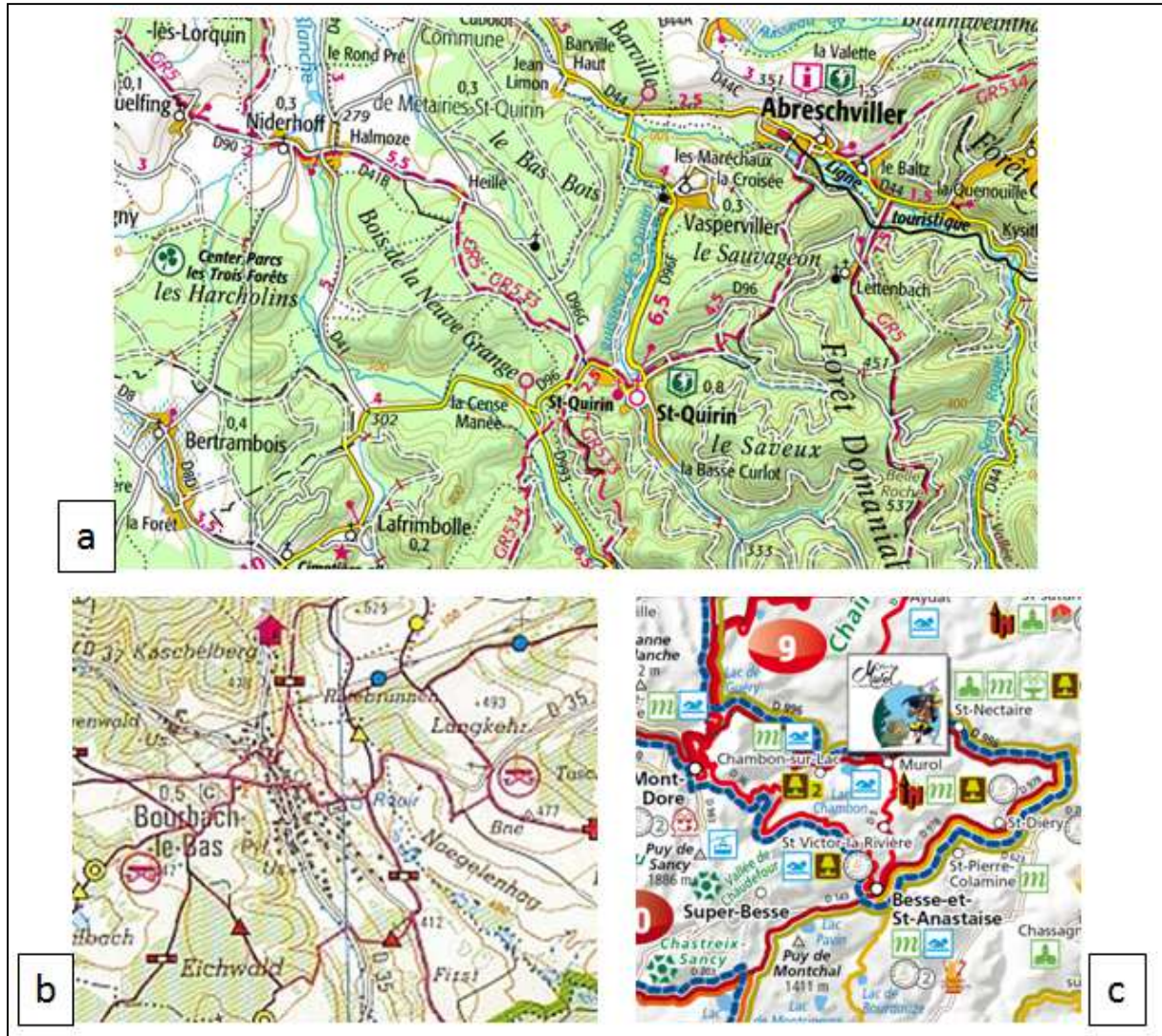


Figure 39 (a) : un extrait d'une carte à l'échelle 1 :100k, IGN-France, où les itinéraires sont décrits par un unique symbole, et distingués les uns des autres par des labels. (b) : un extrait d'une carte à l'échelle 1 :50k, Club Vosgien et IGN, où les itinéraires partagent un même symbole linéaire, et sont distingués par des symboles ponctuels. (c) : un extrait de la carte « L'Auvergne à Moto » à l'échelle 1 :530k, CRDT Auvergne/Actual, où chaque itinéraire a son propre symbole linéaire.

Les pratiques constatées sur les cartes de randonnées ont été rapprochées de celles identifiées dans le travail de Morrison (1996) [74] sur les plans de transport urbain. Ce dernier avait identifié plusieurs styles pour représenter les plans de bus principalement :

- Le style **classique**, avec une symbolisation commune à toutes les lignes avec labellisation différente.
- Le style **français**, avec une symbolisation différente pour chaque ligne.
- Les styles **scandinave** et **néerlandais**, hybrides, permettant dans une certaine mesure des symbolisations différentes (par exemple pour deux compagnies de bus différentes), mais pouvant être partagées par différentes lignes, ce qui implique un recours aux labels pour les distinguer.

Sur la base de pratiques constatées et de la classification de Morrison (1996) [74], Gautrais et Kraus (2015) [41] ont proposé différents exemples de cartes avec des symbolisations dif-



férentes. Un des objectifs de ces cartes prototypes est de permettre des tests auprès d'utilisateurs avec les différentes cartes, entre autres dans un cadre avec objectifs à identifier : des utilisateurs, cartes prototypes en main, devront exécuter un certains nombres de tâches, exécution qui sera évaluée. La Figure 40 montre quatre exemples de cartes prototypes.

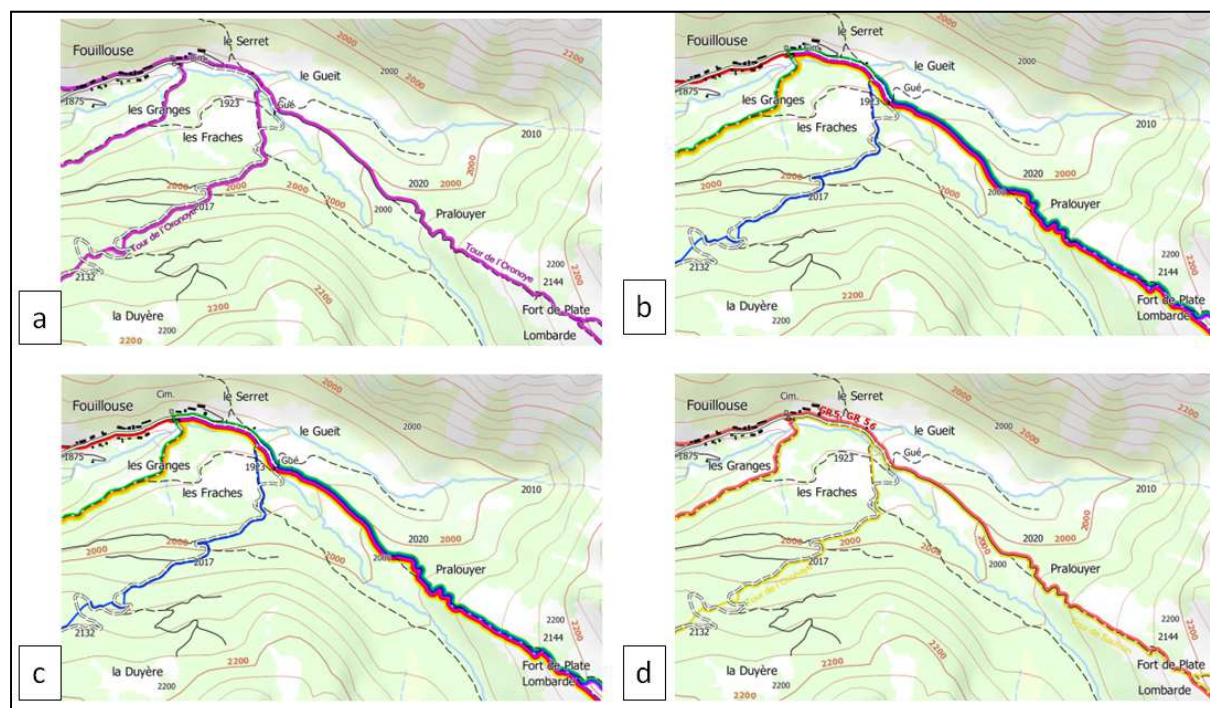


Figure 40 Quatre exemples de cartes proposées : (a) représentation en mode réseau ; (b) représentation à la manière de plan de bus ; (c) représentation à la mode de plan de bus avec bâtiments décalés ; (d) représentation groupée par type. D'après Gautrais et Kraus (2015).

#### IV.1.2 Les problématiques du positionnement automatique d'itinéraires

Nous avons vu que plusieurs choix de représentation sont possibles sur une carte d'itinéraires. Nous nous sommes intéressés au cas de la représentation d'itinéraires différents par symboles linéaires multiples (représentation dans la Figure 39c, la Figure 40b, la Figure 40c et la Figure 40d), que nous appelons « à la manière de plan de bus » en référence aux plans parisiens de la RATP (style français pour Morrison (1996) [74]). Pour cette représentation, se posent des questions de choix de positionnement des différents symboles d'itinéraires les uns par rapport aux autres.

Lorsqu'un tronçon de route porte plusieurs itinéraires, le positionnement des symboles d'itinéraires a son importance. Lorsqu'un itinéraire bifurque sur un autre tronçon, la question de la cohérence se pose. Le trajet de l'itinéraire doit être perçu comme étant continu, sans cassure, par l'utilisateur. De fait, une hypothèse de base est que les croisements non nécessaires d'itinéraires (i.e. qui ne correspondent pas à de véritables croisements dans la réalité) rendent plus difficile la lisibilité de la carte (principe de bonne continuité du Gestalt (Wertheimer, 1923) [119]).

Ainsi, lors du choix de positionnement du symbole d'un itinéraire par rapport à un tronçon le supportant, il est important d'essayer de minimiser ces croisements.

Afin de nous représenter le positionnement d'un symbole d'itinéraire sur un tronçon de route susceptible d'en supporter plusieurs, nous utilisons une notation numérotée. S'il est situé sur la section de route, on considère la position comme 0, s'il est décalé directement sur la gauche, sa position sera 1. S'il est décalé sur la gauche, avec un espace pour un autre itinéraire entre le symbole de section de route, sa position sera 2... Pour les décalages sur la droite, nous utilisons des valeurs négatives. La Figure 41 expose illustre cette modélisation. Positionner un symbole d'itinéraire revient à assigner la valeur de positionnement pour chaque section d'itinéraire portée par une section de route.

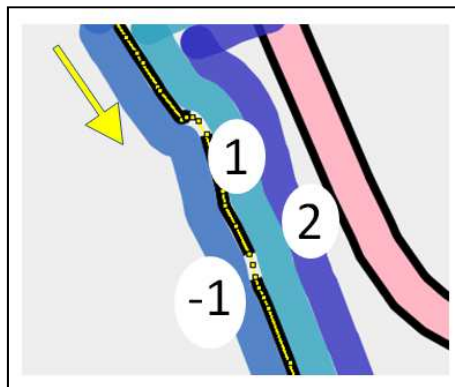


Figure 41 Valeur des positions pour trois itinéraires portés par une même section de route (en noir et jaune). La flèche jaune indique le sens de la section de route.

Ce cas d'étude présente deux intérêts essentiels : d'une part, l'apparition de données utilisateurs, et les besoins de représentation qui y sont associés encourage les initiatives permettant l'automatisation du traitement de ses données ; d'autre part, ce problème laisse entrevoir l'idée de structures hiérarchiques entre différents objets cartographiques, pouvant justifier l'utilisation de DIOGEN pour en automatiser la généralisation : itinéraires « hébergés » par les routes, repère le long des itinéraires « embarqués » par les routes, etc...

Nous allons tout d'abord décrire comment un itinéraire peut être intégré au sein de données topographiques. Sachant que nous avons un réseau support, c'est-à-dire un ensemble de tronçons (par exemple de routes), un itinéraire sera porté par un sous-ensemble de tronçons de ce réseau. Ainsi, un itinéraire est subdivisé en tronçons, chacun porté par un tronçon du réseau support. Un même tronçon du réseau support peut porter plusieurs tronçons d'itinéraires différents. Le diagramme de classe illustré Figure 42 montre la relation entre une section de route (RoadSegment) et une section d'itinéraire (RouteSegment).

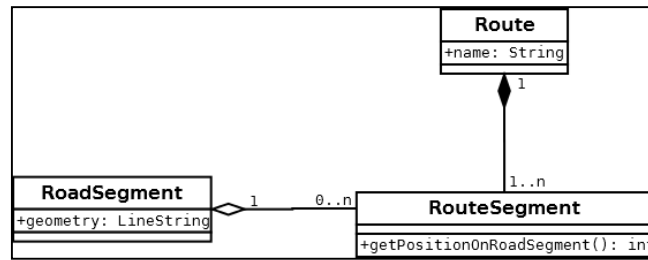


Figure 42 Diagramme de classe représentant les relations entre itinéraires et réseau support. Une section d'itinéraire (**RouteSegment**) est liée à une unique section de route (**RoadSegment**). Une section de route peut être liée à plusieurs sections d'itinéraires. L'ensemble des sections d'un itinéraire est lié à un objet représentant l'itinéraire (**Route**).

### IV.1.3 Contexte de travail : les données utilisées

Pour tester nos propositions, nous nous appuyons sur un ensemble de données. Ces données sont issues de la BD Topo de l'IGN, dont l'échelle de référence est de 1:15k, et pour les itinéraires de randonnées de la Fédération Française de Randonnée Pédestre.

Les données considérées sont les suivantes :

- Données topographiques :
  - o Un réseau routier, composé de routes et de chemins.
  - o Le bâti.
  - o Les données hydrologiques linéaires et surfaciques.
  - o La végétation.
  - o La mer et le trait de côte.
- Données touristiques :
  - o Des points d'intérêts (lieu de repos, point de vue ...).
  - o Des itinéraires de randonnée, dont la géométrie a été recalée sur le réseau routier.

Les itinéraires de randonnées sont fournis avec leur géométrie. Afin de créer une relation entre les tronçons d'itinéraires et les tronçons du réseau support, telle que décrite Figure 42, nous avons dû traiter les données. Ce traitement a pu être fait assez simplement grâce au fait que nous disposions d'une version de la géométrie recalée (automatiquement) des itinéraires sur le réseau routier, recalage effectué par le Service de Développement de l'IGN pour des besoins internes. Il a néanmoins fallu gérer certains cas afin de respecter les contraintes sur la relation section de route/section d'itinéraire :

- d'abord, lorsque la géométrie d'une unique section d'itinéraire est partagée par deux ou plus géométries de section de route, il a été choisi de subdiviser la section d'itinéraire en autant de sections qu'il y a de sections de routes pour couvrir la géométrie, comme décrit dans la Figure 43 ;



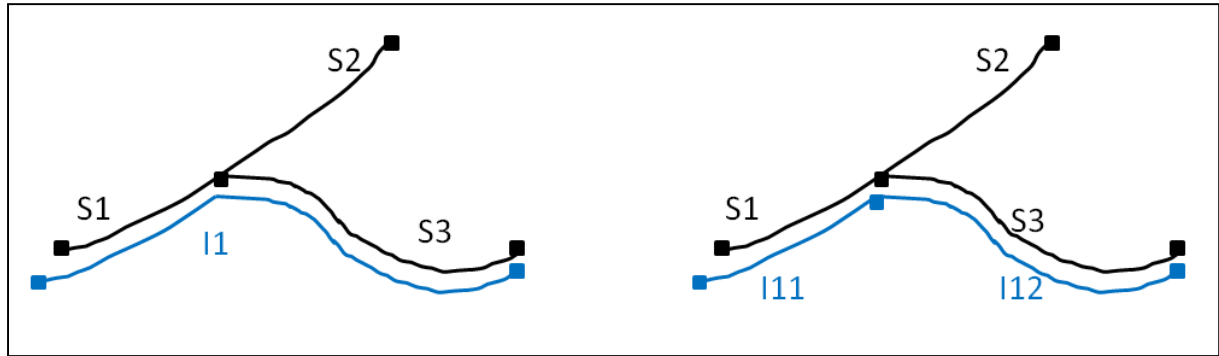


Figure 43 Schématisation du cas de création de deux tronçons d'itinéraires à partir d'un : à gauche la situation initiale, telle que présent dans les données ; à droite l'adaptation, la section I1 a été divisée en deux sous-sections, I11 et I12, chacune correspondant à un unique tronçon de route (respectivement S1 et S3).

- ensuite, lorsque la géométrie de l'itinéraire ne suit aucune section de route, nous créons une section de route fictive que nous rajoutons au réseau, afin de pouvoir systématiquement déduire la géométrie de l'itinéraire de celle de la route ou du chemin qui le supporte (Figure 44).

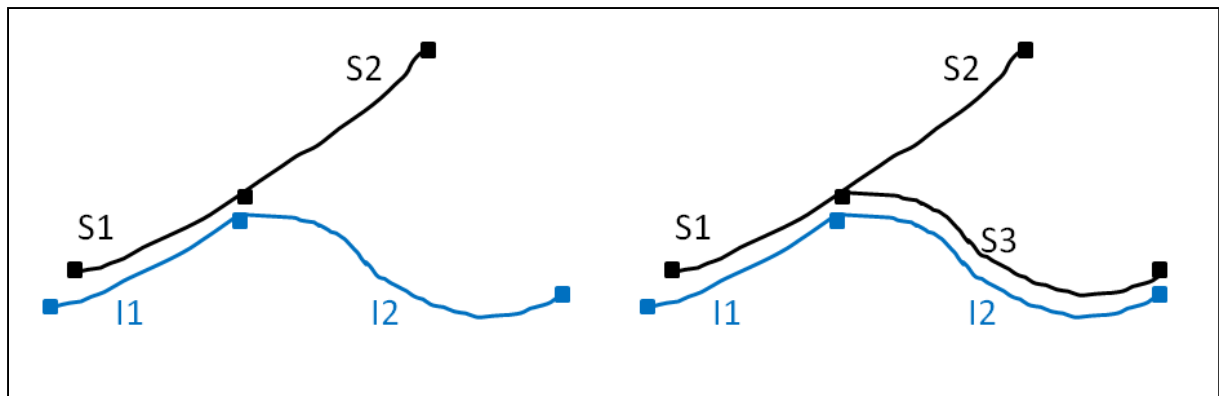


Figure 44 Schématisation du cas de création de section de route à partir d'une section d'itinéraire. À gauche, la situation initiale, avec une section d'itinéraire, I2, associée à aucune route ; à droite, une section de route fictive a été créée afin de permettre l'association de chaque tronçon d'itinéraire à un tronçon de route.

Ces opérations doivent se faire sur des données géométriques cohérentes, avec un recalage parfait, ce qui n'était pas le cas sur nos données. Une détection des incohérences a été faite, et un nettoyage manuel des données a dû être effectué. Ces corrections ont été nécessaires, car les données l'exigeaient. Dans un cadre applicatif, ces corrections peuvent être évitées si les données sont saisies d'une façon à correspondre aux contraintes de modélisation.

Les zones étudiées sont des zones denses en sentier de randonnée. Elles correspondent à des ensembles de dalles de 20km par 20km du territoire français.

L'échelle choisie est 1:40k qui est une échelle suffisamment grande pour qu'apparaissent des conflits à résoudre, et qui s'éloigne des standards existant à l'IGN, et contribue ainsi à inscrire notre proposition dans une démarche de création de cartes innovantes.

#### IV.1.4 Identification des zones impactées par la présence d'itinéraires

La présence d'itinéraires ne concerne pas l'intégralité de la carte. Parfois, les itinéraires peuvent être présents sur une partie de la carte, et d'autres endroits peuvent ne pas être concernés par leur existence. Ceci implique qu'on peut vouloir utiliser un procédé de généralisation ne prenant pas en compte les itinéraires, ou bien des données déjà généralisées pour ces parties de la carte non impactées. Nous proposons un système permettant d'identifier les îlots possiblement impactés par la présence d'itinéraire. Pour cela, nous fabriquons un graphe où chaque itinéraire et chaque îlot urbain sont représentés par des nœuds. Lorsqu'un îlot est adjacent à une section de route portant un itinéraire, une arête est tracée entre les nœuds respectifs des deux objets. De même lorsque deux îlots sont adjacents, une arête est ajoutée au graphe entre les sommets des deux îlots. Les arêtes du graphe sont pondérées par l'inverse de la densité de l'îlot dans le cas d'une arête îlot-section, et de la somme des deux inverses de la densité dans le cas d'une arête îlot-îlot. La Figure 45 montre un extrait de carte avec un graphe en superposition.



Figure 45 Exemple de graphe construit à partir d'une zone urbaine pour estimer l'impact des itinéraires sur les îlots. Les nœuds représentant les itinéraires sont rouges, ceux pour les îlots en jaune, les arêtes sont en bleu.

Une fois ce graphe construit, nous utilisons un algorithme de détection de plus court chemin pour identifier pour chaque nœud d'îlot le nœud d'itinéraire le plus proche, ainsi que la distance (dont la valeur est liée à la densité des îlots urbains reliés). Nous obtenons ainsi pour chaque îlot une distance par rapport aux sections porteuses d'itinéraires. La Figure 46 montre une représentation des îlots selon leur distance aux itinéraires. Le but y est d'établir un seuil, afin de distinguer les zones nécessitant une généralisation nouvelle, pour cause de présence des symboles d'itinéraires, et celle pouvant utiliser des processus, ou des données, de généralisation existants.

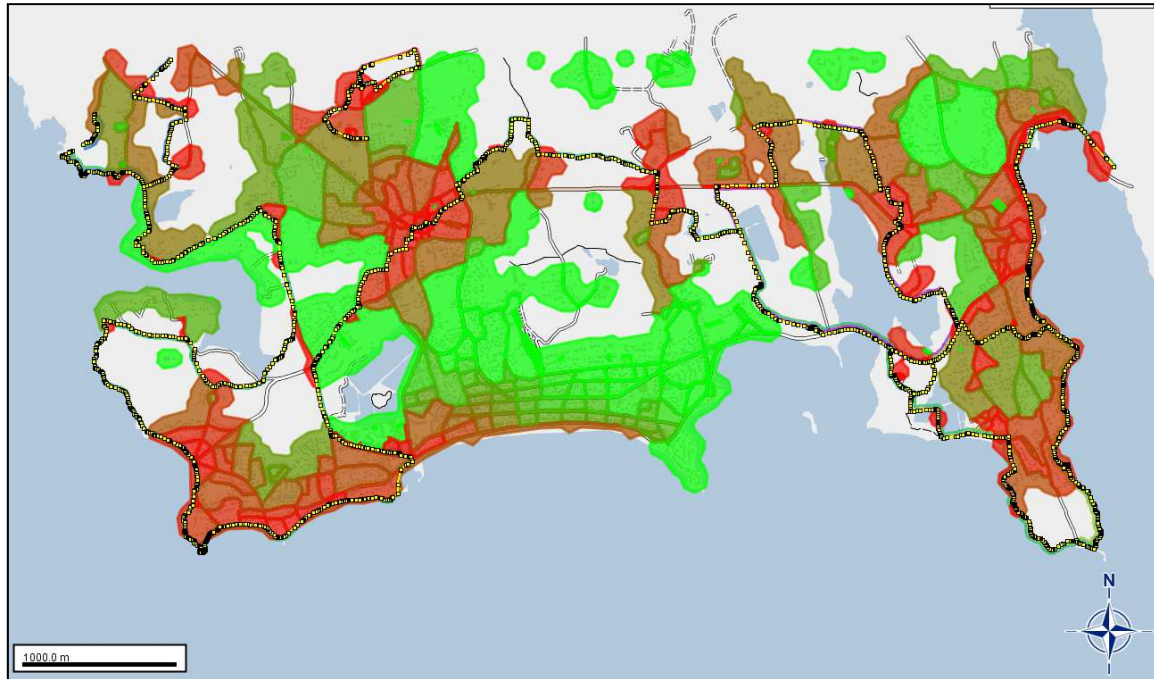


Figure 46 Représentation des distances de chaque îlot aux itinéraires. Les itinéraires apparaissent en pointillés jaune. Plus l'îlot est vert, moins il a de chances d'être impacté par les itinéraires, plus il est rouge, plus il a de chance d'être impacté par les itinéraires. Cette mesure permet d'identifier un seuil à partir duquel la généralisation doit tenir compte des itinéraires.

## IV.2 Optimisation du positionnement d'itinéraires sur un réseau support

Dans cette partie, nous nous intéressons au placement des symboles d'itinéraires sur les tronçons du réseau support de manière à minimiser leur croisement sans considérer les autres objets présents sur la carte. La problématique du placement des itinéraires sur les segments d'un réseau support est proche d'autres problèmes d'optimisation en cartographie, comme la conception de cartes schématiques de transport. Nous nous penchons brièvement sur ce domaine, afin de positionner nos propositions.

### IV.2.1 Approches issues des cartes schématiques

La génération automatique de cartes de métro en style français (1 ruban coloré par ligne, cf. Morrison 1996) est un domaine de recherche important connaissant plusieurs approches. Le but est de générer des cartes de métro similaires à celles actuellement en usage, qui sont réalisées par des opérateurs humains. Cette problématique couvre plusieurs aspects, comme le positionnement idéal des lignes, qui n'est ici pas contraint par des contraintes de géolocalisations fortes, ou encore le placement des labels de noms de ligne et de stations. Nöllenburg (2014) [79] propose un état de l'art des propositions dans ce domaine. Afin d'identifier le cadre de chacune des contributions, il identifie dix principes de design pour ce type de cartes, et identifie, pour chaque proposition, quel principe est concerné. Les dix principes sont les suivants :

- R1 : conserver la topologie du réseau.

- R2 : restreindre les angles entre les différentes arêtes à un ensemble de mesure d'angles (par exemple, seulement des multiples de 45).
- R3 : dessiner chaque ligne le plus possible d'un seul trait, en évitant les angles aigus.
- R4 : éviter les modifications d'orientation des lignes au niveau des stations de correspondances (avec le plus grand nombre de lignes passant).
- R5 : avoir une forte résolution d'angles, c'est-à-dire avoir des angles ayant des valeurs « rondes » (par exemple 30 degrés, 45 degrés, 90 degrés...)
- R6 : rester relativement proche de la réalité géographique.
- R7 : avoir une longueur homogène des arêtes.
- R8 : éloigner les éléments non reliés.
- R9 : éviter les grands espaces vides.
- R10 : placer les noms des stations de façon lisible et non invasive.

À travers la liste de ces principes de design, on remarque que nombre d'entre eux peuvent être transposés à notre problème, mais nécessitent des aménagements dans la mesure où, dans le travail de Nöllenburg (2014) [79], l'objectif est de faire des cartes schématiques, qui caricaturent de façon excessive les objets pour une carte topographique. Par exemple, les principes R7 et R9 ne sont pas des principes pertinents pour une carte topographique (ce qui doit rester une carte d'itinéraires de randonnée). D'autres, comme le placement de la balise R10, relève, dans le cadre de la généralisation, d'un traitement séparé. L'existence d'autres éléments sur la carte rend R8 trop strict : des éléments non reliés dans le réseau d'itinéraires peuvent l'être selon d'autres critères sur la carte. Le R6 est un principe beaucoup plus important pour notre problématique, et R1 est indispensable. R2 et R5 peuvent être des principes pertinents, mais sont excessifs dans notre situation, où nous ne pouvons pas limiter les mesures possibles d'angles à une dizaine de cas particuliers. R3 et R4 sont des principes intéressants. Néanmoins, aucune des approches étudiées par Nöllenburg (2014) [79] ne se focalise sur ces deux derniers. En effet, ils sont essentiellement « constructifs », c'est-à-dire qu'elles partent d'un graphe abstrait à dessiner selon les contraintes déduites des principes de design. Dans notre domaine, nous partons de géométries existantes qu'il est permis de modifier, mais dont les formes caractéristiques et le positionnement doivent être préservés au mieux.

#### **IV.2.2 Le concept de « Stroke » d'itinéraire**

La proposition de cette partie et l'approche décrite dans la suivante ont été étudiées dans le cadre du stage de Master de Jules Teulade-Denantes que j'ai encadré durant la thèse. Les résultats ont fait l'objet d'une publication (Teulade-Denantes et al., 2015) [111]. Cette section présente de façon résumée les contributions du stage.

Selon la théorie du Gestalt (Wertheimer, 1923) [119], l'être humain perçoit comme un tout des ensembles successifs de tronçons de route (ou de tout autre type d'élément linéaire) si une certaine continuité existe entre eux. Sur la base de ce constat, Thomson et Richardson (1999) [112] ont proposé la notion de « stroke » (au sens de « coup de crayon ») pour dé-

crire une telle succession. Ils proposent un algorithme de construction de ces entités géographiques, et proposent de raisonner dessus plutôt que sur les tronçons individuels.

Nous proposons de nous appuyer sur cette notion, et de l'appliquer sur les ensembles de sections de routes, avec des ensembles d'itinéraires communs. C'est-à-dire qu'on construit un stroke à partir d'un ensemble successif de sections de routes, portant les mêmes itinéraires, et présentant des propriétés géométriques de continuité, au sens du principe du Gestalt. Ces propriétés géométriques sont un angle minimum entre les deux sections pouvant appartenir potentiellement au même stroke (angle de déflexion défini empiriquement de 102 degrés) et le même symbole de représentation de la route. La Figure 47 représente des exemples de regroupement par stroke. La Figure 47a montre le découpage initial des 5 tronçons de routes porteurs d'itinéraires, et la Figure 47b les trois strokes identifiés. Ainsi les tronçons 1 et 2 sont regroupés en un stroke 1, les tronçons 3 et 4 sont regroupés dans le stroke 2, et le tronçon 5 correspond au stroke 3.

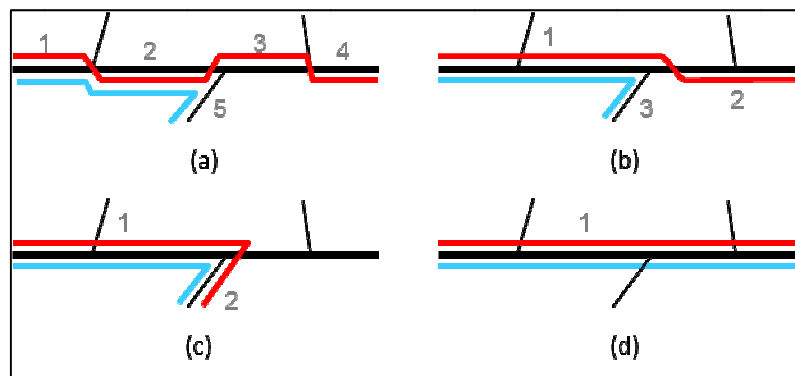


Figure 47 (a) décrit une modélisation initiale. (b) décrit un regroupement par stroke, (c) et (d) décrivent de strokes possibles si les itinéraires avaient eu un cheminement différent. (c) présente deux strokes, car l'angle entre le stoke 1 et le stroke 2 est trop aigü pour supposer que les deux sections puissent appartenir au même stroke. Par contre dans le cas (d), nous ne constatons qu'un seul stroke.

L'intérêt de cette modélisation est de diminuer l'espace de recherche de solutions en établissant un nombre beaucoup plus restreint de choix possibles.

Afin d'intégrer ces strokes dans notre modélisation, nous nous appuyons sur un modèle permettant de décrire la relation d'un stroke de route construit en tenant compte des itinéraires, et d'un « stroke d'itinéraire », qui regroupe l'ensemble des sections d'un même itinéraire portées par un même stroke de route. La Figure 48 présente une modélisation UML de ces relations.

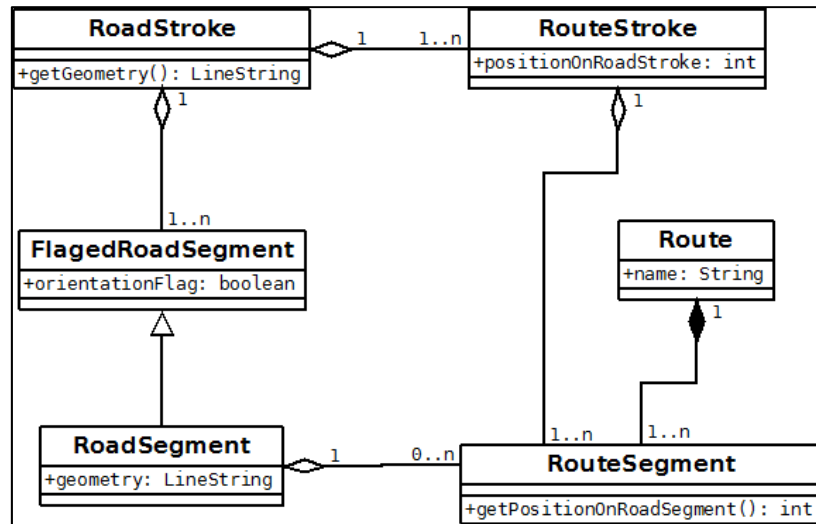


Figure 48 Diagramme de classes représentant les relations entre stroke de route (RoadStroke), stroke d'itinéraire (RouteStroke), section de route (Road Segment) et section d'itinéraire (RouteSegment). Chose importante, les sections de route ne sont pas toutes orientées dans le même sens, il est important, de savoir si elles sont orientées dans le même sens ou non que le stroke. C'est le propos de la classe FlagedRoadSegment.

### IV.2.3 Placement des itinéraires par optimisation de contraintes

Le principe est de s'appuyer sur des méthodes d'Optimisation de Problèmes Contraints (COP) pour identifier le positionnement optimum des itinéraires.

Le principe d'un COP est d'optimiser une fonction d'évaluation globale. Cette fonction d'évaluation couvre l'ensemble des contraintes que l'on souhaite optimiser. Ainsi, il est nécessaire d'identifier les contraintes à satisfaire et de les décrire mathématiquement.

Plusieurs contraintes ont été identifiées :

- C1 : tous les segments d'itinéraires portés par une même section de route doivent être contigus. On s'interdit d'avoir des « trous » entre deux sections d'itinéraires, ou une section d'itinéraire et la section de route.
- C2 : la continuité d'un itinéraire doit être favorisée.
- C3 : le nombre d'intersections entre les tronçons d'itinéraires et entre les tronçons d'itinéraires et les tronçons de routes doit être minimisé.
- C4 : le nombre de cluster d'intersection doit être minimisé. Par cluster d'intersections, nous entendons un ensemble d'intersection qui n'est pas disjoint en termes d'itinéraires impliqués (d'après la définition de Fink et Pupyrev (2013) [32]).

Ces contraintes ont fait l'objet d'une description formalisée décrite Figure 49, nous les présentons d'une façon qualitative ci-dessous.

Formalisme de description des contraintes :

Pour chaque stroke de route  $rd$ , une variable  $x_{rd}$  est définie. Elle représente l'assignation à une position (Figure 41). Le domaine de la variable  $x_{rd}$  est  $D_{n_{rd}}$ , où  $n_{rd}$  est le nombre d'itinéraires portés par la route  $rd$ .

$\bar{x}_{rd} = (\bar{x}_{rd}^1, \dots, \bar{x}_{rd}^{n_{rd}})$  est utilisé pour désigner une assignation de la variable  $x_{rd}$ .  $\bar{x}_{rd} \in D_{n_{rd}}$ .

$X$  est le n-uplet de toutes les variables à assigner. Il contient les variables correspondant à toutes les routes supportant des itinéraires  $X = (x_{rd})_{rd \text{ carrying routes}}$ . Le domaine de  $X$  est  $\prod_{(rd \text{ carrying routes})} D_{n_{rd}}$ . Une assignation de ce n-uplet est notée  $\bar{X}$ ;  $\bar{X} \in \prod_{(rd \text{ carrying routes})} D_{n_{rd}}$ .

Pour une intersection  $j$ ,  $X_j$  est le n-uplet des variables associées aux routes adjacentes à  $j$ :  $X_j = (x_{rd})_{rd \text{ adjacent to } j}$ . Une assignation de  $X_j$  est notée  $\bar{X}_j$ ;  $\bar{X}_j \in \prod_{(rd \text{ adjacent to } j)} D_{n_{rd}}$ .

Une contrainte  $c$  est définie comme un couple  $(X_c, f_c)$ , où:

$X_c$  est le n-uplet des variables associées aux routes impliquées dans  $c$ ;  $\bar{X}_c \in \prod_{(rd \text{ concerned by } c)} D_{n_{rd}}$ .

$f_c: \prod_{(rd \text{ concerned by } c)} D_{n_{rd}} \rightarrow \mathbb{R}$  est la fonction de coût de  $c$ , elle associe une valeur réelle à une assignation  $\bar{X}_c$  de  $X_c$ . Cette fonction est conçue de sorte à ce qu'un faible coût indique une satisfaction élevée de la contrainte.

Les contraintes  $c2$  à  $c4$  identifiées pour notre problème sont formalisées comme suit :

(a)

$c2_j = (X_j, f2_j : \prod_{(rd \text{ adjacent to } j)} D_{n_{rd}} \rightarrow \mathbb{R})$ , avec pour fonction de coût  $f2_j$  calculée comme :

$$f2_j(X_j) = \lambda_2 * \sum_{rt \text{ passing through } j} \ln(\text{minimum\_distance}(rt, j, X_j) + 1),$$

où  $\ln$  représente le logarithme népérien (base  $e$ ), et  $\lambda_2 \in \mathbb{R}^+$ .

(b)

$c3a_{rd} = ((x_{rd}), f3a_{rd}: D_{n_{rd}} \rightarrow \mathbb{R})$ , avec pour fonction de coût  $f3a_{rd}$  calculée comme :

$$f3a_{rd}(x_{rd}) = \lambda_{3a} * (\#_{\text{left}}(rd) - \min(\#_{\text{left}}(rd), \#_{\text{right}}(rd))) \text{ if } \forall i \in [1, n_{rd}], \bar{x}_{rd}^i > 0$$

$$\lambda_{3a} * (\#_{\text{right}}(rd) - \min(\#_{\text{left}}(rd), \#_{\text{right}}(rd))) \text{ if } \forall i \in [1, n_{rd}], \bar{x}_{rd}^i < 0$$

$$\lambda_{3a} * (\#_{\text{left}}(rd) + \#_{\text{right}}(rd) - \min(\#_{\text{left}}(rd), \#_{\text{right}}(rd))) \text{ otherwise}$$

avec  $\#_{\text{left}}(rd)$  (resp.  $\#_{\text{right}}(rd)$ ) le nombre de croisements identifiés à gauche (resp. À droite) de  $rd$ ; et  $\lambda_{3a} \in \mathbb{R}^+$ .

(c)

$c3b_j = (X_j, f3b_j : \prod_{(rd \text{ adjacent to } j)} D_{n_{rd}} \rightarrow \mathbb{R})$ , avec pour fonction de coût  $f3b_j$  calculée comme

$$f3b_j(X_j) = \lambda_{3b} * \text{count\_crossings}(j, X_j) \text{ avec } \lambda_{3b} \in \mathbb{R}^+.$$

(d)

$C4_j = (X_j, f4_j : \prod_{(rd \text{ adjacent to } j)} D_{n_{rd}} \rightarrow \mathbb{R})$ , avec pour fonction de coût  $f4_j$  calculée comme

$$f4_j(X_j) = e^{\lambda_4 * \text{count\_crossing\_clusters}(j, X_j)} - 1 \text{ avec } \lambda_4 \in \mathbb{Z}^+.$$

Figure 49 Formalisme du COP, avec description de chaque type de contraintes, d'après Teulade-Denantes et al., 2015.



La contrainte C1 est gérée implicitement par notre modélisation, qui interdit une assignation à une position, si la position plus proche du symbole de route n'est pas assignée (par exemple, un itinéraire ne peut pas être positionné en position 2 si la position 1 est vacante). Seule exception la position 0 que nous considérons non assignables dans notre situation. Les autres contraintes doivent être décrites de façon mathématique afin d'être évaluées et exploitées dans un modèle d'optimisation de contraintes.

La contrainte C2 doit être évaluée pour chaque croisement entre deux strokes différents. Il existe donc une contrainte  $c_{2j}$  pour chaque intersection  $j$ . Elle est définie comme suit : sa satisfaction se mesure selon la distance constatée entre les positions de part et d'autre de l'intersection. La Figure 50 montre un exemple de mesure de cette distance. Une transformation logarithmique est appliquée pour chaque itinéraire, ainsi qu'une multiplication par un coefficient  $\lambda_2$ . La formule est décrite dans la Figure 49a.

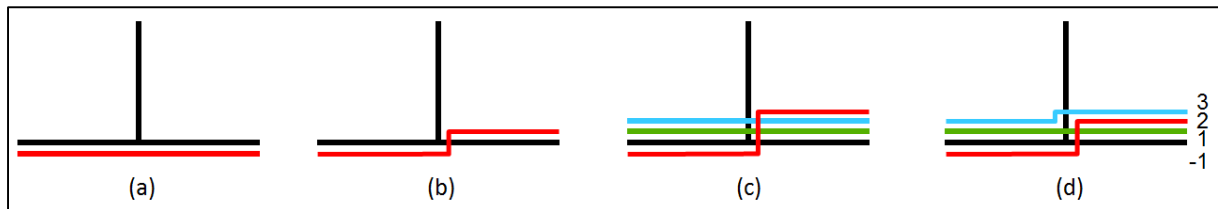


Figure 50 L'itinéraire rouge passe respectivement de la position -1 à (a) -1, (b) 1, (c) 3 et (d) 2. Ce qui signifie que la distance pour l'itinéraire rouge à cette intersection est de respectivement de (a) 0, (b) 2, (c) 4 et (d) 3.

La contrainte C3 est utilisée pour diminuer le nombre de croisements entre les itinéraires. Cette contrainte s'applique à deux situations : les croisements en dehors des intersections (on parlera de la contrainte C3a) et les croisements au sein des intersections (on parlera alors de la contrainte C3b). Par intersection, nous entendons un carrefour auquel aboutissent au moins deux strokes.

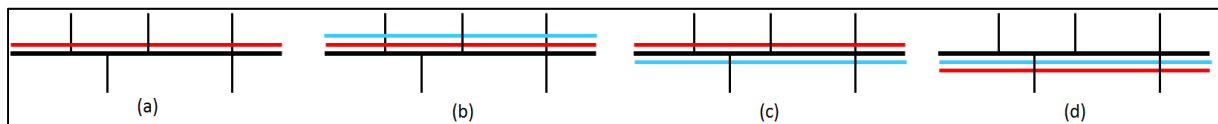


Figure 51 Évaluation du nombre de croisements de routes. Dans les cas (a) et (b) les itinéraires croisent trois routes. Dans le cas (d), ils croisent deux routes seulement. Dans (c), le rouge croise 3 routes et le bleu en croise 2. La mesure de la contrainte C3a donne respectivement (a)  $3-2=1$ , (b)  $3-2=1$ , (c)  $3+2=5$ , (d)  $2-3=-1 > 0$ .

C3a compte le nombre de routes croisées le long des sections de routes traversées par un itinéraire. Elle est donc définie pour chaque stroke de route  $rd$ , avec comme mesure de satisfaction la différence entre le nombre de routes croisées effectivement et le nombre de routes non croisées, le tout multiplié par un facteur de pondération de la contrainte  $\lambda_{3a}$ . La Figure 51 montre 4 cas de figures avec les mesures effectives. La formule est décrite dans la Figure 49b.

La contrainte C3b s'intéresse au nombre total de croisements entre itinéraires et entre itinéraires et routes qui se produisent à une intersection. Décompter le nombre de croisements d'un itinéraire avec les autres itinéraires et les autres routes à une intersection n'est pas



évident. Nous utilisons pour cela une modélisation abstraite de l'interaction et des itinéraires incidents, construit sur la proposition de Fink et Pupyrev (2013) [32], illustrée Figure 52. Pour compter les croisements, nous utilisons un algorithme dont le principe est l'identification, pour un itinéraire donné, du nombre d'intersections d'itinéraires d'une autre couleur. Une fois ce décompte effectué, on retire l'itinéraire, puis on recompte pour un autre, jusqu'à arriver à ne plus avoir d'itinéraire dans notre représentation abstraite. La formule est décrite dans la Figure 49c.

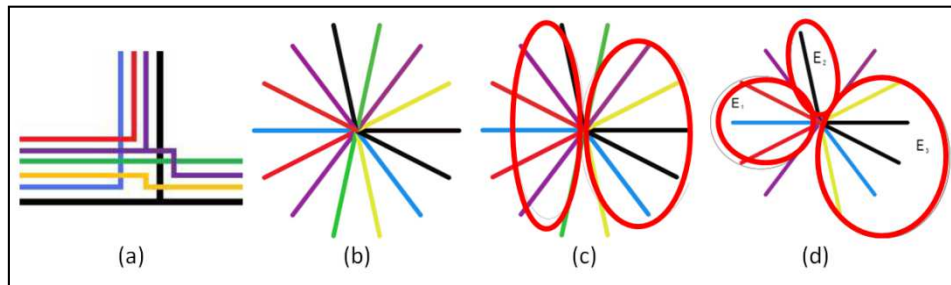


Figure 52 Exemple de décompte de croisements à une intersection. (a) présente l'intersection telle qu'elle peut apparaître sur la carte, (b) montre une vision abstraite « en étoile » de l'intersection, (c) montre les deux premiers clusters identifiés de part et d'autre de l'itinéraire vert. Les couleurs violet, noir et bleu se trouvant dans les deux clusters, on décompte une valeur de 3. Dans (d), nous avons retiré l'itinéraire vert. On effectue la même opération avec l'itinéraire violet. On observe trois clusters. Les routes noires et bleues apparaissent dans deux clusters, on décompte donc un score de 2.

Enfin la contrainte C4 s'intéresse aux clusters de croisements. Elle part du constat que des intersections sont plus facilement lisibles si elles sont le fait d'un petit nombre d'itinéraires avec de nombreux autres. La mesure se fait pour chaque intersection : on identifie l'itinéraire qui croise le plus grand nombre d'itinéraires et de routes (par exemple dans la Figure 52, l'itinéraire bleu est celui qui croise le plus d'itinéraires). Lorsqu'on le retire, le noir en croise le plus, on le retire, ainsi que le violet, et on arrive à une situation sans intersection, pour cela il a fallu retirer trois itinéraires. La valeur de la fonction de C4 pour cette intersection est alors de 3, sur laquelle on applique une fonction exponentielle. La formule est décrite dans la Figure 49d.

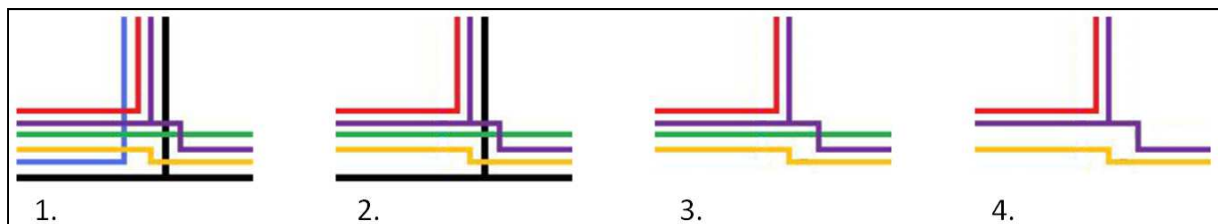


Figure 53 Identification du nombre de clusters. À chaque itération, on détecte l'itinéraire ayant le plus d'intersections, puis on le retire. En (1), l'itinéraire bleu ayant le plus d'intersections, il est retiré, en (2) c'est le noir, en (3) le vert. Arrivé en (4), nous ne constatons plus aucune intersection, il a fallu retirer trois itinéraires pour cela, nous décomptons trois clusters.

Lorsque nous avons identifié toutes les contraintes pour notre carte, nous arrivons à l'étape où nous cherchons à minimiser la somme de toutes ces fonctions. Une approche par backtracking nous amène à avoir des résultats corrects, mais pas assez rapidement pour supporter une application sur une grande zone. Une méthode par recuit-simulé nous a permis d'arriver à une solution rapidement. La Figure 54 montre un exemple d'allocation de posi-

tions d'itinéraire après recherche d'un optimum par recuit-simulé. Les résultats obtenus ont fait l'objet d'une évaluation par les cartographes de l'IGN. Ils ont été jugés satisfaisants d'un point de vue cartographique en termes de placement des itinéraires de manière à limiter leurs croisements. Mais nous y observons que nombre de problèmes ne sont pas traités. Par exemple, suite au placement des itinéraires, une partie d'un cours d'eau n'est pas visible, et il n'est pas possible d'identifier si le chemin croise un de ces cours d'eau. La prise en compte d'autres éléments non intégrés au COP risque d'être trop complexe pour une gestion par COP, d'autant plus que parmi les solutions possibles pour gérer ces problèmes figurent le déplacement d'un itinéraire de l'autre côté de « sa » route, mais aussi l'application d'opération de généralisation aux objets cachés par les itinéraires (déformation, etc...) Nous revenons donc sur une solution par agents pour résoudre ce problème, solution qui intègre les résultats de l'application du COP.

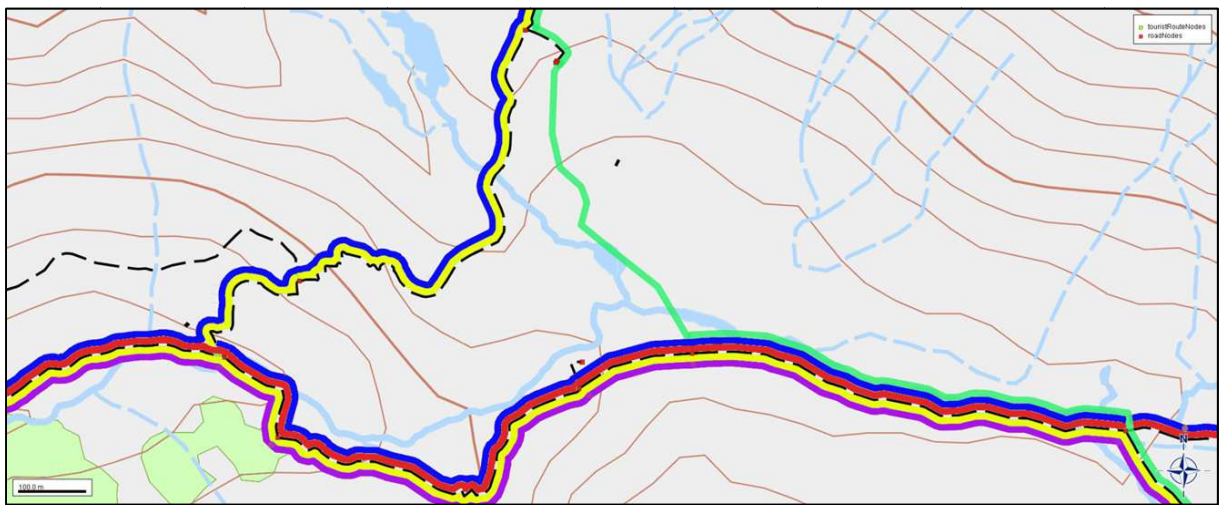


Figure 54 Extrait du résultat du COP basé sur le recuit-simulé. Quelques problèmes, non gérés par l'approche par COP sont ici visibles, comme une partie des tronçons de cours d'eau recouverts par les itinéraires.

### IV.3 Interactions entre agents pour la réalisation de cartes de randonnées

Les interactions entre routes et itinéraires, et itinéraires entre eux lorsqu'ils sont supportés par une même route, peuvent être gérées de façon efficace par l'approche COP, comme expliqué dans la section précédente. Néanmoins, la présence d'itinéraire influe sur les autres éléments de la carte. Ainsi, il doit être possible de changer des éléments de la carte si ces derniers entrent en conflit de superposition avec le positionnement des itinéraires, ou bien influencer sur le choix de positionnement des itinéraires si les objets ne peuvent pas être suffisamment modifiés (pour des raisons autres). Ce genre de situation encourage une approche décentralisée proposée par l'approche SMA pour la généralisation. Nous proposons donc de nous appuyer sur DIOGEN, afin d'apporter des solutions.

### IV.3.1 Interactions d'un itinéraire dans un environnement de route support

Le modèle par COP proposé dans la précédente section s'applique comme un fonctionnement clos. C'est-à-dire qu'il est difficile d'introduire lors de l'exécution d'un tel système des informations extérieures en dehors des données initiales, car une telle introduction implique de recalculer certaines contraintes. Il est possible d'exécuter le positionnement par COP *a priori* de l'exécution de DIOGEN, ce qui est logique en termes de préparation, mais risque de faire perdre beaucoup de décisions prises par le COP lors de l'exécution du SMA, sans véritable possibilité de correction, tout du moins en s'appuyant sur le COP. Une exécution *a posteriori* est néanmoins bien plus problématique, car les modifications apportées causeront des conflits qu'il faudra de nouveau résoudre. Une exécution du COP au sein du SMA peut être envisagée, mais elle implique que, dans un premier temps, nous définissions nos éléments comme des entités agents et environnements de notre modèle. Une modélisation d'un CSP sous forme de comportement d'agent a été proposée par Breton et Jussien (2004) [9]. Bien qu'une telle approche ne se soit pas faite avec l'utilisation concomitante d'autres comportements, cela reste une piste envisageable pour la description des comportements propres au COP au sein d'un SMA.

Dans le cadre de nos expérimentations, nous avons opté pour une exécution du COP *a priori* de l'exécution de DIOGEN.

Dans la mesure où nous travaillons avec des strokes tels que définis dans la précédente section plutôt qu'avec des routes, nous pouvons modéliser le stroke de route comme un agent hébergeant un agent modélisant un stroke d'itinéraire. L'environnement encapsulé par l'agent stroke de route a un fonctionnement reproduisant la modélisation des positions décrite en page 85.

Dans un tel environnement, un itinéraire peut effectuer une interaction de type **Changer de position**. Une telle interaction est influencée par des contraintes partagées avec d'autres éléments de l'espace, comme des contraintes de **non superposition**. Elle pourrait aussi être influencée par d'autres contraintes, inspirée des contraintes décrites dans le COP, dans une perspective d'intégration du COP dans DIOGEN.

### IV.3.2 Interaction des itinéraires avec des objets compacts

La présence d'éléments à proximité des routes supportant des itinéraires implique des interactions entre ces objets et les strokes d'itinéraires. Si ces objets sont relativement plus longs que la route, et déformables, comme un cours d'eau ou un trait de côte, alors l'approche ne sera pas la même que si ce sont des objets petits et compacts, comme un bâtiment. Dans ce dernier cas, il est nécessaire que les objets concernés soient des objets du voisinage de la route supportant les itinéraires.

Nous proposons ici de nous appuyer sur une modélisation du voisinage de la route supportant les itinéraires. La modélisation d'un tel voisinage implique l'existence en tant qu'agent d'un tel voisinage. Cet agent encapsule un environnement qui héberge les agents représentant les objets petits et compacts positionnés à proximité.

Nous construisons ce voisinage en deux temps :

- d'abord par détection des bâtiments à proximité de la route, avec l'aide d'un buffer, de façon similaire à la détection du voisinage des impasses ;
- d'autre part par détection des bâtiments à proximité des bâtiments, de proche en proche, jusqu'à une distance maximale, précédemment détectés comme faisant partie du voisinage.

Nous appelons agent embarqué le voisinage ainsi identifié. La Figure 55 montre les relations d'hébergement entre les agents impliqués.

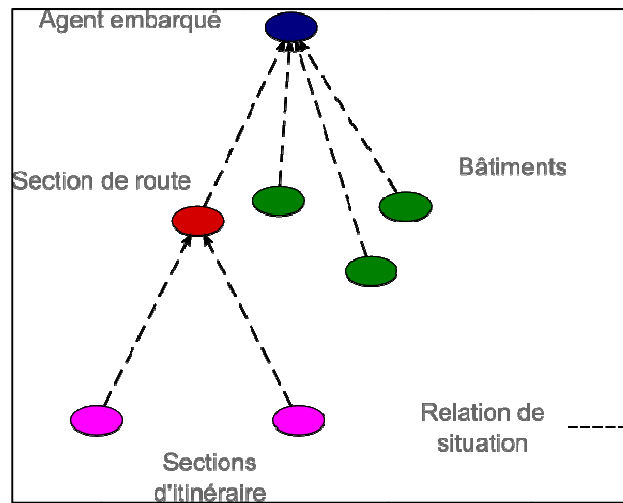


Figure 55 Relations entre agent au sein de la modélisation d'un voisinage.

Cet environnement fonctionne sur une logique de coordonnées locales, similaire au système des environnements embarqués décrit dans la section III.4.2 **Définition d'un espace embarqué pour le voisinage**, p. 75. Ainsi, un agent situé dans cet environnement est positionné à l'intérieur avec des coordonnées propres à cet environnement. Ainsi, une modification de la route supportant les itinéraires (ou le changement de côté d'un itinéraire), entraîne une modification du référentiel de cet environnement, et par conséquent du positionnement absolu sur la carte des agents situés dans cet environnement, ce qui se traduit par un déplacement des agents selon un vecteur de déplacement dont l'orientation joint le projeté du centroïde du compacte sur la route et le centroïde, et dont la valeur dépend de la largeur du symbole d'itinéraire, amortie par la distance de l'objet.

Nous présentons ce système de coordonnées locales, qui permet de caractériser une relation hiérarchique d'hébergement d'objets dans un voisinage auquel ils appartiennent. Pour cela, un système de coordonnées locales est proposé. Un objet possède deux coordonnées dans l'environnement/voisinage auquel il appartient :

- la position en abscisse curviligne du projeté du centroïde sur le stroke.
- la distance des deux symboles : celui de l'objet et celui du stroke (combinant la largeur du symbole de la route et celle des itinéraires).

Lorsque la situation des itinéraires change par rapport au stroke support, les bâtiments doivent bouger afin de maintenir la distance entre les deux symboles. Cette interaction de **Déplacement** de bâtiments reste liée à la modification de la position de l'itinéraire, et est une conséquence directe de ce changement de côté. La Figure 56 montre un scénario type de déclenchement des interactions de déplacement.

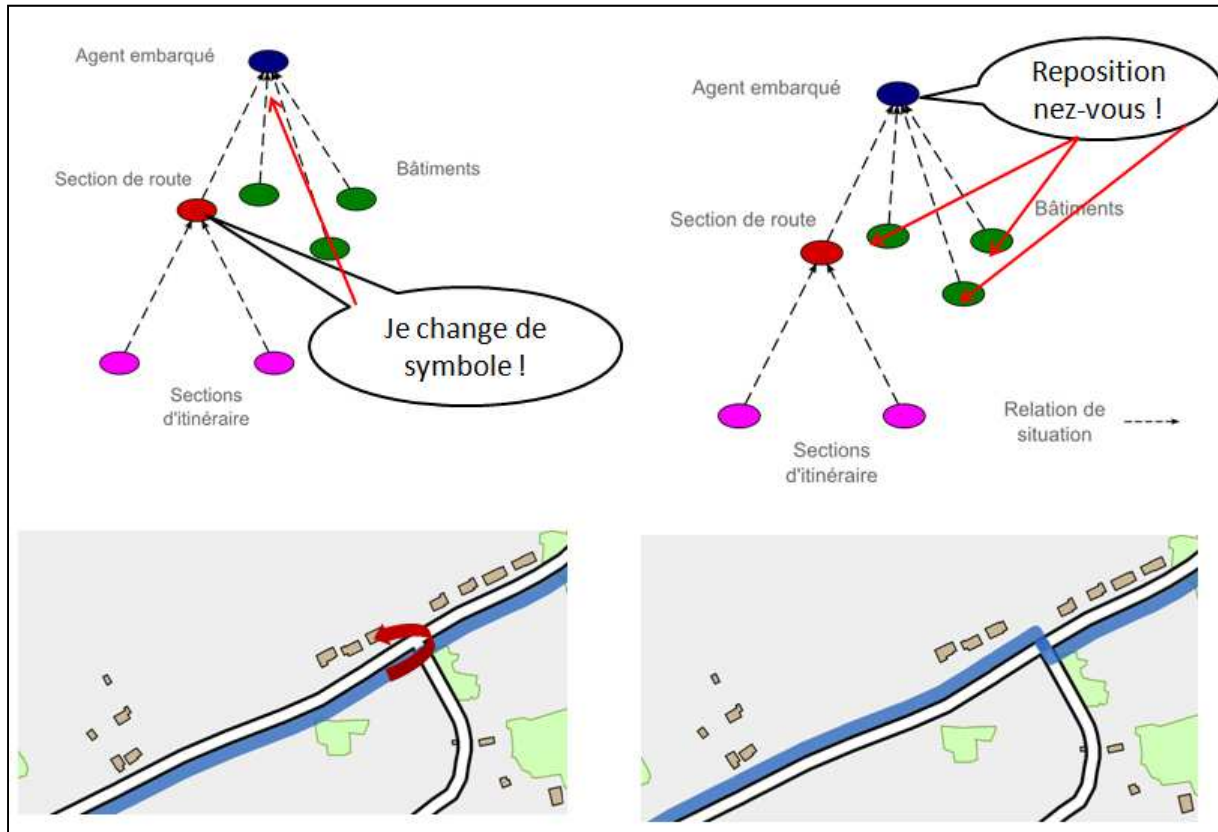


Figure 56 Exemple de scénario. À gauche, l'itinéraire s'apprête à changer de position, informant ainsi la route que l'ensemble route-itinéraire va changer de symbole.

Bien entendu, de telles modifications peuvent impacter les autres objets en relation qui ne font pas partie du voisinage de l'agent en question. Notre approche consiste à s'assurer que les agents des objets concernés sont majoritairement dans l'environnement (en détectant le voisinage de proche en proche). La Figure 57 décrit un exemple de système de coordonnées locales à un environnement, et l'impact d'une modification du positionnement d'un itinéraire.

Afin de permettre une solution en cas d'impossibilité de déplacement des bâtiments (qu'on peut détecter *a posteriori* avec la détection des bâtiments qui se chevauchent), l'itinéraire doit pouvoir changer de côté. Une interaction de **Changement de position d'itinéraire** est implémentée. Afin de détecter *a priori* ce problème, nous nous appuyons sur les relations entre contraintes et interactions. L'interaction de **changement de position d'itinéraire** est encouragée par la contrainte de **respect de la distance initiale** (qui se montre favorable si insatisfaite). Cette contrainte relationnelle, partagée par le stroke et chaque objet compact susceptible d'être recouvert par un symbole d'itinéraire, est pleinement satisfaite si l'objet

compact est à une distance proche de celle de l'objet initial par rapport à la géométrie de la route. Si ce n'est pas le cas, cette contrainte n'est pas satisfaite, et son score de satisfaction est d'autant plus bas que la distance augmente. L'interaction de repositionnement des bâtiments est aussi encouragée par cette contrainte de non superposition, mais peut être découragée par une contrainte de **non superposition** des objets de l'îlot, portée par l'îlot, à la manière du modèle AGENT, contenant les bâtiments susceptibles d'être déplacés. Une contrainte proxy est en fait créée sur les bâtiments concernés, qui pointe vers la contrainte de l'îlot, pour deux raisons :

- d'abord clarifier les relations entre agents, sans ajouter au schéma de relations les îlots urbains. Les relations restent définies entre les bâtiments et la route ;
- ensuite dans une perspective future d'utiliser une contrainte plus fine de mesure de la possibilité de déplacer les bâtiments (sur la base d'une densité locale ne s'étendant pas à tout l'îlot, voire dans un cas où il n'y a pas d'îlot urbain, mais une zone rurale, pouvant se montrer localement dense).

En matière d'ordonnancement, nous organisons la file d'attente de façon à ce que le positionnement de l'itinéraire se fasse une fois les différentes interactions des bâtiments effectués. Si un déplacement de positionnement se fait après coup, une nouvelle activation du voisinage est faite pour valider la contrainte, et agir éventuellement.

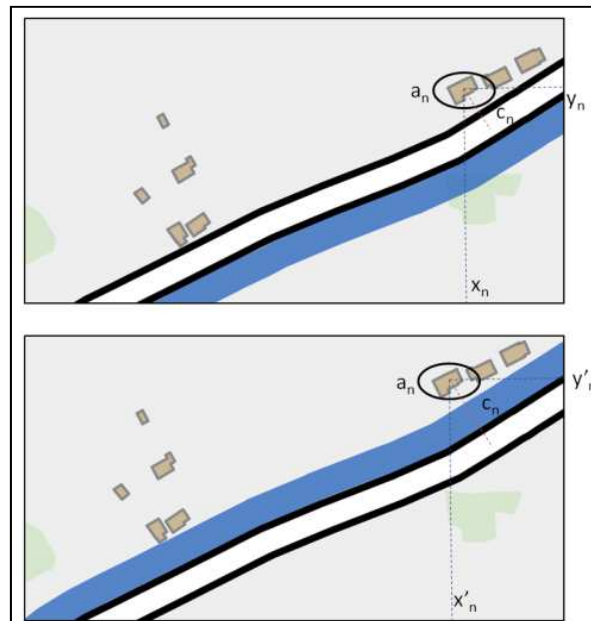


Figure 57 Description du positionnement relatif de bâtiment dans un voisinage. Le bâtiment est positionné aux points de coordonnées  $(a_n, c_n)$  dans le repère du voisinage du stroke. Lorsque le stroke change la position de l'itinéraire, il est nécessaire de changer les coordonnées dans le repère général (passage d'une position  $(x_n, y_n)$  à une position  $(x'_n, y'_n)$ ).

### IV.3.3 Interaction des itinéraires avec un objet déformable

L'existence d'éléments déformables à proximité des routes supportant les itinéraires nécessite de pouvoir modifier ces derniers. Une telle modification de déformable doit se faire en essayant de maintenir la forme générale de l'objet déformable. Pour ce faire, nous nous ap-

puyons sur les agents points issus du modèle GAEL. Les agents points concernés par les déplacements d'itinéraires sont ceux susceptibles d'être recouverts par les itinéraires en question. Si une telle possibilité existe, une relation entre les agents en question est établie. Les agents points sont susceptibles d'être déplacés, et le sont si jamais les itinéraires changent de côté.

Les points du voisinage sont identifiés grâce à un buffer établi autour de la géométrie du stroke. La taille de ce buffer prend en compte la largeur de symbole de route additionnée à la somme des largeurs de symboles de tous les itinéraires. À cela, il faut ajouter une valeur pour le symbole de l'objet déformable. Dans le cadre d'un objet à géométrie linéaire, la moitié de la largeur du symbole sera utilisée. Dans le cadre d'un objet à géométrie surfacique, la moitié de l'épaisseur du contour est retenue, additionnée à un seuil de séparabilité (i.e. la distance minimale attendue entre les symboles, définie dans les spécifications).

Nous avons identifié deux façons d'établir une relation entre les objets :

- par une contrainte relationnelle entre un stroke portant un ensemble d'itinéraires et l'objet déformable, si au moins un de ses points est contenu dans le buffer. Cette méthode permet de proposer des solutions au niveau du déformable et d'y appliquer un algorithme au niveau de l'objet déformable, comme par exemple une méthode de résolution par moindres carrés ;
- par une contrainte relationnelle entre le stroke portant un ensemble d'itinéraires et chaque point contenu dans le buffer. Cette approche implique l'intégration des points de la géométrie comme agents interagissant avec d'autres agents géographiques. Des comportements d'agents points inspirés du modèle GAEL peuvent être utilisés dans ce contexte.

Les deux propositions ont été testées. Elles montrent toutes deux des résultats satisfaisants. L'avantage de la seconde approche est d'explicitier les contraintes et les modifications des points dans le modèle DIOGEN, et d'ouvrir la possibilité à d'autres interactions et à une prise de décision tenant compte de ces aspects.

Pour la première approche, une interaction permettant le déplacement des points en utilisant un système par moindres carrés, implémentant le modèle de Harrie et Sarjakoski (2002) [45], a été utilisée. L'ensemble des points susceptibles de poser problème se voit donner une contrainte de déplacement au sein du système par moindres carrés, que ce dernier va tenter de résoudre, tout en conservant la forme générale de l'objet. Le déclenchement de l'interaction correspond à la résolution de ce problème contraint. Elle est motivée par une contrainte relationnelle de non superposition entre l'objet déformable et le stroke porteur d'itinéraire qui se montre favorable à l'exécution de l'interaction de **Déformation par moindres carrés** si elle est insatisfaisante.

Nous nous focaliserons sur la deuxième approche pour la suite. Pour utiliser DIOGEN de façon à permettre la déformation d'objets déformables selon la position d'itinéraires par rapport à un stroke, nous devons intégrer les contraintes et interactions de GAEL (voir section

**III.3.3 Intégration d'éléments du modèle GAEL**, page 72). Nous devons aussi ajouter une nouvelle contrainte et la rendre compatible avec l'interaction de déplacement du point, afin que le point prenne en compte cette contrainte pour se déplacer. Nous avons défini cette contrainte comme suit. Cette contrainte est satisfaite lorsque le point n'est pas sous un symbole d'itinéraire porté par le stroke. S'il est sous un symbole d'itinéraire, la satisfaction est inversement proportionnelle à la distance qu'il sera nécessaire au point pour « sortir » de la zone recouverte par le symbole. La distance nécessaire est calculée suite au calcul du vecteur nécessaire à effectuer ce déplacement, qui relie la position actuelle du point à son projeté sur la surface fabriquée par le buffer.

L'interaction de déplacement des points tient compte de toutes les contraintes partagées par l'agent point. Pour chaque contrainte, un vecteur de déplacement est calculé (dans le cas de contraintes adaptées de GAEL, voir section **III.3.3 Intégration d'éléments du modèle GAEL**, page 72 ; dans le cas de la contrainte partagée avec l'itinéraire, voir ci-dessus).

Afin de maintenir une cohésion, il est important que les extrémités des segments restent cohérentes les unes avec les autres. Afin d'empêcher les incohérences, nous relierons plusieurs points représentant un nœud au même point, afin de permettre une mise à jour cohérente des géométries de sections adjacentes. Cette mécanique est illustrée dans la Figure 58.

En matière d'ordonnancement, nous organisons la file d'attente de façon à ce que le positionnement de l'itinéraire se fasse avant l'activation des points par le voisinage. Tout changement de position de l'itinéraire active l'agent voisinage, qui lui-même ajoute les points dans la file d'activation.



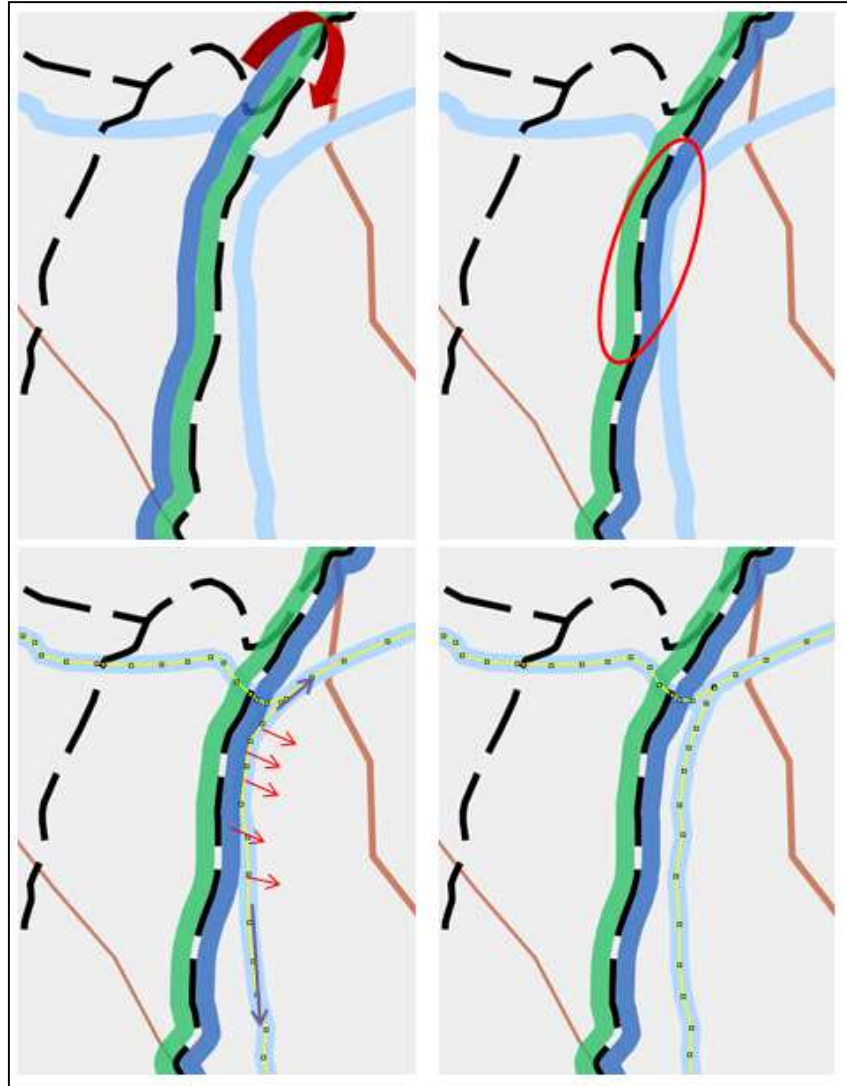


Figure 58 Scénario de déplacement partiel et de déformation d'un objet déformable. En haut à gauche, l'itinéraire bleu souhaite changer de position sur la section de route. En haut à droite, cela cause des conflits. En bas à gauche, un déplacement des points de l'objet déformable est souhaitable, il est par contre nécessaire de répercuter cette déformation aux éléments suivants. Le résultat de la déformation est observable en bas à droite.

Si un tel déplacement de points est produit, la forme est perdue, et doit donc être reconstruite. Les agents submicro et leur contrainte jouent ce rôle. L'exécution des points selon les interactions de GAEL se fait selon une logique propre à GAEL de conservation des submicro, en s'appuyant sur leurs contraintes spécifiques. Ces contraintes sont :

- pour les relations impliquant un agent submicro segment :
  - o préservation de la taille
  - o préservation de l'orientation
- pour les relations impliquant un agent submicro angle :
  - o préservation de la valeur.

La Figure 59 montre les relations entre les agents impliqués dans ce scénario.

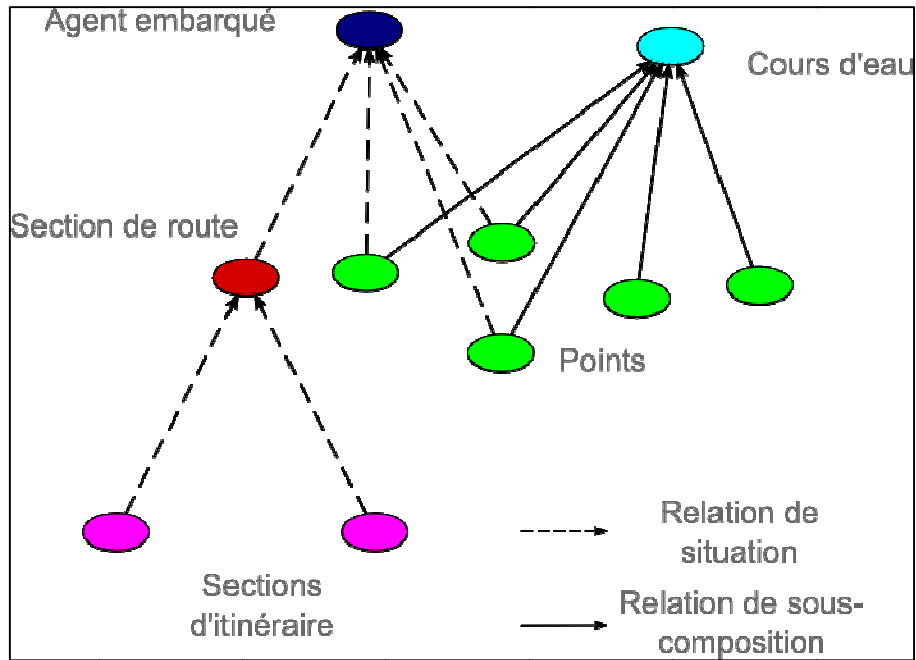


Figure 59 Modélisation des relations entre un cours d'eau et une route portant des itinéraires. Voir section II.2.3 Les aspects multi-niveaux dans les SMA de généralisation existants pour les types de relations.

## IV.4 Résultats et analyse

Pour chacun des situations, nous avons procédé à des expérimentations locales. Ainsi, nous avons identifié des situations particulières, pour lesquelles des calibrage et ajustements ont été nécessaires.

### IV.4.1 Déplacement de compacts

Les expérimentations sur le déplacement d'objet compact ont mis en avant les risques de répercussions appliqués sur les objets non directement liés aux itinéraires. La plupart des expérimentations ont été effectuées en zone de faible densité sur des ensembles de petits bâtiments. La Figure 60 montre plusieurs cas de bâtiments déplacés. Pour ce faire, il a été nécessaire de décaler plusieurs bâtiments, et d'avoir une définition large de voisinage.

Dans le cadre de zone avec beaucoup d'espace, les itinéraires réussissent à pousser les bâtiments, et à maintenir une position cohérente.

Le principal obstacle à un déplacement des bâtiments, est l'absence de place dans les îlots. Nous ne détectons pas ce genre de problème *a priori*, mais nous évaluons *a posteriori* si les bâtiments se chevauchent au sein de l'îlot, si c'est le cas, nous annulons l'opération, et le stroke d'itinéraire change de côté.



Figure 60 Exemples de bâtiments déplacés. À gauche la situation initiale. Au milieu l'itinéraire a changé de côté. À droite les bâtiments ont été déplacés.

#### IV.4.2 Déplacement et déformation de déformables

Le déplacement de déformable a suivi deux approches : une approche avec système contraints dans son ensemble, utilisant les moindres carrés, et une autre avec système utilisant exclusivement les agents. Pour l'approche avec les moindres carrés, nous avons testé sur des traits de cotes. Pour l'approche avec agents, nous avons testé sur des sections de cours d'eau. La Figure 61 montre plusieurs situations de déplacement et déformation de trait de côte. D'un point de vue cartographique ces déplacements restent acceptables, mais nécessitent néanmoins une validation à l'utilisation de cartes en cadre réel.

Dans le cadre des cours d'eau, nous obtenons des résultats convenables. Un travail de calibrage a été effectué, afin d'assurer une convergence plus rapide du déplacement vers une solution convenable.

Néanmoins, les tests se sont effectués sur des régions espacées. Des tests doivent être menés dans le cadre de régions plus contraintes, avec plus d'éléments.

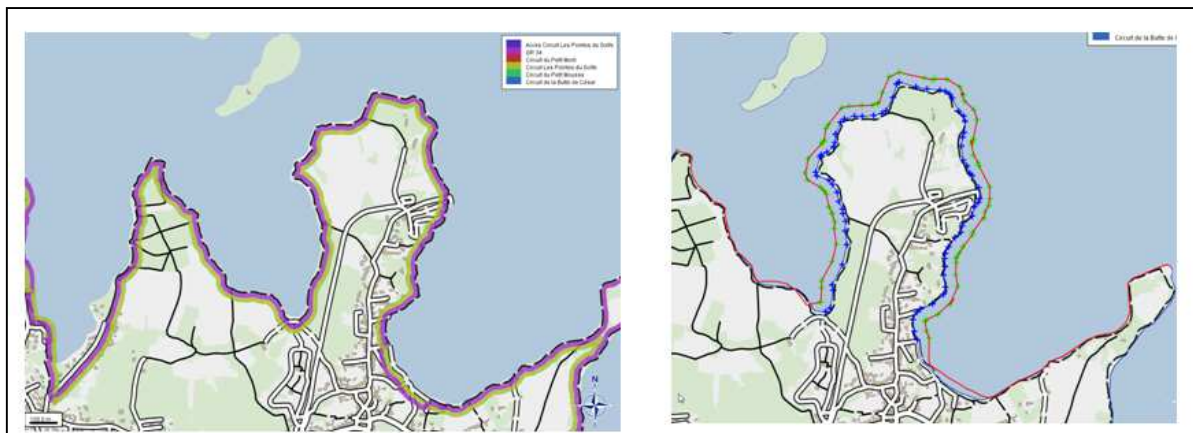


Figure 61 Illustration du déplacement de trait de côte par moindres carrés. À gauche la situation initiale. À droite, une illustration du déplacement du trait de côte (ancienne position avec des points bleus, nouvelle position avec des points verts reliés en rouge).

### IV.5 Synthèse du chapitre

Dans ce chapitre, nous avons identifié les problèmes propres au cas des cartes thématiques, et tout particulièrement des cartes d'itinéraires de randonnées. Après avoir commencé à cerner les types d'utilisation de telles cartes, nous avons proposé une méthode de position-

nement des itinéraires à base de COP. Puis, nous avons proposé d'utiliser notre modèle DIOGEN pour gérer les conséquences de la présence de plusieurs itinéraires. Nous avons proposé plusieurs interactions, que nous avons implémenté et évaluée.

Afin de compléter ce travail, et proposer une méthode complète et fonctionnelle, nous identifions les pistes et perspectives suivantes :

- Dans un premier temps, la gestion de l'ordonnancement des agents et de l'orchestration de leurs interactions, afin de permettre une généralisation globale d'une carte avec itinéraires.
- Ensuite, la prise en compte d'autres contraintes visuelles sur des éléments variables, comme les couleurs.
- Dans une perspective d'intégration du COP comme comportement d'agent, nous pourrions considérer que l'ensemble des strokes appartient à un réseau, le réseau de routes portant les itinéraires. Modélisé comme un agent, ce réseau est l'entité qui peut être amenée à appeler l'exécution du COP. Une telle approche est néanmoins problématique, car elle amène à identifier ce qui pourrait déclencher une telle interaction. Selon notre modèle, l'existence d'une contrainte globale, construite sur la base de fonction de satisfaction présentée dans la section précédente est la base de la mesure de satisfaction de cette contrainte.
- Enfin, la résolution locale d'interactions, et l'identification de problèmes particuliers dynamiquement, pouvant notamment conduire à des solutions nouvelles, comme l'utilisation de coalition de points au sein d'un élément déformable.



# Chapitre V :

## Vers des *patterns* multi-niveaux

---

Dans les précédentes parties, nous avons étudié les différents modèles multi-agents pour la généralisation, et nous en avons fusionné le comportement dans un modèle unique, DIOGEN, mettant en avant les aspects multi-niveaux à la façon du modèle PADAWAN. Nous avons ensuite utilisé ce modèle pour nous intéresser à la généralisation de cartes contenant des données thématiques, et plus particulièrement aux cartes d'itinéraires. Durant ce processus, nous avons identifié plusieurs structures communes dans les différents modèles, et pour différents problèmes. L'identification de ces structures nous amène à la possibilité de les formaliser. C'est sous la forme de *patterns* d'analyse que nous présentons ce travail.

Dans une première section, nous expliciterons nos motivations à la conception de *patterns*, puis dans une deuxième section, nous reviendrons sur le cadre formel dans lequel nous exprimerons les *patterns*. Enfin, dans une troisième partie, nous exposerons les différents *patterns* identifiés. Ce chapitre est un développement de l'article (Maudet et al., 2015) [67].

### V.1 Motivations à l'identification de *patterns*

Les évolutions dans la conception de modèles multi-niveaux ont conduit à l'élaboration de modèles visant la généricité des architectures. Un autre type de généricité est d'établir des liens entre les solutions apportées à différents problèmes. Cette généricité porte sur les fonctionnalités d'un modèle. Pour cela, nous identifions des situations génériques récurrentes dans plusieurs modèles d'application, et leur formalisation sous forme de patrons, ou *patterns*. Ainsi, Couturier et al. (2010) [15], proposent des *patterns* d'analyse multi-agents. Comme nous l'avons déjà discuté dans le second chapitre, certains des modèles génériques pour les SMA s'intéressent particulièrement à leurs aspects multi-niveaux, ce qui est le cas de PADAWAN. D'ailleurs, un travail d'identification de *patterns* multi-niveaux en simulation a déjà été entamé (Mathieu et al., 2014 [65], Mathieu et al., 2016 [80]).

Lors de la résolution de problèmes multi-niveaux dans le domaine de la généralisation, nous avons rencontré des motifs structurels ou comportementaux récurrents. Nous supposons que ces motifs sont susceptibles d'être utilisés pour l'analyse et la modélisation d'autres situations dans d'autres domaines, voire pour l'implémentation de solutions. Pour cela, nous essayons de décrire ces situations récurrentes d'une façon abstraite (en les caractérisant *a minima* et indépendamment du domaine d'où elles sont issues), d'identifier des exemples où

elles apparaissent hors du domaine d'origine, et de montrer les liens qu'elles entretiennent les unes avec les autres. Nous sommes confrontés à un problème spatialisé impliquant des définitions variées de ce qui peut constituer un groupe d'agents et une multiplicité des aspects multi-niveaux, tant dans la construction des différents niveaux que dans la nature de ces derniers. En termes de construction de groupes, nous rappelons ici plusieurs cas :

- la création d'agents représentant des entités cohérentes d'un point de vue cartographique, et sur lesquelles l'application d'algorithmes particuliers peut s'avérer efficace (par exemple, en ville, élimination de bâtiments décidée au niveau de l'îlot urbain pour garder les plus représentatifs la relation meso/composant, voir section **I.2 Les systèmes multi-agents pour la généralisation**) ;
- la division d'un agent, menant à la création d'agents du même type, pour permettre la généralisation de ses sous-parties de manière individuelle, tout en lui laissant un contrôle sur la situation (par exemple, la division des sections de routes en sous-section, voir section **I.3.2.2 Applications dans le cadre du projet AGENT**, p. 21) ;
- la création d'agents-points composant la géométrie d'un objet, agents-points pouvant interagir avec d'autres objets de la carte (voir les sections **I.3.5 Le modèle GAEL**, p24 ; **IV.3.3 Interaction des itinéraires avec un objet déformable**, p101).
- l'émergence de cas problématiques dont la non-résolution prolongée, voire l'aggravation, peut amener à préférer une résolution à un niveau plus global (voir section **I.3.7 Conflits persistants**, p27);
- des objets dont la position relative à un autre est importante, ce qui nous amène à considérer l'ensemble de ces objets comme appartenant à un même espace ayant un référentiel propre (voir section **IV.3.2 Interaction des itinéraires avec des objets compacts**, p98).

Nous observons aussi des types d'environnements différents :

- l'environnement de base, cartésien 2D, où la position des agents est exprimée selon leurs coordonnées  $(x, y)$  ;
- des environnements relationnels, où les agents identifient leur voisinage à partir d'un graphe de relations, établi *a priori* à partir de l'environnement cartésien, en mesurant la proximité des agents entre eux ;
- des environnements en abscisse curviligne et distance au projeté permettant de positionner un objet particulier vis-à-vis d'un objet linéaire (voir section **IV.3.2 Interaction des itinéraires avec des objets compacts**, page 98) ;
- des environnements décrits de façon à tenir compte de la forme même de cet environnement et de points saillants (par exemple, intersections ou virages d'une route) pouvant servir de points de repère.

De la diversité de ces situations, nous pensons pouvoir extraire des cas suffisamment génériques. Avant de décrire les patterns que nous proposons, nous revenons sur le cadre formel dans lequel nous les définissons.

## V.2 Cadre formel

L'approche par *patterns* n'est intéressante que si elle permet au minimum une description univoque et opérationnelle des situations rencontrées, ce qui suppose de disposer d'un cadre formel suffisamment abstrait pour être appliqué à divers contextes. C'est en général UML qui assure cette fonction dans le cas des *patterns* de conception en programmation par objets. Ici, nous proposons d'utiliser une formalisation particulière des relations entre agents (et environnements), que nous illustrons graphiquement. Dans la continuité de l'approche de la conception du modèle DIOGEN, basé sur PADAWAN, nous nous appuyons pour cela sur les concepts introduits dans le modèle PADAWAN dont nous reprenons les notations. Notre objectif étant toutefois de nous abstraire des choix de modélisation et d'implémentation de l'architecture multi-niveaux, nous n'avons conservé de PADAWAN que la part la plus abstraite, susceptible de donner lieu à des réalisations différentes selon les contextes de déploiement visés. Nous revenons rapidement sur les aspects retenus du modèle, déjà exposé dans la section II.4.1 **Le modèle PADAWAN**, p51).

En particulier, la décomposition d'un système complexe multi-niveaux dans PADAWAN s'appuient sur deux sortes d'éléments : les agents  $A$  bien sûr mais également les environnements  $E$ . Ces derniers peuvent désigner indifféremment des groupes sociaux ou des portions de l'espace physique, et peuvent eux-mêmes être décrits par des *patterns* (Mathieu et Picault, 2014). Nous précisons, par souci de s'abstraire de la diversité des implémentations, que nous désignons par environnement un espace doté d'une métrique, dans lequel les agents peuvent se percevoir et interagir. Deux relations fondamentales sont définies entre les agents et les environnements. D'une part, la situation d'un agent  $a$  dans un environnement  $e$  exprime le fait que l'agent  $a$  peut percevoir, être perçu, agir ou subir des actions dans  $e$ . Un agent peut être situé dans un nombre quelconque d'environnements. D'autre part, l'encapsulation d'un environnement  $e$  par un agent  $a$  signifie que l'agent  $a$  « contient »  $e$  : il est notamment capable de percevoir tous les agents qui y sont situés et en est l'hôte.

À partir de ces deux relations nous définissons la relation d'hébergement qui permet de s'affranchir du choix fait dans PADAWAN de multiplier les environnements. Un agent  $a_1$  est hébergé par un agent  $a_2$ , (où  $a_2$  est l'hôte de  $a_1$ ), si et seulement si : il existe un environnement  $e$ , tel que  $a_2$  encapsule  $e$  et  $e$  héberge  $a_1$ . Les relations d'hébergement (avec la situation et l'encapsulation sous-jacentes) sont représentées par la Figure 62.

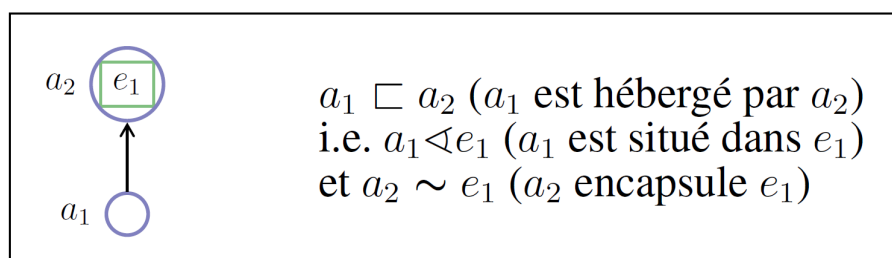


Figure 62 Représentation graphique de la relation d'hébergement.



Une autre hypothèse de travail liée au multi-niveaux en général est que les comportements d'un agent dépendent de sa situation, autrement dit un agent n'agira pas de la même façon selon qu'il est hébergé par  $a_1$  ou par  $a_2$ . Dans PADAWAN, cela se traduit par le fait que chaque environnement contient une matrice d'interaction au sens de l'approche IODA qui définit qui peut faire quoi avec qui. Ici, il suffit de supposer plus généralement que chaque hôte doit spécifier les comportements des agents qu'il héberge. Par ailleurs, ces comportements peuvent s'appuyer sur un certain nombre de primitives, certaines étant directement destinées à assurer des transformations cohérentes des relations d'hébergement. Par exemple dans la suite nous utiliserons *put*( $a, b$ ) pour indiquer que l'agent  $a$  est désormais hébergé par  $b$  ainsi que *remove*( $a, b$ ) pour supprimer la relation d'hébergement de  $a$  par  $b$ ; *del*( $a$ ) pour désigner la destruction d'un agent ; et enfin *merge*( $a, b$ ) pour noter la fusion de deux agents. Ces primitives, qui ont une définition univoque dans PADAWAN, ont évidemment vocation à être implémentées de façon adéquate dans chaque modèle multi-niveaux où l'on souhaite utiliser les *patterns* que nous avons identifiés.

Outre ces relations, nous notons dans la suite comme indiqué sur la Figure 63, deux cas utiles pour la description des *patterns* : d'une part, un lien d'hébergement optionnel, qui indique que l'agent concerné peut ou non être hébergé par l'autre sans affecter la définition du *pattern* ; d'autre part, un lien de dépendance qui exprime le fait que l'existence de l'agent  $a_1$  est conditionnée à celle de l'agent  $a_2$  (autrement dit, la suppression de  $a_2$  entraîne par défaut celle de  $a_1$ ).

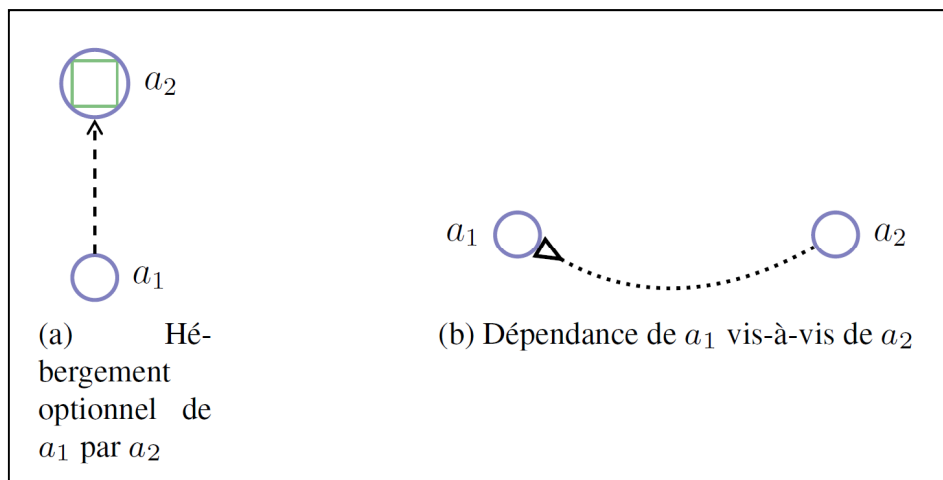


Figure 63 Notation des liens d'hébergement optionnel et de dépendance entre agents.

### V.3 Présentation des *patterns* identifiés

L'identification de *patterns* suppose de délimiter des situations univoques, caractérisées par une certaine structuration et d'éventuels effets sur le comportement. Pour chacun des *patterns* proposés, nous décrivons la situation de notre cas d'application qui en est à l'origine, une formalisation et d'autres situations où il peut s'appliquer.

### V.3.1 Pattern Agrégation

**Origine** : Lors de la généralisation automatisée, nous avons besoin, en plus des objets géographiques de base, d'objets qui en sont déduits. Par exemple les villes, qui sont construites par enrichissement des données initiales, par agrégation spatiale de bâtiments proches les uns des autres. Ces agents villes sont appelés à interagir avec les agents qu'ils hébergent : bâtiments, routes, etc...

Une autre situation est l'identification de groupes constitués d'agents ne réussissant pas à satisfaire leurs contraintes. Duchêne et Touya (2010) [27] proposent deux méthodes pour identifier les conflits résiduels et créer des entités intermédiaires ayant en charge la résolution de ces conflits.

De tels cas nous amènent à considérer le fait qu'un agent construit à partir d'autres agents, peut inclure plusieurs agents différents de ceux qui ont permis sa construction. Nous proposons donc de formaliser cet aspect en introduisant deux fonctions agissant sur un ensemble d'agents :

- une fonction **build** :  $A_x \rightarrow a_{gg}$  qui construit un agent agrégé  $a_{gg}$  à partir d'un ensemble d'agents  $A_x$  ;
- une fonction **populate** :  $A_x \rightarrow a_{gg}$  qui sélectionne les agents qui seront hébergés par l'agent ainsi créé.

**Formalisation** :

*compose* ( $A_x, build, populate$ ):

- Prérequis
  - Tous les agents de  $A_x$  sont hébergés par un même agent.
- Caractérisation :
  - $a_{gg} \leftarrow build(populate(A_x))$

La Figure 64 expose la représentation graphique du pattern Agrégation.

**Cas d'application** : cette situation évoque des cas fréquemment rencontrés lors de la simulation à savoir la réification de phénomènes émergents et la création de clusters. Ainsi la fonction **build** peut être construite à partir de la fonction d'émergence proposée par Camus et al. (2012) [13] ou de la fonction de clusterisation proposée par Caillou et Gil-Quijano (2012) [12]. Plus généralement, ce pattern permet de réifier par un agent toute structure émergente, tout groupe dans lesquels des agents estiment pouvoir appartenir. Il est à noter que c'est autour de cette problématique de création de niveaux d'abstraction supérieurs que se sont développés les premiers travaux sur la simulation multi-niveaux, à travers le projet RIVAGE (Servat et al., 1998) [101] en particulier. Depuis, tout un pan de recherche porte sur la façon d'écrire des fonctions **build** et **populate** de façon générique.

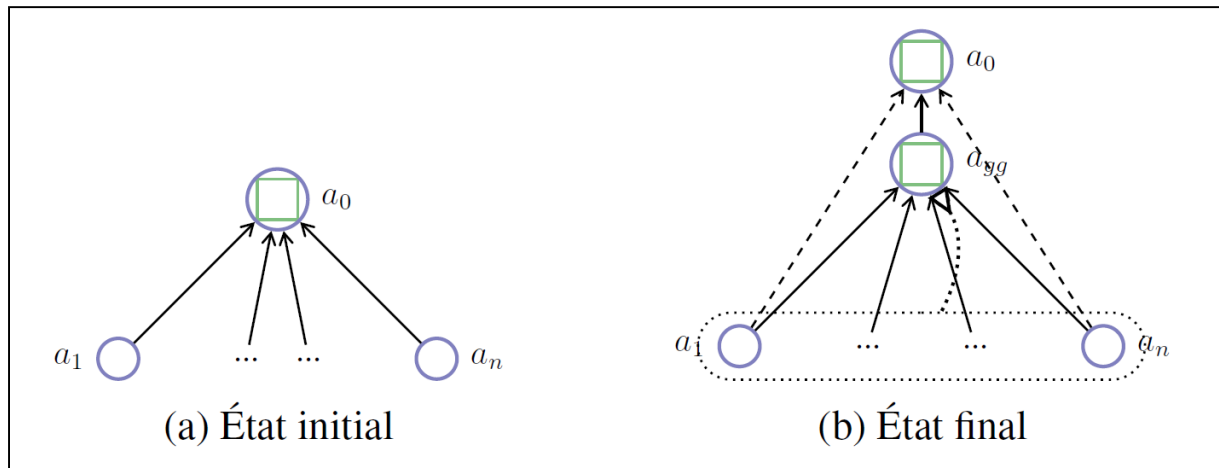


Figure 64 Représentation graphique du pattern Agrégation. L'état initial montre un agent qui encapsule plusieurs autres. L'état final montre la même situation, avec en plus un agent agrégé issu de l'agrégation de plusieurs agents de l'agent initial, qui héberge ce nouvel agent.

### V.3.2 Pattern Décomposition

**Origine** : Dans le modèle GAEL, les objets géographiques considérés comme déformables (e.g. courbes de niveaux) ont leur géométrie décomposée en agents points. Dans notre adaptation au sein de DIOGEN (voir section III.3.3 **Intégration d'éléments du modèle GAEL**, p72), ces points sont formalisés comme des agents, et sont situés dans un environnement appelé environnement déformable, tout comme les agents points. Ces derniers interagissent de manière à permettre des déformations locales des objets, tout en gardant au mieux leur forme.

De même, dans le cadre de la généralisation du réseau routier, un tronçon de route peut être divisé en sous-tronçons qui seront plus faciles à généraliser individuellement. Néanmoins, afin de maintenir la cohérence de l'ensemble, il est nécessaire que les agents issus de cette subdivision restent liés ensemble. L'agent initial encapsule alors les agents créés, et coordonne leurs actions.

De ces situations, nous constatons que, d'une manière symétrique à l'agrégation, nous pouvons proposer une fonction **build** :  $A_x \rightarrow a_{agg}$ , pour construire les agents du niveau inférieur. Notons que dans ce cas il n'est pas nécessaire de proposer une fonction **populate** séparée.

**Formalisation** :

*decompose* ( $a, build$ )

- Prérequis : Aucun
- Caractérisation :
  - o  $A_x \leftarrow build(a)$  pour tout  $ai \in A_x$  : **put**(ai; a)

La Figure 65 expose la représentation graphique du pattern Décomposition.

**Cas d'application** : les cas de décomposition s'effectuent lorsqu'une entité de niveau supérieur doit laisser la place à une entité de niveau moindre. Cela se produit par exemple dans (Vo et al. 2012) [117], lorsqu'un agent est libéré, ou dans (Camus et al. 2012) [13], lorsqu'une fonction d'immersion est appelée.

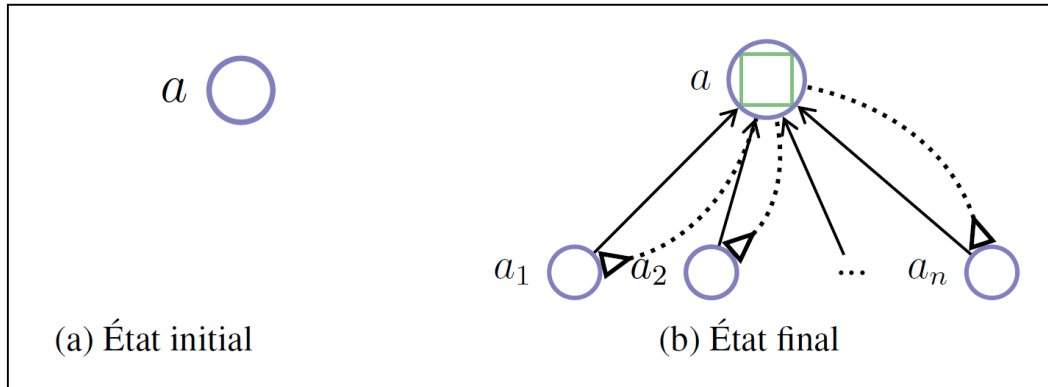


Figure 65 Représentation graphique du pattern Décomposition. À partir d'un unique agent initial sont créés plusieurs agents représentant certaines de ses parties, qui se retrouvent hébergés dans un nouvel environnement qu'il encapsule.

### V.3.3 Pattern Hiérarchisation

**Origine** : Le cas de la hiérarchisation intervient lorsqu'on veut établir une relation hiérarchique entre deux agents. En généralisation, nous pouvons être amenés à définir ce genre de situation lorsqu'un objet semble clairement situé sur un autre objet (e.g. un point d'intérêt comme une table d'orientation indiquée sur le parcours d'un itinéraire décrit sur la carte). Il est important que, suite à la modification de l'objet support, la relation soit préservée, afin que l'information reste cohérente.

**Formalisation** :

*hierarchise* ( $a_1, a_2$ )

- Prérequis :
  - o  $a_1$  et  $a_2$  sont hébergés pas un même agent.
- Caractérisation :
  - o  $put(a_2, a_1)$

La Figure 66 expose la représentation graphique du pattern Hiérarchisation.

**Cas d'application** : la hiérarchisation s'applique souvent dans les cas les plus simples où un agent entre dans un autre agent qui se déplace dans le même environnement, comme par exemple un individu qui entre dans un véhicule.

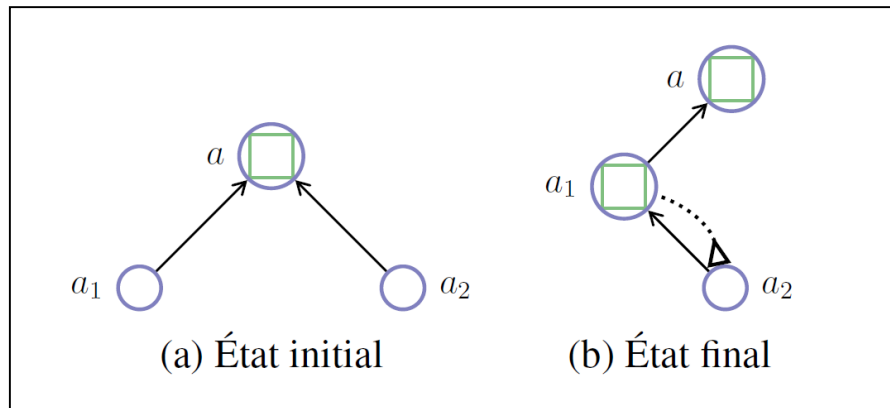


Figure 66 Représentation graphique du pattern Hiérarchisation. Dans l'état initial, on part d'une relation d'hébergement impliquant deux agents. Après avoir identifié une relation hiérarchique possible entre deux des agents hébergés, un environnement est créé et encapsulé par l'agent en position de supériorité, et pour que ce dernier héberge l'agent en situation d'infériorité.

### V.3.4 Pattern Frontière

**Origine :** Dans le cadre de la généralisation, deux îlots urbains adjacents séparés par une route doivent être fusionnés lorsque cette route séparatrice disparaît pour une raison tierce, comme la volonté de diminuer la densité du réseau routier dans une ville. Le résultat de cette fusion, un nouvel îlot, doit héberger l'ensemble des bâtiments qui étaient hébergés dans les îlots initiaux. Dans Atrash (2011) [1], cette situation a pu être intégrée par l'introduction de la notion de frontière. Nous parlons de frontière lorsque deux environnements sont séparés par un objet appartenant aux deux environnements. Le rôle de la frontière est d'interdire le passage d'agents d'un environnement à l'autre.

**Formalisation :**

*frontiere* ( $f, \{a_1, a_2\}$ )

- Prérequis :
  - o  $f$  héberge  $a_1$  et  $f$  héberge  $a_2$
  - o  $a_1$  et  $a_2$  sont des agents « compatibles » (on peut les « fusionner »).
- Caractérisation : **del** ( $f$ ) → **merge** ( $a_1, a_2$ )

La Figure 67 expose la représentation graphique du pattern Frontière.

**Cas d'application :** l'agent frontière peut être utilisé dans les situations où des environnements sont séparés par un élément dont le dynamisme (qui peut, comme dans notre exemple, se limiter à sa disparition) justifie sa modélisation sous forme d'agent. Nous pouvons nous interroger ici sur le fait que, dans les exemples proposés, la frontière est hermétique. Or, certaines situations impliquent une certaine porosité, comme la modélisation de la paroi d'une cellule, ou la modélisation d'un sas. Pour tenir compte de ce genre de situation, nous proposons un autre pattern spécifique : porte.

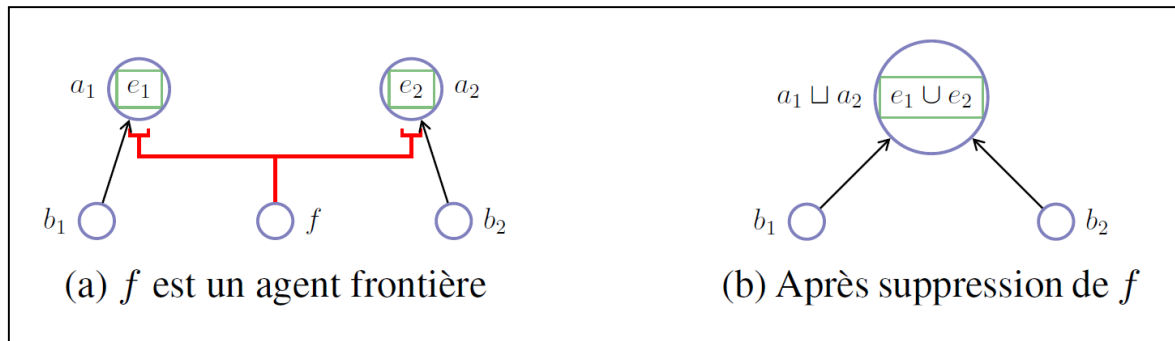


Figure 67 Représentation graphique du pattern frontière. L'agent frontière, lorsqu'il est supprimé, réunit les deux environnements qu'il séparait en un seul, et il fusionne les agents qui les encapsulaient.

### V.3.5 Pattern Porte

**Origine :** Nous ne sommes pas confrontés à cette situation dans le cadre de la généralisation, mais la caractérisation du pattern frontière nous amène à identifier un pattern porte.

**Formalisation :**

**porte** ( $p, \{ a_1, a_2 \}$ ) :

- Prérequis :
  - o  $p$  héberge  $a_1$  et  $p$  héberge  $a_2$
- Caractérisation : Le pattern porte permet de définir une primitive **cross**( $a, p$ ) qui permet à un agent  $a$  situé dans  $a_1$  (respectivement  $a_2$ ) de demander à l'agent porte  $p$  de le déplacer vers  $a_2$  (respectivement  $a_1$ ). Cette primitive peut être assujettie à des conditions propres à la porte et aux agents hôtes concernés. La suppression d'une porte n'a pas d'effets sur les hôtes.

La Figure 68 expose la représentation graphique du pattern Frontière.

**Cas d'application :** un exemple d'application du modèle PADAWAN, la pompe sodium-potassium, peut être modélisé selon ce *pattern* porte. La simulation d'un système impliquant un sas est un autre exemple.

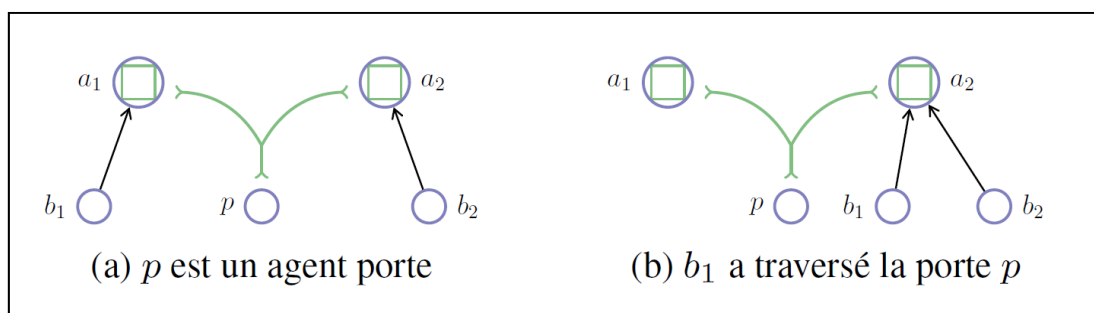


Figure 68 Représentation graphique du pattern Porte. La porte permet de contrôler de façon explicite le déplacement d'agents d'un environnement à un autre.

### V.3.6 *Pattern* Embarquement

**Origine** : En généralisation, lorsque nous modifions la géométrie ou le symbole d'un objet, il est souhaité que les relations que cet objet possède par ailleurs avec d'autres objets soient préservées. Par exemple dans le cas d'étude de déplacement des impasses (voir section III.4.2 **Définition d'un espace embarqué pour le voisinage**, p75), une impasse peut glisser le long de la route la supportant afin de laisser de la place aux bâtiments. Si cela se produit les bâtiments dans son voisinage doivent être déplacés afin de garder leur position relative par rapport à l'impasse. De même pour le cas des objets compacts à proximité de routes portant des itinéraires (voir IV.3.2 **Interaction des itinéraires avec des objets compacts**, p98), les objets à proximité d'une route doivent garder une position relative identique lorsque la route change de symbole, par exemple lorsque le symbole d'un itinéraire change de position. Pour résoudre cette situation dans le cadre de PADAWAN, nous nous appuyons sur la notion d'« environnement embarqué ». Lorsqu'un agent possède, dans son voisinage, un ensemble d'objets dont la position relative par rapport à lui est importante, un environnement embarqué est créé, ainsi qu'un agent encapsulant cet environnement. Cet environnement embarque à la fois l'agent à l'origine de sa création, et les agents de son voisinage.

L'environnement embarqué définit son propre référentiel, qui s'appuie sur l'agent à l'origine de sa création, et dans lequel les autres agents ont des coordonnées relatives. Ainsi, si l'agent à l'origine de l'embarquement se voit modifié, la relation entre le référentiel de l'environnement embarqué et le référentiel de l'environnement dans lequel il se situait initialement change aussi. Par conséquent, comme les agents doivent préserver leurs positions relatives dans l'environnement embarqué, ils devront modifier leur position dans l'environnement où ils se situaient auparavant. Dans le cadre de nos exemples, un référentiel simple consiste à utiliser des coordonnées composées de l'abscisse curviligne du projeté du centroïde de l'agent embarqué, et de sa distance à l'objet linéaire (route) à l'origine de l'embarquement. Lors de modifications de l'objet linéaire, le référentiel local est modifié par rapport au référentiel cartésien de la carte, ce qui implique une modification de la position des objets de l'environnement embarqué dans le référentiel de la carte.

**Formalisation** :

*embarque* ( $a^*$ ,  $A_x$ ) :

- Prérequis :  $a^*$  identifié comme jouant un rôle clef pour le référencement spatial dans son voisinage  $a^*$ ,  $A_x$ .
- Caractérisation :
  - o Création de l'environnement embarqué  $e_{emb}$  et de l'agent hôtebergeant : c'est un cas particulier d'agrégation, où l'agent à l'origine de l'embarquement a une relation de dépendance avec l'agent agrégat (encapsulant l'environnement embarqué) créé.
  - o modification de  $a^*$  → mise à jour de la position des agents  $a$  de l'ensemble  $A_x$ .

- modification de la position dans l'environnement initial d'un agent  $a \rightarrow$  mise à jour des coordonnées de  $a$ .

La Figure 69 illustre le *pattern* Embarquement.

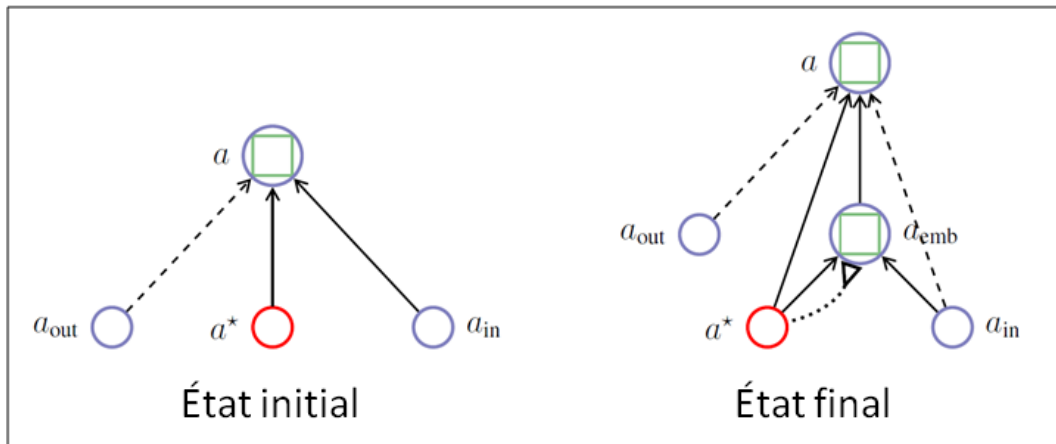


Figure 69 Représentation graphique du pattern Embarquement. D'un état initial, où un agent,  $a^*$ , s'avère jouer un rôle significatif pour une partie des agents ( $a_{in}$ ) situés dans son environnement, un agent  $a_{emb}$  est créé, encapsulant un environnement qui hébergera  $a^*$ , ainsi que les agents  $a_{in}$ .  $A^*$  joue un rôle important dans l'environnement encapsulé par  $a_{emb}$ , car il sert de base au référentiel dans lequel les agents  $a_{in}$  ont leur coordonnées.

**Cas d'application :** Plusieurs situations de simulation peuvent être modélisées selon la notion d'agent embarqué. Par exemple, une simulation de déplacement d'astres peut modéliser leurs champs de gravitation. Dans une telle modélisation, la Terre définirait un environnement embarqué dans lequel se situerait la Lune. Le déplacement de la Terre autour du Soleil impliquerait une modification de la position de la Lune dans le référentiel héliocentrique.

### V.3.7 Patterns Meso

Les interactions hiérarchiques, c'est-à-dire impliquant un hôte et ses hébergés, sont particulières, dans la mesure où elles peuvent impliquer des prises de décisions fortes d'un niveau supérieur pour les agents hébergés.

**Origines :** Dans le modèle AGENT, un meso désigne un agent qui est composé d'autres agents (ces derniers pouvant être soit micro, c'est-à-dire des agents existant initialement, soit meso eux aussi). Les meso ont plusieurs rôles vis-à-vis des autres agents. Plusieurs de ces rôles sont décrits dans Ruas (2000) [94] : le Législateur, le Contrôleur et le Coordinateur. Le Coordinateur décrit une situation particulière qui fait l'objet d'un pattern Ordonnanceur décrit à part.

Le Législateur agit directement sur les connaissances des agents qu'il héberge, ainsi que leurs motivations. Dans le cadre de la généralisation, cela se traduit par le fait que le meso peut modifier directement les contraintes de ses composants, et ainsi influencer leur comportement. Le Contrôleur assure que les actions effectuées par les agents vont dans le sens de ses intérêts (dans le cadre de la généralisation, dans le sens de la satisfaction de ses con-



traintes, mais aussi de celle de ses composants). Il peut décider d'annuler une action effectuée par un de ses agents hébergés. Pour Jabeur et al. (2006) [52], les agents hébergés sont amenés à interagir eux-mêmes pour se généraliser, mais peuvent parfois s'en remettre à un agent hôte pour décider d'une action qu'ils ne peuvent résoudre eux même. Nous proposons la dénomination d'Arbitre pour ce cas de figure. Pour Renard (2012) [87], les agents bâtiments appartenant à un alignement de bâtiments peuvent décider de s'en remettre à l'agent alignement pour effectuer une action qui sera mieux effectuée par ce dernier. Nous proposons ici le terme de Délégitaire pour décrire ce rôle de meso.

**Cas d'application** : Dans le cadre d'une simulation avec un groupe hiérarchisé, comme un orchestre où le chef choisit l'ordre d'activation des membres du groupe, nous pouvons modéliser des situations impliquant certains des rôles du meso.

### V.3.8 *Pattern* Ordonnanceur

**Origine** : Dans le modèle AGENT, les agents meso ont la possibilité d'activer leurs composants, afin qu'ils se généralisent eux-mêmes. Cette activation s'effectue lors du cycle de vie de l'agent meso. Ce rôle d'activation appelé Coordinateur par Renard (2012) [87], permet d'identifier le meilleur moment pour la généralisation individuelle des bâtiments. Les connaissances qu'ont les îlots de l'intégralité de ces composants en font les agents les plus appropriés pour identifier l'ordre d'activation le plus opportun.

**Cas d'application** : Dans Gaud et al. (2008) [40], une approche pour la simulation propose de donner aux holons un rôle équivalent à celui de Coordinateur.

## V.4 Discussion autour des *patterns*

Les *patterns* proposés sont des premières propositions dans notre démarche. Ces *patterns* permettent l'analyse et la comparaison de situations récurrentes. Leur intérêt réside aussi dans la possibilité de les combiner, et dans leur utilisation lors de la conception de modèle. Par exemple, plusieurs situations particulières propres à la généralisation cartographique peuvent être reproduites à partir d'une combinaison de ces *patterns*. Ainsi, la création d'îlots urbains, et leur relation avec les bâtiments qu'ils hébergent, peut s'exprimer à l'aide des trois *patterns* de création de niveau utilisés séquentiellement, comme montré dans la Figure 70. Ensuite, à partir des agents îlots ainsi obtenus, des relations hiérarchiques spécifiques sont appliquées : ce sont les relations de Contrôleur et de Législateur décrites lors de la définition du *pattern* meso. Ainsi les *patterns* de création de niveau peuvent s'accompagner d'autres *patterns* spécifiques. L'utilisation conjointe de *patterns* est aussi pertinente en dehors de la généralisation. Par exemple, la modélisation d'une membrane cellulaire nécessite l'utilisation conjointe des *patterns* frontière et porte.

Outre qu'ils peuvent être utilisés conjointement, nous constatons aussi que les *patterns* proposés peuvent présenter des liens. Ainsi le *pattern* Agrégation et le *pattern* Embarquement présentent des similitudes. Néanmoins, les spécificités propres à l'agent embarqué, et

la définition de comportement de maintien de la position relative qui motive sa création justifie sa définition de manière indépendante.

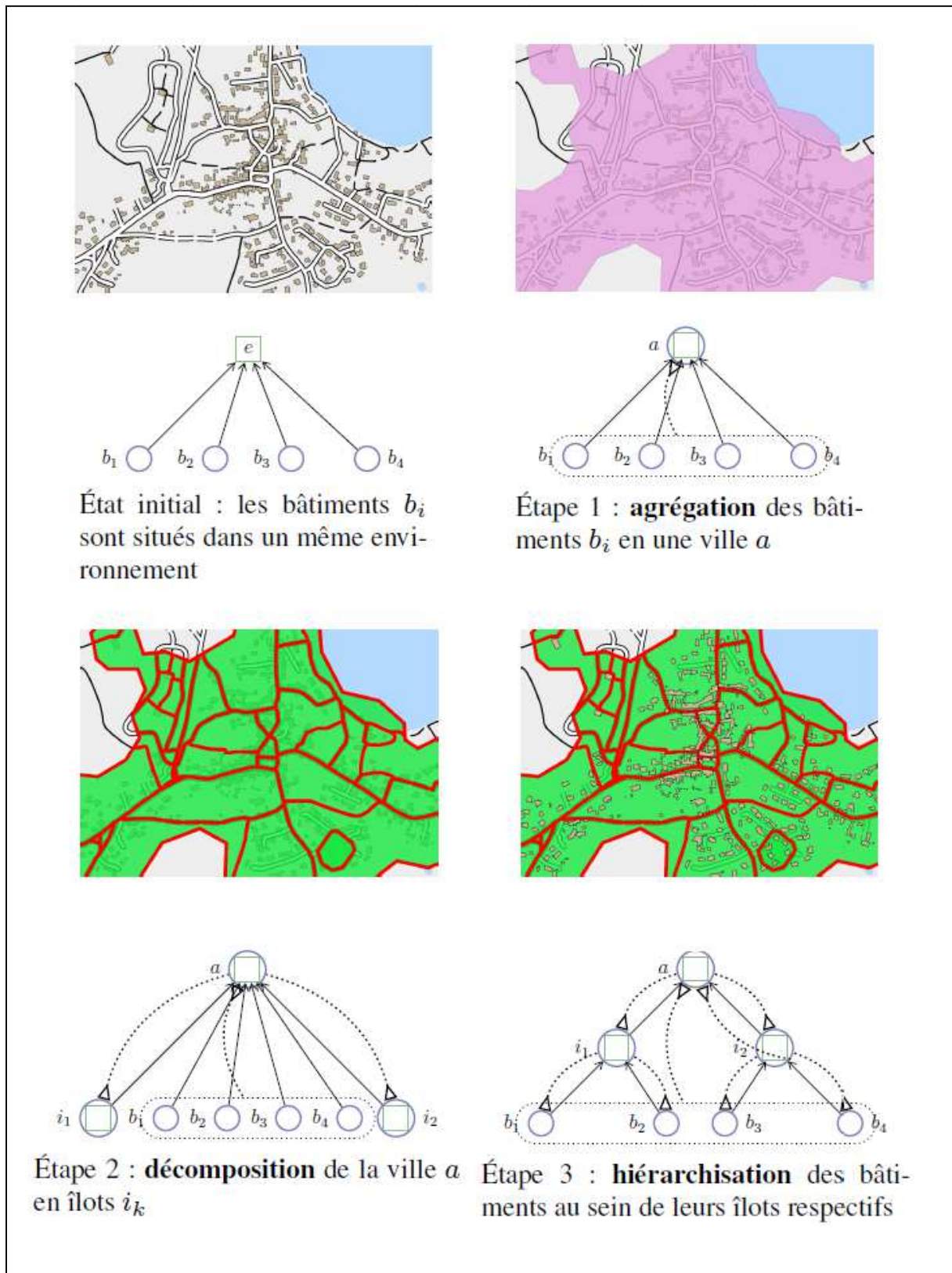


Figure 70 Explication de la construction d'îlots urbains proposée par Boffet (2002) [8] à l'aide d'une séquence de patterns.

En termes d'exhaustivité, les *patterns* proposés sont issus des recherches effectuées dans le cadre de la généralisation s'appuyant sur des modèles orientés agents. Une des contraintes de la généralisation étant de préserver relativement la position des objets sur la carte, le procédé n'implique pas une grande mobilité des agents. Il est donc très probable que les *patterns* pouvant être extraits des SMA pour la généralisation cartographique ne constituent pas une liste exhaustive, et que de nombreux *patterns* multi-niveaux intéressants restent à être identifiés. En revanche, en termes de généralité nous avons cherché à nous affranchir autant que possible du modèle d'origine dans lequel les *patterns* proposés ont été mis en œuvre, afin de ne retenir que les caractéristiques les plus générales d'une décomposition multi-niveaux.

## V.5 Synthèse du chapitre

Nous avons proposé un début de formalisation de *patterns* multi-niveaux pour des SMA. Ces *patterns*, identifiés à partir de cas particulier rencontrés lors de la mise en commun de modèles pour la généralisation cartographique, portent sur trois aspects spécifiques de relations multi-niveaux : la création de niveaux différents, des aspects structurels de la relation entre niveaux et les relations spécifiques d'un agent avec un agent d'un niveau supérieur dans lequel il est hébergé. Ces *patterns* d'analyse ont, à moyen terme, vocation à être étendus pour servir également à la conception de systèmes multi-agents multi-niveaux, tant dans le cadre de la résolution de problèmes que dans celui de la simulation. Plusieurs pistes peuvent être envisagées pour des cas d'utilisations de tels *patterns* :

- dans le cadre d'un prétraitement d'une situation à modéliser : comme dans de nombreuses situations de généralisation, les relations multi-niveaux peuvent être établies préalablement à l'exécution de système. Les solutions les plus efficaces et éprouvées peuvent être automatisées pour de telles utilisations ;
- pour la spécification de transformations des structures hiérarchiques. Dans un système qui évolue, les relations hiérarchiques peuvent apparaître, disparaître, se fusionner ... Des réponses génériques appropriées peuvent être alors proposées ;
- en identifiant des comportements particuliers lors de certaines opérations influentes sur les structures multi-niveaux du problème (comme le déplacement d'agents situés, en conséquence de la création d'un environnement).

La liste des *patterns* proposés n'est pas exhaustive et gagnerait à être enrichie de nouvelles situations, à identifier à partir d'autres SMA multi-niveaux. Pour Mathieu et al. (2016) [80] quatre types de situations multi-niveaux ont été identifiées selon la nature de la relation hiérarchiques entre agents (Vue, Zoom, Cohabitation, Marionnettiste). Le cas de la Cohabitation, où plusieurs agents de niveaux différents interagissent, est le cadre d'identification des *patterns* que nous avons décrits ici. En effet, le caractère autonome de tous les agents, et la possibilité d'interagir entre niveaux différents dans cette situation induit une grande complexité de situations et de possibilités de modélisation. Ceci incite à identifier des cas et des comportements génériques et récurrents afin d'y apporter des solutions éprouvées.

# Conclusion et perspectives

---

Nous avons, dans ce mémoire, étudié la généralisation automatique de carte, et notamment l'utilisation des approches orientées agents pour l'effectuer. Nous avons pour ce faire focalisé nos efforts sur les aspects multi-niveaux des modèles pour la généralisation.

## Rappel des objectifs

Les objectifs de notre travail de thèse étaient d'exprimer de manière générique les caractéristiques des modèles SMA pour la généralisation cartographique, et tout particulièrement leurs aspects multi-niveaux et de voir quelles problématiques de généralisation automatisée nous pourrions résoudre. Nous avons identifié plusieurs verrous à cela que sont l'organisation des objets, l'orchestration des agents, le choix des actions à effectuer, les répercussions des modifications d'un contexte à un autre et l'identification de nouvelles situations et structures pour les résoudre.

## Résumé des contributions

Nous avons proposé les contributions suivantes :

- La fusion de plusieurs modèles pour la généralisation au sein d'un même paradigme. En nous appuyant sur le paradigme PADAWAN, nous avons proposé notre propre modèle, DIOGEN, intégrant les comportements des modèles multi-agents pour la généralisation.
- Afin de permettre cette adaptation, il a fallu tenir compte des spécificités propres à la résolution distribuée de contraintes par rapport à la simulation (comme l'inversion de la finalité) en introduisant les contraintes de façon explicite dans le modèle. Ce travail de formalisation de connaissances a consisté en l'introduction d'un modèle d'opinion de contraintes (favorable, indifférente, défavorable, opposée à une interaction), introduit dans DIOGEN, ainsi que la synthèse de ces opinions afin de choisir l'interaction à effectuer.
- DIOGEN a été implémenté dans la plate-forme Géoxygène. De plus, des comportements d'autres modèles ont aussi dû être intégrés dans cette plate-forme afin de permettre leur intégration dans DIOGEN, mais aussi la comparaison entre modèle, pour nous assurer que nous pouvions reproduire l'existant.
- Nous avons validé DIOGEN dans des situations de généralisation déjà résolues par les précédents modèles, puis nous avons proposé de nouvelles solutions pour des situations spécifiques de généralisation nécessitant d'emprunter des comportements à plusieurs modèles conjointement (parallélisation des bâtiments aux routes au sein d'un espace urbain) ou spécifiquement multi-niveaux avec des interactions parfois

- non hiérarchiques (généralisation des bâtiments adjacents et généralisation des zones environnant les impasses).
- Nous nous sommes ensuite intéressés à la généralisation de cartes comportant des données thématiques en plus des données topographiques : les cartes d'itinéraires de randonnées à la façon des plans de bus. Dans de telles cartes, le positionnement de symboles pour les itinéraires implique d'une part une décision quant à leur positionnement le long d'un tronçon de route, d'autre part une décision vis-à-vis des autres objets sur la carte, qui peuvent être occultés par le symbole des itinéraires. Dans ce cas, les autres objets peuvent être amenés à se déplacer, ou à influencer le positionnement du symbole de l'itinéraire. Pour positionner les symboles d'itinéraire de façon optimale, nous avons encadré un stage explorant une solution de type COP, basée sur un ensemble de contraintes exprimant la lisibilité des tracés d'itinéraires et sur une résolution exploitant le recuit-simulé. Pour gérer les conséquences sur les objets voisins des routes portant des itinéraires, nous avons construit dans DIOGEN un ensemble d'interactions permettant de négocier le déplacement des symboles d'itinéraire, ou d'agir sur les autres objets géographiques qu'ils soient compacts comme les bâtiments (déplacement groupé d'objets) ou déformables comme les cours d'eau (déformation locale d'un objet linéaire).
  - Enfin, nous avons cherché à rendre générique notre approche. À travers les divers cas d'application en généralisation cartographique que nous avons étudiés, nous avons pu identifier des situations récurrentes. Nous avons isolé plusieurs de ces situations, puis nous les avons exprimées de façon générique sous forme de patterns d'analyse. Pour chaque pattern identifié, nous avons exposé la ou les situations qui l'ont inspiré, une formalisation générique affranchie des spécificités propres au contexte cartographique, et ouvert sur des possibles applications dans d'autres situations.

La Figure 71 expose ces contributions et comment elles s'articulent entre elles.

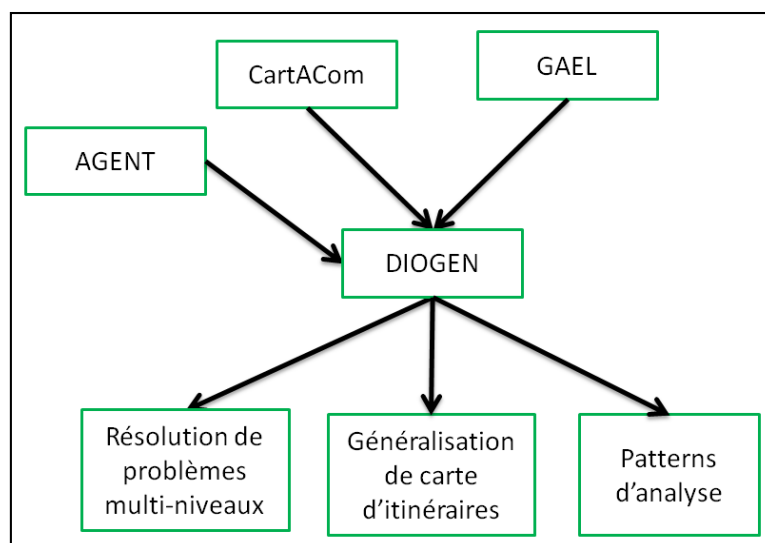


Figure 71 Résumé des contributions et de leur imbrication.

## Perspectives

Pour notre travail, nous avons identifié un ensemble de perspectives variées, tant pour les aspects techniques en matière d'application pour la généralisation, qu'en matière de cadre générique de résolution distribuée de contraintes, ou en matière de contribution dans la recherche en SMA.

### Perspectives immédiates

Afin de finaliser les réponses apportées aux problématiques de notre thèse, plusieurs points restent encore à traiter. Dans un premier temps, il est nécessaire, dans une optique de mise en œuvre conjointe des agents et d'une résolution plus automatisée de la carte, d'apporter des solutions en matière d'orchestration de l'activation des agents. Cette orchestration peut être centralisée, mais, certaines situations problématiques arrivent au cours de la généralisation, nécessitant un ordonnancement dynamique des agents. Plusieurs pistes existent pour permettre cet ordonnancement. Par exemple, des systèmes de vote, où les agents encourageront l'activation des agents les plus à même de résoudre leurs contraintes peuvent être envisagés. Une autre piste porte sur l'exploitation de méthodes multicritères, tenant compte de plusieurs aspects des agents et de leurs contraintes, aspects pouvant être définis initialement lors de la spécification, ou bien émergeant au cours de la résolution (par exemple, passé un certains temps sans avoir été activé, un agent non satisfait devrait pouvoir être prioritaire).

Nous avons effectué plusieurs tests sur plusieurs zones. Néanmoins, certaines situations limites n'ont pas été abordées (par exemple, des routes avec de nombreux virages). Effectuer des tests intensifs sur un ensemble plus large et plus diversifié de zones permettra de valoriser nos propositions.

Un point important de la généralisation est la possibilité qui doit être donnée au système de pouvoir revenir à une situation antérieure, partiellement ou complètement. Si ces retours arrière sont possibles dans des modèles arborescents comme AGENT, et que nous les avons retranscrits tant qu'on en reste à des modélisations arborescentes, il est envisageable de le faire de manière générale. Une sauvegarde des états successifs n'est par contre pas envisageable, en raison de leur trop grand nombre, d'où la nécessité d'envisager une solution intermédiaire, s'appuyant sur des mécaniques collectives des agents (par exemple, la construction d'une coalition qui sauvegarderait des situations ponctuellement d'un point de vue de tous ses membres).

Autre point, nous avons intégré des éléments issus des modèles AGENT, CartACom et GAEL, mais nous avons validé la reproduction des résultats des deux premiers modèles seulement. La validation de la reproduction du comportement de GAEL reste à effectuer. De plus, d'autres modèles multi-agents existent (nous les avons présenté en section **I.3 Présentation des différents modèles**, p19), et l'intégration de leurs comportements est un plus intéressant.

## **Perspectives cartographiques**

L'essor des données utilisateur dans le cadre de la géomatique amène à vouloir des solutions de représentation de ces données sous formes de cartes. Dans la mesure où les données utilisateur s'appuient pour partie d'entre elles sur des données topographiques de base, des relations hiérarchiques peuvent être identifiées et explicitées au niveau de la saisie des données thématiques (e.g. accidents routiers, données touristiques), comme proposé notamment par Jaara (2015). Les interactions et contraintes proposées pour la généralisation des cartes d'itinéraires, sont utilisables dans ce genre de situation. De plus, d'autres interactions et comportements variés peuvent être conçus et développés pour gérer d'autres aspects : par exemple, des interactions permettant une coalition d'un ensemble de points afin de résoudre des problèmes récurrents et localisés. Pour cela, de nouvelles contraintes peuvent être définies et intégrées dans DIOGEN, premièrement en développant une méthode d'évaluation de la satisfaction de contrainte, et ensuite en exprimant le positionnement de ces contraintes par rapport aux interactions de déplacement et de déformations.

L'utilisateur, en plus de spécifier les données en entrée et le résultat attendu sous forme de contraintes, peut en fonction de son expertise, vouloir intervenir dans le paramétrage des connaissances qui guident le système (par exemple, le lien entre les contraintes et les interactions). Il serait intéressant de proposer un système d'aide au paramétrage assistant l'utilisateur dans ce type de démarche.

L'utilisation émergente de cartes multi-échelles interactives utilisées notamment sur de nouveaux supports (téléphone portable, tablette) amène à considérer de nouvelles contraintes afin de maintenir une cohérence entre les échelles. Par exemple, lors d'un zoom avant ou arrière sur une carte de randonnées avec itinéraires à la façon des plans de bus, le positionnement de l'itinéraire doit pouvoir rester fixe, indépendamment des autres objets pouvant entrer dans la cadre de la zone visualisée. L'introduction de contraintes de persistance de certains choix aidera à résoudre ce genre de problématique : des choix fait sur une portion réduite de l'espace doivent se maintenir lorsque l'utilisateur observe une partie plus large de l'espace. Ces contraintes auront donc une opinion défavorable vis-à-vis des interactions de déplacement d'itinéraires.

## **Perspectives en résolution de problèmes contraints**

Les perspectives de notre travail en résolution de problèmes contraints portent sur la généralité de notre approche. La séparation entre déclaratif et procédural est une démarche pouvant être appliquée à d'autres situations. La spécificité de DIOGEN, l'explicitation des contraintes dans le modèle, peut probablement être réutilisée pour d'autres problématiques de satisfaction de contraintes. De plus, certains choix faits pour notre modèle peuvent être rendus plus génériques. Par exemple, la manière dont nous agrégeons les différents avis des différentes contraintes pour une même interaction, utilisant une formule empirique tenant compte de plusieurs critères propres à la généralisation (importance d'une contrainte à être satisfaite, priorité d'une contrainte ...), pourrait être rendu générique, et permettre

l'utilisation, par exemple, de méthodes multicritères ou de mesure de bien-être, qui ont déjà montré leur intérêt dans des contextes différents (Touya, 2001) [113].

### Perspectives dans le domaine des SMA

La contribution à la recherche multi-agents par la conception de patterns d'analyse peut être étendue à l'identification d'autres patterns, notamment dans le cadre de modèles pour la simulation. De façon similaire au développement des patterns proposés, l'identification de ce genre de situation conduira à leur expression de façon générique, et à leur transposition dans d'autres situations. Pour identifier ces nouveaux patterns, une exploration de l'application concrète des différents modèles est une première base.

Une autre perspective est le développement de patterns de conception sur la base des patterns d'analyse que nous avons identifiés. Une piste pour le développement de tels patterns est d'une part la définition de situations récurrentes dans le cadre des modèles multi-agents (par exemple, du pattern Embarquement, nous pouvons proposer des méthodes d'identification d'agents pouvant embarquer d'autres), d'autre part à l'ajout automatique de certains comportements (par exemple, du pattern Frontière, nous pourrions établir une relation Frontière entre un agent et deux environnements, et systématiser la création d'un nouvel environnement et le déplacement des agents des environnements initiaux dans comme conséquence de la suppression de l'agent Frontière).

La Figure 72 expose schématiquement certaines de ces perspectives.

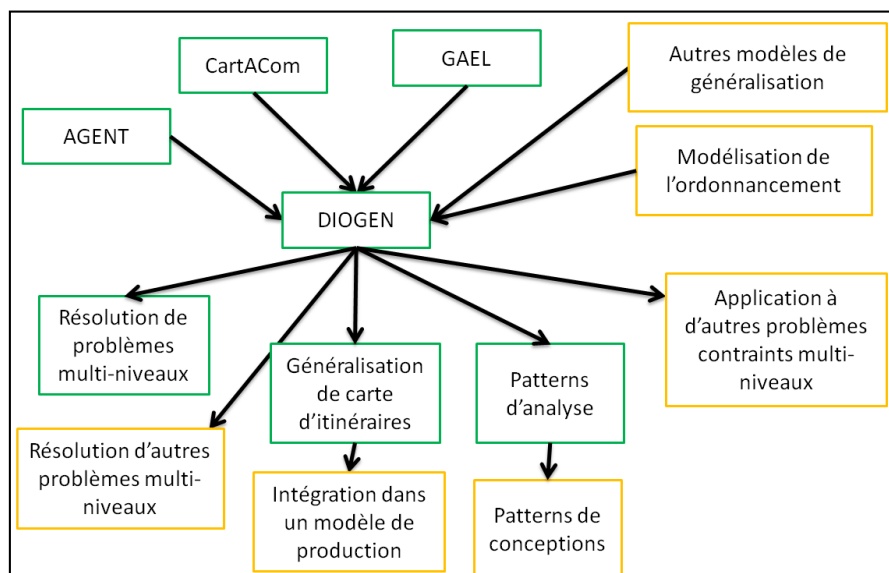


Figure 72 Résumé de certaines perspectives (encadrées en jaune) et de leur imbrication avec les contributions de la thèse (encadrées en vert).





# Bibliographie

---

- [1] A. Atrash. Adaptation of PADAWAN multi-agent model for the solution of a spatial problem: the cartographic generalization. Rapport de stage, Univ. Paris Sud, 2011.
- [2] M. Bader, M. Barrault, and R. Weibel. Building displacement over a ductile truss. *International Journal of Geographical Information Science*, 19(8-9):915–936, 2005.
- [3] C. Baeijs. *Fonctionnalité émergente dans une société d'agents autonomes : étude des aspects organisationnels dans les systèmes multi-agents réactifs*. PhD thesis, Grenoble, INPG, Grenoble, 1998. Th. : informatique.
- [4] C. Baeijs, Y. Demazeau, and L. Alvares. SIGMA: application of multi-agent systems to cartographic generalization. In *Agents Breaking Away, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996, Proceedings*, pages 163–176, 1996.
- [5] M. Barrault, N. Regnauld, C. Duchêne, K. Haire, C. Baeijs, Y. Demazeau, P. Hardy, W. Mackaness, A. Ruas, and R. Weibel. Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalization. In *20th International Cartographic Conference (ICC'01)*, volume 3, pages 2110–2116. International Cartographic Association, 2001.
- [6] K. M. Beard. Constraints on rule formation. In B. Buttenfield and R. McMaster, editors, *Map Generalization: Making Decisions for Knowledge Representation*, pages 121–135. Longmans, London, 1990.
- [7] J. Bertin. *Sémiologie graphique*. Gauthier Villars, 1967.
- [8] A. Boffet. Analyse multi-niveaux des espaces urbains. *Revue Internationale de Géomatique*, 12(2):215–260, 2002.
- [9] L. Breton and N. Jussien. Un csp comme comportement d'agent. application à la résolution d'équations en physique des milieux granulaires. *Journal Electronique d'Intelligence Artificielle*, 3:jedai–26, 2004.
- [10] D. Burghardt and S. Meier. Cartographic displacement using the snakes concept. *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, Birkhaeuser Verlag, pages 59–71, 1997.
- [11] D. Burghardt and M. Neun. Automated sequencing of generalisation services based on collaborative filtering. In *4th international conference GIScience*. Citeseer, 2006.

- [12] P. Caillou and J. Gil-Quijano. Description automatique de dynamiques de groupes dans des simulations à base d'agents. In P. Chevaillier and B. Mermet, editors, *Actes des 20e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2012)*, pages 23–32. Cépaduès, 2012.
- [13] B. Camus, J. Siebert, C. Bourjot, and V. Chevrier. Modélisation multi-niveaux dans AA4MM. In P. Chevaillier and B. Mermet, editors, *Actes des 20e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2012)*, pages 43–52. Cépaduès, 2012.
- [14] O. Chaudhry and W. A. Mackaness. Utilising partonomic information in the creation of hierarchical geographies. In *10th ICA Workshop on Generalisation and Multiple Representation. Moscow, Russia, 2007*.
- [15] V. Couturier, D. Telisson, and M.-P. Huget. Patterns d'analyse pour l'ingénierie des systèmes multi-agents. In M. Occello and L. Rejeb, editors, *Actes des 18e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2010)*, pages 55–64. Cépaduès, 2010.
- [16] D. David, D. Payet, and R. Courdier. Réification de zones urbaines émergents dans un modèle simulant l'évolution de la population à La Réunion. In E. Adam and J.-P. Sansonnet, editors, *Actes des 19e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2011)*, pages 63–72. Cépaduès, 2011.
- [17] A. DeLucia and R. Blacke. Comprehensive approach to automatic feature generalization. In *Proceedings of the 13th International Cartographic Conference, Morelia, Mexico, 1987*.
- [18] A. Drogoul, E. Amouroux, P. Caillou, B. Gaudou, A. Grignard, N. Marilleau, P. Taillandier, M. Vavasseur, D.-A. Vo, and J.-D. Zucker. Gama: A spatially explicit, multi-level, agent-based modeling and simulation platform. In *Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 271–274. Springer, 2013.
- [19] A. Drogoul and C. Dubreuil. Eco-Problem-Solving Model: Results of the N-Puzzle. 1992.
- [20] R. Duboz, É. Ramat, and P. Preux. Scale transfer modeling: Using emergent computation for coupling an ordinary differential equation system with a reactive agent model. *Syst. Anal. Model. Simul.*, 43(6):793–814, June 2003.
- [21] C. Duchêne. Making a map from “thematically multi-sourced data”: the potential of making inter-layers spatial relations explicit. In *Proceedings of 17th ICA Workshop on Generalisation and Multiple Representation*. ICA, 2014. Vienna, Austria, 22 sept. 2014, [http://generalisation.icaci.org/images/files/workshop/workshop2014/genemr2014\\_submission\\_14.pdf](http://generalisation.icaci.org/images/files/workshop/workshop2014/genemr2014_submission_14.pdf).
- [22] C. Duchêne, M. Barrault, and K. Haire. Road network generalization: A multi-agent system approach. In Chinese Society of Geodesy Photogrammetry and Cartography, editors,

---

20th International Cartographic Conference (ICC'01), 6-10 August , Beijing (China), volume 3, pages 2166--2177. International Cartographic Association, 2001.

[23] C. Duchêne and J. Gaffuri. Combining three multi-agent based generalisation models : Agent, cartacom and gael. In C. Gold A. Ruas, editor, *Headway in Spatial Data Handling : 13th International Symposium on Spatial Data Handling*, pages 277-296. Springer, jun 2008.

[24] C. Duchêne. *Automatisation de la généralisation cartographique - Relations et interactions, orchestration et approches multi-agents*. Habilitation à diriger des recherches, spécialité sciences et technologies de l'information géographique, Université Paris-Est, jun 2016.

[25] C. Duchêne. *Généralisation par agents communicants : Le modèle CARTACOM. Application aux données topographiques en zone rurale*. Thèse de doctorat en informatique, Université Paris 6, 2004.

[26] C. Duchêne, A. Ruas, and C. Cambier. The CARTACOM model: transforming cartographic features into communicating agents for cartographic generalisation. *International Journal of Geographical Information Science*, 26(9):1533-1562, 2012.

[27] C. Duchêne and G. Touya. Emergence de zones conflits dans deux modèles de généralisation cartographique multi-agents. In M. Occello and L. Rejeb, editors, *Actes des 18e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2010)*, pages 33-42. Cépaduès, 2010.

[28] M. S. El Hmam, H. Abouassïa, and D. Jolly. Contribution à la modélisation et la simulation du flux de trafic: approche hybride basée sur les systèmes multi-agents. *Actes INRETS*, pages 93-111, 2008.

[29] J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1999.

[30] J. Ferber and A. Gutknecht. Un méta-modèle organisationnel pour l'analyse, la conception et l'exécution de systèmes multi-agents. In *Proceedings of Third International Conference on Multi-Agent Systems ICMAS*, volume 98, pages 128-135, 1998.

[31] J. Ferber, F. Michel, and J. Báez. AGRE: Integrating environments with organizations. In D. Weyns et al., editors, *Environments for Multi-Agent Systems: 1st Int. Workshop (E4MAS'2004)*, volume 3374 of LNCS, pages 48-56. Springer, 2005.

[32] M. Fink and S. Pupyrev. Ordering metro lines by block crossings. In *International Symposium on Mathematical Foundations of Computer Science*, pages 397-408. Springer, 2013.

[33] J. Gaffuri. Field deformation in an agent-based generalisation model: the gael model. In *GI-days 2007 - young researches forum*, volume 30, pages 1-24, 2007.

- [34] J. Gaffuri. *Généralisation automatique pour la prise en compte de thèmes champs: le modèle GAEL*. Thèse de doctorat, spécialité sciences de l'information géographique, Université Paris-Est, 2008.
- [35] J. Gaffuri. *Généralisation automatique, cours*. École Nationale des Sciences Géographiques, IGN, 2008.
- [36] J. Gaffuri. Three reuse example of a generic deformation model in map generalisation. In *24th International Cartographic Conference*, 2009.
- [37] M. Galanda. *Automated Polygon Generalization in a Multi Agent System*. Ph.d. thesis, department of geography, University of Zurich, 2003.
- [38] M. Galanda and R. Weibel. Algorithmes de généralisation des données catégorielles. *Généralisation et représentation multiple*, 2002.
- [39] M. Galanda and R. Weibel. Using an energy minimization technique for polygon generalization. *Cartography and Geographic Information Science*, 30(3):263–279, 2003.
- [40] N. Gaud, S. Galland, V. Hilaire, and A. Koukam. An organisational platform for holonic and multiagent systems. In *International Workshop on Programming Multi-Agent Systems*, pages 104–119. Springer, 2008.
- [41] O. Gautrais and Kraus A. Cas d'utilisation des cartes de randonnées avec représentation à symboles multiples. Rapport de stage, ENSG, 2015.
- [42] C. C. Gibson, E. Ostrom, and T.-K. Ahn. The concept of scale and the human dimensions of global change: a survey. *Ecological economics*, 32(2):217–239, 2000.
- [43] J. Gil-Quijano, M. Piron, and A. Drogoul. Mechanisms of automated formation and evolution of social-groups: A multi-agent system to model the intra-urban mobilities of bogotá city. *Idea Group Inc*, pages 151–168, 2007.
- [44] E. Guilbert. Multi-level representation of terrain features on a contour map. *Geoinformatica*, 17(2):301–324, 2013.
- [45] L. Harrie and T. Sarjakoski. Simultaneous graphic generalization of vector data sets. *Geoinformatica*, 6(3):233–261, 2002.
- [46] A. Hassane, Y. Kubera, and M. Gildas. Modélisation hybride dynamique de flux de trafic. Technical report, December 2013. version française de <http://arxiv.org/abs/1401.6773>.
- [47] C. S. Holling. The functional response of invertebrate predators to prey density. *Memoirs of the Entomological Society of Canada*, 98(S48):5–86, 1966.
- [48] R. L. Hughes. A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, 36(6):507–535, 2002.

- 
- [49] T. Huraux, N. Sabouret, and Y. Haradji. A multi-level model for multi-agent based simulation. In *ICAART (2)*, pages 139–146, 2014.
- [50] K. Jaara. *Prise en compte des dépendances entre données thématiques utilisateur et données topographiques lors d'un changement de niveau de détail*. PhD thesis, Université Paris-Est, Paris, France, mar 2015.
- [51] N. Jabeur. *A Multi-Agent System for "On-the-Fly" Web Map Generation and Spatial Conflict Resolution*. Thèse de doctorat, Université de Laval, 2006.
- [52] N. Jabeur, B. Boulekrouche, and B. Moulin. Using multiagent systems to improve real-time map generation. In *Advances in Artificial Intelligence*, pages 37–48. Springer, 2006.
- [53] A. Kolácný. Cartographic information—a fundamental concept and term in modern cartography. *The cartographic journal*, 6(1):47–49, 1969.
- [54] Y. Kubera, P. Mathieu, and S. Picault. Everything can be agent! In W. van der Hoek et al., editors, *9th Int. Joint Conf. on Auton. Agents and Multi-Agent Systems (AAMAS)*, pages 1547–1548, 2010.
- [55] D. Lane. Hierarchy, complexity, society. In *Hierarchy in natural and social sciences*, pages 81–119. Springer, 2006.
- [56] T. Lang. Rules for robot draughtsmen. 1969.
- [57] J. Lepagnot and G. Hutzler. A multiscale agent-based model for the simulation of avascular tumour growth. *Jour. of Biol. Phys. and Chem*, 9:17–25, 2009.
- [58] M. J. Lighthill and G. B. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 229, pages 317–345. The Royal Society, 1955.
- [59] T. Louail. De la géométrie des réseaux viaires à l'organisation spatiale intra-urbaine. une approche de la comparaison des formes de villes par la simulation orientée agents. *Revue Internationale de Géomatique*, 19(4):443–469, 2009.
- [60] W. A. Mackaness and N. Gould. The role of geography in automated generalisation. In *17th ICA Workshop on Generalisation and Multiple Representation. Vienna, Austria*, 2014.
- [61] W.A. Mackaness. Understanding geographic space. *Generalisation of geographic information: Cartographic modelling and applications*, 2007.
- [62] P. Marcenac, R. Courdier, S. Caldéroni, and J. C. Soulié. Towards an emergence machine for complex systems simulations. In *Tasks and Methods in Applied Artificial Intelligence*, pages 785–794. Springer, 1998.
- [63] P. Marcenac and S. Giroux. Geamas: A generic architecture for agent-oriented simulations of complex processes. *Applied Intelligence*, 8(3):247–267, 1998.
-

- [64] P. Mathieu and S. Picault. An interaction-oriented model for multi-scale simulation. In T. Walsh, editor, *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'2011)*, pages 332–337. AAAI, 2011.
- [65] P. Mathieu, S. Picault, and Y. Secq. Les environnements : en avoir ou pas ? Formalisation du concept et patterns d'implémentation. In Jean-Paul Jamont Rémy Courdier, editor, *Actes des 22e Journées Francophones sur les Systèmes Multi-Agents*, Systèmes Multi-Agents. Principe de parcimonie, pages 55–64, Loriol-sur-Drôme, France, October 2014. Cépaduès.
- [66] P. Mathieu and Y. Secq. Environment updating and agent scheduling policies in agent-based simulators. In *ICAART (2)*, pages 170–175, 2012.
- [67] A. Maudet, G. Touya, C. Duchêne, and S. Picault. Patterns multi-niveaux pour les sma. In Laurent Vercoeur and Gauthier Picard, editors, *23es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'15)*, pages 19–28, Rennes, France, 2015. Cépaduès.
- [68] A. Maudet, G. Touya, C. Duchêne, and S. Picault. Improving multi-level interactions modelling in a multi-agent generalisation model: first experiments. In D. Burghardt, editor, *Proc. of the 16th ICA Workshop on Generalisation and Map Production*, 2013.
- [69] A. Maudet, G. Touya, C. Duchêne, and S. Picault. Multi-agent multi-level cartographic generalisation in cartagen. In *Advances in Practical Applications of Heterogeneous Multi-Agent Systems: Proceedings of the 12th International Conference, PAAMS 2014*, Lecture Notes in Computer Science, pages 355–358. PAAMS, Springer, jun 2014.
- [70] A. Maudet, G. Touya, C. Duchêne, and S. Picault. Representation of interactions in a multi-level multi-agent model for cartography constraint solving. In Yves Demazeau et al., editors, *Proc. of the 12th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2014)*, number 8473 in LNCS, pages 183–194. Springer, 2014.
- [71] F. Michel. The IRMAS model: the influence/reaction principle for multiagent based simulation. In IFAAMAS, editor, *6th international joint conference on Autonomous agents and multiagent systems*, pages 903–905, Honolulu, Hawai'i, USA, May 2007. ACM.
- [72] F. Michel, A. Gouaich, and J. Ferber. Weak interaction and strong interaction in agent based simulations. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 43–56. Springer, 2003.
- [73] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute Santa Fe, 1996.
- [74] A. Morrison. Public transport maps in western european cities. *The Cartographic Journal*, 33(2):93–110, 1996.
- [75] G. Morvan. Multi-level agent-based modeling - a literature survey. CoRR, abs/1205.0561, 2013.

- 
- [76] G. Morvan and D. Jolly. Multi-level agent-based modeling with the influence reaction principle. *arXiv preprint arXiv:1204.0634*, 2012.
- [77] G. Morvan, A. Veremme, and D. Dupont. Irm4mls: The influence reaction model for multi-level simulation. In Tibor Bosse, Armando Geller, and CatholijnM. Jonker, editors, *Multi-Agent-Based Simulation XI*, volume 6532 of *Lecture Notes in Computer Science*, pages 16–27. Springer Berlin Heidelberg, 2011.
- [78] S. Mustière and B. Moulin. What is spatial context in cartographic generalization? In *Proceedings of the ISPRS Commission IV Symposium, Ottawa, July 2002*, pages 274–278, 2002.
- [79] M. Nöllenburg. A survey on automated metro map layout methods.
- [80] G. Morvan P. Mathieu and S. Picault. Simulations multi-agents multi-niveaux : quatre patterns de conception. In *24es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'15)*, 2016 », publisher =.
- [81] H.J. Payne. Models of freeway traffic and control. 1971.
- [82] D. Pumain, L. Sanders, A. Bretagnolle, B. Glisse, and H. Mathian. The future of urban systems: exploratory models. In *Complexity perspectives in innovation and social change*, pages 331–360. Springer, 2009.
- [83] J. Gil Quijano, G. Hutzler, and T. Louail. De la cellule biologique à la cellule urbaine : retour sur trois expériences de modélisation multi-échelles à base d'agents. In *Systèmes Multi-Agents, Génie logiciel multi-agents - JFSMA 09 - Dix Septièmes Journées Francophones sur les Systèmes Multi-Agents, Lyon, France, October 19-21, 2009*, pages 187–196, 2009.
- [84] J. Gil Quijano, G. Hutzler, and T. Louail. Accroche-toi au niveau, j'enlève l'échelle. éléments d'analyse des aspects multiniveaux dans la simulation à base d'agents. *Revue d'Intelligence Artificielle*, 24(5):625–648, 2010.
- [85] C. Ratzé, F. Gillet, J.-P. Müller, and K. Stoffel. Simulation modelling of ecological hierarchies in constructive dynamical systems. *Ecological complexity*, 4(1):13–25, 2007.
- [86] N. Regnauld and R.B. McMaster. A synoptic view of generalisation operators. *Generalisation of geographic information: Cartographic modelling and applications*, 2007.
- [87] J. Renard. Introduction de structures réactionnelles à activation ascendante dans une organisation hiérarchique descendante d'agents - application à la généralisation des alignements urbains. In P. Chevaillier and B. Mermet, editors, *Actes des 20e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2012)*, pages 139–149. Cépaduès, 2012.
- [88] J. Renard, J. Gaffuri, and C. Duchêne. Capitalisation problem in research - example of a new platform for generalisation: Cartagen. In *13th ICA Workshop on Generalisation and Multiple Representation*, sept 2010.
-



- [89] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34, 1987.
- [90] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.
- [91] A. Ruas. Oo-constraint modelling to automate urban generalisation process. In *8th International Symposium on Spatial Data Handling (SDH'98), 11-15 July, Vancouver (Canada)*, 1998.
- [92] A. Ruas. *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. Thèse de doctorat, spécialité sciences de l'information géographique, Université de Marne-la-Vallée, apr 1999.
- [93] A. Ruas. The role of meso level for urban generalisation. In *2nd ICA Workshop on Generalisation and Multiple Representation, 7-8 July, Ottawa (Canada)*, 1999.
- [94] A. Ruas. The role of meso objects for generalisation. In *9th Int. Symposium on Spatial Data Handling (SDH'00), Beijing (China)*, 2000.
- [95] A. Ruas and J.-P. Lagrange. Modélisation pour l'automatisation de la généralisation cartographique. In *4th European Conference on GIS (EGIS'94), 29 mars - 30 avril, Paris (France)*, 1994.
- [96] A. Ruas and C. Plazanet. Strategies for automated map generalisation. In *7th International Symposium on Spatial Data Handling (SDH'96), Delft (Netherlands), 12-16 August*, pages 319–336, 1996.
- [97] M. N. Sabo and E. Bernier. Methodology for developing a database of geometric patterns to better support on-the-fly map generalization. In *International Cartographic Conference*. Citeseer, 2005.
- [98] MN Sabo. *Intégration des algorithmes de généralisation et des patrons géométriques pour la création des objets auto-généralisants (SGO) afin d'améliorer la généralisation cartographique à la volée*. Thèse de doctorat, Université de Laval, 2007.
- [99] MN Sabo, A Cardenas, Y Bédard, and E Bernier. Introduction du concept de patrons géométriques et application aux bâtiments afin de faciliter leur généralisation cartographique à la volée. *Geomatica*, 59(3):295–311, 2005.
- [100] M. D Seck and H. J. Honig. Multi-perspective modelling of complex phenomena. *Computational and Mathematical Organization Theory*, 18(1):128–144, 2012.
- [101] D. Servat, E. Perrier, J.-P. Treuil, and A. Drogoul. When agents emerge from agents: Introducing multi-scale viewpoints in multi-agent simulations. In J. S. Sichman, R. Conte, and N. Gilbert, editors, *Proc. of the 1st Int. Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS'98)*, volume 1534 of LNCS, pages 183–198, Paris, France, 1998. Springer.

- 
- [102] M. Sester. Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science*, 19(8-9):871–897, 2005.
- [103] J. Sewall, D. Wilkie, and M. C. Lin. Interactive hybrid simulation of large-scale traffic. In *ACM Transactions on Graphics (TOG)*, volume 30, page 135. ACM, 2011.
- [104] K. S. Shea and R. B. McMaster. Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings of AutoCarto*, volume 9, pages 56–67, 1989.
- [105] J. Siebert, L. Ciarletta, and V. Chevrier. Agents and artefacts for multiple models co-evolution. Building complex system simulation as a set of interacting models. In *Autonomous Agents and Multiagent Systems - AAMAS 2010*, volume 1, pages 509–516, Toronto, Canada, May 2010. ACM.
- [106] J.-B. Soyez, G. Morvan, D. Dupont, and R. Merzouki. A methodology to engineer and validate dynamic multi-level multi-agent based simulations. In *MABS*, pages 130–142. Springer, 2012.
- [107] L.V. Stanislawski, B.P. Buttenfield, P. Bereuter, S. Savino, and C.A. Brewer. Generalisation operators. *Abstracting Geographic Information in a Data Rich World*, 2014.
- [108] P. Taillandier. *Révision automatique des connaissances guidant l'exploration informée d'arbres d'états. Application au contexte de la généralisation de données géographiques*. Thèse de doctorat, spécialité informatique, Université Paris-Est, 2008.
- [109] P. Taillandier, D.-A. Vo, E. Amouroux, and A. Drogoul. Gama: bringing gis and multi-level capabilities to multi-agent simulation. In *European Workshop on Multi-Agent Systems*, 2010.
- [110] J. T Tanner. The stability and the intrinsic growth rates of prey and predator populations. *Ecology*, 56(4):855–867, 1975.
- [111] J. Teulade-Denantes, A. Maudet, and C. Duchêne. Routes visualization: Automated placement of multiple route symbols along a physical network infrastructure. *Journal of Spatial Information Science (JoSIS)*, 11:53–79, 2015.
- [112] R. C Thomson and D. E Richardson. The 'good continuation' principle of perceptual organization applied to the generalization of road networks.
- [113] G. Touya. *Le Modèle CollaGen : Collaboration de processus automatiques pour la généralisation cartographique de paysages hétérogènes*. Thèse de doctorat, spécialité sciences de l'information géographique, Université Paris-Est, 2011. Thèse de doctorat.
- [114] G. Touya and C. Duchêne. Collagen: Collaboration between automatic cartographic generalisation processes. In Anne Ruas, editor, *Advances in Cartography and GIScience*, volume 1 of *Lecture Notes in Geoinformation and Cartography*, pages 541–558, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
-

- [115] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.
- [116] M. Visvalingam and J.D. Whyatt. Line generalisation by repeated elimination of points. *The Cartographic Journal*, 30(1):46–51, 1993.
- [117] D. A. Vo, A. Drogoul, and J.-D. Zucker. An operational meta-model for handling multiple scales in agent-based simulations. In *Proc. of the IEEE Int. Conf. on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*. IEEE Press, 2012.
- [118] J. M. Ware, C. B. Jones, and N. Thomas. Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8):743–769, 2003.
- [119] M. Wertheimer. A brief introduction to gestalt, identifying key theories and principles. *Psychol. Forsch*, 4:301–350, 1923.
- [120] D. Weyns and T. Holvoet. A formal model for situated multi-agent systems. *Fundamenta Informaticae*, 63(2-3):125–158, 2004.
- [121] D. Weyns, H V D Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems state-of-the-art and research challenges. In *International Workshop on Environments for Multi-Agent Systems*, pages 1–47. Springer, 2004.
- [122] D. Weyns, V. D. H. Parunak, and F. Michel. Environments for multi-agent systems. iii. In *International Workshop E4MAS 2006. 3. 8 May 2006. Hakodate. JP.*, 2006.
- [123] X. Zhang and E. Guilbert. A multi-agent system approach for feature-driven generalization of isobathymetric line. In Anne Ruas, editor, *Advances in Cartography and GIScience. Volume 1*, Lecture Notes in Geoinformation and Cartography, pages 477–495. Springer Berlin Heidelberg, 2011.
- [124] X. Zhang and E. Guilbert. A multi-agent system approach for feature-driven generalization of isobathymetric line. In *Advances in Cartography and GIScience. Volume 1*, pages 477–495. Springer, 2011.



Les cartes représentent l'information géographique d'une zone donnée de manière d'autant plus simplifiée que l'échelle de la carte est petite. Le procédé de simplification, appelé généralisation cartographique, est soumis au respect de contraintes de lisibilité, d'adéquation de la représentation avec le niveau d'abstraction souhaité et de cohérence avec la réalité. La volonté d'automatiser le processus de création de cartes à partir de bases de données géographiques, a conduit à la création d'algorithmes permettant d'effectuer cette simplification objet par objet. Néanmoins, les choix des algorithmes, tout comme leur paramétrage, sont autant influencés par l'objet sur lequel ils s'appliquent que par les autres objets en relation (e.g. bâtiment à proximité d'un autre, route parallèle à un alignement de bâtiments). Ce constat a motivé l'utilisation de modèles multi-agents pour la généralisation automatisée de cartes. Le principe de ces modèles multi-agents repose sur la modélisation des objets (e.g. bâtiment, tronçon de route, îlot urbain) sous forme d'agents qui cherchent à se généraliser de façon à satisfaire leurs contraintes. Plusieurs modèles multi-agents ont été proposés, chacun ayant une approche différente des interactions entre niveaux. Ici, nous entendons par niveau, par exemple, la distinction entre les agents individuels comme un bâtiment, des agents représentant un groupe d'autres agents, comme un îlot urbain composé des routes l'entourant et des bâtiments inclus dans l'îlot.

Nous étudions l'unification de ces modèles en nous appuyant sur le paradigme multi-niveaux PADAWAN, afin de faciliter les interactions entre agents de niveaux différents. Nous proposons ainsi le modèle DIOGEN, adaptant les principes d'interaction entre agents de niveaux différents à la généralisation cartographique guidée par des contraintes, ce qui a permis d'unifier les précédents modèles AGENT, CartACom et GAEL, tout en disposant de nouvelles capacités prometteuses.

Nous avons évalué notre proposition sur un ensemble de cas d'étude. Parmi ces cas, nous nous sommes penchés sur la généralisation de carte de randonnée, où les itinéraires sont symbolisés individuellement avec des symboles différents, à la manière des plans de bus. La présence de plusieurs symboles d'itinéraires sur une même route support amène des problèmes de généralisation particuliers, comme le choix du positionnement des itinéraires de part et d'autre de la route, ou les implications pour les autres objets de la carte (e.g. points d'intérêts, bâtiments) se retrouvant sous le symbole de l'itinéraire, problèmes que nous essayons de résoudre en nous appuyant sur notre proposition de représentation formelle multi-niveaux.

Ce travail nous a ensuite conduits à identifier des comportements multi-niveaux récurrents. Nous les avons exprimés de façon générique sous forme de *patterns* d'analyse, affranchies des spécificités de la généralisation cartographique, et de la résolution de problèmes contraints

Maps show geographic information of a given area in a simplified way, particularly when the scale is small. The simplification process, called cartographic generalisation, is submitted to several constraints: legibility, adequation to the abstraction level, and consistency with reality. The will to automate the maps creation process from geographical databases led to the creation of algorithms allowing the simplification object by object. However the choice of the algorithms, as their settings, are influenced by the object on which it is applied, and by the other objects in relation with this object (e.g. a building close to another one, a road parallel to a buildings alignment). This motivates the use of multi-agents models for automated map generalisation. Several multi-agent models were proposed, each of them having a different approach to manage multi-levels relations. Here, what we call a level is, for instance, the distinction between individual agents, like a building, and agents representing a group of other agents, like a urban block composed by the surrounding roads and buildings inside.

We study the unification of existing models, using the multi-level paradigm PADAWAN, in order to simplify interactions between agents in different levels. We propose the DIOGEN model, in which the principle of interactions between agents of different levels is adapted to cartographic generalisation guided by constraints, those allowing to unify the existing models AGENT, CartACom and GAEL, and giving promising features.

We evaluate our proposal on different case studies. Among them, we study the generalisation of trekking maps, where the routes are symbolized individually by a different coloured line symbols, like on bus maps. The presence of several route symbols on a same road leads to specific generalisation issues, like the choice of the side of each route symbol position, or the implications for the other objects on the map (e.g. points of interest, buildings) under the route symbol - issues tackled using our proposal of formal multi-levels representation.

This work leads us to the identification of recurrent behaviours. We express them as analysis patterns, in a way that is independent from cartographic generalisation and constraint solving problems.