



HAL
open science

Le Modèle CollaGen: collaboration de processus automatiques pour la généralisation cartographique de paysages hétérogènes

Guillaume Touya

► **To cite this version:**

Guillaume Touya. Le Modèle CollaGen: collaboration de processus automatiques pour la généralisation cartographique de paysages hétérogènes. Géographie. Université Paris-Est, 2011. Français. NNT: 2011PEST1052 . tel-00664464

HAL Id: tel-00664464

<https://theses.hal.science/tel-00664464>

Submitted on 30 Jan 2012

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 : Sciences de l'information géographique

Guillaume TOUYA

Le Modèle CollaGen

Collaboration de processus automatiques pour la généralisation cartographique de paysages hétérogènes



Soutenue le 14 juin 2011 devant le jury composé de :

| | |
|--------------------------|--------------|
| Pr. Christophe Claramunt | Rapporteur |
| Dr. Cécile Duchêne | Encadrante |
| Pr. Thérèse Libourel | Examinatrice |
| Dr. Annabelle Mas | Examinatrice |
| Pr. Philippe Mathieu | Examinateur |
| Dr. Anne Ruas | Directrice |
| Dr. Nicolas Sabouret | Rapporteur |

REMERCIEMENTS

Je remercie, tout d'abord, les membres de mon jury, Christophe Claramunt, Thérèse Libourel, Annabelle Mas, Philippe Mathieu et Nicolas Sabouret, qui ont pris le temps de lire ce (long !) mémoire pour évaluer mon travail de thèse.

Je remercie Anne Ruas, pour avoir dirigé ma thèse et pour m'avoir transmis le virus de la généralisation cartographique. Ses idées toujours innovantes et sa confiance ont été indispensables pour mener à bien cette thèse.

Je remercie Cécile Duchêne pour l'encadrement de cette thèse. Elle a toujours été présente pour répondre à mes interrogations et pour canaliser mes propositions, tout au long de ces trois années et demie de travail. Elle m'a surtout appris à être (je l'espère) un bon chercheur, et j'espère être capable un jour de faire aussi bien qu'elle pour transmettre moi aussi ces connaissances.

Je remercie Bénédicte Bucher pour m'avoir supporté si longtemps. En plus de sa gentillesse, son ouverture et son sens de la discussion sont à l'origine d'une grande partie de ce travail de thèse. Je lui dois aussi beaucoup.

Je tiens à remercier Mickael Brasebin et Jérémy Renard qui ont essuyé les plâtres lors de ma première répétition et ont été de bon conseil pour l'améliorer.

Un grand merci pour Jean-François Girres et Patrick Taillandier qui se sont relayés pour contribuer à la bonne ambiance qui a régné dans mon bureau pendant ma thèse. A ton tour maintenant, Jean-François !

Je tiens à remercier, bien évidemment, tous les chercheurs que j'ai croisés au COGIT ou dans le couloir K200 (la liste est trop longue) : grâce à eux, j'ai toujours eu un grand plaisir à venir travailler au labo. Une mention pour François qui m'a soufflé le nom du modèle et m'a ainsi déchargé d'une tâche ardue !

Je remercie énormément Loric, Sylvain et Lisa d'avoir fait un si long voyage pour m'encourager lors de ma soutenance. Je remercie bien sûr Clément, l'autre Dr Touya, (notamment pour m'avoir évité d'écrire des remerciements plus longs !) et mes parents pour leur présence indispensable.

Enfin, merci pour tout à Caroline.

Table des matières

| | |
|--|----|
| Introduction..... | 9 |
| CHAPITRE A. La Généralisation cartographique et ses processus automatiques | 13 |
| A.1. La généralisation cartographique | 13 |
| A.1.1. Information géographique et notion d'échelle..... | 13 |
| A.1.2. La Généralisation de données géographiques..... | 14 |
| A.2. Des Processus automatiques de généralisation cartographique..... | 19 |
| A.2.1. De L'Algorithme au processus de généralisation..... | 20 |
| A.2.2. Les Modèles par séquence prédéfinie | 22 |
| A.2.3. Les Modèles globaux..... | 24 |
| A.2.4. Les Modèles pas à pas locaux | 29 |
| A.2.5. Les Processus spécialisés..... | 34 |
| A.3. Objectifs de la thèse..... | 37 |
| CHAPITRE B. Un cadre formel pour la généralisation collaborative | 41 |
| B.1. Orchestration et Collaboration de processus | 41 |
| B.1.1. Orchestration et chorégraphie dans les services Web..... | 41 |
| B.1.2. Collaboration et coopération dans les systèmes multi-agents..... | 43 |
| B.1.3. L'Orchestration et la collaboration en généralisation cartographique..... | 44 |
| B.2. Définition de la Généralisation Collaborative | 46 |
| B.3. Analyse fonctionnelle de la Généralisation Collaborative | 46 |
| B.4. Un nouvel espace de travail : les espaces géographiques | 56 |
| B.4.1. Définitions | 56 |
| B.4.2. Modélisation d'un espace géographique..... | 60 |
| B.4.3. Quel est le meilleur contour pour espace territorial ?..... | 64 |
| B.5. Dynamique de la généralisation collaborative..... | 65 |
| CHAPITRE C. Mise en œuvre : le modèle CollaGen | 68 |
| C.1. Cas d'étude : généraliser la BD Topo® au 1 : 50 000..... | 68 |
| C.2. Qui fait quoi dans CollaGen ? | 69 |
| C.3. Dynamique du modèle CollaGen..... | 71 |
| C.4. Une ontologie de la généralisation cartographique | 75 |
| C.4.1. Qu'est-ce qu'une ontologie en informatique ? | 75 |
| C.4.2. Les Ontologies en généralisation | 77 |

| | |
|--|-----|
| C.4.3. La Problématique du schéma adapté..... | 78 |
| C.4.4. Que doit-on mettre dans une ontologie de la généralisation cartographique ? | 82 |
| C.4.5. Modélisation de l'ontologie de généralisation | 84 |
| C.4.6. Discussion | 85 |
| C.5. Un modèle formel de contraintes de généralisation | 86 |
| C.5.1. Les Classifications de contraintes de généralisation | 87 |
| C.5.2. Les Formalismes de contraintes existants..... | 89 |
| C.5.3. Proposition de modèle formel de contraintes de généralisation | 90 |
| C.5.4. Vers des contraintes de généralisation génériques | 95 |
| C.6. Un modèle formel de règles opérationnelles | 96 |
| C.7. Décrire les capacités d'un processus de généralisation..... | 98 |
| C.7.1. La Description des traitements géographiques | 98 |
| C.7.2. Les Capacités d'un processus de généralisation cartographique | 100 |
| C.7.3. Un Modèle formel pour décrire un processus de généralisation | 101 |
| C.8. Un modèle formel de règles d'orchestration..... | 104 |
| C.8.1. Les Grandes étapes de la généralisation..... | 105 |
| C.8.2. Quelles règles souhaite-t-on formaliser pour guider l'orchestration ? | 107 |
| C.8.3. Un Modèle formel de règle d'orchestration | 108 |
| C.8.4. Bilan de la formalisation des connaissances en généralisation | 109 |
| C.9. Espaces géographiques et processus dans CollaGen | 110 |
| C.9.1. Modélisation des espaces géographiques dans CollaGen | 110 |
| C.9.2. Les Processus disponibles dans CollaGen | 112 |
| C.10. Les Moniteurs de contrainte | 113 |
| C.10.1. Définition et modélisation..... | 114 |
| C.10.2. Fonctionnement des moniteurs de contrainte | 116 |
| C.11. Composant d'enrichissement géographique | 117 |
| C.11.1. Enrichir les données par des concepts meso et procéduraux..... | 117 |
| C.11.2. Expliciter les relations géographiques..... | 119 |
| C.11.3. Construire les espaces géographiques..... | 121 |
| C.11.4. Création des moniteurs de contrainte | 122 |
| C.11.5. La Boîte à outils | 122 |
| C.12. Composant de traduction | 123 |
| C.12.1. Annotation sémantique de la base de données initiale..... | 124 |
| C.12.2. Appariement entre ontologie et boîte à outils | 126 |

| | |
|--|-----|
| C.12.3. Appariement entre contraintes formelles et moniteurs..... | 127 |
| C.12.4. Les Fonctions de traduction | 129 |
| C.12.5. La Gestion des unités | 133 |
| C.13. Composant <i>registre</i> | 134 |
| C.13.1. Fonctionnalités d'un registre | 134 |
| C.13.2. Modélisation du composant registre | 136 |
| C.13.3. Comment le registre répond-il à une requête de généralisation ?..... | 137 |
| C.13.4. Un Registre qui apprend de ses erreurs..... | 141 |
| C.13.5. Autres capacités du registre..... | 143 |
| C.14. Composant d'observation et évaluation..... | 143 |
| C.14.1. L'Évaluation en généralisation | 144 |
| C.14.2. Evaluation des conflits par les moniteurs | 145 |
| C.14.3. Observation en ligne d'une généralisation | 147 |
| C.14.4. Emergence d'un espace géographique..... | 151 |
| C.14.5. Evaluation finale de la généralisation d'un espace | 155 |
| C.14.6. Interactions entre émergence et évaluation finale..... | 161 |
| C.15. Composant d'orchestration | 161 |
| C.15.1. Fonctions du composant d'orchestration | 162 |
| C.15.2. Gestion des états dans CollaGen..... | 163 |
| C.15.3. Choisir un type d'espace géographique | 164 |
| C.15.4. Ordonner les instances d'espaces géographiques | 168 |
| C.16. Composant de gestion des effets de bord | 169 |
| C.16.1. Taxonomie des effets de bord..... | 170 |
| C.16.2. Un Modèle de contraintes de cohérence..... | 174 |
| C.16.3. Détection des effets de bord..... | 176 |
| C.16.4. Correction des effets de bord | 178 |
| C.17. Bilan..... | 181 |
| CHAPITRE D. Mise en Œuvre De CollaGen par un Système Multi-Agents..... | 183 |
| D.1. Vers une perspective SMA pour CollaGen | 183 |
| D.2. Qui sont les agents dans CollaGen ?..... | 186 |
| D.2.1. Schéma général du Système Multi-Agents CollaGen..... | 186 |
| D.2.2. L'Environnement du SMA | 187 |
| D.2.3. Les Agents situés..... | 189 |
| D.2.4. Les Agents logiciels | 190 |

| | |
|--|-----|
| D.2.5. Les Agents coordinateurs..... | 191 |
| D.2.6. Communication et perception | 194 |
| D.3. Répartition des capacités entre les agents | 195 |
| D.3.1. L'Agent espace géographique..... | 195 |
| D.3.2. L'Agent processus | 196 |
| D.3.3. Le Chef d'orchestre | 197 |
| D.3.4. Le Registre..... | 198 |
| D.3.5. L'Agent observateur..... | 198 |
| D.3.6. L'Agent correcteur d'effets de bord | 199 |
| D.3.7. Bilan de la répartition des capacités entre agents..... | 199 |
| D.3.8. Déroulement d'une généralisation CollaGen du point de vue SMA..... | 200 |
| D.4. Discussion..... | 201 |
| D.4.1. Analyse critique des agents de CollaGen | 201 |
| D.4.2. En quoi est-il intéressant de mettre en œuvre CollaGen par un SMA..... | 203 |
| D.4.3. Autres considérations SMA..... | 205 |
| CHAPITRE E. Implémentation et Résultats..... | 208 |
| E.1. Développements Connexes à CollaGen..... | 208 |
| E.1.1. Description de notre plateforme de travail..... | 208 |
| E.1.2. Un Modèle de généralisation par moindres carrés..... | 209 |
| E.1.3. Amélioration de processus disponibles..... | 213 |
| E.1.4. Des Processus spécialisés | 215 |
| E.1.5. Algorithmes de création d'espaces géographiques | 216 |
| E.1.6. Correction d'effets de bord par moindres carrés..... | 222 |
| E.2. Des outils pour faciliter l'usage de CollaGen..... | 223 |
| E.2.1. Faciliter les saisie des ressources | 223 |
| E.2.2. Outils de traduction interactive | 225 |
| E.3. Description du protocole de test..... | 227 |
| E.3.1. Description des données utilisées | 227 |
| E.3.2. Description des ressources formelles utilisées | 228 |
| E.4. Résultats de tests du prototype | 233 |
| E.4.1. Résultats individuels des composants de CollaGen | 234 |
| E.4.2. Etude de sensibilité de CollaGen au choix de la limite des zones urbaines | 244 |
| E.4.3. Résultats de généralisations complètes par le prototype..... | 247 |
| E.4.4. Expérimentations sur l'ordre induit par l'orchestration des espaces | 251 |

| | |
|--|-----|
| E.5. Evaluation des résultats | 252 |
| E.5.1. Evaluation de la formalisation des spécifications | 252 |
| E.5.2. Comparaison avec des logiciels du commerce | 253 |
| E.5.3. Comparaison avec un modèle unique | 257 |
| E.5.4. Comparaison avec un processus ad-hoc | 259 |
| E.6. Discussion | 260 |
| Conclusion et perspectives..... | 263 |
| Rappel de l'objectif..... | 263 |
| Contributions de notre thèse | 263 |
| Enseignements | 265 |
| Perspectives..... | 266 |
| Bibliographie..... | 270 |
| Glossaire | 287 |
| Publications | 288 |

INTRODUCTION

Contexte Général

Les avancées technologiques nous plongent dans un monde dans lequel l'information géographique n'a jamais été aussi présente au quotidien. Le nombre toujours plus important de services mobiles localisés (location based services) utilisables sur des Smartphones fait de plus en plus appel aux représentations cartographiques. Nous avons aussi accès aux outils de navigation par GPS, aux géoportails et autres outils pour réaliser des cartes « à la carte ». Les citoyens peuvent même contribuer eux-mêmes à la construction de l'information géographique avec le développement très important de plateformes de *Volunteered Geographic Information (VGI)* (Goodchild, 2007) dont la plus emblématique est OpenStreetMap. Ces nouveaux domaines d'applications proposent des conditions de lecture de l'information géographique plus difficiles qu'avec une carte traditionnelle en papier. Par exemple, l'écran d'un Smartphone laisse peu de place à l'affichage de la page, la navigation embarquée doit proposer une lecture adaptée à la conduite alors que les données collaboratives proposent le défi de la visualisation d'un plus grand nombre de type de données sur la carte.

Nous avons donc tous accès à une quantité, inimaginable il y a quelques années, de données géographiques mais ces données brutes ne peuvent être affichées directement sur nos téléphones ou nos ordinateurs car elles sont nombreuses et pas forcément faites pour être affichées ensemble. Il faut d'abord les traiter pour en retirer ce qui est utile au besoin de l'utilisateur et ce qui est graphiquement compatible. Le traitement le plus évident est la représentation de ces données sous une forme cartographique. Alors, que le terrain de jeu des cartographes est en train de se diversifier, il est fondamental de pouvoir garantir dans toutes ces applications des représentations cartographiques justes, lisibles, belles et adaptées au contexte d'affichage. Dans ce contexte, les changements d'échelle peuvent se faire très rapidement par les fonctions de zoom et réclament des outils de généralisation cartographique pour garantir la lisibilité et la justesse.

La généralisation cartographique a pour objectif la représentation simplifiée de l'information géographique sur une carte suite à une réduction d'échelle. La généralisation cartographique est traditionnellement liée à la production de cartes en séries par des producteurs de données géographiques tels l'Institut Géographique National (IGN) en France. Elle était réalisée autrefois au crayon et au couteau à graver par des artistes cartographes. Elle est encore aujourd'hui souvent réalisée par des cartographes de manière interactive sur ordinateur. Dans un cas comme dans l'autre, il s'agit d'un processus très long, surtout dans le contexte technologique actuel où l'on réclame les cartes dans l'instant sur son téléphone. Depuis près de vingt-cinq ans, des travaux de recherche, partout dans le monde, se sont intéressés à l'automatisation de la généralisation, principalement dans le contexte de la production de cartes en séries mais les besoins s'étant élargis, l'automatisation est encore plus importante aujourd'hui.

La recherche sur l'automatisation de la généralisation a beaucoup progressé ces dernières années et des cartes en série sont déjà produites en utilisant des processus automatiques, notamment à l'IGN. Les travaux présentés dans cette thèse s'inscrivent dans la continuité de ces travaux et visent à franchir un nouveau pas vers l'automatisation de la généralisation cartographique pour des cartes en série mais aussi des cartes « à la carte ».

Problématique de la thèse

Il existe aujourd'hui dans la littérature scientifique de nombreuses propositions pour des processus automatiques de généralisation cartographique, répondant à des logiques, des besoins différents et reposant sur des techniques différentes. Ils ont tous prouvé plus ou moins d'efficacité sur des problèmes spécifiques que l'on peut exprimer en termes de paysages précis (une zone urbaine ou zone de montagne par exemple), de thème de données traité (l'occupation du sol ou le réseau hydrographique par exemple), de type de conflits résolus (les conflits de superposition entre symboles par exemple) ou plus généralement un mélange des trois. Seulement, aucun n'est réellement capable de prendre en charge la généralisation d'une carte complète faite de paysages, de thèmes et de conflits hétérogènes. Pour aller plus loin dans l'automatisation de la généralisation cartographique, deux choix sont possibles : soit on cherche à concevoir un nouveau modèle encore plus générique et efficace que les autres capable de gérer les différents paysages, thèmes et conflits ; soit on cherche à profiter de cette diversité de processus existants et on essaie de trouver un moyen pour les faire travailler ensemble. La première solution paraissant illusoire, nous considérons qu'il faut plutôt essayer de faire collaborer les processus automatiques existants pour progresser vers l'automatisation complète de la généralisation cartographique.

Dans cette thèse nous proposons donc un cadre formel pour ce que nous appelons la généralisation collaborative, capable de répondre à cette problématique. Puis nous proposons un modèle générique dans ce cadre, CollaGen, ainsi qu'une implémentation prototype qui permet la généralisation de cartes topographiques à partir de bases de données de l'Institut Géographique National.

Plan

Dans le chapitre A, nous situons notre travail dans le contexte de la recherche en généralisation cartographique. Après avoir présenté la définition et les principes de base de la généralisation cartographique et de son automatisation, nous dressons un état de l'art des processus automatiques de généralisation cartographique pour mettre en lumière leur diversité et leurs spécificités. Nous concluons ce chapitre en exposant les objectifs détaillés de cette thèse.

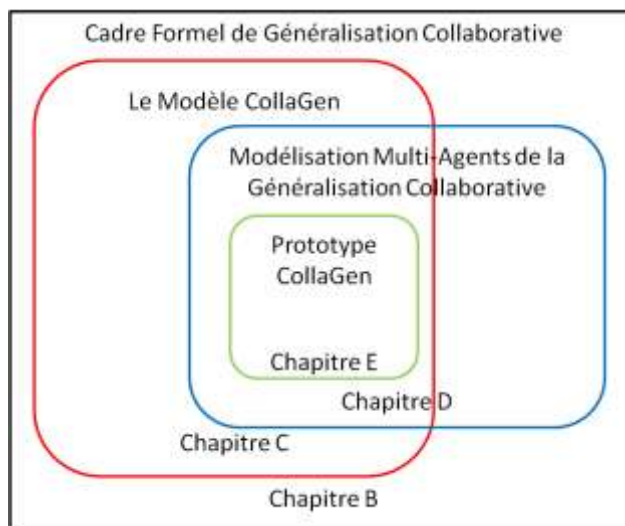


Figure 0-1. Structuration imbriquée de notre proposition et du plan de ce mémoire : du plus générique au chapitre B avec le cadre formel de généralisation collaborative au plus spécifique avec le prototype de CollaGen dédié à la généralisation de carte topographiques type 1 :50000 IGN au chapitre E.

Les chapitres suivants suivent une progression dans la description de la solution que nous proposons en partant du très générique cadre formel jusqu'au prototype que nous avons implémenté (Figure 0-1). Le chapitre B présente donc un cadre formel de Généralisation Collaborative permettant d'utiliser ensemble différents processus de généralisation. Dans un premier temps, nous définissons les notions d'orchestration et de collaboration par analogie avec des domaines informatiques connexes. Puis nous définissons ce cadre formel et nous analysons ses fonctions afin d'identifier ses besoins en composants et ressources : une ressource est un élément utilisé par le système de généralisation et un composant est un élément logiciel qui manipule les ressources pour réaliser une action précise dans la généralisation collaborative. Puis nous décrivons une des ressources qui va de pair avec ce cadre formel : les espaces géographiques. Enfin, nous présentons les différentes possibilités de dynamique qui peuvent être choisies pour mettre en œuvre la généralisation collaborative.

Dans le chapitre C, nous exposons en détail le modèle CollaGen par lequel nous mettons en œuvre le cadre formel de généralisation collaborative défini au chapitre B. Nous définissons, d'abord, les rôles que peut adopter un humain en interaction avec CollaGen. Nous présentons, ensuite, les choix que nous avons effectués pour la dynamique du modèle CollaGen. Puis, nous présentons nos choix de modélisation pour chacune des ressources du cadre formel puis pour chacun des six composants définis au chapitre B. Ces ressources et composants permettent de répondre aux problèmes posés par la généralisation collaborative : le partitionnement pertinent des données par rapport aux processus à utiliser, l'interopérabilité des processus de généralisation, le choix de la meilleure orchestration des processus, la gestion d'effets de bord causés par la généralisation d'une partie seulement des données sur les autres données, l'évaluation pas à pas des généralisations et la gestion de l'homogénéité de traitement sur toute la carte. Nous introduisons notamment la notion clé pour CollaGen de moniteur de contrainte.

Dans le chapitre D, nous approfondissons notre description de CollaGen en exposant pourquoi et comment ce modèle peut être modélisé d'un point de vue Système Multi-Agents. Nous expliquons d'abord en quoi il est naturel et pratique d'intégrer CollaGen dans le paradigme multi-agents. Puis, nous exposons quels sont les éléments de CollaGen qui sont vus comme des agents dans notre mise

en œuvre. Nous détaillons ensuite comment sont réparties les capacités des différents composants du modèle entre les agents précédemment définis. Enfin, nous discutons ce choix de mise en œuvre en nous interrogeant sur son apport et en nous demandant si nous avons bien défini des agents autonomes.

Dans le chapitre E, nous présentons une implémentation prototype de CollaGen pour la généralisation de cartes pour une gamme d'échelle autour du 1 : 50 000 de type IGN à partir de la base de données BDTopo®. Nous présentons d'abord les protocoles de test établis puis les outils conçus pour faciliter la mise en place de ces tests. Nous revenons également sur les processus de généralisation et les outils d'analyse spatiale développés dans le cadre de la thèse. Puis, nous présentons une série de résultats issus du prototype permettant de mettre en valeur chacun des composants. Nous proposons une illustration des possibilités de CollaGen en faisant varier ses paramètres. Enfin, nous proposons une évaluation des résultats du prototype.

CHAPITRE A. LA GENERALISATION CARTOGRAPHIQUE ET SES PROCESSUS AUTOMATIQUES

A.1. LA GENERALISATION CARTOGRAPHIQUE

Avant de s'intéresser aux processus de généralisations que nous allons essayer de faire collaborer dans notre travail de thèse, cette partie présente les notions essentielles à la compréhension de nos travaux et notamment celles concernant l'information géographique et la généralisation cartographique.

A.1.1. Information géographique et notion d'échelle

L'information géographique est l'information recueillie dans une base de données géographique qui comprend une **information** relative à un phénomène du monde représenté par son aspect et ses caractéristiques, et sa **localisation** décrite dans un système de coordonnées de référence (Denègre et Salgé, 1996 ; Ruas et Libourel, 2002). Dans une base de données géographique, l'information géographique est souvent représentée en mode vecteur : on associe à chaque objet une primitive géométrique (un point, une polyligne ou un polygone). Pour représenter l'espace dans une base de données géographiques, il faut répondre à deux questions, quoi, et comment ? (Ruas, 2004) :

- Quoi ? : il s'agit de sélectionner l'information qui sera utile pour répondre aux objectifs de la base de données.
- Comment ? : il s'agit par abstraction de savoir comment on va représenter ces informations de manière géométrique et sémantique dans la base de données.

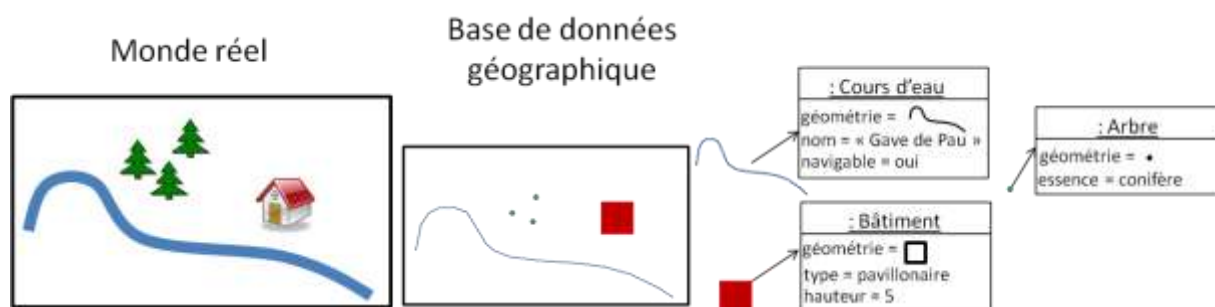


Figure A-1. Exemple de modélisation d'une base de données géographique avec ses classes, géométries et attributs dans le formalisme UML.

Si l'on prend l'exemple de la Figure A-1, la réponse au quoi est « on sélectionne les rivières, les bâtiments et les arbres » et la réponse au comment est « on représente les arbres par des objets ponctuels, les rivières par des objets linéaires et les bâtiments par des objets surfaciques ». L'information géographique stockée dans une base de données géographique peut être symbolisée comme elle l'est sur une carte grâce à des outils logiciels appelés *Systèmes d'Information Géographique* (SIG).

Dans une base de données géographique, les objets sont classifiés selon leur nature : dans notre exemple de la Figure A-1, il existe trois classes, 'Arbre', 'Bâtiment' et 'Cours d'eau'. Chaque objet est caractérisé par une *géométrie* vectorielle qui peut être un point, une polyligne ou un polygone et des *données attributaires* qui permettent de compléter la description de l'objet (Figure A-1).

L'échelle est une notion indissociable de l'information géographique. Dans le cas des cartes, on la définit mathématiquement comme la taille d'un élément sur la carte divisée par la taille réelle de cet élément. Il n'existe pas une seule échelle pour voir et représenter le monde et il en va de même de l'information géographique : à chaque échelle, on peut percevoir les objets géographiques différemment. (Chaudhry, 2007) illustre ce propos en disant : "You can not tell you are on an island buy looking at the pebbles on a beach" ("On ne peut pas deviner que l'on est sur une île en regardant simplement les galets sur la plage").

Dans le cas des bases de données géographiques, (Ruas, 2004, p.29) définit *l'échelle caractéristique* (ou *échelle de référence*) d'une base de données géographique comme "l'échelle la plus appropriée pour rendre sensible un phénomène en fonction de son étude". On définit ainsi le *niveau de détail* d'une base de données à partir de son échelle caractéristique : il traduit la finesse de contenu d'une base de données géographique. (Ruas, 2004) distingue deux types de niveau de détail dans une base de données géographique : le niveau de détail sémantique qui traduit la finesse du processus d'abstraction de l'information géographique et le niveau de détail géométrique qui traduit la finesse de description de la géométrie des objets géographiques.

A.1.2. La Généralisation de données géographiques

(Ruas, 2004) définit une Base de Données Cartographique (BDC) comme une base qui "regroupe des données géographiques dont la position de chaque objet est proche de la précision géométrique de saisie mais dont la taille et la position de chaque objet est conforme à l'échelle mathématique de représentation. Une base de données cartographique possède un espace de signification qui est constitué de concepts géographiques décrits dans la base. Ces concepts sont décrits par le schéma de données et les spécifications." Cette distinction est issue des travaux de (Brassel et Weibel, 1988), prolongeant ceux de (Brüggemann, 1985; Grünreich, 1985; Meyer, 1986) qui ont défini la notion de Digital Cartographic Model (DCM) proche de la BDC par opposition au Digital Landscape Model (DLM) qui se rapproche de la base de données géographique (BDG). **La généralisation cartographique peut être définie comme l'opération permettant d'obtenir une BDC à partir d'une base de données géographique.**

Plus généralement, (Ruas, 2004) définit la généralisation de données géographiques de la manière suivante : "La généralisation est un processus de synthèse d'information. Ce processus peut être comparé avec celui d'un résumé de texte dont les objectifs sont de réduire le nombre de mots, de

retenir les idées principales, de ne pas faire de faux sens et, si possible, conserver le style de l'auteur (...). Pour bien généraliser des données géographiques, il faut également :

- réduire la quantité d'information,
- mettre en valeur l'information la plus importante,
- rester fidèle à l'information initiale,
- et dans le cas d'une carte, respecter les règles de sémiologie qui permettent une bonne lecture de l'information."

La Figure A-2 montre comment la production brute d'une BDC en adaptant les symboles de la carte à la nouvelle échelle provoque des problèmes de lisibilité liés aux contraintes de perception de l'œil humain et des problèmes d'interprétation liés aux limites d'analyse du cerveau humain (au delà des limites visuelles, le cerveau humain n'est pas capable d'analyser le message transmis par un ensemble de symboles trop petits ou trop nombreux) (Weger, 1998). Dans notre exemple, les bâtiments sont trop petits (lisibilité) et les bâtiments chevauchent les symboles de routes (interprétation) donc il est difficile de les voir mais aussi de comprendre leur sens. La généralisation cartographique consiste dans cet exemple en un grossissement et une simplification des bâtiments pour garantir une bonne lisibilité, des éliminations et des déplacements de bâtiments pour garantir une bonne interprétation. Les zones trop denses en routes et bâtiments subissent même un changement d'abstraction en devenant des zones bâties représentées par un aplat coloré car il n'est plus possible à cette échelle de les représenter individuellement de manière lisible et interprétable. Cette dernière transformation est appelée CUSP (Change Under Scale Progression) (Müller, 1990 ; Buttenfield, 1991).

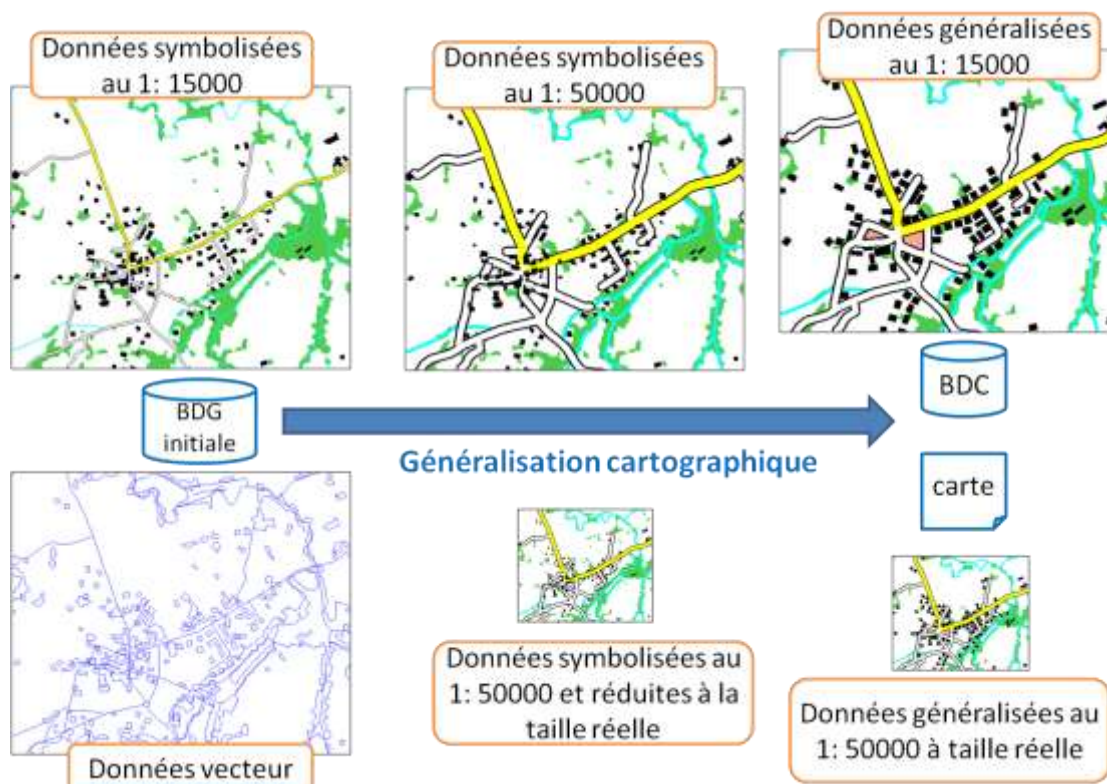


Figure A-2. Définition illustrée de la généralisation cartographique : représenter les données d'une BDG à une échelle plus petite que l'échelle de référence engendre des problèmes de lisibilité et d'interprétation. La généralisation permet de rendre les données lisibles.

La complexité de ce processus de généralisation est qu'il ne s'agit pas simplement de manipuler des points, des lignes et des surfaces mais plutôt de modéliser et comprendre et préserver une part de l'espace géographique et les interactions de ses objets (Mackaness, 2007). Un exemple de cette complexité dans la Figure A-2 réside dans la transformation des deux zones où la densité de bâtiments et de routes est trop grande, qui ont été interprétées comme des zones bâties. En lisant une carte, un géographe est capable d'identifier des phénomènes géographiques non représentés de manière explicite dans les données servant à la carte. Il est donc essentiel que la généralisation facilite ces interprétations géographiques et il est même souhaitable qu'elle fasse émerger des phénomènes géographiques relatifs à la nouvelle échelle. La Figure A-3 illustre par deux exemples cette complexité accentuée par les phénomènes géographiques sous-jacents dans les données.

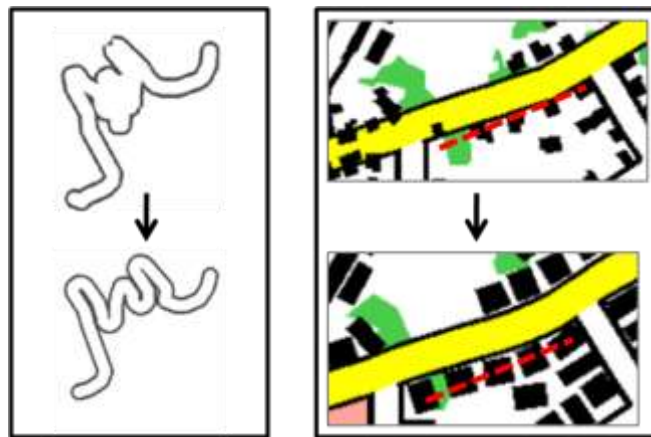


Figure A-3. Illustration de la complexité de la généralisation. (a) il ne faut pas uniquement simplifier la ligne représentant la route mais transmettre la présence d'une série de virages en épingle. (b) la généralisation doit ici garantir que l'on pourra toujours interpréter cet ensemble de bâtiments comme un alignement le long d'une route (représenté par la ligne pointillée).

Les opérateurs de généralisation

Pour généraliser des données géographiques, il faut les modifier en leur appliquant différentes transformations que l'on peut classer suivant le type d'opération. Beaucoup d'auteurs ont proposé des classifications d'opérateurs de généralisation cartographique (McMaster & Shea, 1992; Hake et al, 1994; Yaolin et al, 2001; Cecconi, 2003; Foerster et al, 2007 ; Roth et al, 2008). Nous adopterons ici la classification et le vocabulaire de (Mustière, 2001) qui nous apparaît comme le meilleur compromis entre clarté et complétude. Cette classification distingue trois catégories d'opérateurs de généralisation : les opérateurs de *simplification*, de *caricature* et d'*harmonisation*. On peut voir en Figure A-4 deux exemples d'opérateurs définis dans cette classification, l'agrégation (de la catégorie *caricature*) et la squelettisation (de la catégorie *simplification*), que nous utiliserons plus loin dans cette thèse pour illustrer notre propos.



Agrégation (fusion ou amalgamation) Squelettisation (polygone vers ligne ou point)

Figure A-4. Deux exemples d'opérateurs de généralisation d'après la classification de (Mustière, 2001).

Différentes façons de voir la généralisation

Dans le passé, de nombreux chercheurs (Bertin, 1967 ; Nyerges, 1991 ; Beguin et Pumain, 1994 ; Müller et al, 1995 ; Weibel et Dutton, 1999) ont distingué deux types de généralisation :

- la *généralisation de modèle* ou *conceptuelle* qui propose une nouvelle conceptualisation des données géographiques en fonction du nouveau niveau de détail.
- La *généralisation cartographique* ou *graphique* ou *structurale* qui regroupe les opérations qui intègrent les contraintes liées à la représentation graphique (i.e. les symboles, la lisibilité).

Seulement, (Ruas, 2004) nous montre que ces distinctions sont difficiles à utiliser car souvent ambiguës. Il est en effet souvent difficile de dire si une opération de généralisation tient de la généralisation de modèle ou de la généralisation graphique. Si on prend l'exemple d'une opération d'agrégation, dans le cas (a) de la Figure A-5, on peut dire que l'agrégation fait partie de la généralisation de modèle. Mais dans le cas (b) où deux bâtiments sont agrégés car trop proches et encombrés (ils ne peuvent être correctement déplacés), l'opération d'agrégation fait clairement partie de la généralisation graphique. (Ruas, 2004) fait la même démonstration d'ambiguïté à partir de l'opération d'élimination/sélection.

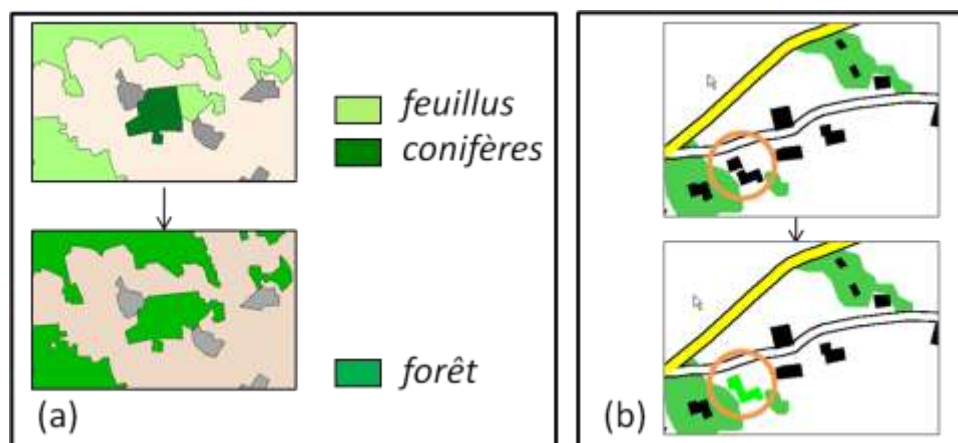


Figure A-5. Deux exemples d'opérations d'agrégation, la première plutôt associée à la généralisation de modèle et la seconde à la généralisation graphique.

Comme (Ruas, 2004) nous considérons que cette distinction n'est pas assez claire. Nous utiliserons donc plutôt la distinction suivante :

- Nous appelons *généralisation de base de données* une généralisation qui vise à la production d'une base de données dérivée de niveau de détail différent, sans objectif de visualisation cartographique.
- Nous appelons *généralisation cartographique* une généralisation qui vise à la production d'une base de données cartographique (BDC) d'échelle de symbolisation inférieure au niveau de détail de la BDG source.

Ainsi, un processus de généralisation cartographique peut contenir des phases de généralisation de modèle et des phases de généralisation graphique au sens de (Harrie et Sarjakoski, 2002). La Figure A-6 montre cette distinction avec la possibilité pour une généralisation cartographique de contenir

des opérations plutôt reliées à la généralisation de modèle et d'autres plutôt reliées à la généralisation graphique.

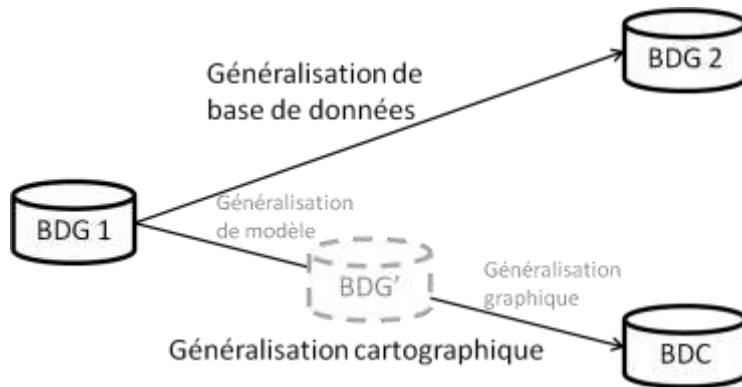


Figure A-6. Distinction entre généralisation de base de données et généralisation cartographique qui peut comporter une phase de généralisation de modèle et une phase de généralisation graphique.

Des algorithmes de généralisation

Quand la cartographie est manuelle, ces opérations de généralisation sont réalisées manuellement par des cartographes. Si l'on souhaite automatiser la généralisation, il est nécessaire de développer des algorithmes qui réalisent ces opérations automatiquement sur de l'information géographique numérique. La conception d'algorithmes de généralisation a donc été une des premières tâches des chercheurs en généralisation. Depuis le premier algorithme de filtrage d'une ligne (Douglas et Peucker, 1973), il existe ainsi de nombreux algorithmes en fonction des opérateurs, des sauts d'échelle et des données sur lesquelles on les applique. On peut voir un aperçu de cette diversité dans (Regnauld et McMaster, 2007). La Figure A-7 montre des exemples d'algorithmes réalisant des opérations de caricature sur des bâtiments et des routes de montagne.

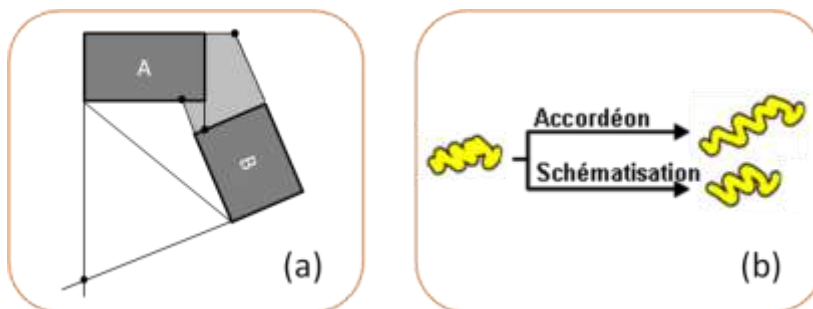


Figure A-7. Exemples d'algorithmes de généralisation. (a) algorithme d'amalgamation de bâtiments (Regnauld, 1998). (b) algorithmes de caricature de routes de montagne (Lecordix et al, 1997).

L'Analyse spatiale pour caractériser et enrichir les données

La généralisation qu'elle soit cartographique ou de base de données est une opération fortement contextuelle dans le sens où un cartographe va toujours généraliser en analysant le contexte géographique de la situation qu'il traite pour pouvoir réaliser les opérations appropriées. En effet, d'après (Mackaness et Edwards, 2002), la carte est vue comme un palimpseste de structures

émergeant des symboles représentés et le cartographe prend en compte ces structures quand il généralise. On peut ainsi considérer la généralisation comme un processus holistique car il considère l'ensemble des entités de la carte et leurs interactions (Bundy et al, 1995).

Dès les premiers modèles de généralisation, le besoin en enrichissement est identifié : (Brassel et Weibel, 1988) évoquent comme première phase de la généralisation la "reconnaissance de structure" ; (McMaster et Shea, 1988) parlent plutôt d'évaluation cartométrique (par mesures d'analyse spatiale) pour répondre à la question "quand généraliser?". Ensuite (Ruas et Plazanet, 1996) intègrent la caractérisation d'une situation comme première étape de leur processus de généralisation. Enfin, (Weibel et Dutton, 1998) parlent clairement d'enrichissement de la base de données pour permettre la généralisation. Il est maintenant clairement établi que l'enrichissement de données est un préalable indispensable à tout processus de généralisation et de nombreux travaux s'y sont intéressés, par exemple (Boffet, 2000 ; Mackaness et Edwardes, 2002, Steiniger et al, 2006, 2008; Lüscher et al, 2008a; Touya, 2008).

Ainsi, en plus des algorithmes de généralisation, de nombreux travaux ont été effectués sur des outils de mesure par analyse spatiale pour mieux caractériser les données géographiques et les enrichir. (Uni-Zh, 1999) propose un état de l'art de ces mesures dans le cadre du projet européen AGENT et (Barillot, 2002 ; Heinzle et Anders, 2007) proposent des vues complémentaires sur les algorithmes d'analyse spatiale nécessaires à la généralisation. La Figure A-8 montre différents algorithmes pour mesurer l'orientation d'un bâtiment (Duchêne et al, 2003).

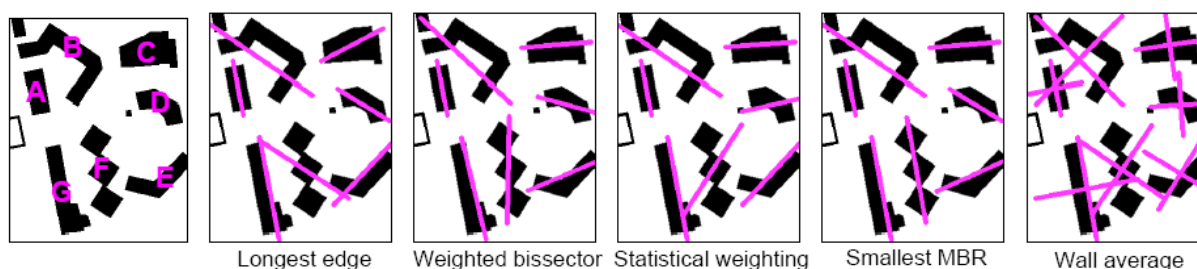


Figure A-8. Différentes mesures existantes pour déterminer l'orientation d'un bâtiment (Duchêne et al, 2003).

A.2. DES PROCESSUS AUTOMATIQUES DE GENERALISATION CARTOGRAPHIQUE

Maintenant que nous avons défini la généralisation cartographique et ses principes de base, nous allons nous intéresser aux processus automatiques de généralisation cartographique. Dans un premier temps, nous décrivons comment la recherche en généralisation est passée de l'algorithme au processus automatique puis nous présentons un état de l'art des processus automatiques de généralisation cartographique.

A.2.1. De L'Algorithme au processus de généralisation

Afin de clarifier les concepts utilisés dans cette partie, nous commençons par quatre définitions.

- Nous appelons *algorithme de généralisation* un processus qui réalise une opération de généralisation parmi celles qui sont présentées dans la partie A.1.2 grâce à un programme informatique automatique.
- Nous appelons *processus de généralisation* un processus qui enchaîne et/ou combine différentes opérations de généralisation de manière automatique.
- Nous appelons *modèle de généralisation* un modèle conceptuel (éventuellement implémenté) qui permet d'enchaîner et/ou combiner différentes opérations de généralisations pour des cas d'applications différents. La mise en œuvre d'un modèle de généralisation pour un cas d'application particulier est considérée ici comme un *processus de généralisation*.
- Nous appelons enfin *approche de généralisation* un ensemble de modèles de généralisations dont le point de vue est le même.

Pour illustrer ces définitions, prenons le cas d'AGENT (Ruas, 1999 ; Barrault et al, 2001) dont on peut voir une application en Figure A-9. Il s'agit selon nos définitions d'un modèle de généralisation car il permet aux objets géographiques quelque soit leur type d'être des agents qui s'appliquent des algorithmes de généralisation pour se généraliser, ceux-ci étant choisis en fonction du contexte local. Il se situe dans l'approche de généralisation dite pas à pas locale car on résout les conflits un par un localement sur chaque objet ou groupe d'objets géographiques. Sa mise en œuvre pour la généralisation des zones urbaines denses illustrée en Figure A-9 est, elle, considérée comme un processus de généralisation : dans ce processus les agents du modèle sont les bâtiments, les routes et les structures urbaines.

La distinction entre approche, modèle, processus et algorithmes est détaillée plus loin dans cette partie en l'illustrant avec un des modèles de généralisation décrits dans cet état de l'art.

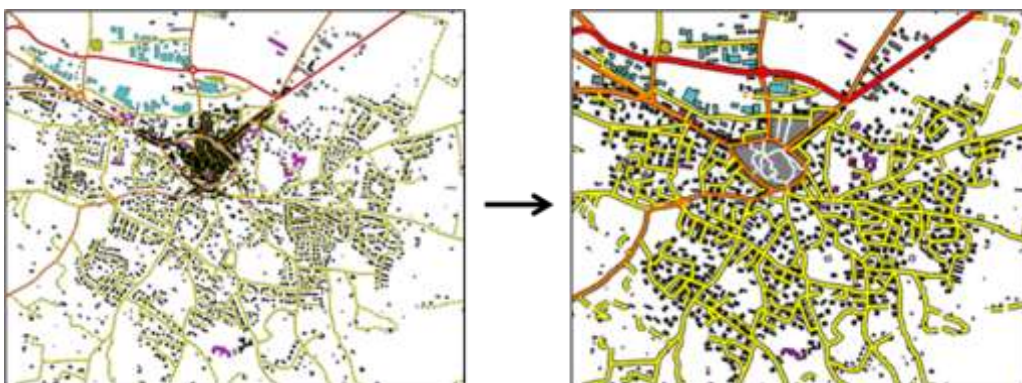


Figure A-9. Généralisation de données de la BD Topo © de l'IGN au 1 : 50 000 sur une ville par un processus issu du modèle AGENT. Figure issue de (Barrault et al, 2001).

La mise au point de processus ou de modèles de généralisation nécessite donc de savoir enchaîner des algorithmes de généralisation, ce qui a été un des premiers problèmes identifiés par les chercheurs en généralisation. (Brassel et Weibel, 1988) et (McMaster et Shea, 1988 ; Shea et McMaster, 1989) traitent de ce problème, ce dernier se posant les questions : "Quand, comment et

pourquoi généraliser ?". Il en ressort que l'on ne peut spécifier une séquence de généralisation qu'en prenant en compte les spécifications du produit final et les données elles-mêmes car la généralisation va dépendre du contexte géographique (caractéristiques et relations) de chaque situation (ou configuration particulière des données) présente dans les données. Il s'agit alors de modéliser la composante holistique de la généralisation.

Les premiers processus de généralisation ont été développés dans les années 90 pour la généralisation d'un thème géographique particulier. Ainsi (Müller et Wang, 1992) proposent un processus pour la généralisation d'un ensemble de surfaces non connexes comme des lacs qui se base sur des opérations dilatation, érosion, élimination et agrégation (Figure A-10). Le processus GALBE (Mustière, 1998) est dédié à la généralisation des routes de montagnes : il caractérise les routes et leur sinuosité, les découpe en portions homogènes et applique l'algorithme adapté à chaque portion. (Schylberg, 1992) est un autre exemple de processus de généralisation dédié aux données d'occupation du sol. Au fur et à mesure des avancées de la recherche en généralisation, la conception de modèles de généralisation a été privilégiée car ces modèles permettent la résolution d'un plus grand nombre de situations particulières de généralisation.

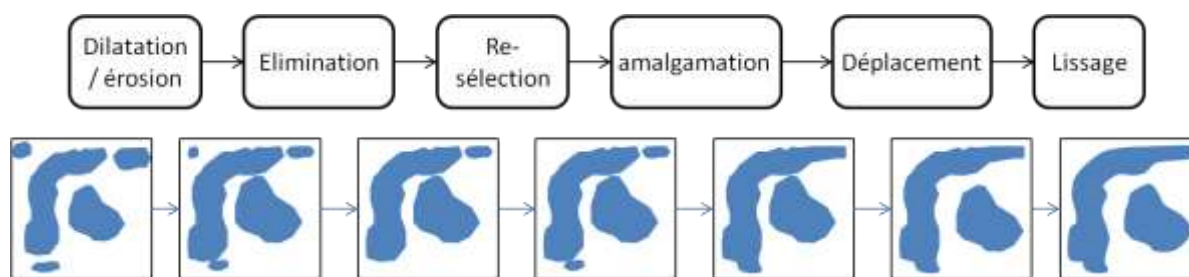


Figure A-10. Illustration d'un des premiers processus automatiques de généralisation dédié à la généralisation des groupes de lacs.

Les premiers processus et modèles développés visaient principalement à la réalisation de systèmes-experts copiant le raisonnement d'un cartographe, dans une approche condition-action (Harrie et Weibel, 2007). Parmi ces premiers travaux, on peut citer (Mackaness et al, 1986 ; Mackaness et Fisher, 1987) dont l'objectif était la cartographie automatique en général et pas seulement la généralisation (choix des classes à afficher, des symboles, des textes sur la carte, etc.). Un système expert contenant des règles issues de la littérature en cartographie définit pour chaque classe le type de symbole à appliquer, les conflits susceptibles d'arriver et éventuellement la méthode à utiliser pour résoudre le conflit. Les travaux de Nickerson (Nickerson et Freeman, 1986 ; Nickerson, 1988) ont abouti à la réalisation d'une plate-forme de généralisation des routes (et autres linéaires) basée sur un système-expert. Il a défini des règles pour la sélection, la simplification des lignes et pour détecter des conflits de superposition entre symboles, conflits qui sont résolus les uns après les autres. Un grand nombre de règles sont produites pour couvrir tous les cas de superposition avec des méthodes adaptées ce qui permet d'obtenir automatiquement des résultats très intéressants (Figure A-11). On peut également citer les travaux suivants (Buttenfield et Mark, 1991 ; Schylberg, 1992). Selon les recommandations de (Beard, 1991), les modèles plus récents ont principalement suivi une approche basée sur la satisfaction de contraintes qui définissent les besoins d'une carte bien généralisée et lisible. La plupart des modèles que nous allons présenter dans cette partie suivent plutôt une approche orientée contraintes.

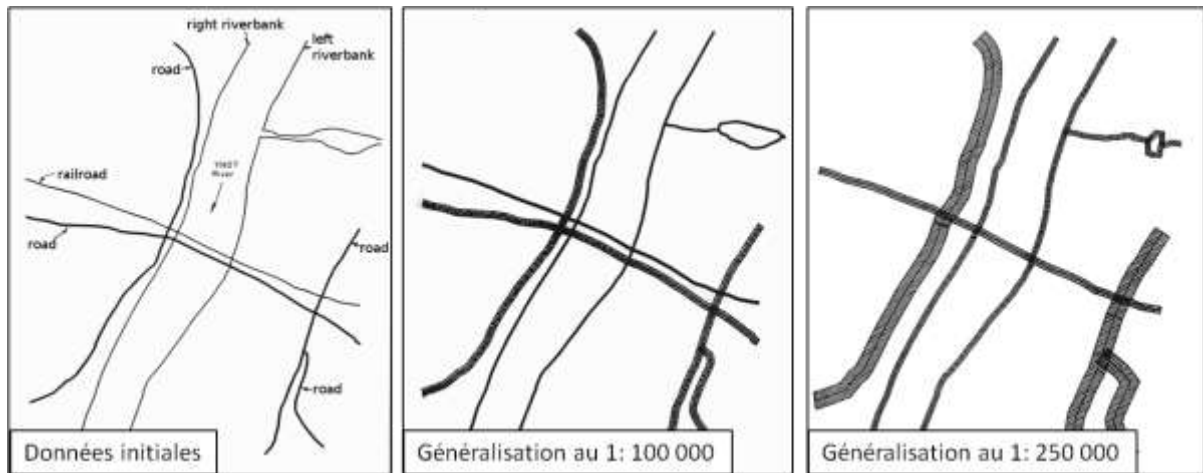


Figure A-11. Résultats de généralisation sur des objets linéaires obtenus avec le processus de Nickerson fondé sur un système-expert (Nickerson, 1988).

La fin de ce chapitre présente un état de l'art représentatif des modèles (et certains processus particuliers) de généralisation qui sont classés suivant leur approche. Par ailleurs, pour chaque modèle ou processus présenté, nous détaillerons les situations pour lesquelles le processus donne de bons résultats en termes de paysages géographiques, de thèmes des données traitées ou de types de conflits cartographiques gérés. Nous mentionnerons également les paramètres que le processus prend en entrée afin de mettre en valeur les questions d'interopérabilité de ces différents processus.

A.2.2. Les Modèles par séquence prédéfinie

L'approche par séquences prédéfinies consiste à définir des séquences d'opérations sur des populations. Par exemple, on filtre les routes, puis on les lisse puis on simplifie les bâtiments. Cette approche a fait ses preuves pour de faibles changements d'échelles en zone urbaine par exemple comme avec la plate-forme CHANGE développée par l'Université de Hanovre (Bobrich, 2001).

Le logiciel SIG ArcGIS® de ESRI contient un module de généralisation par séquences prédéfinies, Model Builder (Lee, 2003). Il permet de définir graphiquement des séquences d'algorithmes fournis par le logiciel comme le montre l'exemple sur les courbes de niveau (Figure A-12). Ce modèle se prête bien aux cas simples pour lesquels des adaptations locales des séquences en fonction du contexte ne sont pas nécessaires. Les processus de sélection des routes (séquence de sélection en zone rurale puis sélection en villes puis recollage du réseau par exemple) ou d'autres thèmes en sont de bons exemples. Il faut noter que bien que postérieurs à (Beard, 1991) ces deux modèles ne formalisent pas les contraintes.

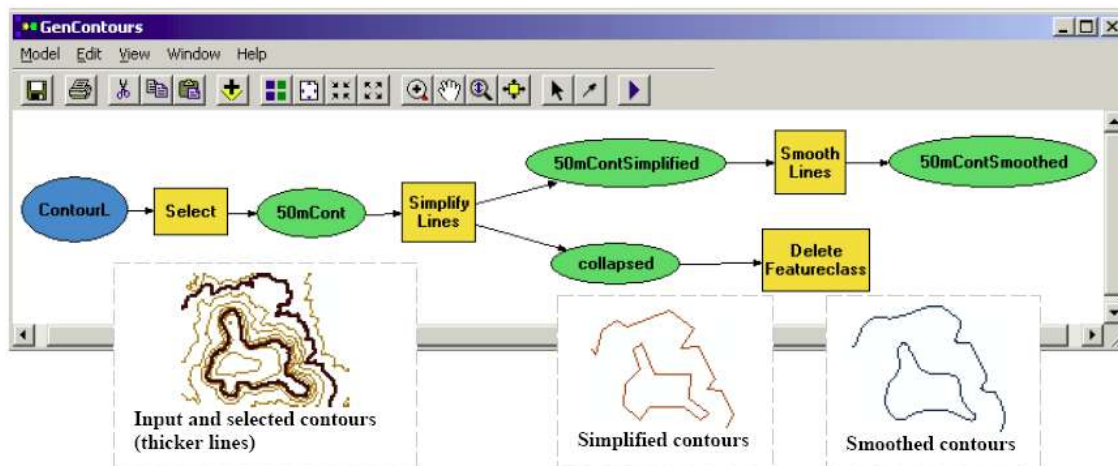


Figure A-12. Définition d'une séquence prédéfinie d'algorithmes avec l'outil Model Builder (Lee, 2003)

En s'appuyant sur le concept des "workflows", il est possible de définir des séquences d'opérations plus complexes et plus adaptées aux problématiques de la généralisation. Les workflows sont des graphes de séquençements autorisant des sauts conditionnels, des boucles, des tests et des processus concurrents (Figure A-13). (Petzold *et al*, 2006) présente une application des workflows à la modélisation de processus de généralisation. Le modèle par workflow dispose *a priori* de possibilités plus larges que le Model Builder même si aucune application concrète n'est présentée. En plus des paramètres individuels de chaque composant, un processus basé sur un modèle par workflow peut également prendre en entrée la mise en place des différents composants, c'est-à-dire la présence et l'ordre des mesures et algorithmes utilisés (suivant le type de mise en œuvre du modèle dans le processus, voir encart sur la généralisation par moindres carrés).

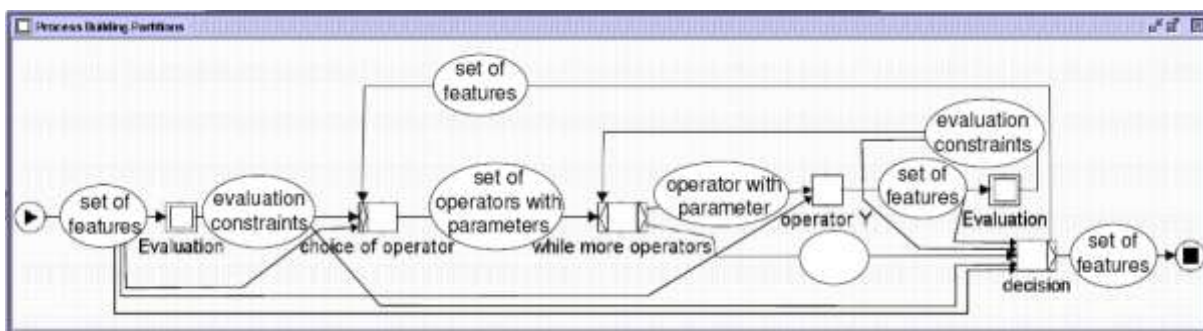


Figure A-13. Exemple de séquence prédéfinie par un workflow complexe (Petzold *et al*, 2006).

Le modèle par workflow permet également de définir des processus de granularité différente en prenant comme composants du workflow des processus de généralisation à la place des algorithmes. Le Global Master Plan de (Ruas et Plazanet, 1996) qui définit des grandes étapes comme 'on commence par généraliser les routes principales' puis 'on généralise les zones urbaines denses', peut être vu comme un workflow de processus de généralisation.

A.2.3. Les Modèles globaux

Les modèles présentés dans cette partie ont le point commun de réaliser la généralisation en cherchant une solution globale optimale. Toutes les contraintes sont prises en compte en même temps et le modèle vise l'optimisation de leur satisfaction globale.

Généralisation par Moindres Carrés

L'idée de la généralisation par Moindres Carrés est de traduire l'ensemble des contraintes en un très grand système d'équations dont les inconnues sont les variations des coordonnées des objets géographiques (Harrie, 1999). Un tel système contient plus d'équations que d'inconnues et ne possède pas de solution exacte. Les moindres carrés sont une technique mathématique pour obtenir une solution optimale de ce système qui minimise l'écart-type des résidus (Bjerhammar, 1973). (Harrie, 1999; Harrie et Sarjakoski 2002; Sester, 2000 ; Sester, 2005) proposent des méthodes pour modéliser des contraintes sous forme d'équations et (Harrie, 2003) des méthodes pour assigner des poids efficaces aux différentes contraintes (par exemple pour éviter que les bâtiments et les routes se déforment). Ce modèle est particulièrement efficace pour résoudre des conflits de proximité (Harrie et Sarjakoski 2002), alors que les opérations de simplification et de caricature ont déjà été effectuées (Figure A-14). En effet, il faut noter que la limite de ce modèle est que certaines opérations nécessaires ne concernent pas les coordonnées mais les objets (élimination, agrégation). Les paramètres d'un processus fondé sur le modèle par moindres carrés sont le système d'équations sur les points ainsi qu'une matrice pondérant chacune des équations (donc chacune des contraintes).



Figure A-14. *Généralisation par moindres carrés. Les objets initiaux apparaissent grisés (Harrie et Sarjakoski, 2002).*

Nous proposons ici un encart pour illustrer, sur l'exemple concret de la généralisation par moindres carrés, les différences entre approche, modèle, processus et algorithme de généralisation. Nous illustrons également la démarche de spécialisation d'un modèle pour concevoir un processus de généralisation sur cet exemple.

Illustration du passage approche/modèle/processus/algorithme

Dans le cas de la généralisation par moindres carrés, l'approche est une approche globale car il s'agit bien de résoudre l'ensemble des conflits d'un seul coup en résolvant une équation matricielle tenant compte de toutes les contraintes sur les coordonnées sous forme d'équations.

Ici, le modèle est celui présenté dans (Harrie et Sarjakoski, 2002) où des contraintes internes sont associées aux objets suivant leur type : un bâtiment a une contrainte de rigidité alors qu'une route a une contrainte pour conserver la courbure. Des contraintes externes sont aussi ajoutées

entre des objets comme des contraintes de proximité ou de maintien des intersections entre routes. Ces contraintes sont traduites en équations sur les points composant chaque objet. Chacune des contraintes est pondérée en fonction de l'importance des résidus : les résidus sont les écarts à une résolution parfaite d'une des équations et plus le poids d'une contrainte est important plus les résidus après résolution des moindres carrés sont faibles.

Réaliser un processus de généralisation à partir de ce modèle entraîne plusieurs choix. Ici, le processus est dédié à la généralisation des routes et des bâtiments concernés par des conflits de proximité. Tout d'abord, on limite le choix des contraintes à appliquer sur les objets, une dizaine dans notre cas. On doit également choisir les méthodes utilisées pour analyser spatialement les proximités entre objets afin de calculer les équations à partir des contraintes : nous utilisons une triangulation de Delaunay dans notre processus. Pour utiliser le processus il reste alors à déterminer si les objets sont rigides ou malléables (non rigides), les distances minimales entre les objets en mm carte et le poids de chacune des contraintes sur une échelle de 1 à 20, l'ensemble des poids étant transformé automatiquement en matrice de poids.

Dans le cas des moindres carrés, il n'y a pas d'algorithmes mis en jeu, un exemple de passage de modèle à processus utilisant des algorithmes est présenté plus loin dans cette partie.

Un processus n'est donc pas simplement une implémentation logicielle d'un modèle de généralisation mais plutôt une spécialisation de ce modèle qui réduit en quelque sorte l'espace des paramètres du modèle. En se spécialisant, un processus réduit également les capacités d'un modèle à traiter différents types de problèmes de généralisation : dans notre cas, le processus de généralisation par moindres carrés n'est plus dédié qu'aux problèmes de proximité entre bâtiments et routes.

Généralisation par modèles physiques

Certaines opérations de généralisation peuvent être vues comme des déformations de l'espace géographique donc des analogies avec des modèles physiques ont été très fertiles en modèles de généralisation. Ainsi, (Hojholt, 1998, 2000) propose une approche basée sur des triangles flexibles. Les conflits cartographiques assignent des forces d'expansion aux triangles pour pousser ou déformer les voisins. Ces forces sont compensées par la rigidité des objets. La position d'équilibre est déterminée par la méthode des éléments finis. Du fait de la nécessité de traduire les contraintes en rigidité de triangles, ce modèle est dédié à la résolution de contraintes de proximité entre objets géographiques qui sont les plus simples à traduire. Il paraît en effet ardu de traduire des contraintes sur la granularité du contour des bâtiments par des rigidités et des forces sur des triangles.

(Baeijs, 1998) a proposé un modèle dans lequel des forces de répulsion, suivi ou changement sont définies sur les points composant les géométries des objets. La position d'équilibre est obtenue par un système multi-agent dans lequel les agents sont les points des géométries. Dans ce modèle, la position d'équilibre n'est pas obtenue par éléments finis mais par une résolution itérative par petits déplacements des agents, d'où une difficulté de maintenir la convergence du système. Ce modèle de généralisation est dédié dans les applications présentées aux contraintes de proximité entre objets linéaires (routes et cours d'eau par exemple) et objets surfaciques (bâtiments par exemple). Les

paramètres sont assez simples consistant en une classification des objets en fonction de l'importance et en un seuil de lisibilité entre les objets.

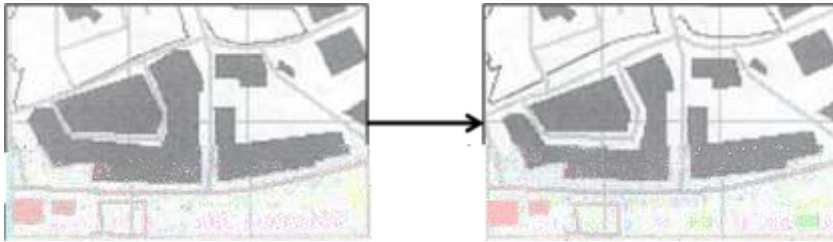


Figure A-15. Avant et après la généralisation des bâtiments et des routes par les agents points (Baeijs et al, 1996).

La méthode des Snakes introduite pour la généralisation dans (Burghardt et Meier, 1997) modélise les objets linéaires sous forme d'une chaîne de segments rigides articulés soumis à des forces internes et externes. Les forces internes correspondent aux contraintes de préservation de forme qui incitent les segments à garder leur configuration initiale. Les forces externes correspondent aux conflits de superposition avec les autres objets qui incitent les segments à se déplacer pour résoudre le conflit. Le champ des forces correspond à une énergie que l'on cherche à minimiser pour obtenir la position d'équilibre des forces, qui correspond à la meilleure position des objets linéaires à généraliser. La minimisation des forces externes et internes est calculée par la méthode des éléments finis qui est utilisée de manière itérative.

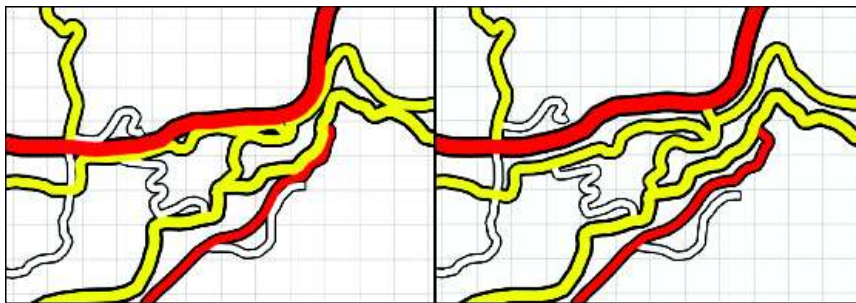


Figure A-16. A gauche, un réseau routier avec superpositions de symboles. A droite, les superpositions sont résolues par les poutres élastiques (Bader et Barrault, 2001).

Les « Poutres élastiques » ou Elastic Beams (Bader et Barrault, 2001 ; Bader et al, 2005) étendent le modèle des Snakes en remplaçant les parties rigides du modèle précédent par des poutres élastiques déformables en fonction de propriétés de résistance à l'étirement et la compression (Figure A-16). Ce modèle est particulièrement adapté aux conflits de superposition entre symboles d'objets linéaires comme les routes ou les cours d'eau. Son application sur des graphes de flexibilité, sous-réseaux adaptés à la résolution par les Beams a prouvé son efficacité (Lemarié, 2003) : autour des conflits de superposition entre symboles d'objets linéaires, le graphe de flexibilité est étendu aux tronçons de réseau qui permettent un amortissement du déplacement des objets en conflit sans trop de déformations. Ce graphe de flexibilité est plus ou moins étendu suivant les propriétés géométriques des tronçons de réseau (une portion droite amortit plus facilement une déformation qu'une portion sinueuse). Les Beams peuvent aussi être adaptés aux conflits de proximité entre routes et bâtiments (Bader et al, 2005). Les paramètres, dans le modèle des Beams, sont les valeurs de résistance affectées à chacun des types d'objets (par exemple, les routes sont plus résistantes à la compression que les rivières mais moins résistantes que les voies ferrées).

(Joubran et Doytsher, 2004 ; 2005) proposent un autre modèle de généralisation basé sur des principes physiques avec une analogie au champ électrique. Comme dans un champ électrique, les objets géographiques ont une certaine puissance qui guide les opérations à effectuer. À travers cette puissance, chaque objet a une influence sur ses voisins et est influencé par ses voisins. Les objets sont soumis à des contraintes cartographiques relatives à certaines de leurs propriétés : les caractéristiques internes des objets (taille, forme, granularité) sont mesurées et des relations de l'objet avec son voisinage sont retrouvées par triangulation et par buffer pour mesurer la congestion de l'objet. La détermination de la puissance qui est une fonction de ces caractéristiques se fait en utilisant un réseau de neurones (Joubran et Doytsher, 2008). Suivant la configuration du champ de forces dérivées des puissances, plusieurs actions peuvent être effectuées : élimination, agrégation, déplacement, grossissement ou simplification (Figure A-17) pour trouver une situation d'équilibre du champ électrique. Les résultats proposés sont toutefois relativement discutables, notamment dans certains choix d'agrégaions de bâtiments. Ce modèle semble adapté aux situations dans lesquelles les objets géographiques cherchent au mieux à occuper l'espace (comme dans la Figure A-17) sans avoir de transformations drastiques à effectuer (éliminations, typification) car elles engendreraient des configurations de forces trop complexes et trop variées pour en déduire les opérations à effectuer.

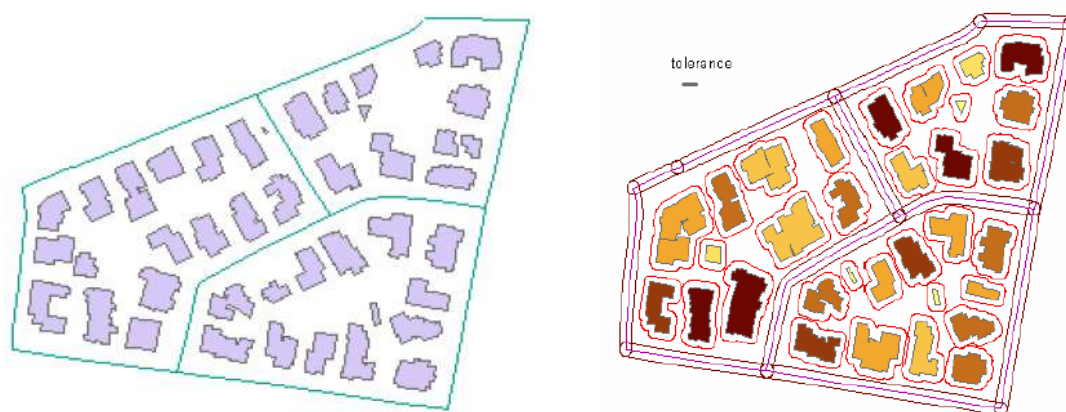


Figure A-17. Résultats obtenus sur 3 îlots de bâtiments par le modèle d'analogie avec un champ électrique. Plus les bâtiments sont forcés, plus leur puissance est grande (Joubran et Doytsher, 2008).

Généralisation par minimisation d'une fonction de coût

Dans cette approche, une solution globale est recherchée par exploration locale de l'espace des solutions à partir d'une solution initiale. La qualité générale de la généralisation est calculée de manière globale sous la forme d'une fonction de coût que l'on va chercher à minimiser par l'exploration de ces solutions par des techniques d'optimisation issues de l'intelligence artificielle. (Ware et Jones, 1998) proposent ainsi de tester à chaque itération une séquence d'opérations avec des paramètres différents et de trouver la séquence minimisant le coût. La difficulté est de trouver la bonne configuration sans tout essayer ce qui serait trop coûteux. (Ware et Jones, 1998) proposent la méthode de descente de gradient (Rich et Knight, 1990) ou le recuit simulé (Kirkpatrick et al, 1983). Dans (Ware et al, 2003) une transition vers une solution voisine correspond à une opération de généralisation, sur un objet et avec des paramètres, chacun étant tiré au sort dans l'espace des

solutions voisines, améliorant sensiblement les résultats par rapport aux modèles précédents (Figure A-18). Le modèle est adapté aux situations dans lesquelles le nombre de contraintes différentes est assez faible (seulement proximité et maintien des alignements dans l'application de la Figure A-18) pour permettre la construction de la fonction de coût. Même avec un nombre de contraintes faible, ce modèle est délicat à utiliser car il pose le problème de l'agrégation dans une seule fonction de coût : que faire quand on améliore à un endroit et simultanément on dégrade à un autre endroit ? Le paramètre principal est, dans ces modèles comme dans les suivants, la fonction de coût globale évaluant la généralisation. A un niveau de granularité plus grand, on peut considérer comme paramètres la stratégie d'exploration locale (descente de gradient, recuit simulé etc.) et la définition d'une transition vers une solution proche.

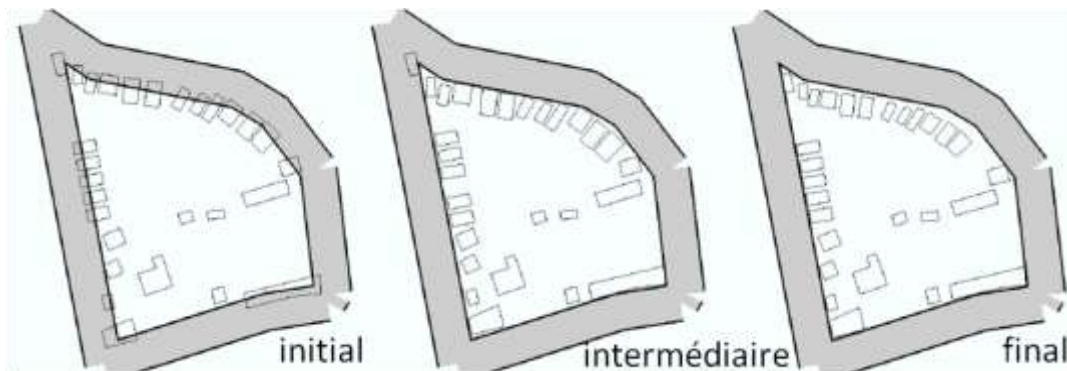


Figure A-18. Généralisation d'un îlot par recuit simulé (données initiales à gauche et résultat à droite) (Ware et al 2003).

Dans (Wilson et al, 2003) une transition vers une solution voisine correspond au choix d'une transformation pour chaque objet. Un algorithme génétique (Friedberg, 1958; Russel et Norvig, 2006, p.132) est utilisé pour obtenir la combinaison optimale de transformations. Le champ d'application de ce modèle est relativement le même que pour le précédent mais (Brosset et Devogele, 2004) montrent son application à la généralisation des sondes bathymétriques (équivalents des points cotés dans les cartes marines). Le problème de sélection des sondes convient bien à la modélisation par génome car chaque sonde peut être un gène avec deux allèles, sélectionné ou non. De plus, les contraintes mises en jeu sont très en relation avec le voisinage de chaque sonde ce qui convient bien au fonctionnement par croisement génétique.

(Richards et Ware, 2010 ; Ware et Richards, 2010) proposent une autre méthode pour rechercher la configuration optimale parmi les solutions avec un algorithme de colonies de fourmis. Cet algorithme classique de l'intelligence artificielle (Dorigo et al, 1996) cherche à reproduire le comportement d'une colonie de fourmis pour se guider vers une destination en déposant des phéromones sur leur chemin. Ce modèle est appliqué à la schématisation de réseaux de transport qui convient bien car la schématisation peut être ramenée à l'optimisation du positionnement de chaque nœud du réseau pour former des tronçons horizontaux, verticaux ou diagonaux de longueur suffisante. Son extension à d'autres problèmes de généralisation dépend donc de l'abstraction de ces problèmes sous la forme d'un ensemble de solutions spatialisées auxquelles des phéromones peuvent être assignées. Cependant, les fourmis ne cherchant pas par cycle, l'analogie semble plus adaptée à la hiérarchisation qu'à la plupart des problèmes de généralisation.

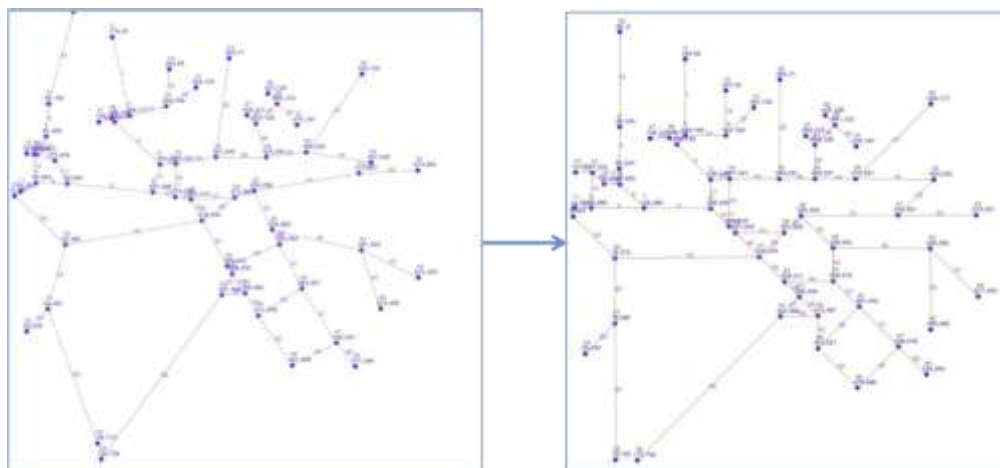


Figure A-19. Schématisation d'un réseau routier par optimisation par colonies de fourmis (Richards et Ware, 2010).

A.2.4. Les Modèles pas à pas locaux

L'approche pas à pas locale est celle qui applique littéralement la proposition de (McMaster et Shea, 1988 ; Shea et McMaster, 1989) en répondant aux questions *quand*, *comment*, et *pourquoi généraliser* ? C'est la présence locale de conflits cartographiques, que l'on peut définir comme des contraintes non satisfaites selon les principes de (Beard, 1991), qui déclenche dans cette approche les opérations de généralisation. Ces conflits sont détectés par des mesures d'analyse spatiale et dépendent des spécifications du produit final recherché. Ces mesures servent également à analyser le contexte géographique et les caractéristiques du conflit pour déterminer quel algorithme sera utilisé et avec quels paramètres.

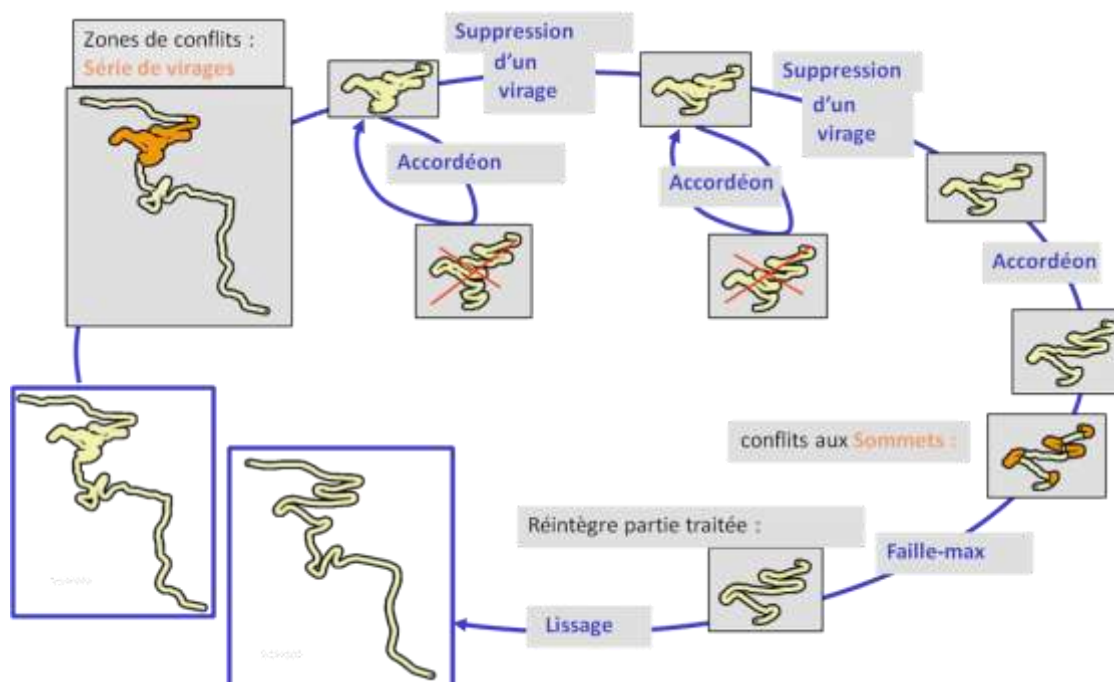


Figure A-20. Description du processus GALBE de généralisation pas à pas locale des routes (Mustière, 1998).

Des premiers processus ont adopté cette approche pas à pas locale pour traiter des problèmes de généralisation spécifiques comme la généralisation des groupes de lacs (Müller et Wang, 1992) ou pour la généralisation des routes de montagne (Figure A-20) avec le processus GALBE (Mustière, 1998).

Plusieurs modèles théoriques ont été proposés pour détailler cette approche pas à pas locale. Ainsi, le modèle de (Ruas et Plazanet, 1996) détaille l'étape de focalisation sur une situation détectée comme conflictuelle (*quand*, en introduisant la nuance du *où*) quand le modèle de (Weibel et Dutton, 1998) détaille le rôle des contraintes dans le guidage du processus répondant au *pourquoi* et au *comment* (Figure A-21). Les modèles présentés dans la suite de cette partie se sont appuyés sur ces deux modèles théoriques pour mettre en œuvre l'approche pas à pas locale.

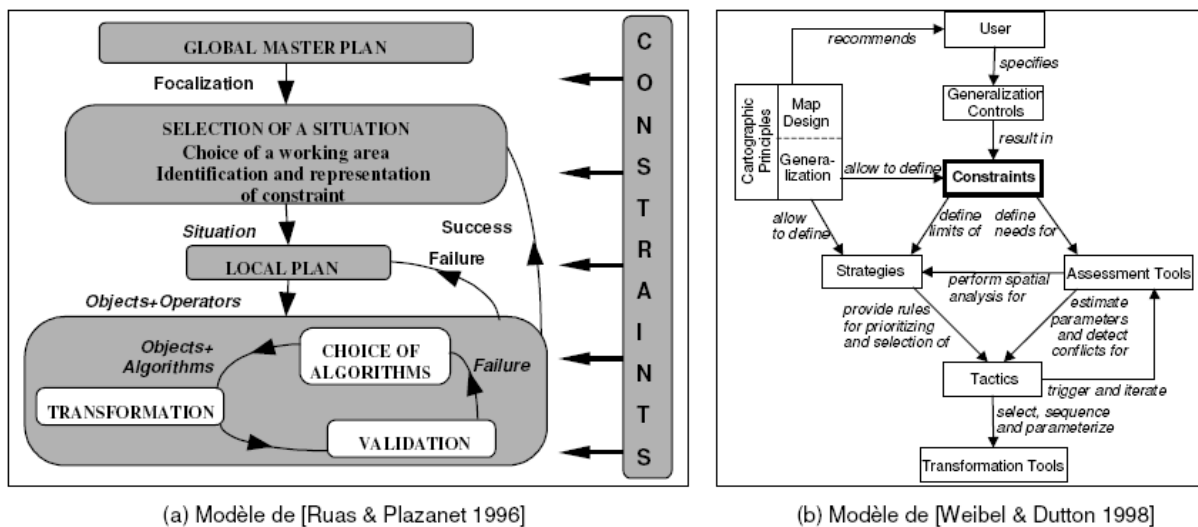


Figure A-21. Les modèles pas à pas locaux de (Ruas et Plazanet, 1996) et (Weibel et Dutton, 1998).

Le modèle AGENT est issu des travaux de (Ruas, 1999 ; 2000) repris dans le cadre du projet européen AGENT (Lamy *et al*, 1999 ; Barrault *et al*, 2001). Afin qu'ils se généralisent localement avec autonomie, les objets géographiques (bâtiments, routes par exemple) sont vus comme des agents. Les agents sont des entités autonomes capables de raisonnement et de prise de décision pour atteindre un but individuel et/ou collectif (voir les définitions précises en B.1.2). Dans AGENT, les agents cherchent à se généraliser en satisfaisant leurs contraintes de généralisation par application d'algorithmes. Le modèle est basé sur une structure hiérarchique d'agents : il existe des agents micros (par exemple un bâtiment ou une route), des agents meso composés d'agents micros (par exemple un îlot de bâtiments) ou d'autres agents meso (par exemple une ville composée d'îlots). AGENT est donc particulièrement adapté à la généralisation des villes dont la structure est bien hiérarchisée et à celle des routes de montagnes (Figure A-22) parce que l'approche hiérarchique correspond bien au principe de découpage des routes en parties homogènes comme dans GALBE, même si quelque que soit l'espace, il peut y avoir la nécessité d'une approche locale hiérarchique (un groupe de lacs par exemple dans la généralisation d'une couche de lacs). Mais AGENT n'est pas uniquement hiérarchique car à chaque étape, dans un niveau donné, l'agent se généralise individuellement par un moteur non hiérarchique (par exemple, un bâtiment peut se généraliser seul à partir de ses contraintes micro). Un tel modèle est paramétré par le choix des contraintes à affecter aux agents et le choix des valeurs à atteindre pour satisfaire les contraintes.

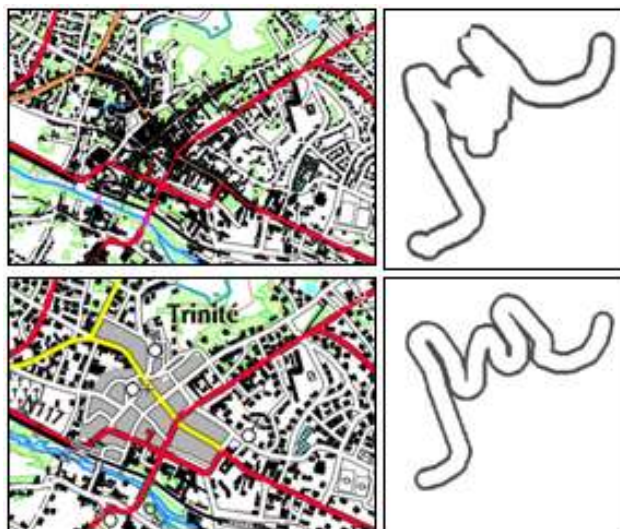


Figure A-22. Des résultats obtenus à l'aide du modèle AGENT sur une ville et une route de montagne.

Illustration du passage approche/modèle/processus/algorithmme

Dans le cas du modèle AGENT, le passage de modèle à processus est un peu différent que dans le cas du modèle de généralisation par moindres carrés car la spécialisation met en jeu le concept d'algorithmes de généralisation. Comme dans le cas des moindres carrés, il faut spécialiser les contraintes prises en compte par le processus en définissant dans ce cas des méthodes pour déterminer la valeur courante du caractère contraint (par exemple un algorithme calculant l'aire d'un bâtiment), la valeur but ou la satisfaction de la contrainte. Il faut également spécifier des connaissances de contrôle (Taillandier, 2008) qui permettent notamment de déterminer si l'état global d'un agent a été assez amélioré par une action pour valider l'état ou revenir en arrière. Il faut spécialiser également les types d'agents pouvant être manipulés par le processus, par exemple des agents bâtiment, route et îlot de bâtiments pour un processus dédié à la généralisation des zones urbaines. On peut ensuite déterminer les contraintes qui peuvent s'appliquer sur chacun des types d'agent. Il faut enfin spécialiser le modèle en désignant les algorithmes de généralisation pouvant être utilisés par chaque agent pour résoudre chacune de ses contraintes : par exemple, l'agent îlot de bâtiments peut utiliser l'algorithme de déplacement de (Ruas, 1998) pour tenter de satisfaire sa contrainte de congestion de ses bâtiments.

Un processus spécialisant le modèle AGENT a pour paramètres restant le choix des types d'objets géographiques qui sont des agents et un certain nombre de paramètres qui sont utilisés pour déterminer les valeurs but des contraintes comme l'aire minimale d'un bâtiment (en fonction des spécifications du produit final). On peut toutefois imaginer la spécialisation du modèle avec un processus dans lequel le choix des contraintes assignées aux agents est laissé libre comme paramètre supplémentaire du processus. Le processus spécialise moins le modèle : il est plus générique mais peut-être moins efficace car les choix faits pour l'assignation des contraintes par un utilisateur seront peut-être moins pertinents que ceux effectués par le développeur du processus, plus spécialiste du modèle.

Le modèle CartACom (Duchêne 2003 ; Duchêne 2004) est complémentaire au modèle AGENT car il est dédié aux conflits partagés entre deux objets géographiques et non aux relations hiérarchiques ; il

fonctionne par interactions transversales entre agents de même niveau (i.e. un bâtiment et une route) qui s'échangent des messages (i.e. « peux-tu te pousser ? ») pour résoudre les conflits partagés. Les relations entre deux objets comme la proximité des symboles, le positionnement relatif (par exemple, le bâtiment est au bout à droite de l'impasse) ou l'orientation relative sont contraintes sous la forme de contraintes relationnelles et les conflits partagés reviennent à des contraintes relationnelles non satisfaites. Le modèle CartACom est adapté à la généralisation des zones peu denses comme les zones rurales (Figure A-23). Les zones plus denses nécessiteraient de raisonner sur des entités de plus haut niveau comme dans AGENT pour résoudre certains conflits et génèreraient un trop grand nombre de dialogues dans CartACom. Son paramétrage repose sur le même principe qu'AGENT : il faut donner les valeurs permettant aux contraintes de calculer leur satisfaction et éventuellement désigner les contraintes à appliquer sur chaque type d'agent suivant le niveau de granularité du passage du modèle au processus.

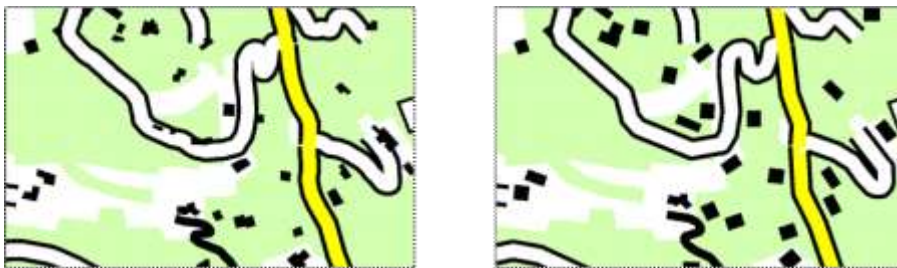


Figure A-23. Résultat d'une généralisation d'une zone rurale au 1/50000 par CartACom (Duchêne, 2004).

Dans (Jabeur et al, 2003 ; Jabeur et al 2006), les objets géographiques sont encore des agents, qui sont en compétition pour la place disponible sur la carte afin d'être lisibles. Ils sont plus ou moins forts suivant leurs caractéristiques et peuvent réaliser des coalitions. L'application proposée pour ce modèle est la génération de cartes avec une couche de symboles thématiques sur des données de référence car ce problème représente vraiment une forme de compétition entre les données de références (bâtiments, routes, etc.) et les données thématiques (hôtels, restaurants, magasins par exemple) ce qui n'est pas vraiment le cas pour les cartes topographiques où l'ensemble des données sont plutôt en coopération pour donner le sens géographique de la carte. Cependant, peu de résultats concrets sont présentés dans ces travaux. Le paramétrage est fait par la saisie de besoins utilisateurs sous une forme ad-hoc proposée dans (Jabeur et al 2006). On y définit des priorités d'affichage entre les couches thématiques, des tailles de symboles pour les données thématiques ainsi que des contraintes plus classiques pour la lisibilité des objets de la carte comme la distance minimum entre objets.

Le modèle GAEL (Gaffuri 2007 ; 2009 ; Gaffuri et al, 2008) est aussi basé sur un système multi-agents, dédié à la gestion des relations entre les objets géographiques discrets et les champs continus comme le relief (thème ayant une valeur en tout point de l'espace). Des contraintes objet-champ (par exemple « un bâtiment doit garder son altitude » ou « une rivière doit couler dans son talweg ») sont définies. Le champ est décomposé en agents (qui sont les points du champ triangulé) qui vont chercher un équilibre global du champ entre sa forme initiale et les contraintes objet-champ (Figure A-24). Pour réaliser cet équilibre, les contraintes objet-champ sont traduites en contraintes spatiales sur des objets 'submicros' du champ qui sont les segments, les triangles et les angles de la triangulation du champ. Des contraintes permettant au champ de maintenir au mieux sa forme initiale sont également définies sur les objets submicros. Toutes ces contraintes sur les objets

submicros sont ensuite traduites en forces sur les agents points qui vont chercher une position d'équilibre entre toutes leurs forces.

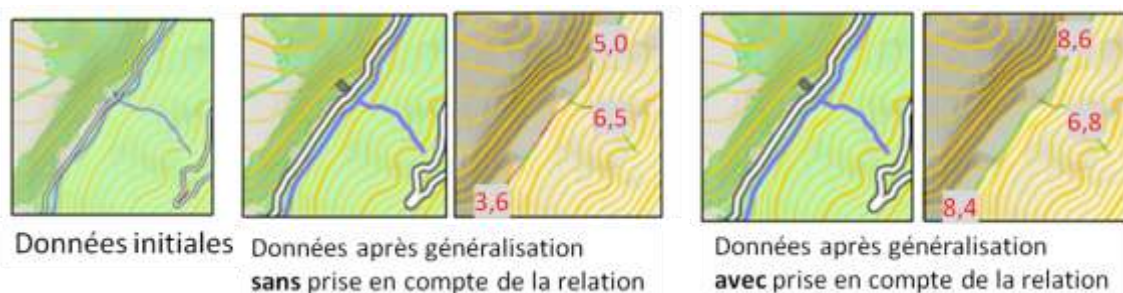


Figure A-24. Le relief est déformé grâce au modèle GAEL pour rétablir la relation entre la rivière et son talweg détériorée par la généralisation (Gaffuri, 2007). Les valeurs en rouge représentent la satisfaction de la contrainte objet-champ entre le relief et chaque tronçon de cours d'eau sur une échelle de 1 à 10.

D'autres modèles pas à pas locaux proposent de déterminer le choix et l'ordre des algorithmes de manière stochastique (en introduisant une part de hasard) : les connaissances procédurales qui permettent de guider un modèle pas à pas locale (choix du prochain algorithme, validation de la progression, décision d'arrêter la généralisation car on ne peut pas avoir mieux) sont remplacées partiellement ou totalement par un procédé stochastique. Contrairement aux modèles globaux présentés en A.2.3, les décisions stochastiques ne sont pas globales mais locales pour la généralisation de chaque objet ou groupe d'objets. (Neun et al, 2009) proposent trois méthodes de résolution stochastique pour choisir les opérateurs à appliquer dans un cas bien caractérisé par ses contraintes : le recuit simulé, un algorithme d'escalade (Russell et Norvig, 2006, p. 126) et une recherche génétique. La société ESRI propose avec l'Optimizer une approche similaire par recuit simulé (Hardy et al, 2006; Monnot et al, 2007). Le modèle de (Neun et al, 2009) a été spécialisé pour la généralisation des villes que l'on peut donc considérer comme son domaine d'application. C'est le cas également du modèle d'Optimizer, appliqué à la généralisation des villes dans (Punt et Watkins, 2010). Ici, il faut paramétrer en plus des contraintes, l'algorithme de recherche stochastique (gestion du refroidissement de la température du recuit simulé par exemple).

Enfin, (Burghardt et Neun, 2006) propose d'utiliser une méthode d'apprentissage par renforcement (le filtrage collaboratif) pour apprendre petit à petit quels algorithmes appliquer à un îlot caractérisé par un ensemble de mesures. Chaque généralisation contribue à améliorer les connaissances en attribuant une note au choix d'algorithme en fonction de l'évaluation du résultat de la généralisation. Les choix postérieurs d'algorithmes pour des îlots ayant les mêmes caractéristiques seront filtrés par cette note. Comme ce modèle s'améliore à chaque fois qu'il rencontre une situation similaire (la similarité est calculée par agrégation de mesures d'analyse spatiale), il est particulièrement adapté aux zones répétitives (structurées et fréquentes) comme les zones de lotissement. Comme souvent, les paramètres comptent les valeurs permettant d'évaluer les contraintes mais aussi dans ce cas des choix sur les algorithmes proposés par défaut et sur le mécanisme de filtrage collaboratif (à quel point on fait confiance au filtrage par rapport aux connaissances initiales). Une approche équivalente d'apprentissage de connaissances sur le choix des algorithmes a été proposée pour AGENT dans (Taillandier, 2008).



Figure A-25. Résultats obtenus par la méthode de filtrage collaboratif en utilisant des algorithmes disponibles par services web (Burghardt et Neun, 2006).

A.2.5. Les Processus spécialisés

Certains modèles de généralisation ne visent pas la résolution générique de différents problèmes de généralisation mais se concentrent sur des problèmes particuliers qui nécessitent des solutions adaptées. Dans cette partie, nous présentons d'abord des modèles de généralisation de couches d'occupation du sol, puis des modèles dédiés à la généralisation de réseaux et enfin des processus, trop spécifiques pour être qualifiés de modèles, dédiés à la généralisation d'un thème particulier (par exemple les points cotés, les archipels ou le trait de côte).

Modèles de généralisation de données catégorielles

Les données catégorielles dans les bases de données géographiques sont des partitions de l'espace pour lesquelles on affecte une valeur d'occupation à chaque élément de la partition. Il peut s'agir de couches d'occupation du sol mais aussi de couches de pédologie ou de géologie. Parmi les nombreux modèles existants, on peut distinguer les modèles :

- **Guidés par la géométrie** : les petits polygones sont agrégés à de plus gros, les fins sont élargis, etc. (van Oosterom, 1995).
- **Guidés par le schéma de données** : la généralisation est guidée par le schéma de données (par exemple les polygones de "tissu urbain continu" et de "tissu urbain discontinu" deviennent des "zones urbanisées" et sont agrégés quand ils sont voisins) (Le Men, 1996 ; Yaolin et al, 2003 ; Revell, 2007a ; Revell, 2007b). Il s'agit dans ce cas de généralisation méronymique.
- **Guidés par la fonctionnalité** : de nouveaux concepts sont créés par agrégation. Par exemple, un hôpital est l'agrégation de bâtiments et différentes autres surfaces (Ruas et Lagrange, 1995). (van Smaalen, 2003) propose un modèle pour l'agrégation par fonctionnalité. Contrairement au cas précédent, il s'agit de généralisation partonomique
- **Mixtes** : ces modèles sont à la fois guidés par la géométrie, le modèle et la fonctionnalité. On peut citer les travaux suivants (Galanda et Weibel, 2002 ; Galanda, 2003 ; Gao et al, 2004 ; Haurert, 2007a ; 2007b (Figure A-26)).

Suivant les modèles proposés, les paramètres peuvent être très différents : par exemple, une taxonomie de concepts pour les généralisations méronymiques, la taille ou la largeur minimum des

éléments de la partition ou un paramètre définissant une heuristique dans le modèle d'optimisation de (Hunert, 2007a).

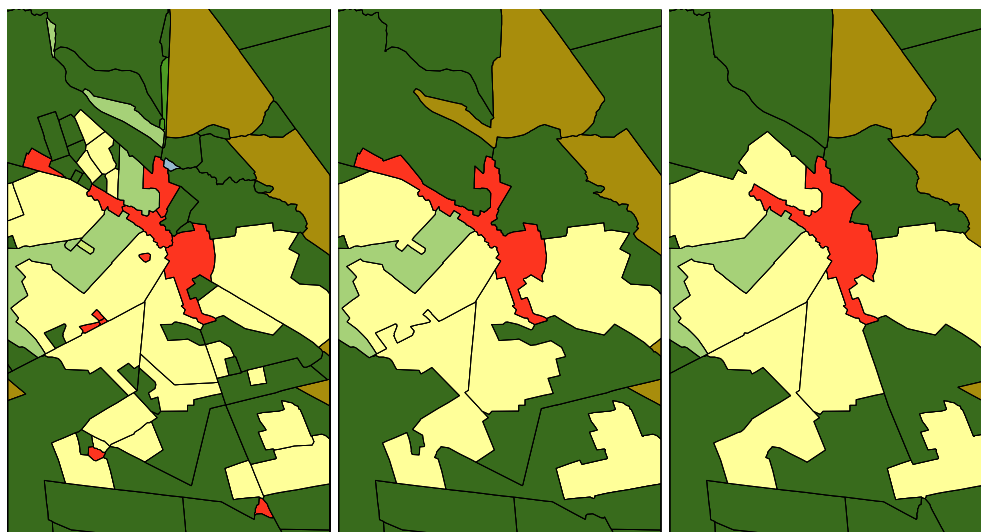


Figure A-26. Exemple d'agrégations mixtes par un modèle d'optimisation issu de l'intelligence artificielle (Hunert, 2007a).

Modèles de sélection de réseaux

La production d'une carte (BDC) à partir d'une base de données géographique (BDG) nécessite souvent la création d'une BDG intermédiaire par sélection avant la généralisation cartographique proprement dite afin de filtrer préalablement les objets affichés sur la carte (Weibel et Dutton, 1998). Certaines classifications d'opérateurs de généralisation considèrent la sélection comme une opération de simplification mais du fait de sa complexité et de la nécessité d'opérations contextuelles, notamment dans le cas des réseaux routiers, hydrographiques ou ferrés, la sélection peut être considérée plutôt comme un processus de généralisation dédié à une problématique. Les réseaux routiers et hydrographiques (Figure A-27) étant des thèmes topographiques fondamentaux, plusieurs modèles ont été proposés pour leur sélection automatique. On peut ainsi citer par exemple (Thomson et Richardson, 1999; Thomson et Brooks, 2007) qui repose sur l'identification de 'strokes', c'est-à-dire des ensembles de tronçons dont on perçoit qu'ils se suivent par continuité géométrique ; la sélection est alors effectuée plutôt sur les strokes que sur les simples tronçons de route ou de cours d'eau. Dans cet exemple, les paramètres utilisés sont la longueur des strokes et ainsi que le nombre de carrefours pour les routes et l'ordre de Horton (Horton, 1945) du stroke pour les cours d'eau. Un autre exemple propose un processus combinant l'utilisation des strokes avec des mesures issues de la théorie des graphes et avec la reconnaissance de structures particulières (Touya, 2007 ; Touya, 2010). Dans cet exemple, les paramètres sont un nombre important de valeurs numériques comme la longueur minimum des impasses ou la fréquentation minimum des routes en termes de plus courts chemins, mais également le choix des structures meso qui seront considérées dans la sélection. (Buttenfield et al, 2010) est un exemple récent dans lequel le processus de sélection du réseau hydrographique dépend de la structure géologique du réseau, qui est alors segmenté en zones homogènes géologiquement. Dans cet exemple, les paramètres sont ceux de la méthode de

segmentation et ceux des méthodes spécifiques de chaque zone géologique (comme la longueur minimale d'un cours d'eau ou des seuils de surface pour l'agrégation des marais dans les zones marécageuses).

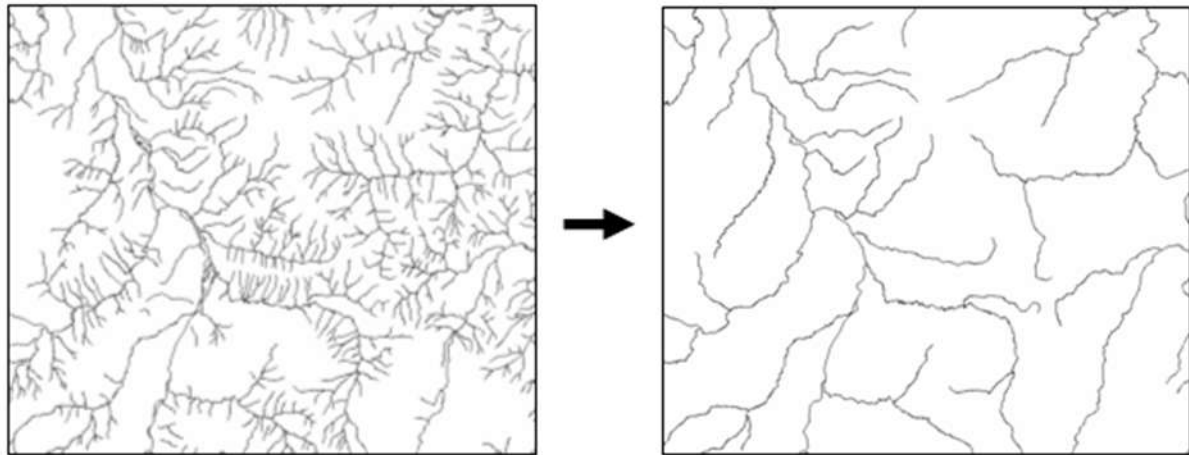


Figure A-27. Exemple de résultat de sélection du réseau hydrographique de la BD Topo® de l'IGN (Touya, 2007b).

Processus de généralisation d'un thème particulier

Il existe plusieurs autres processus de généralisation issus de recherches pour la généralisation d'un thème cartographique particulier. On peut citer par exemple (Baella et al, 2007) dédié à la généralisation des points cotés (Figure A-28), (Brosset et Devogele, 2005) qui s'intéressent aux courbes bathymétriques, (Steiniger et al, 2006) dédié à la généralisation d'archipels ou encore (Mackaness et al, 2007) qui s'intéresse aux forêts.

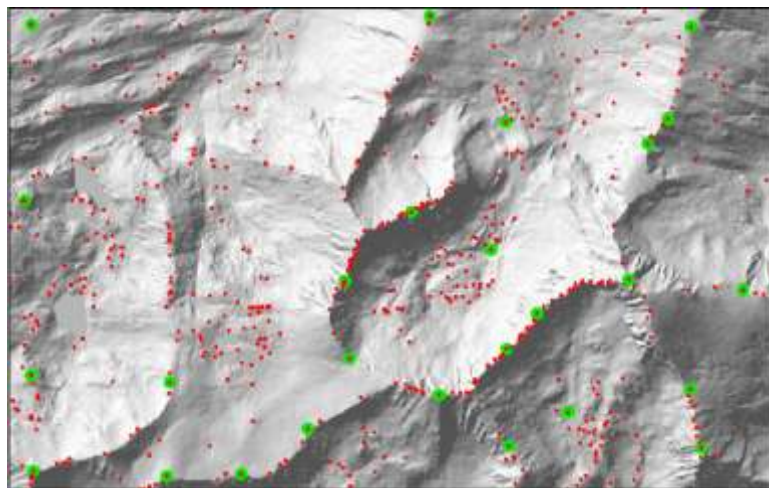


Figure A-28. Résultat de la sélection des points cotés, en rouge les points initiaux et en verts les points sélectionnés (Baella et al, 2007).

Pour conclure cet état de l'art des modèles et processus de généralisation, on peut noter qu'il en existe une grande diversité. Qu'ils soient plus ou moins efficaces, ils ne sont pleinement adaptés

qu'à une partie du problème de la généralisation, partie que nous pouvons décomposer en un paysage particulier (par exemple les villes), en un sous-ensemble des thèmes (par exemple les routes, l'occupation du sol) et en un sous-ensemble des types de conflits (par exemple la proximité entre bâtiment et route). On peut également noter la très grande diversité des types de paramétrage de ces processus : il n'existe pas de standard d'entrée pour les processus de généralisation.

A.3. OBJECTIFS DE LA THESE

Maintenant que nous avons présenté un état de l'art des processus automatiques de généralisation cartographique développés au cours des dernières années, nous pouvons exposer avec plus de précision les objectifs de notre travail de thèse. Nous allons donc présenter la problématique de cette thèse puis ses objectifs et enfin les hypothèses de départ de notre travail.

Problématique

Cette thèse est motivée par les conclusions que l'on peut tirer de l'état de l'art précédent : il existe de nombreux modèles et processus de généralisation et à eux tous, ils permettent de répondre à une grande diversité de problèmes de généralisation ; mais aucun n'est capable du fait de ses spécificités de résoudre tous les problèmes rencontrés lors de la généralisation d'une carte. En effet, les exemples de réalisation de chaînes de productions de cartes topographiques utilisant des outils de généralisation automatique (Baella et Pla, 2003 ; Lemarié 2003 ; Lecordix et al, 2007 ; Revell et al, 2007) montrent la nécessité de gérer une grande diversité de thèmes de données et de paysages (ville, campagne, montagne, littoral, etc.). Suivant les pays et les cultures, il existe une grande diversité des contenus (Hopfstock, 2007 ; Dukaczewski et al, 2007) et des représentations (Jolivet, 2009). Ainsi, le nombre de problèmes de généralisation spécifiques générés par cette diversité est presque infini : chercher à concevoir un nouveau modèle de généralisation indépendant des autres, parfaitement générique, et capable de résoudre tout problème nous paraît illusoire.

De plus, les processus de généralisation sont souvent assez lourds et ne peuvent pas toujours gérer la quantité de données correspondant à la généralisation d'une carte complète. Il est donc nécessaire de partitionner l'espace si on ambitionne de généraliser automatiquement des cartes sur de grands territoires (Chaudhry et Mackaness, 2008). Si nous voulons aller plus loin dans l'automatisation de la généralisation, nous considérons donc qu'il faut trouver un moyen pour utiliser au mieux les processus existants et par ailleurs, trouver un moyen pertinent pour partitionner l'espace de travail sur lequel appliquer les processus.

Objectif

Nous nous fixons pour objectif de proposer un nouveau modèle permettant de faire collaborer des processus de généralisation pour réaliser une généralisation complète optimale. Ce modèle doit permettre d'enchaîner des applications de processus sur des parties de l'espace adaptées à ce processus pour finalement couvrir au mieux tous les problèmes (Figure A-29). Il s'agit donc de répondre aux questions de recherche suivantes :

- Comment découper l'espace pertinemment vis-à-vis des processus disponibles ?
- Quel est le meilleur processus pour généraliser une partie de l'espace donnée à un moment donné du processus complet ?
- Que se passe-t-il aux limites des parties généralisées ?
- Comment assurer une homogénéité de traitement sur l'ensemble de l'espace ?
- Comment techniquement enchaîner deux processus aux entrées/sorties différentes ?
- Quelle est la séquence optimale d'applications de processus sur des espaces pour un problème de généralisation donné comme la généralisation de cartes topographiques au 1 : 50000 ?

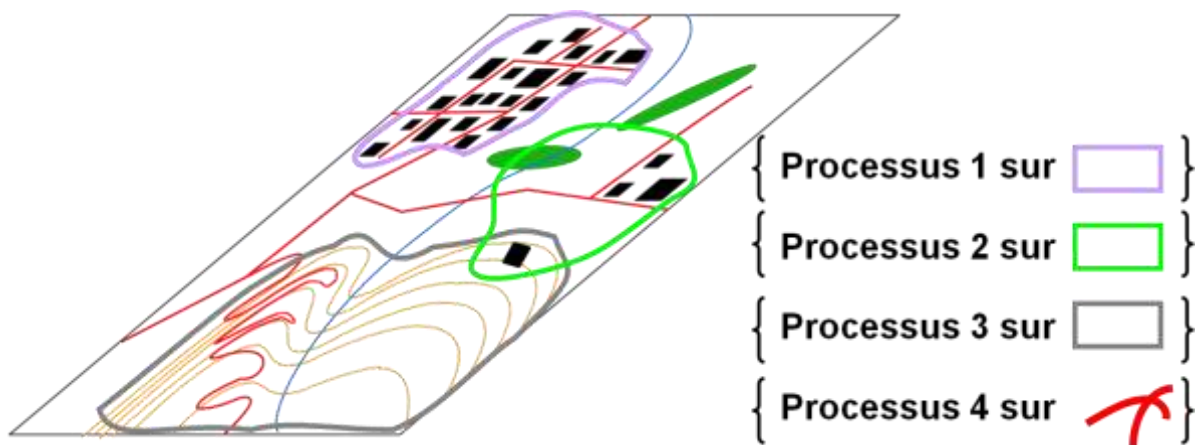


Figure A-29. Objectif de la thèse : permettre l'application orchestrée de différents processus sur les espaces adaptés à ces processus (le processus 1 sur la ville, le 2 sur la zone rurale, le 3 sur la zone de montagne et le 4 sur le réseau routier).

Un dernier objectif de notre travail est de s'insérer de manière cohérente dans les travaux du laboratoire qui ont mené au développement des modèles AGENT, CartACom et GAEL.

Hypothèses



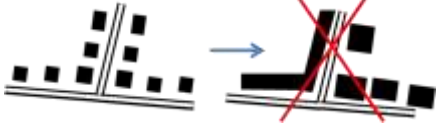
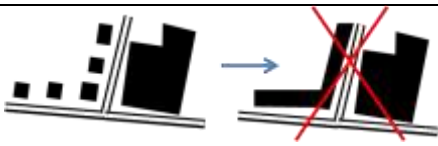

Dans le cadre de ce travail de thèse, nous nous limitons uniquement à la collaboration de processus disponibles sur une même plate-forme logicielle. Cela signifie que tous les processus utilisés doivent avoir une implémentation appelable par du code sur notre plate-forme de travail, le logiciel Radius Clarity™ développé par la société 1Spatial. Nous ne nous intéressons pas à toutes les questions d'appel de traitement à distance ou de service web. Ce serait une simple extension de notre travail sans que cela doive remettre en cause le modèle proposé.

Par ailleurs, nous limitons nos cas d'application à la réalisation de cartes topographiques de type séries IGN à partir de la base de données BDTopo® de l'IGN (résolution métrique, échelle de référence d'environ 1/15 000) mais le modèle proposé se veut plus générique.

Guide de lecture

Dans les prochains chapitres, beaucoup de nos exemples, de nos illustrations et de nos réflexions vont s'appuyer sur la connaissance implicite des spécifications d'une carte topographique de l'IGN. Afin de permettre au lecteur de suivre notre raisonnement dans ces chapitres, quelque soit son expérience des spécifications des cartes topographiques, nous proposons dans ce guide de lecture un récapitulatif des spécifications classiques que nous prenons en compte quand nous parlons de bonne ou de mauvaise généralisation. Ce tableau n'est bien sûr pas exhaustif mais présente les spécifications les plus importantes pour la suite de cette thèse.

| Description de la spécification | Thèmes concernés | Illustration |
|---|---|--------------|
| Les objets surfaciques doivent être d'une taille suffisante pour être lisibles (par exemple une aire de 0.2 mm ² carte pour les bâtiments) | Bâtiments, zones arborées, lacs | |
| Proximité : les symboles des objets doivent être séparés d'une distance minimale de lisibilité (par exemple 0.1 mm carte) | Bâtiments, routes, cours d'eau, voies ferrées | |
| Granularité : la taille du plus petit détail des objets doit être assez grande pour être lisible | Bâtiments, zones arborées, lacs, cours d'eau, routes | |
| Maintenir la forme générale des objets | Tous | |
| Maintenir au mieux la position initiale des objets | Tous | |
| Eviter l'empâtement des symboles linéaires | Routes, cours d'eau, voies ferrées, chemins | |
| Maintenir les relations d'orientation relative | Bâtiments, bâtiment et route, bâtiment et cours d'eau | |
| Maintenir les relations de positionnement relatif | Bâtiment et impasse, bâtiment et forêt, etc. | |
| Maintenir les relations de cohérence topographique | Bâtiments avec le relief, cours d'eau avec le relief... | |
| Maintenir les relations de partage topologique | Routes et forêts, limites communales et cours d'eau, etc. | |

| | | |
|---|---|---|
| Maintenir les structures morphologiques ayant un sens géographique | Alignements de bâtiments, formes du relief (crêtes, vallées...), |  |
| Maintenir la densité initiale dans un groupe d'objets | Groupes de bâtiments, de lacs | |
| Maintenir la répartition spatiale des objets dans un groupe | Groupes de bâtiments, de lacs, structures routières complexes |  |
| Maintenir et accentuer les ressemblances | Tous les thèmes dont les objets présentent des ressemblances de forme, de taille, de nature |  |
| Maintenir et accentuer les différences | Tous les thèmes dont les objets présentent des différences de forme, de taille, de nature |  |
| Ne pas changer le sens géographique de l'information cartographiée (changer des habitations pavillonnaires en immeubles collectifs par exemple) | Tous |  |

CHAPITRE B. UN CADRE FORMEL POUR LA GENERALISATION COLLABORATIVE

Dans le chapitre précédent, nous avons vu illustré la diversité des processus de généralisation et détaillé les objectifs de notre thèse qui consiste à faire collaborer des processus pour leur faire généraliser une carte complète. Ce chapitre présente notre proposition d'un cadre formel pour la généralisation collaborative. Ce cadre formel permet de concevoir des modèles permettant de d'orchestrer des processus de généralisation sur des parties de l'espace adaptées aux différents processus. La première partie de ce chapitre propose des analogies avec des domaines scientifiques connexes qui permettent de définir les concepts clés de notre proposition à savoir l'orchestration et la collaboration. Les parties suivantes de ce chapitre définissent et détaillent le cadre formel pour la généralisation collaborative que nous proposons dans cette thèse.

B.1. ORCHESTRATION ET COLLABORATION DE PROCESSUS

Nous avons vu dans le chapitre précédent que l'objectif de cette thèse est d'optimiser l'utilisation des nombreux processus de généralisation automatique existants en utilisant chacun uniquement sur la partie de l'espace pour laquelle il est adapté. Nous souhaitons donc orchestrer l'application de processus de généralisation. Si l'on se place du point de vue des processus, nous souhaitons faire collaborer les processus pour généraliser au mieux une carte. Dans cette partie, nous essayons de clarifier les termes d'orchestration et de collaboration et d'expliquer pourquoi nous les avons choisis, en nous appuyant sur deux domaines dans lesquels on les retrouve. Tout d'abord, nous allons voir comment sont définies les notions d'orchestration et de chorégraphie dans le domaine de la composition de services web. Puis, nous allons clarifier les notions de collaboration et de coopération dans le domaine des systèmes multi-agent. Enfin, nous analyserons les travaux qui ont abordé ce problème de collaboration de processus en généralisation cartographique.

B.1.1. Orchestration et chorégraphie dans les services Web

Nous considérons dans cette partie une analogie entre les objectifs de notre thèse et la composition automatique de services Web car dans les deux cas, il s'agit de résoudre une portion du problème global en faisant appel de manière orchestrée à des processus de portée plus locale. Selon (W3C,

2004) un service est « une ressource abstraite capable de réaliser une tâche qui représente une fonction cohérente ». Cette définition de service est floue car elle se situe entre les notions de *logiciel* et d'*utilisation* (Bucher et Jolivet, 2008). Toutefois, nous allons surtout nous attacher ici à la notion de service web que nous pouvons définir comme un composant logiciel distribué qui fournit des services machine-machine interopérables à travers le Web (Haas et Brown, 2004 ; Lemmens, 2006, p. 22). Un exemple classique de service Web est celui de la météo fourni par Météo France pour des applications qui utilisent des données météorologiques comme paramètre.

Les services Web sont dédiés à l'utilisation du service par une application donc on peut vouloir utiliser de plusieurs services différents pour bâtir une application, c'est ce que l'on appelle la composition de services Web. Si nous comparons nos processus de généralisation à des services Web, notre problème peut être vu par analogie comme un problème de composition de services. Il existe principalement deux méthodes pour réaliser de manière automatique une composition de services Web, l'*orchestration* et la *chorégraphie*.

– L'*orchestration* de services décrit la manière par laquelle les services web peuvent s'enchaîner par un canevas prédéfini incluant l'ordre d'exécution des interactions en services (Peltz, 2003 ; Jamal, 2005 ; Sabouret, 2010). Dans le cas de l'orchestration, un des services prend en charge la création du canevas de composition, les autres n'étant que fournisseurs ; la communication se fait uniquement entre l'entité décideuse et les fournisseurs et pas entre les fournisseurs ; la prise de décision est centralisée et elle concerne à la fois le choix du service et l'enchaînement des services. Donc, si on se place dans le cadre de cette thèse, nous pouvons parler d'orchestration de processus de généralisation si l'on adopte le point de vue d'une entité du système qui décide de façon centralisée de l'ordre d'application des processus sans que ceux-ci puissent communiquer entre eux (Figure B-1).

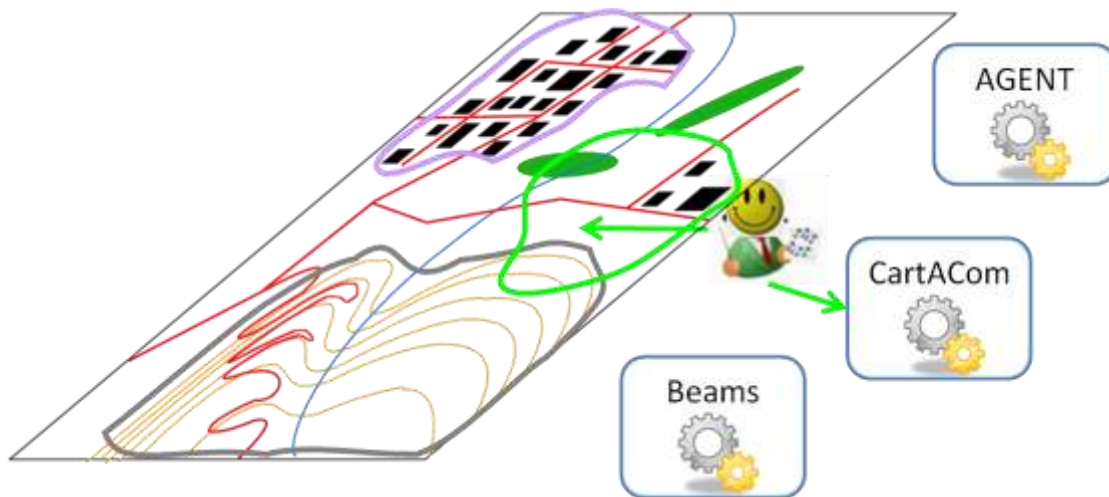


Figure B-1. Vision orientée orchestration de notre objectif : une entité chef d'orchestre choisit les couples (processus, espace) (ici CartACom sur une zone rurale) et l'enchaînement de ces couples.

– La chorégraphie décrit une intelligence plus collective, chaque service impliqué décrivant son rôle dans les interactions entre services pour permettre une décision plus collaborative (Peltz, 2003 ; Sabouret, 2010). La chorégraphie se traduit principalement en la définition de standards de communication pair-à-pair entre services Web. Ainsi, le langage de communication WSCI a été développé pour permettre la chorégraphie de services Web (Jamal, 2005). Si l'on reprend les mêmes

critères de description que pour l'orchestration, pour la chorégraphie, tous les acteurs sont au même niveau en tant que fournisseurs de service ; les communications s'effectuent de manière pair-à-pair entre les services ; la prise de décision pour le choix du service à utiliser dans un cas est collaborative, chaque service jouant un rôle dans la décision ; la méthode pour déterminer de l'enchaînement des prises de décision sur le choix n'est pas définie, elle peut être de différents types et même centralisée. Dans le cadre de cette thèse, on pourrait parler de chorégraphie si les processus communiquaient entre eux pour décider qui s'applique sur quelle partie de l'espace, quelque soit la manière de choisir et d'enchaîner ces parties de l'espace (centralisée ou distribuée) (Figure B-2).

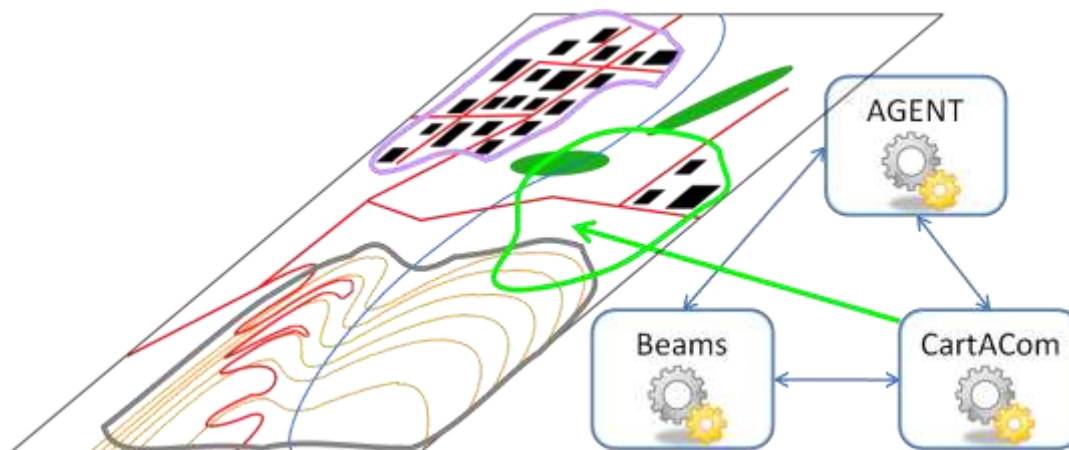


Figure B-2. Vision orientée chorégraphie de notre objectif : les processus interagissent pour prendre la décision du prochain processus à appliquer sur une partie de l'espace.

Par analogie avec le problème de la composition de services Web, nous choisissons d'utiliser la notion d'orchestration de processus car il nous paraît plus simple d'utiliser une entité *chef d'orchestre* plutôt que de permettre la communication chorégraphique entre les processus de généralisation. De plus, la généralisation étant souvent vue comme un processus holistique, il nous paraît plus judicieux de gérer la prise de décision et l'enchaînement des couples (espace, processus) de manière plus centralisée, avec une vue plus complète des problèmes.

B.1.2. Collaboration et coopération dans les systèmes multi-agents

Nous considérons dans cette partie une analogie entre les objectifs de notre thèse et la coopération dans les systèmes multi-agents car dans les deux cas, il s'agit de résoudre un problème global par des entités dont l'objectif individuel est un sous-ensemble des objectifs collectifs. Le concept d'agent et de systèmes multi-agents provient du domaine de l'intelligence artificielle. Il n'existe pas de réel consensus sur la définition du concept d'agent mais nous considérons, en synthétisant différentes définitions, qu'un agent est une entité informatique autonome, capable de percevoir son environnement, de communiquer, de réflexion et de prise de décision pour atteindre son objectif. Plusieurs définitions peuvent être trouvées dans (Ferber, 1995 ; Wooldridge, 1999 ; Briot et Demazeau, 2001). Un système multi-agents (SMA) est alors un système dans lequel évoluent et interagissent plusieurs agents (Weiss 1999).

Les SMA situés (i.e. l'environnement du SMA possède un système de référence permettant de situer les agents par des coordonnées), ont deux grandes familles d'applications : la résolution de problème et la simulation. On peut citer quelques exemples d'applications des SMA à la résolution de problèmes : gagner une partie d'échecs (Campbell, 2002), la construction d'une ontologie (Sellami et al, 2010) ou comme nous l'avons vu précédemment la généralisation cartographique (Barrault et al, 2001, Duchêne, 2003). Dans les applications de simulation, on peut noter la simulation de la consommation énergétique (Gil-Quijano et al, 2010), du comportement des fourmis (Drogoul et Ferber, 1994) ou plus proche de notre domaine, la simulation de peuplement urbain (Sanders et al, 1997) et la simulation de l'évolution urbaine (Perret et al, 2009). La généralisation cartographique fait donc partie des applications déjà connues des SMA.

Au travers de ces applications, on voit qu'à l'intérieur d'un système multi-agents, les agents peuvent interagir. Suivant les objectifs et la modélisation du SMA, les agents peuvent *coexister*, être en *concurrence* ou *coopérer* (Chaib-Draa et al, 2001). Si l'on se replace dans le contexte de notre thèse et de ses objectifs, seul le cas de la *coopération* va nous intéresser. Il y a coopération quand les buts des agents sont compatibles et dépendants les uns des autres (Ferber, 1995) ce qui est le cas pour nous car par analogie, les processus ont pour but de bien généraliser la partie de la carte qui leur est assignée. Parmi les types de comportements coopératifs des agents, (El Fallah Segrouchni, 2001) distingue en parlant plutôt de coordination que de coopération :

- la *coordination centralisée* qui se rapproche l'orchestration de services Web car la coordination repose sur l'existence d'un agent central coordinateur.
- La *coordination distribuée* qui se rapproche de la chorégraphie de services Web car les agents cherchent à trouver un équilibre ou un compromis entre leurs propres intérêts.

De plus, (Weiss, 1999) définit la *collaboration* entre agents comme une coopération de haut niveau qui nécessite une compréhension mutuelle et une vue partagée du problème à résoudre.

Si l'on se place dans le contexte de notre thèse, nous souhaitons généraliser une carte complète en utilisant différents processus sur des sous-parties de la carte adaptées aux capacités des processus qui couvrent l'ensemble des problèmes de généralisation de la carte. Si l'on assimile les processus à des agents, ils possèdent un objectif commun qui est de généraliser correctement l'ensemble de la carte sans pouvoir le réaliser individuellement. Par ailleurs, nous avons vu dans la partie précédente que l'orchestration était une bonne analogie pour modéliser la coopération des processus et l'entité centralisée réalisant l'orchestration permet à l'ensemble des processus d'avoir une compréhension mutuelle centralisée du problème.

Par analogie avec la définition de la collaboration par (Weiss, 1999) nous proposons donc d'identifier notre objectif à celui de faire collaborer des processus automatiques de généralisation par une orchestration centralisée.

B.1.3. L'Orchestration et la collaboration en généralisation cartographique

L'orchestration n'est pas un terme nouveau en généralisation cartographique car il a depuis longtemps été utilisé pour décrire le procédé d'enchaîner les algorithmes de généralisation. Il s'agit

même d'un des mots clés les plus utilisés dans l'histoire des ateliers organisés annuellement par la commission sur la généralisation de l'Association de Cartographie Internationale (Burghardt, 2010). Si ces travaux traitent principalement d'orchestration d'algorithmes pour concevoir un processus automatique de généralisation, (Regnauld, 2007) propose un système générique reposant sur l'orchestration de services Web de généralisation fournissant des algorithmes de mesure, de généralisation mais aussi des processus tels que nous les avons définis dans le chapitre précédent (Figure B-3). Le contexte de cette architecture est la cartographie « à la carte » (i.e. en réponse à des demandes non standardisées d'un utilisateur). Le système proposé repose sur l'utilisation de trois ontologies, une ontologie étant un ensemble organisé de concepts, propriétés et relations partagés au sein d'un domaine restreint (voir C.4.1 pour une définition plus précise). Le système générique d'orchestration doit remplir notamment quatre fonctions importantes qui sont : l'extraction des données (i.e. aller chercher dans les bonnes BDG les données initiales de la généralisation), le raisonnement (i.e. être capable d'analyser les informations reçues des trois ontologies pour réaliser l'orchestration), la sélection des services (i.e. le choix du service à appliquer à un moment donné, sur certaines données) et enfin la traçabilité (i.e. en plus de la carte, le système doit rendre des comptes sur les choix qu'il a effectué). L'architecture proposée met aussi en avant la difficulté fondamentale de notre travail, qui est celle de l'interopérabilité des processus (des services dans son cas).

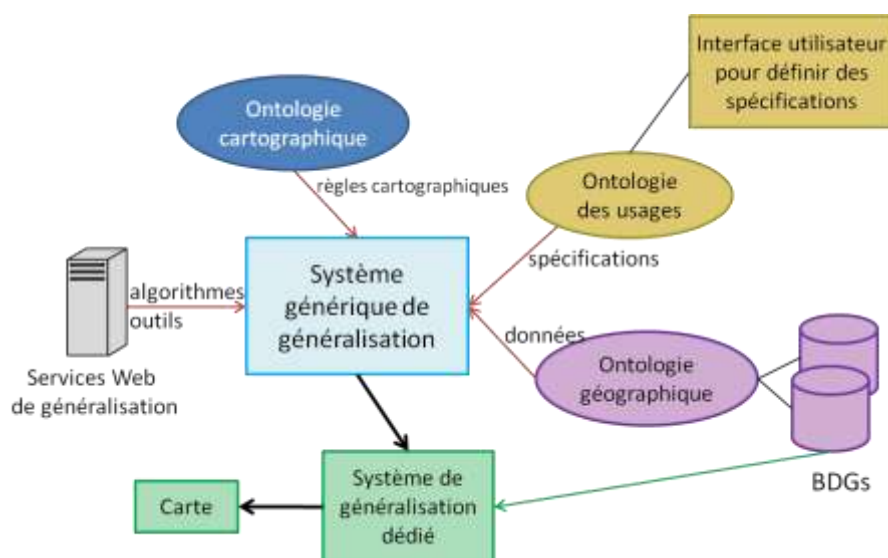


Figure B-3. Architecture du système permettant l'orchestration de services de généralisation proposé par (Regnauld, 2007).

Par ailleurs, (Duchêne et Gaffuri, 2008) proposent des pistes pour coupler les modèles de généralisations AGENT, CartACom et GAEL. Le premier scénario proposé pour réaliser ce couplage repose sur l'application séparée des processus issus de ces modèles : AGENT sur les zones urbaines, CartACom sur les zones rurales (complémentaires des zones urbaines) et GAEL sur les données de relief. Ils ont par ailleurs proposé d'intégrer cette application séparée des trois modèles à un processus plus global de type « Global Master Plan » de (Ruas et Plazanet, 1996) qui proposait une planification globale du type « 1. Sélection dans tous les thèmes 2. Simplification des réseaux 3. Traitement des zones denses ... ». Il s'agit d'une orchestration prédéfinie pour des processus connus, mais l'objectif de cette thèse va plus loin en ne se limitant pas à ces trois processus : le modèle collaboratif à développer doit pouvoir orchestrer n'importe quel type de processus de généralisation.

B.2. DEFINITION DE LA GENERALISATION COLLABORATIVE

A partir de l'état de l'art présenté dans les trois parties précédentes, nous proposons une définition de la généralisation collaborative comme le cadre générique pouvant répondre aux objectifs de notre thèse :

La *Généralisation Collaborative* est un cadre générique dans lequel la généralisation est effectuée par la collaboration de différents processus de généralisation appliqués sur des parties de l'espace pour lesquelles ils sont adaptés, appelées espaces géographiques. La Généralisation Collaborative est capable de gérer les effets de bord à la limite de ces espaces géographiques. Le choix des processus de généralisation appliqués se fait parmi ceux qui sont disponibles pour le système collaboratif (i.e. qui peuvent être exécutés automatiquement, d'une manière ou d'une autre sur les données généralisées). En extension, du modèle de (McMaster et Shea, 1988 ; Shea et McMaster, 1989), un modèle s'inscrivant dans le cadre générique de la Généralisation Collaborative apporte des réponses automatiques aux questions « Où, quand et pourquoi généraliser avec quel processus ? ».

Le principe de la Généralisation Collaborative est le suivant :

- Le système découpe la base de données initiale en espaces géographiques en fonction des processus de généralisation que l'on peut utiliser.
- La collaboration permet de déclencher itérativement des applications adaptées d'un processus sur un espace géographique.
- Le système vérifie si la généralisation de l'espace n'a pas engendré d'effets de bord (i.e. un conflit cartographique à la limite de l'espace qui n'existait pas auparavant). Si c'est le cas, le système essaie de corriger ces effets de bord.

Dans la suite de ce chapitre, nous allons décrire plus précisément le cadre générique de la Généralisation Collaborative en s'attachant (1) à ses composants principaux, (2) à la définition de cet espace de travail particulier qu'est l'espace géographique et enfin (3) à sa dynamique. A partir de maintenant, le mot espace fera toujours référence à une unité de découpage pertinente pour y appliquer un processus de généralisation et ne fera plus référence à l'ensemble de l'espace représenté par les données géographiques.

B.3. ANALYSE FONCTIONNELLE DE LA GENERALISATION COLLABORATIVE

Dans cette partie, nous exposons une analyse fonctionnelle de la Généralisation Collaborative. Ainsi, nous allons identifier les fonctions impliquées par un tel cadre et en déduire à chaque fois les besoins en composants spécifiques et en ressources. Nous définissons une *ressource* comme un ensemble d'éléments requis qui peuvent être considérés comme une entrée de la généralisation car ils sont utilisés durant la généralisation collaborative ou la guident. Un *composant* est un élément logiciel qui

agit durant la généralisation collaborative pour fournir un service précis et qui utilise ou crée des ressources. Nous considérons que la fonction principale est la généralisation complète d'une carte et nous allons présenter dans cette partie les fonctions contraintes (i.e. qui dérivent de la définition de la généralisation collaborative) dans un premier temps puis les fonctions complémentaires (i.e. qui facilitent ou améliorent la fonction principale) de la généralisation collaborative.

Découper l'espace avec pertinence

La première fonction contrainte dans un système de généralisation collaborative est celle du découpage des données en espaces géographiques pertinents. Comme le montrent la Figure B-4 et la Figure B-5, il existe plusieurs possibilités pour ce découpage. Il faut d'un côté choisir les types d'espaces que le système souhaite identifier : contrairement au cas (a), le système ne cherche pas (ou ne trouve pas) d'espace périurbain dans le cas (b). D'un autre côté, il faut déterminer, pour chacun des types d'espaces à identifier (ici urbain, périurbain, rural), une méthode pour les délimiter.

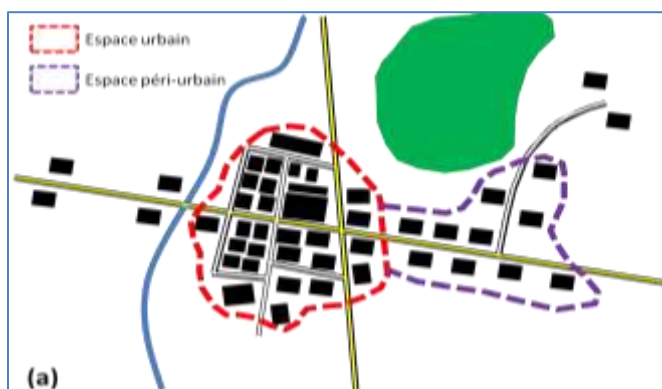


Figure B-4. Un exemple de découpage en espaces géographiques avec un espace urbain, un espace périurbain et un espace rural autour.

Cette fonction nécessite l'existence d'un premier composant, le composant de **d'enrichissement géographique** qui utilise une bibliothèque d'algorithmes d'analyse spatiale pour identifier et délimiter les espaces géographiques. Une première ressource est également indispensable, il s'agit de connaissances permettant de choisir dans chaque cas les types d'espaces géographiques que le composant d'enrichissement géographique va délimiter.

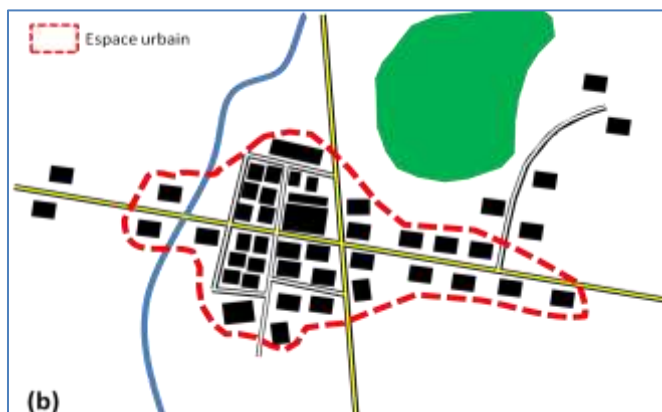


Figure B-5. Autre exemple de découpage en espaces géographiques avec un espace urbain plus étendue et juste un espace rural autour.

Orchestrer les généralisations

La deuxième fonction contrainte de la généralisation est d'orchestrer les généralisations d'un espace géographique par un processus. Cette fonction d'orchestration doit permettre de choisir des couples pertinents (espace géographique, processus de généralisation) et de les séquencer. En ce qui concerne le choix des couples pertinents, il existe trois méthodes pour les associer :

- On essaie de déterminer le meilleur couple à un moment donné.
- On cherche l'espace le plus adapté à un processus donné à un moment donné.
- On cherche le processus le plus adapté à un espace donné à un moment donné.

Si la première méthode prend en charge en plus le séquencement, les deux autres nécessitent un mécanisme complémentaire pour séquencer les espaces ou les processus. Dans le cadre de la généralisation collaborative, la troisième solution nous paraît la plus intéressante. Nous écartons tout d'abord la première solution car plus complexe que les deux autres : décider du séquencement et du couple (espace, processus) pertinent en même temps mettrait en jeu trop de variables pour que la prise de décision soit efficace. Nous écartons enfin la deuxième (partir d'un processus et chercher l'espace adapté) en guidant notre choix par les objectifs d'une généralisation. En effet, dans l'approche pas à pas locale dans laquelle nous inscrivons le cadre générique de généralisation collaborative, la généralisation est guidée localement par les conflits cartographiques restants. Nous considérons donc plus logique de partir d'un espace géographique à généraliser (car des conflits ont été identifiés dans l'espace) pour trouver le processus adapté que l'inverse. De plus, une situation dans laquelle plusieurs processus peuvent être adaptés à un même type d'espace peut arriver dans ce cadre : par exemple, AGENT (Barrault et al, 2001), le modèle stochastique de (Neun et al, 2009) et la généralisation globale par algorithme génétique (Wilson et al, 2003) sont trois modèles *a priori* adaptés à la généralisation des zones urbaines. Si des processus issus de ces modèles sont disponibles dans une généralisation collaborative, il est plus efficace de choisir le processus le plus adapté des trois à la généralisation d'une zone urbaine donnée. Ce qui compte dans notre orchestration, c'est que chaque espace soit généralisé et non que chaque processus soit utilisé. Toutefois, nous ne nous interdisons pas d'avoir recours ponctuellement à la deuxième méthode dans des cas où il est plus naturel de désigner le prochain processus à être utilisé puis chercher l'espace adapté.

Afin d'assurer cette fonction de choix d'un processus, un système de généralisation collaborative doit posséder un composant que nous appelons **registre** par analogie avec un annuaire des Pages Jaunes car il va recenser et répertorier les processus de généralisations et leurs capacités. De plus, ce composant devra, telle la version moderne des Pages Jaunes et son moteur de recherche internet, fournir la liste ordonnée des processus les plus adaptés à la généralisation d'un espace donné. Un tel composant nécessite également l'exploitation d'un nouveau type de ressource, la description formelle des capacités des processus de généralisation (de la même manière qu'un plombier décrit ses activités aux pages jaunes pour être répertorié) pour permettre la réponse du registre.

Puisque nous avons décidé de ne pas retenir la première méthode de choix, la partie séquencement de l'orchestration doit être assignée à une fonction complémentaire qui est d'automatiquement déterminer l'espace géographique à traiter à un moment donné du processus. Par exemple, après avoir généralisé une zone urbaine, doit-on généraliser une autre zone urbaine, le réseau routier ou

les zones périurbaines ? La Figure B-6 montre deux exemples pour lesquels le choix de l'ordre de généralisation des espaces donne des résultats très différents. Dans le premier exemple de la figure, on a le choix entre généraliser d'abord l'espace périurbain ou le réseau routier : si l'on généralise d'abord l'espace périurbain en éloignant les bâtiments de l'impasse avec CartACom par exemple, généraliser ensuite le réseau routier avec les Beams (qui déplacent l'impasse en conflit), recrée des conflits. Ces conflits n'apparaissent pas si on inverse les deux espaces. Dans le deuxième exemple, inverser l'ordre n'engendre pas de conflit mais donne des résultats différents. En effet, si l'on généralise avec AGENT l'espace urbain avant la sélection du réseau routier, les îlots concernés sont très denses et transformés en zone de centre de ville. Si l'on fait la sélection d'abord, les rues concernées sont éliminées et la densité diminue ; les îlots sont donc généralisés par élimination, grossissement et déplacement, générant un résultat bien différent ! Suivant les spécifications choisies, un des deux résultats peut d'ailleurs s'avérer meilleur que l'autre.

Pour obtenir un séquençement efficace et éviter les problèmes illustrés en Figure B-6, une fonction d'orchestration de la généralisation des espaces géographiques est donc indispensable pour permettre une généralisation optimale. Nous proposons d'introduire un deuxième composant pour remplir cette fonction d'orchestration en plus du registre, que nous appelons composant d'**orchestration** car le service rendu est assez différent de celui du registre. Ce composant permet de choisir quel espace il faut généraliser à un moment donné (le réseau routier après une zone urbaine par exemple) et éventuellement quel processus utiliser à un moment donné. Afin de guider un tel composant dans ses choix, des connaissances, sous la forme de *règles d'orchestration formelles*, sont nécessaires et s'ajoutent aux ressources de la généralisation collaborative.

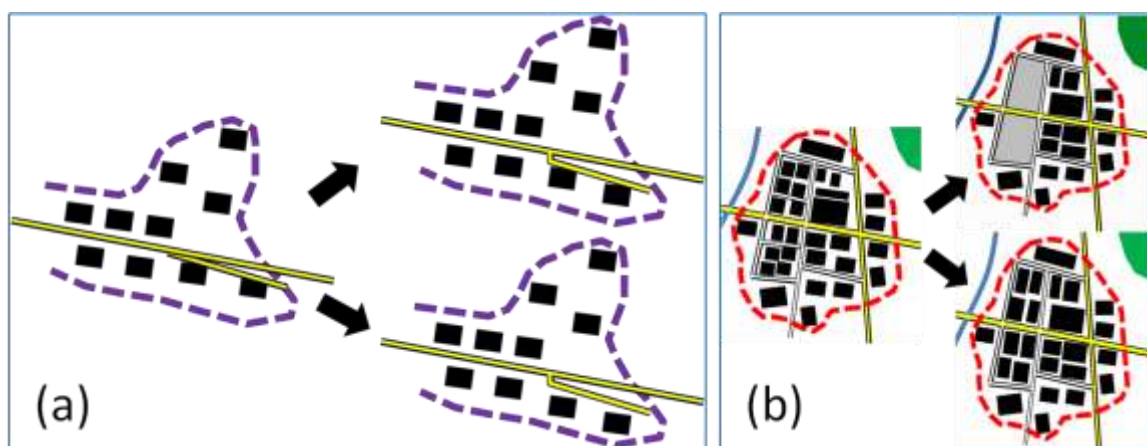


Figure B-6. Importance de l'ordre des généralisations d'espaces : (a) l'espace périurbain a été généralisé avant le réseau routier (des conflits restent) et dans le deuxième cas, c'est l'inverse ; (b) dans le premier cas, la ville a été généralisée avant le réseau routier et dans le deuxième cas c'est l'inverse (ici le résultat est différent car la densité a baissé après l'élimination des petites rues).

Gérer l'interopérabilité des processus

(Brodie, 1992) définit l'interopérabilité comme "la capacité de deux composants logiciels (i.e. services) à interagir effectivement durant leur exécution pour atteindre des objectifs partagés, soit, une activité commune. L'inter opération de deux services X et Y nécessite que X soit capable de procéder à une requête R à Y en se basant sur une compréhension mutuelle de R et que Y soit capable de proposer des réponses S à X en se basant sur une compréhension mutuelle de S comme

des réponses à R." (Lemmens, 2006) précise que les problèmes d'interopérabilité émergent quand il existe de l'hétérogénéité dans l'information contenue dans les données ou les processus. (Stuckenschmidt et van Harmelen, 2005) classe l'hétérogénéité en trois niveaux :

- hétérogénéité syntaxique (différences de formats de données)
- hétérogénéité syntactique (différences de modélisation des concepts)
- hétérogénéité sémantique (différence sur le sens et le nom des concepts)

La Figure B-7 montre les problèmes d'hétérogénéité qui se posent au niveau des paramètres dans la généralisation collaborative. On peut dire qu'il s'agit à la fois d'hétérogénéité syntaxique (format des paramètres), sémantique (sémantique des paramètres, définir « la taille d'un bâtiment » par exemple) et surtout syntactique. Pour illustrer la figure avec des exemples réels, un rond peut être un ensemble de contraintes dans le formalisme d'AGENT, le carré un système d'équations pour faire des Moindres Carrés (Harrie et Sarjaskoski, 2002) et le triangle peut être une séquence d'algorithmes et des paramètres numériques pour un processus par séquence prédéfinie (Lee, 2003). Dans cet exemple, il y a de l'hétérogénéité syntaxique dans les unités utilisées (des millimètres sur la carte pour définir des distances minimales dans AGENT et des mètres terrain pour les moindres carrés). Il y a également de l'hétérogénéité sémantique : AGENT manipule des « *building_agents* », les moindres carrés des « objets rigides » et le processus par séquence prédéfinie une classe d'objets « bâtiments » qui concernent tous le même concept de bâtiment modélisé différemment. Il y a enfin de l'hétérogénéité syntactique, par exemple dans la définition de la position initiale des objets qui correspond aux coordonnées de chaque point de la géométrie pour les moindres carrés et aux coordonnées du centre de gravité pour AGENT.

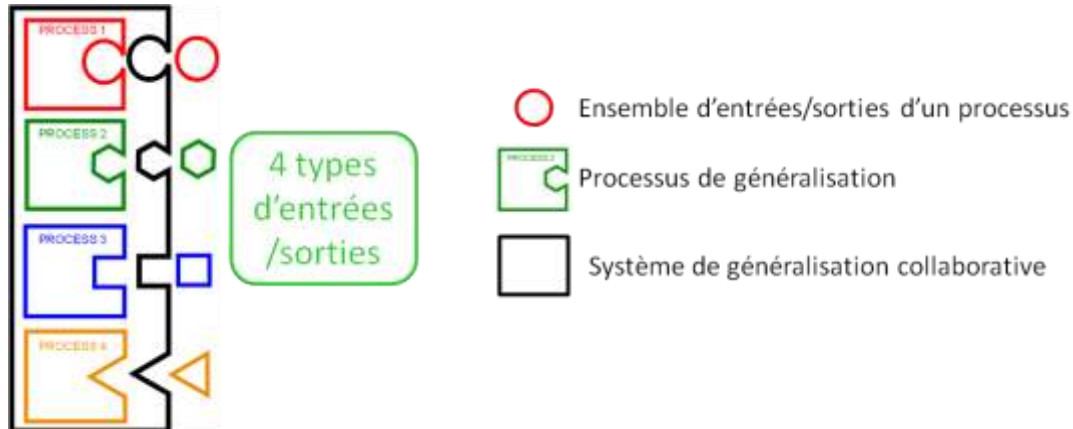


Figure B-7. Un système collaboratif avec des paramètres hétérogènes : il doit savoir gérer les paramètres au format rond pour utiliser les processus 1 et ainsi de suite.

La gestion d'un système comme celui de la Figure B-7 où de nombreux formalismes de paramètres sont maintenus est très compliquée en termes d'utilisations et d'évolutions. Une solution serait d'imposer un seul formalisme de paramètres pour les processus de généralisation mais cela nécessiterait de redévelopper l'ensemble des processus n'utilisant pas ce formalisme. La solution que nous proposons, inspirée des solutions proposées par (Lemmens, 2006) dans la composition de services Web, est d'utiliser un format pivot qui sert d'entrée unique et qui sera transformé à la volée dans le format spécifique d'un processus (Figure B-8).

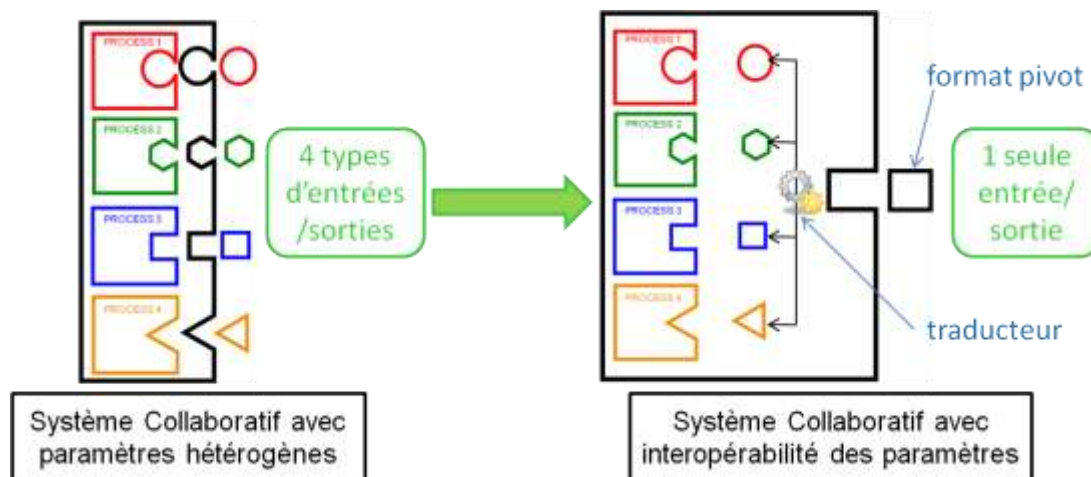


Figure B-8. Différences entre un système interopérable et un autre non.

Comme l'illustre la Figure B-8, la gestion de l'hétérogénéité syntaxique dans un système de généralisation collaborative pour aller vers des processus interopérables passe par le choix d'un format pivot. La conséquence de l'utilisation d'un format pivot est la nécessité d'un composant de **traduction** du type de paramètre pivot vers les types spécifiques des différents processus. Ce format pivot et son composant de traduction permettent aussi de résoudre les problèmes d'interopérabilité syntaxique (un seul format est manipulé).

D'autre part, La gestion de l'interopérabilité sémantique nécessite un traitement à part entière. Dans ce cas là, il est communément admis que le recours à une *ontologie du domaine de la généralisation* comme ressource supplémentaire est une bonne solution. (Regnauld, 2007 ; Neun et al, 2008) proposent cette solution pour résoudre les problèmes d'interopérabilité sémantique dans le cadre de la généralisation.

Détecter et corriger les effets de bord

Découper les données en espaces pour y appliquer de façon limitée (dans l'espace géographique et dans l'espace des thèmes) un processus de généralisation pose la question de ce qui va se passer autour de ces limites. Que les espaces géographiques se touchent (un espace rural complétement ensembliste d'un espace urbain), s'intersectent (comme le réseau routier qui traverse une zone urbaine) ou non, la généralisation des objets proches de ou formant cette limite peut engendrer des effets de bord, notamment avec les objets situés juste à l'extérieur de l'espace qui vient d'être généralisé. Pour illustrer ce problème, nous avons recensé quatre situations types qui engendrent des effets de bord après la généralisation d'espaces voisins (Figure B-9).

Déplacer des objets à l'extérieur des limites de l'espace généralisé pour créer des conflits avec des objets extérieurs non pris en compte (1). Des relations géographiques comme celles décrites par (Duchêne, 2004b) (par exemple, le positionnement relatif de deux bâtiments) entre un objet à l'intérieur et un objet à l'extérieur de l'espace peuvent être détruites (2) ou créées faussement (3). Or ces relations sont très importantes dans la compréhension d'une carte et la généralisation doit les transcrire de manière juste. Enfin, des objets inclus dans plusieurs espaces géographiques peuvent être généralisés plusieurs fois et de manière incohérente par des processus différents (4).

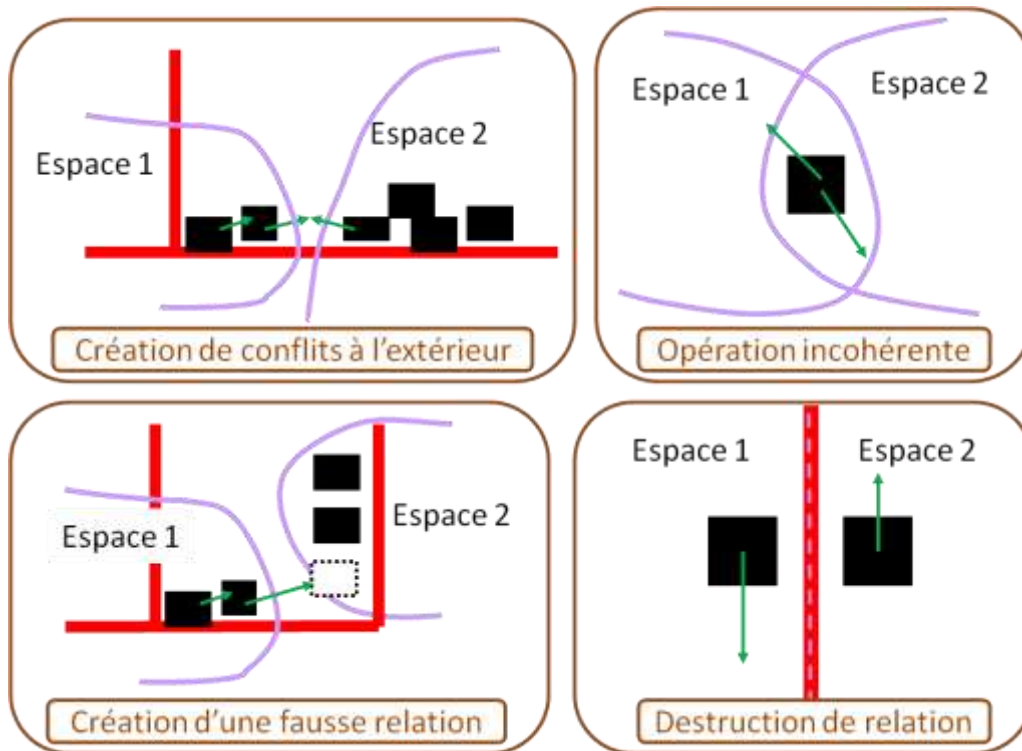


Figure B-9. Quatre exemples d'effets de bord possibles lors d'une généralisation collaborative.

Plutôt que d'essayer de modifier les processus et l'orchestration afin d'éviter leur apparition, ce qui nous paraît très compliqué, nous proposons d'accepter la présence occasionnelle des effets de bords. Ainsi, la dernière des fonctions d'un système de généralisation collaborative est de détecter l'apparition de ces effets de bord après une généralisation et d'essayer de les corriger au mieux sans défaire les opérations de généralisation précédentes, censées avoir amélioré la lisibilité de la carte au niveau de l'espace. Nous ajoutons donc un nouveau composant de **gestion d'effets de bord** chargé de les identifier et de trouver une solution équilibrée entre leur état initial (avant la généralisation causant l'effet de bord) et la situation généralisée de l'espace concerné.

Garantir une généralisation homogène

Après avoir présenté les fonctions de la généralisation collaborative, quelques fonctions complémentaires méritent d'être mentionnées. Tout d'abord, une des contraintes d'une généralisation cartographique réussie est l'homogénéité des transformations appliquées sur l'ensemble de la carte. Si pour une même situation, des transformations différentes sont adoptées à des endroits différents de la carte, plusieurs des principes de base de la généralisation (voir définitions en A.1.2) ne seront pas respectés : d'une part, on ne reste pas fidèle à l'information initiale car deux situations initialement semblables seront perçues différemment dans la carte généralisée ; d'autre part, on brouille la compréhension de la carte car le lecteur de la carte ne peut pas interpréter correctement ces différences de traitement.

En utilisant plusieurs processus différents pour généraliser différents espaces de la carte, la généralisation collaborative s'expose fortement à une généralisation hétérogène. Garantir une généralisation de résultat homogène devient donc une fonction complémentaire primordiale d'un système de généralisation collaborative pour éviter les situations illustrées par la Figure B-10 : la ville

est généralisée par un processus 1 qui, notamment, grossit, élimine et déplace les bâtiments ; l'espace périurbain est généralisé par un processus 2 qui agrège les bâtiments ; l'espace rural est généralisé par un processus 3 qui symbolise les bâtiments par une croix. Ces différences entraînent des incohérences dans l'interprétation des données (y a-t-il des bâtiments collectifs en zone périurbaine et des pavillons en zone urbaine ?) ce qui est contraire aux spécifications classiques d'une carte topographique que nous avons présenté en fin de chapitre A.

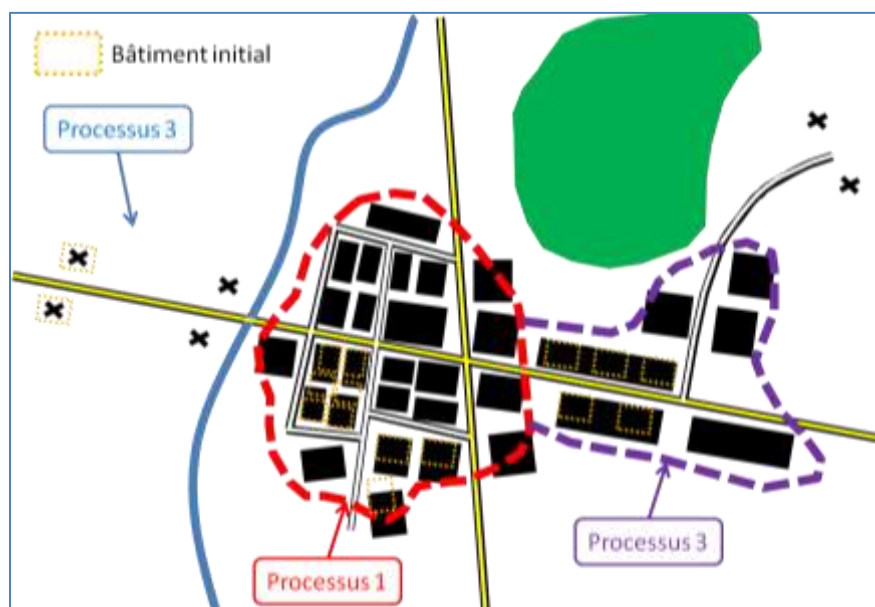


Figure B-10. Illustration d'une généralisation collaborative hétérogène pour les bâtiments : le processus 1 généralise les bâtiments par des opérations de grossissement, élimination et déplacement tandis que le processus 2 a recours à des agrégations et le processus 3 à des opérations de symbolisation. Seuls quelques bâtiments initiaux sont représentés pour ne pas encombrer la lecture.

Afin d'assurer cette fonction de garantir l'homogénéité en fonction des spécifications, nous proposons l'utilisation d'un jeu de contraintes unique pour évaluer les processus de généralisation d'une manière uniforme. Ce jeu de contraintes pourrait aussi être utilisé pour configurer les paramètres de chaque processus. En effet, si tous les processus sont paramétrés à partir d'un jeu unique de contraintes, ils donneront des généralisations un peu plus homogènes. Mais s'ils sont surtout tous évalués à partir des mêmes contraintes, le risque de l'appariation d'hétérogénéités telles celles de la Figure B-10 est diminué. Comme dans les modèles AGENT, CartACom ou GAEL, les contraintes sont utilisées à la fois pour guider la généralisation et comme support de son évaluation pas à pas.

Evaluer la généralisation

La généralisation collaborative suit les préceptes de l'approche de généralisation pas à pas locale et notamment du modèle de (Ruas et Plazanet, 1996) qui propose d'identifier une situation locale en conflit (ici un espace géographique), de trouver l'algorithme adapté à la résolution du conflit (dans notre cas, il s'agit du processus le plus adapté) et d'évaluer le résultat afin de déterminer si on continue (résultat acceptable) ou si l'on essaie un autre algorithme pour cette situation. De même, un système de généralisation collaborative doit posséder une fonction d'évaluation qui évalue la réussite de la généralisation d'un espace par un processus afin de déterminer si on revient en arrière

pour trouver un meilleur processus ou si on juge que la situation a été améliorée. Cette fonction répond au sens de (Mackaness et Ruas, 2007) à l'évaluation pour *contrôler* car elle vise à valider l'enchaînement des processus et l'évaluation pour *estimer* car elle cherche à identifier les erreurs pour permettre de les corriger. De plus, spécifiquement dans le cadre de la généralisation collaborative, cette fonction doit évaluer la qualité interne du processus (s'il répond bien à ce qu'il devait faire) et la qualité globale de la généralisation de l'espace. La Figure B-11 montre la différence entre une généralisation qui doit être évaluée comme complètement mauvaise et une généralisation moyenne mais de bonne qualité interne : le processus n'a pas traité la forêt car il ne sait pas le faire mais il a bien généralisé les bâtiments et les routes. L'intérêt d'évaluer la qualité interne d'une généralisation est de permettre le recours à plusieurs processus à la suite pour généraliser un espace quand aucun des processus disponibles ne peut traiter tous les problèmes présents dans l'espace.

Cela impose donc la présence d'un nouveau composant d'*évaluation* qui utilise les contraintes formelles comme ressource pour assurer les fonctions de contrôle et d'estimation. Ce composant doit être capable d'évaluer globalement la généralisation d'un espace ce qui est encore un problème de recherche : on sait évaluer individuellement la généralisation d'un objet ou d'un groupe d'objet (Bard, 2004 ; Zhang et al, 2010) mais estimer si l'ensemble ou une partie (i.e. un espace) est bien généralisé est aussi complexe.

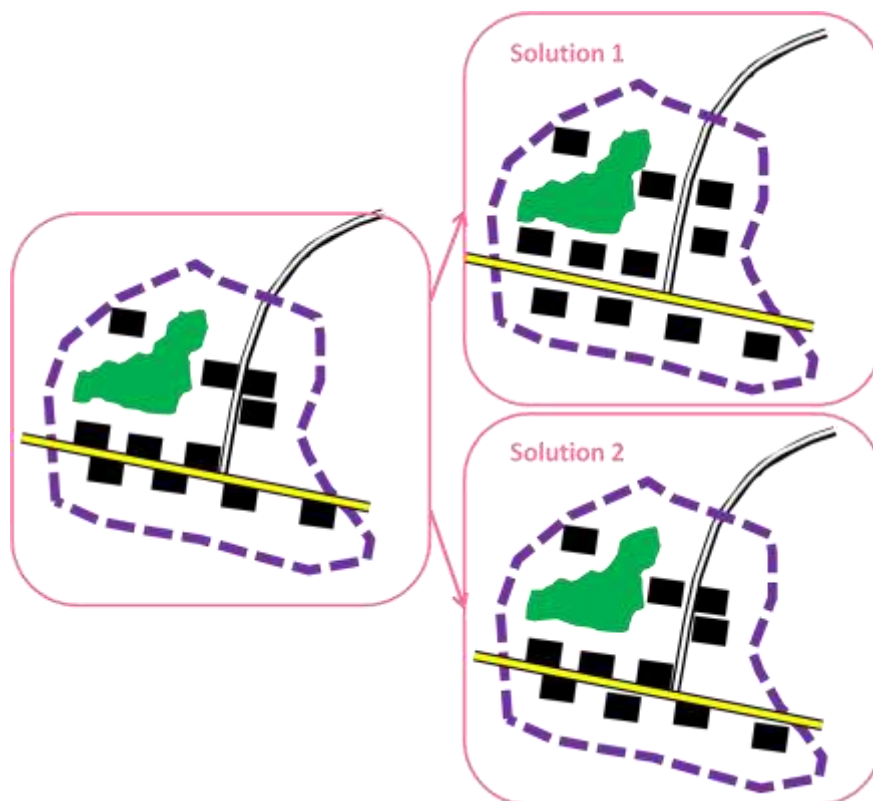


Figure B-11. Différence entre évaluation globale et interne : la solution 1 est issue d'un processus purement urbain ne traite pas la forêt trop détaillée mais bien les bâtiments (résultat meilleur mais non fini). La solution 2 est issue d'un processus traitant forêt et bâtiment donc elle doit être évaluée comme mauvaise (résultat dégradé donc à refuser).

Observer la généralisation

Lors de la conception des modèles AGENT et CartACom, il est apparu que l'approche pas à pas locale pouvait rencontrer des situations d'impasses pour lesquelles seul un autre point de vue pourrait apporter une solution (Ruas, 1999 ; Duchêne, 2004). Par exemple, CartACom peut générer des situations qu'il ne peut résoudre sans faire appel à une structure meso telle celles du modèle AGENT (Figure B-12). En plus des mauvais résultats qu'elles provoquent, ces situations peuvent aussi parasiter le reste de la généralisation autour en modifiant de manière erronée le contexte des situations voisines.

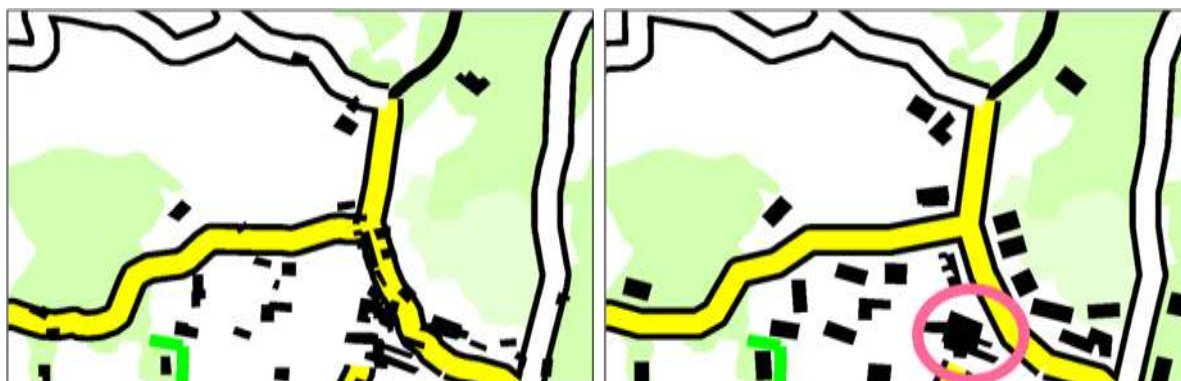


Figure B-12. Du fait de la trop forte densité, un conflit insoluble (entouré) entre les deux grands bâtiments est généré durant l'exécution de CartACom (Duchêne, 2004).

Afin de prévenir ces situations de groupes de conflits locaux qui perdurent, un système de généralisation collaborative devrait disposer d'un système d'observation en temps réel de la généralisation en cours. Durant la généralisation, les conflits sont observés et si une telle situation intervient, le système doit pouvoir interrompre la généralisation pour résoudre le problème et relancer la généralisation là où elle en était.

Cette fonction d'observation est très proche de la précédente fonction d'évaluation car elle cherche à identifier les conflits cartographiques dans un espace. La différence réside dans le facteur temps pris en compte dans la fonction d'observation alors que l'évaluation est effectuée à la fin de l'exécution d'un processus. Nous faisons le choix de regrouper ces deux fonctions dans un composant d'**observation et d'évaluation**.

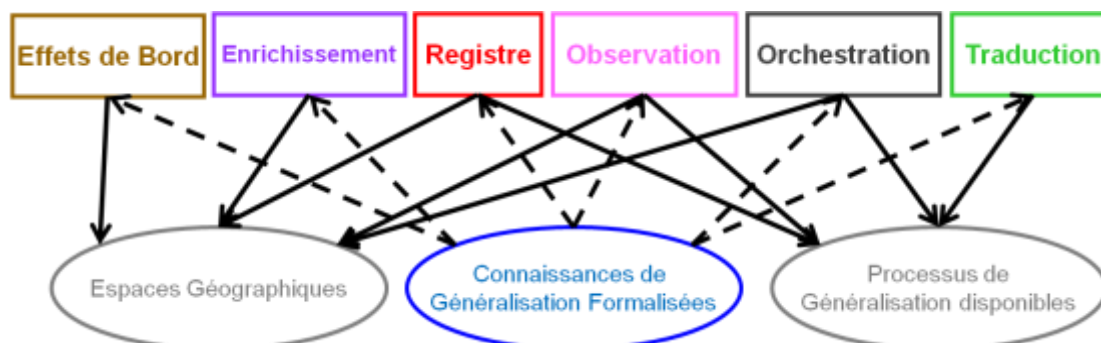


Figure B-13. Composants (rectangles) et Ressources (ellipses) de la Généralisation Collaborative. Les flèches pleines indiquent qu'un composant manipule une ressource et les flèches pointillées indiquent qu'un composant est guidé par des connaissances formalisées.

La Figure B-13 regroupe l'ensemble des composants requis d'après l'analyse fonctionnelle précédente ainsi que les ressources et leurs interactions avec les composants: les espaces géographiques et les processus de généralisation sont manipulés par les composants alors que les connaissances de généralisation formalisées guident les six composants. Nous regroupons ici sous le terme de connaissances de généralisation formalisées toutes les ressources requises par l'analyse fonctionnelle autres que les espaces géographiques et les processus (ontologie de généralisation, spécifications formelles, règles d'orchestration et description formelles des processus).

En résumé, le cadre générique de Généralisation Collaborative nécessite la mise en œuvre de six composants : un composant d'enrichissement géographique, un composant de traduction, un composant registre, un composant d'orchestration, un composant d'observation et d'évaluation et enfin un composant de gestion des effets de bord.

Ces six composants vont manipuler des espaces géographiques et des processus de généralisation disponibles et vont être guidés par un ensemble de connaissances de généralisation formalisées comprenant : des connaissances de découpage en espaces, des spécifications formalisées, des règles d'orchestration formelles, une ontologie du domaine et une description formelle des capacités des processus.

B.4. UN NOUVEL ESPACE DE TRAVAIL : LES ESPACES GEOGRAPHIQUES

Nous avons vu dans les parties précédentes de ce chapitre que le cadre formel de Généralisation Collaborative repose sur le découpage en *espaces géographiques* sur lesquels sont appliqués des processus de généralisation. Contrairement aux autres composants et ressources dont une modélisation est présentée dans le chapitre suivant, nous présentons la modélisation des espaces géographiques dans cette partie. En effet, nous considérons que la modélisation des espaces géographiques telle que nous la proposons dans cette partie est indissociable du cadre formel de la généralisation collaborative et que tout modèle s'intégrant dans ce cadre devrait suivre les recommandations de cette modélisation. Le principe de la généralisation collaborative repose sur le découpage des données en espaces géographiques donc la définition de ce cadre repose sur la définition des espaces géographiques que nous présentons dans cette partie.

B.4.1. Définitions

De manière pragmatique par rapport au contexte de la généralisation collaborative, nous définissons un espace géographique comme ***une partie finie des données géographiques, dans l'espace euclidien de la représentation cartographique et/ou dans l'espace des thèmes de la carte, qui est pertinente pour l'application d'un processus automatique de généralisation.*** Avant de préciser notre définition, nous allons faire le lien entre notre vision pragmatique des espaces géographiques et les différentes définitions données par des géographes.

Comment définir l'espace géographique

Nous présentons d'abord le point de vue des géographes sur l'espace géographique. Selon (Braud et al, 1997) la définition de l'espace géographique fait débat parmi les géographes mais on peut le résumer à « un espace aménagé, modelé, 'produit' par les sociétés, par leurs activités mais aussi par le milieu naturel et les héritages historiques. ». L'espace peut être **représenté** (Braud et al, 1997) ce qui correspond à notre cadre de travail où la représentation est celle de la base de données initiale utilisée pour la généralisation.

(Brunet et al, 1993, p. 194) définit l'espace géographique de la manière suivante : "*l'étendue terrestre utilisée et aménagée par les sociétés en vue de leur production : non seulement pour se nourrir et s'abriter, mais dans toute la complexité des actes sociaux. Il comprend l'ensemble des lieux et leurs relations. (...) L'espace géographique ne se limite donc pas à la distribution du visible, encore moins des seuls éléments auxquels on le réduit parfois : des accidents topographiques, des frontières et des villes ; mais, il inclut cela.*" Il s'agit d'une définition générale de l'espace géographique dans l'absolu soit de l'objet d'étude de la géographie.

(Cauvin, 1984, p.184) distingue trois types d'espaces géographiques :

- "*L'espace physique : c'est l'espace concret, celui que l'on peut toujours mesurer d'une manière identique, quel que soit le lieu ou le moment considéré...*"
- "*Les espaces fonctionnels : ils constituent la deuxième grande famille d'espaces et correspondent à ceux que l'homme utilise, où l'homme vit. Il s'agit des espaces où les relations sont évaluées en termes de temps, de coûts...*"
- "*Les espaces cognitifs : les espaces cognitifs sont des espaces "obtenus", "construits", après modifications, transformations par nos filtres personnels et culturels, des espaces physiques et fonctionnels. Ces espaces diffèrent pour chacun de nous. Ils ne sont pas homogènes...*"

Cette définition fait apparaître que s'il existe un seul espace physique, plusieurs espaces fonctionnels et cognitifs peuvent cohabiter sur un même espace physique.

(Pumain et Saint-Julien, 1997) envisage l'espace géographique de deux manières différentes :

- "*Soit comme un simple contenant, repère immuable dans lequel on situe les objets et analyse leurs relations : c'est l'espace support, ou espace absolu, il est isotrope et homogène, c'est-à-dire qu'il a les mêmes propriétés dans toutes les directions.*"
- "*Soit comme un ensemble de relations dont les propriétés, variables dans le temps et dans l'espace, sont définies par la nature et la forme des interactions entre les objets et les unités spatiales : c'est l'espace relatif ou espace produit.*"

Enfin, (Boffet, 2001, p.20) écrit "*La description de l'espace géographique ne se limite pas à l'analyse de l'espace physique ou espace support. Il faut également considérer l'espace relatif ou fonctionnel vécu par les sociétés ainsi que l'espace cognitif perçu plus ou moins consciemment.*"

Ces définitions de géographes montrent que les espaces géographiques de la généralisation collaborative peuvent être vus comme des espaces par les géographes : ils sont des parties de l'espace support représenté possédant généralement une identité physique et/ou fonctionnelle (comme les espaces urbains ou les espaces de montagne) ou au moins une identité cognitive relative au processus de généralisation.

Le concept d'espace de travail en généralisation

La notion d'espace de travail a toujours été un problème identifié par les chercheurs en généralisation cartographique : sur quels objets, groupes d'objets ou parties d'objets faut-il appliquer un traitement de généralisation pour être vraiment efficace ? Ainsi, (Shea et McMaster, 1989) pose le problème de la surface unitaire sur laquelle on doit analyser les conflits et chercher un algorithme. (Buttenfield, 1991) a ensuite proposé d'adapter l'espace de travail en fonction de ses caractéristiques dans le cas de la généralisation d'une ligne. En allant un peu plus loin, (Ruas et Plazanet, 1996) suggère d'adapter localement à chaque situation l'espace de travail au cours de la généralisation. Cette approche est mise en œuvre dans (Mustière, 1998) qui propose un processus de généralisation des routes qui commence par découper les routes en parties représentant des espaces de travail adaptés aux algorithmes du processus.

Notre définition de l'espace géographique est donc une extension naturelle de la question de l'espace de travail en généralisation : les espaces géographiques sont des espaces de travail adaptés aux processus de généralisation automatiques. Il s'agit d'un découpage par le but et non *a priori*.

Les espaces géographiques dans la généralisation collaborative

Nous distinguons trois types d'espaces géographiques dans le cadre de la généralisation collaborative :

- Les espaces **territoriaux** : ils sont délimités par une frontière et contiennent tous les objets situés à l'intérieur des limites. Les zones urbaines, périurbaines (Figure B-14a), rurales ou côtières sont des exemples d'espaces territoriaux.
- Les espaces **thématiques** : ils sont délimités par un ou plusieurs types de données et contiennent tous les objets de ce type dans la base de données comme une couche de données dans un SIG (Denègre et Salgé, 1996). Le réseau routier, la couche de végétation (Figure B-14b) ou l'occupation du sol sont des exemples d'espaces thématiques.
- Les espaces **mixtes** : ils sont à la fois thématiques et territoriaux. Les routes de montagne (Figure B-14c) ou les bâtiments urbains sont des exemples d'espaces mixtes.

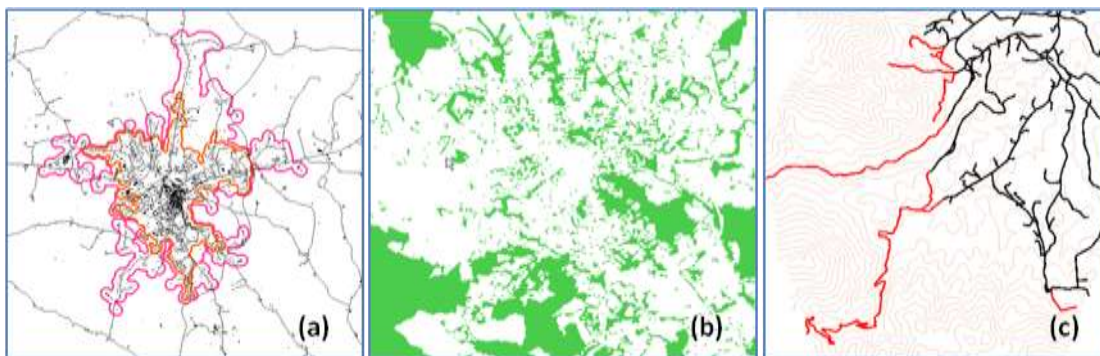


Figure B-14. (a) des espaces urbains (en orange) et périurbains (en rose) délimités à partir des bâtiments (en noir). (b) l'espace couche de végétation. (c) Un espace de type routes de montagne (en rouge).

Il est important de noter qu'une telle définition des espaces géographiques fait qu'ils ne forment pas une partition de l'espace. Les espaces territoriaux peuvent se recouvrir comme une zone de montagne et une zone rurale et les espaces thématiques recouvrent tout ou partie des espaces territoriaux. Plus particulièrement, les réseaux traversent les espaces territoriaux et les délimitent parfois car ils structurent souvent un territoire.

Nous proposons aussi de classer les espaces géographiques selon la raison de leur création au cours du processus de généralisation collaborative, ce qui traduit leur rôle :

- **Espace clé** : il s'agit d'un espace qui est considéré comme pertinent quelques soient les processus disponibles car son identité géographique est différente des autres. L'espace clé doit donc être généralisé par un processus qui peut gérer cette identité de manière efficace (en traitant les conflits particuliers tout en maintenant la différence avec l'extérieur). Il apparaît alors comme un candidat idéal à la généralisation collaborative. Par exemple, les espaces urbains, ruraux et de montagne sont considérés comme clé dans le contexte de la généralisation de cartes topographiques. En effet, ils présentent une identité très différente qui implique des traitements différents et donc éventuellement l'application de processus différents. La liste des espaces clés fait partie des connaissances formelles de généralisation qui sont des ressources de la généralisation collaborative d'après la partie précédente (B.3).
- **Espace facilitateur** : il s'agit d'un espace utile pour optimiser l'utilisation d'un processus disponible donné. Si le processus est très décrit comme très efficace sur ce type d'espace donc l'espace est intégré à la généralisation collaborative pour utiliser au mieux ce processus. Par exemple, si le processus pas à pas par filtrage collaboratif (Burghardt et Neun, 2006) est disponible, l'expérience montre qu'il est très efficace sur les espaces à structures assez répétitives (dans lesquels on retrouve les mêmes structures urbaines) donc les espaces de lotissements résidentiels répétitifs sont considérés comme un espace facilitateur. L'occupation du sol est un autre exemple d'espace facilitateur car il va être considéré seulement quand un processus adapté comme celui de (Hauert, 2007a) est disponible.
- **Espace spécifié** : il s'agit d'un espace utile pour prendre en compte certaines des spécifications de la carte. On verra dans la partie C.5 que les spécifications peuvent n'être définies que sur un territoire particulier que l'on ramène à un espace géographique dans notre cas. Si de telles spécifications existent, la création des espaces requis est indispensable à la prise en compte de ces spécifications. Par exemple, certaines cartes contiennent des spécifications spéciales en zone côtière permettant de mieux maintenir des éléments pouvant servir de repères sur la côte. Si de telles spécifications sont utilisées dans la généralisation collaborative, les espaces côtiers sont considérés comme espaces spécifiés.
- **Espace émergent** : il s'agit d'un sous-espace créé lorsque le composant d'observation identifie un groupe de conflits qui ne peut pas être résolu par le processus en cours ou de manière générale au cours de la généralisation (entre deux processus) (Figure B-15).

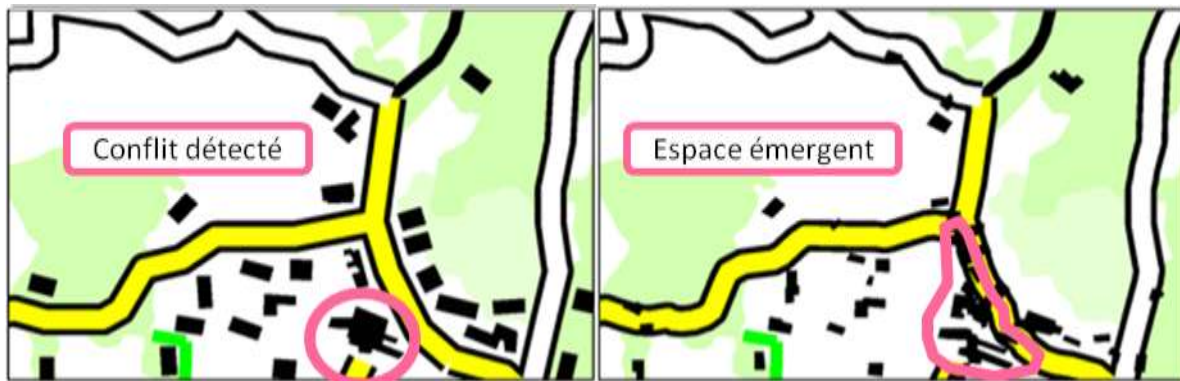


Figure B-15. Exemple d'espace émergent dans CartACom suite à l'observation d'un groupe de conflit.

Il faut noter que cette classification est floue. Si l'on prend l'exemple des graphes de flexibilité (Lemarié, 2003) qui sont des espaces de type mixte au sens de notre première classification (sous-graphes des réseaux routier et hydrographique), ils peuvent d'abord être considérés comme des espaces facilitateurs reliés au processus des « Beams » (Bader et al, 2005) car ils sont conçus pour amortir au mieux les déformations générées par ce processus. Mais ils pourraient être considérés comme clés si on suppose que quelques soient les processus disponibles, on aura besoin de résoudre les conflits de superposition entre éléments des réseaux routier et hydrographique en les amortissant. Ils pourraient alors être traités par un autre processus que les Beams si ceux-ci ne sont pas disponibles. Le choix de la classification en espace clé doit alors répondre à la question suivante : est-ce que je souhaite vraiment utiliser cet espace quels que soient les espaces disponibles et cet espace possède-t-il une identité géographique propre ? Pour les espaces urbains, la réponse sera plutôt oui, pour les graphes de flexibilité, elle serait plutôt non ce qui explique nous les classifications comme espace facilitateur. L'intérêt d'introduire cette classification est de permettre au système collaboratif de déterminer quels espaces géographiques il doit utiliser lors de la généralisation d'une carte.

En résumé, les espaces géographiques sont des portions de la base de données initiale, délimitées spatialement et/ou thématiquement, qui peuvent être des zones d'application adéquates d'un ou plusieurs processus de généralisation.

B.4.2. Modélisation d'un espace géographique

La Figure B-16 montre le diagramme de classe UML qui représente notre modélisation des espaces géographiques dans le cadre de la généralisation collaborative. Nous proposons notamment de relier un espace géographique avec le concept qui correspond à son type dans l'ontologie du domaine, ressource nécessaire pour assurer l'interopérabilité (voir C.4). Les espaces géographiques partagent également une relation partonomique (Chaudhry, 2007) avec les objets de la base de données : un espace géographique contient des objets géographiques. La modélisation de ces espaces géographiques en tant qu'objets complexes de la base de données est essentielle pour permettre une généralisation collaborative. (Steiniger, 2007) considère ces objets complexes comme des *connaissances structurelles* et on peut les intégrer à ce que (Ruas, 2004) appelle le schéma de travail qui comprend tout les enrichissements faits dans la base de données initiales pour bien décrire le contexte géographique des données.

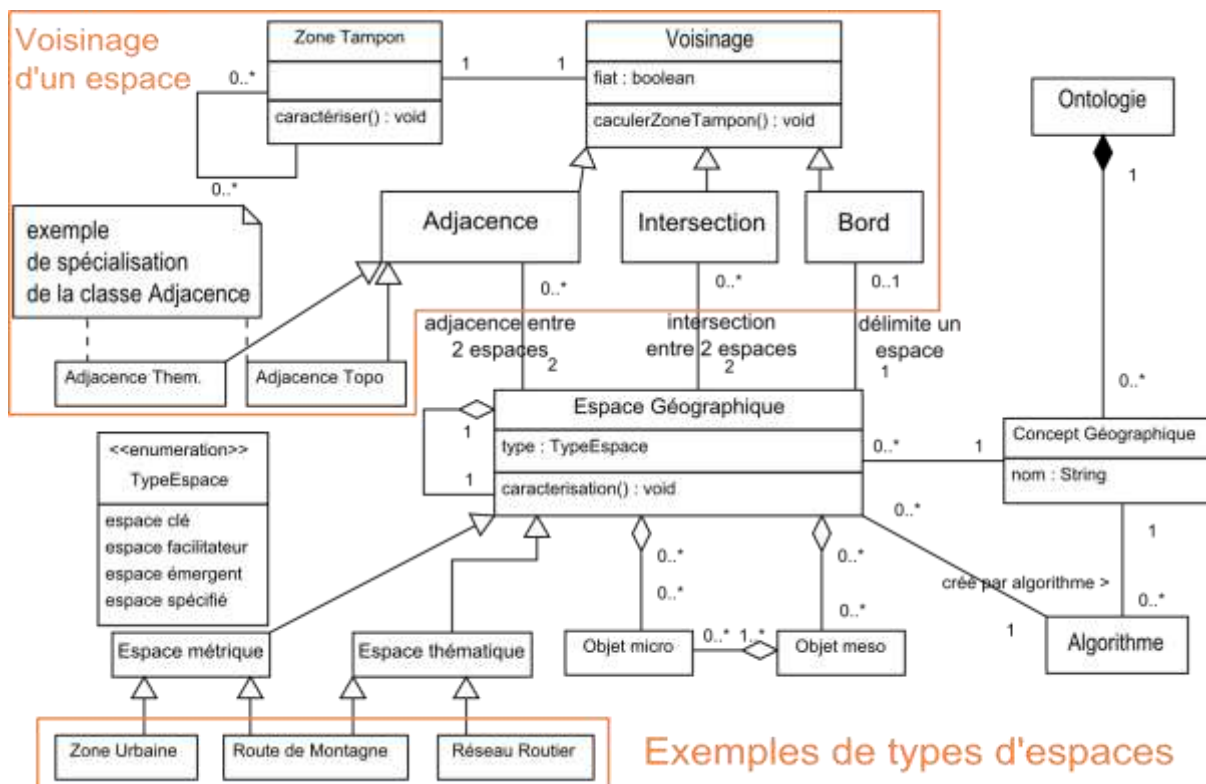


Figure B-16. Diagramme de classe UML du modèle complet d'un espace géographique dans le cadre de la généralisation collaborative.

Nous avons vu dans l'état de l'art que l'espace d'application pertinent de certains processus ne se décrivait pas forcément sous la forme d'espaces territoriaux spécifiques comme l'espace urbain mais parfois par des caractéristiques comme *hiérarchique* (pour AGENT), ou *peu dense en bâtiments* (pour CartACom). Il faut donc non seulement créer ces espaces (les détecter, les représenter) mais aussi les caractériser par des propriétés comme la densité d'objets, le type de structure (hiérarchique ou transversal), la quantité d'espace vide, la diversité des thèmes des objets ou le thème dominant.

Le modèle proposé définit également le voisinage des espaces géographiques. Nous définissons ce voisinage comme la partie de l'espace autour laquelle on doit guetter d'éventuels effets de bord après la généralisation de l'espace géographique. Ces effets de bord peuvent avoir lieu quand deux espaces se recouvrent, autour de limites partagées entre deux espaces (ce qui est très fréquent quand les réseaux sont utilisés pour délimiter les espaces) et juste à l'extérieur des limites d'un espace avec des objets non pris en compte par le processus appliqué à l'espace. Nous définissons donc trois types de voisinage, le bord, l'adjacence et l'intersection. L'adjacence et l'intersection correspondent aux relations classiques de topologie du même nom (Egenhofer, 1989) entre deux espaces géographiques. Le bord est inclus dans le modèle de voisinage afin de pouvoir identifier des effets de bord juste à l'extérieur de l'espace quand il n'y a pas d'adjacence ou d'intersection avec un autre espace. Ces voisinages sont munis d'une zone tampon qui délimite la recherche des effets de bord dans chaque voisinage (Figure B-17). Ces zones tampon ne sont pas de mêmes dimensions pour chaque type de voisinage : celle de l'intersection est la plus petite car les effets de bord sont plus susceptibles d'apparaître à l'intérieur de l'intersection et celle du bord est la plus grande pour pouvoir chercher assez loin à l'extérieur car les objets conflictuels peuvent être assez loin (s'ils étaient près de la limite, ils auraient été vraisemblablement inclus dans l'espace).

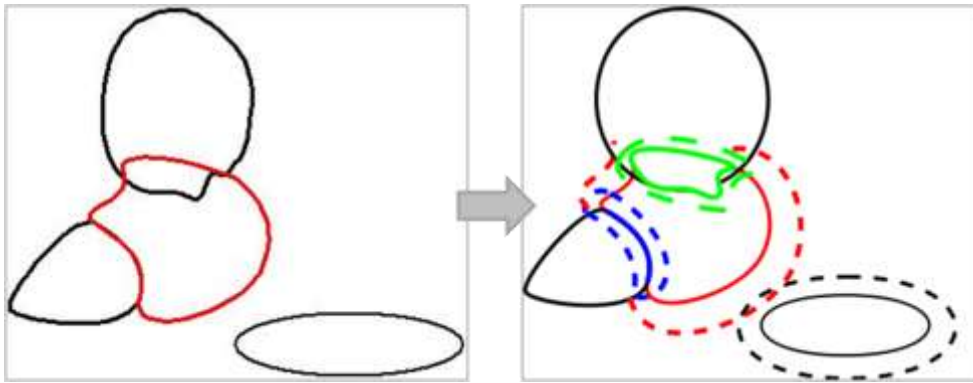


Figure B-17. Les trois types de voisinage d'un espace géographique et leur zone tampon (en pointillés) : le bord en rouge, l'adjacence en bleu et l'intersection en vert.

Nous avons vu qu'en plus des espaces géographiques possédant une géométrie surfacique, comme les espaces territoriaux et mixtes, pour lesquels cette définition du voisinage est valable, il existe aussi des espaces thématiques comme le réseau routier, la couche de végétation ou l'occupation du sol. Notre définition précédente du voisinage étant basée sur une limite et des relations topologiques, il paraît difficile de l'appliquer à des espaces aussi différents qu'un réseau ou une couche d'objets ponctuels : quel est le contour du réseau routier ou hydrographique ? Ce contour a-t-il un sens ? Quand peut-on dire que l'on est dans le voisinage des points remarquables du relief ? De plus, si l'on trouve une réponse à l'une de ces questions, la réponse sera-t-elle la même pour d'autres thèmes ? En d'autres termes, peut-on délimiter le réseau ferré de la même manière que la végétation ou les points remarquables du relief ? Il existe de nombreuses méthodes pour délimiter des ensembles d'objets géométriques (Figure B-18) mais pour chacun des thèmes la méthode la plus adaptée doit être choisie en tenant compte du fait que l'on ne cherche pas à définir l'espace couvert par les objets du thème mais son voisinage donc les endroits susceptibles de connaître des effets de bord.

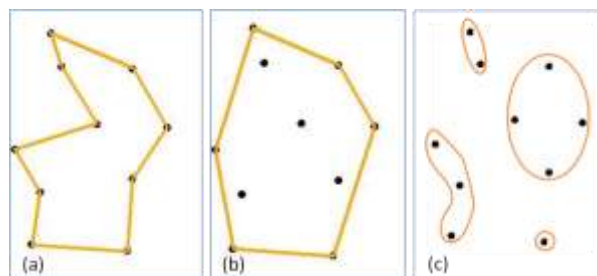


Figure B-18. Trois méthodes pour délimiter un ensemble de points : (a) une enveloppe concave, (b) une enveloppe convexe, (c) une classification non supervisée.

En topologie mathématique, l'adhérence d'une partie d'un espace topologique est le plus petit ensemble fermé contenant cette partie. Ce concept d'adhérence correspond bien à notre recherche d'ensemble des objets susceptibles de subir des effets de bord suite à la généralisation d'un espace géographique thématique. Plutôt que de chercher les voisinages définis précédemment pour les espaces territoriaux, il nous paraît plus intéressant de définir l'équivalent de l'adhérence des espaces thématiques pour déterminer un ensemble fermé dans lequel les objets géographiques sont susceptibles de subir des effets de bord. Prenons l'exemple de l'espace thématique réseau routier. Nous pouvons l'assimiler à un ensemble ouvert car sa fonction est de connecter des lieux géographiques (comme des bâtiments ou des sites touristiques par exemple) qui ne font pas partie

du réseau (uniquement composé des routes reliant ces lieux). Nous pouvons donc considérer que l'adhérence du réseau routier (le plus petit ensemble fermé contenant le réseau routier) est le réseau lui-même plus toutes les entités géographiques auxquelles le réseau permet d'accéder directement. Avec un point de vue analyse spatiale, l'adhérence du réseau routier serait l'union de zones tampons créées autour de chacune des routes, ces zones tampons délimitant les endroits auxquels on peut accéder depuis les tronçons de route (Figure B-19a).

Afin de déterminer quelle méthode doit être choisie pour délimiter l'adhérence d'un espace géographique donné, nous proposons de revenir à la raison pour laquelle nous avons défini le voisinage des espaces géographiques, la gestion des effets de bord après application d'un processus de généralisation sur cet espace. Du fait de ses particularités, chacun des espaces thématiques va être généralisé d'une manière si ce n'est unique, du moins particulière. Par exemple, on sait que le réseau hydrographique va subir des opérations de simplification, de sélection et de déplacement et l'étendue des effets de bord provoqués par de telles opérations, peut être anticipée. La Figure B-19 propose deux exemples de détermination de limites d'espaces thématiques à partir des effets de bord anticipés. Dans le cas du réseau routier, la généralisation va principalement déformer les routes en cas de virages serrés ou de routes trop proches, voire enlever des virages dans le cas des routes en lacets. Ces déformations provoqueront des effets de bord sur les réseaux et les équipements à côté d'une route. Donc le choix d'une adhérence par zone tampon autour du réseau (établie en analysant la fonction de l'espace) convient également en analysant les transformations potentielles. Dans le cas de la couche de cheminées industrielles (constituée de points), la généralisation va principalement réaliser des opérations de typification (qui maintiennent la forme et la structure des groupes tout en simplifiant le contenu) et de déplacement. Les effets de bord vont donc être liés aux bords des groupes plutôt qu'à l'intérieur donc le bord de l'espace sera l'union du bord des groupes de cheminée. Le choix de l'adhérence de chaque espace thématique doit donc être effectué en tenant compte de la fonction des objets du thème et des transformations qu'ils peuvent subir durant la généralisation.



Figure B-19. Deux exemples bien différents de propositions de voisinage (adhérence) pour des espaces thématiques : à gauche, chaque élément du réseau routier forme le bord de l'espace et on voit en jaune sa zone tampon ; à droite, la couche de cheminée possède un bord par cluster.

Enfin, le modèle de la Figure B-16 fait apparaître la possibilité de définir entre deux espaces thématiques un voisinage particulier, que nous appelons *voisinage* (ou *adjacence*) *thématique fonctionnel* (le). Ce voisinage thématique est utile pour prendre en compte plus facilement les effets de bord dus à des relations fonctionnelles entre deux thèmes. Par exemple, une couche de points représentant les équipements routiers comme les stations services est fonctionnellement liée au réseau routier : si l'on bouge l'un suite à une généralisation, la relation fonctionnelle est perdue par

effet de bord et il faut bouger l'autre pour maintenir la relation (Figure B-20). Les voisinages thématiques ainsi définis permettent d'identifier des effets de bord potentiels dont la correction est simple et systématique : on rétablit la relation par diffusion (Burghardt et Meier, 1997, Legrand et al, 2005). La différence par rapport aux autres cas de voisinage est le systématisme des effets de bord : il n'est pas utile d'inspecter localement la présence d'effets de bord mais simplement diffuser la modification du premier thème au deuxième. Il s'agit d'effets de bord *a priori* alors que les autres sont identifiés *a posteriori*.

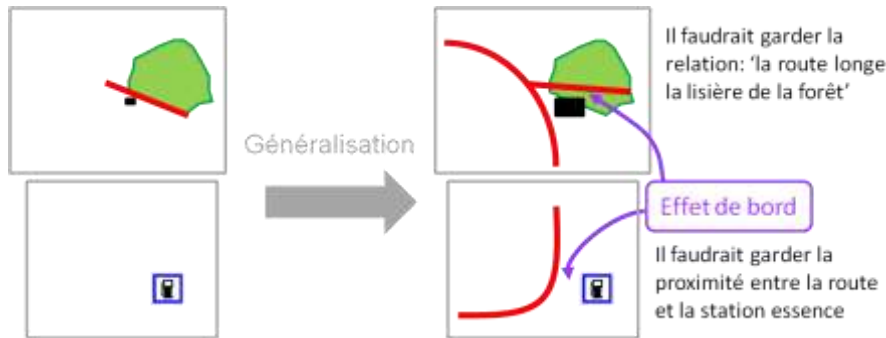


Figure B-20. Deux cas d'adjacence thématique au réseau routier : la végétation et la couche de stations services. Bouger le réseau routier entraîne des effets de bord sur les objets de ces thèmes.

En résumé, les espaces géographiques sont munis de voisinages avec les autres espaces qui permettent d'identifier les objets touchés par des effets de bord. Pour les espaces thématiques, il convient de définir l'adhérence de l'espace, qui varie en fonction de la géométrie de chaque thème, des fonctions géographiques de ce thème et des transformations qu'il peut subir. Il est possible de déterminer des voisinages thématiques fonctionnels qui permettent d'explicitier les relations fonctionnelles entre deux espaces thématiques.

B.4.3. Quel est le meilleur contour pour espace territorial ?

Nous avons vu qu'il était difficile de déterminer la limite d'un espace thématique mais choisir celle d'un espace territorial n'est pas non plus une tâche évidente. En effet, certaines entités géographiques ont des limites dites *bona fide* (Smith et Varzi, 2000) c'est-à-dire nettes car dues à la présence de ruptures physiques comme les lacs. Mais beaucoup d'autres entités ont des limites dites *fiat* (Smith et Varzi, 2000) c'est-à-dire floues comme les zones urbaines ou rurales. Dans le cas des espaces géographiques de type *fiat*, il existe une infinité de possibilités pour les délimiter comme on peut en avoir un aperçu dans le cas des espaces urbains sur la Figure B-21. En effet, la définition d'un espace (ou d'une zone) urbain(e) ne fait pas consensus, les géographes, les cartographes ou l'INSEE donnant des définitions différentes du fait de points de vue différents. Ainsi, plusieurs méthodes ont été développées avec leur propre point de vue dépendant principalement de l'application visée : par exemple la méthode issue de (Boffet, 2001) adopte un point de vue orienté généralisation cartographique, (Chaudhry et Mackaness, 2008b) un point de vue orienté plutôt généralisation de base données avec un grand saut d'échelle alors que (Walter, 2008) adopte centré sur le réseau routier. Mais même dans le cas de limites *bona fide*, la limite de l'espace géographique dépend aussi de l'échelle d'observation et de représentation (ici celle de la carte généralisée) (Mackaness, 2007).

D'une part la granularité du contour est intimement liée à l'échelle de généralisation (on ne va pas délimiter un espace avec des contours possédant des détails de l'ordre du mètre pour généraliser une carte au 1 : 75000 par exemple), et d'autre part notre perception de ce qu'est une forêt (et donc de la délimitation d'un espace forestier par exemple) change suivant l'échelle. Le choix du meilleur contour pour un espace est donc dépendant de l'objectif et donc essentiellement de l'échelle de la carte à généraliser.

La mise en œuvre d'un système de généralisation collaborative nécessite donc de mettre au point une méthode pour lever les ambiguïtés sur la limite des espaces (ou au moins de connaître les caractéristiques des limites par rapport aux processus disponibles) et choisir l'algorithme d'analyse spatiale qui convient pour construire la limite.

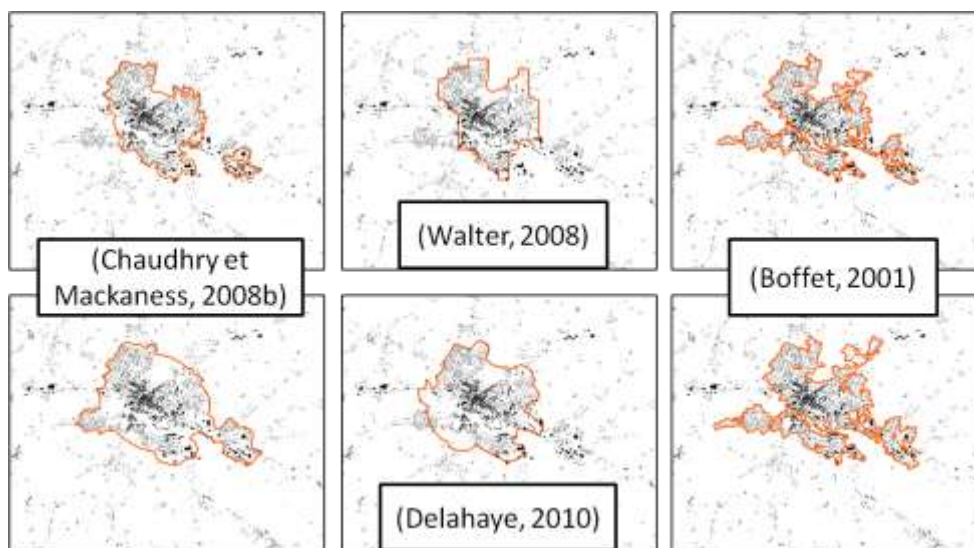


Figure B-21. Six délimitations différentes de l'espace urbain autour de la ville d'Orthez en faisant varier les méthodes et paramètres.

B.5. DYNAMIQUE DE LA GENERALISATION COLLABORATIVE

Après avoir décrit les composants et ressources requis par la mise en œuvre d'un système de généralisation collaborative ainsi que la manière de modéliser les espaces géographiques dans ce cadre, cette partie aborde les aspects de la dynamique d'un tel système. Nous nous concentrons sur les aspects de la dynamique que l'on peut considérer comme génériques et proposons des choix possibles pour les aspects moins génériques. Nous considérons que le cadre formel implique trois contraintes concernant la dynamique du système (Figure B-22). Tout d'abord, la création des espaces géographiques (clés, facilitateurs et spécifiés) doit se faire dans un premier temps. Il y a ensuite un enchaînement de cycles de généralisation jusqu'à ce que l'ensemble des espaces géographiques soit généralisé au mieux. Quelle que soit leur utilisation des différents composants, ces cycles doivent se conclure par une évaluation pour respecter pour permettre un fonctionnement par essai/erreur.

Enfin, nous ajoutons après l'évaluation, si celle-ci s'est avérée positive, l'étape de gestion des effets de bord.

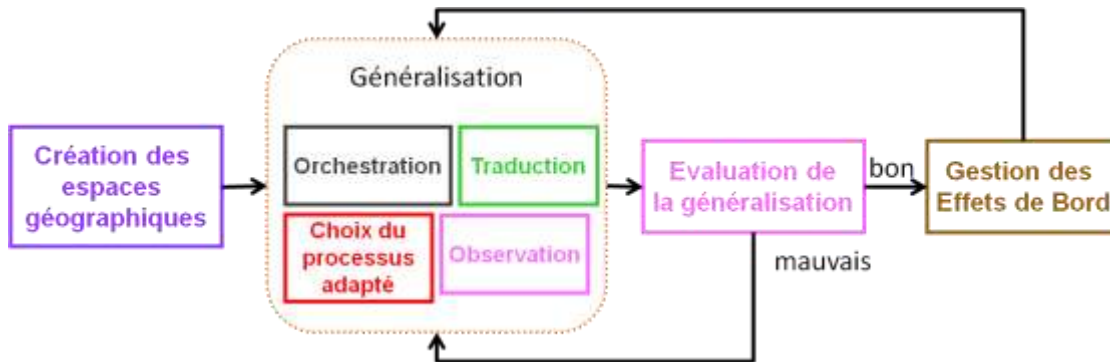


Figure B-22. Les trois parties génériques de la dynamique de la généralisation collaborative : d'abord la création des espaces géographiques puis après chaque généralisation, le système évalue pour respecter l'approche pas à pas locale puis si la progression est validée, il gère les effets de bord.

Les choix de dynamique à l'intérieur du bloc dit de « généralisation » ne sont pas contraints. Il existe plusieurs possibilités pour enchaîner les appels aux composants d'orchestration, registre, traduction et observation. Il faut notamment choisir si l'on souhaite avoir recours à une architecture plus ou moins parallèle ou plus ou moins séquentielle comme le montre la Figure B-23 restreinte au cas de l'orchestration. Dans cet exemple où on fait l'hypothèse que l'on généralise ensemble toutes les instances d'un type d'espace, le scénario parallèle propose pour un type d'espace donné de généraliser toutes les instances en même temps (pour des espaces territoriaux ou mixtes et non thématiques), la gestion des effets de bord ne se faisant qu'une fois toutes les généralisations terminées. Le scénario séquentiel réalise ensuite une par une des séquences de généralisation, évaluation, gestion des effets de bord tant qu'il reste des instances à traiter, l'ordre pouvant être déterminé *a priori* ou à la volée. En faisant d'autres hypothèses sur la généralisation des instances d'un même espace, d'autres scénarios de dynamique sont imaginables.

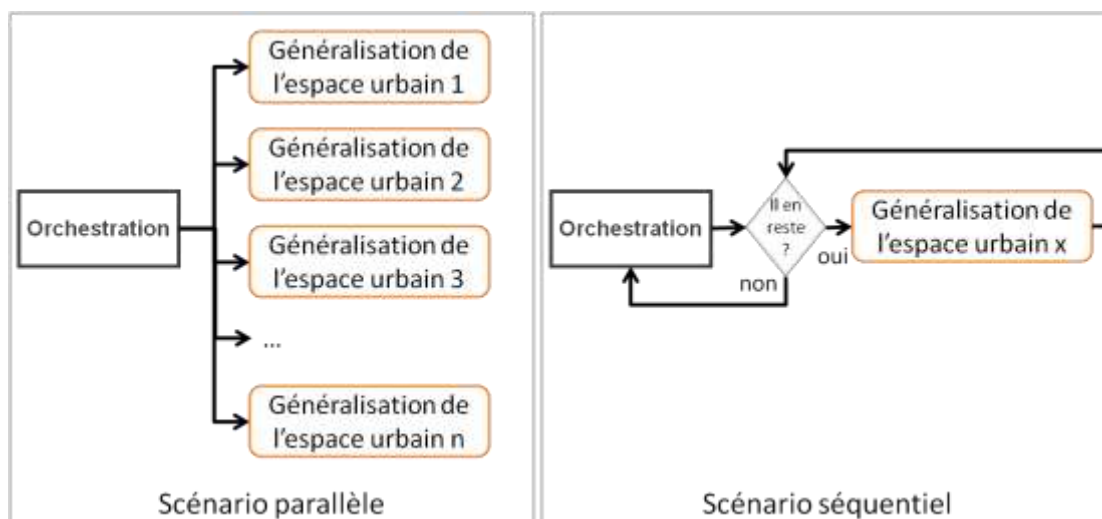


Figure B-23. Deux scénarios de dynamique d'orchestration indépendamment des autres composants : en parallèle ou en séquentiel.

Faire le choix du parallèle par rapport au séquentiel implique des questionnements différents. Par exemple, généraliser deux espaces géographiques voisins entraîne des interactions particulières qu'on ne rencontre pas dans le cas d'un déclenchement séquentiel. En effet, si une route traversant

les deux espaces est généralisée d'abord dans l'un des espaces, faut-il annuler sa généralisation dans l'autre espace ou guider l'autre généralisation en temps réel ou encore envisager des mécanismes de réconciliation après coup ? Il n'est sûrement pas possible de ne faire que du parallèle donc il faut envisager de mixer une approche séquentielle à une approche parallèle par exemple selon les thèmes ou l'étape dans le processus collaboratif.

La généralisation collaborative commence par la création des espaces géographiques et la généralisation doit être suivie de l'évaluation puis de la gestion des effets de bord. Les autres aspects de la dynamique sont des choix de mise en œuvre.

CHAPITRE C. MISE EN ŒUVRE : LE MODELE COLLAGEN

Dans ce chapitre, nous allons présenter le modèle CollaGen (pour Collaborative Generalisation) que nous proposons pour mettre en œuvre le cadre formel de généralisation collaborative. Nous allons notamment décrire les choix que nous avons effectués pour mettre en œuvre la dynamique de généralisation collaborative, les ressources et les composants identifiés au chapitre précédent : l'ontologie de généralisation, les contraintes formelles, les règles opérationnelles, les descriptions formelles d'un processus et les règles d'orchestration ; le composant d'enrichissement, le composant de traduction, le composant registre, le composant d'observation et d'évaluation, le composant d'orchestration et le composant de gestion des effets de bord. Pour rappel, le chapitre D traitera de la mise en œuvre de ce modèle CollaGen comme un système multi-agent et le chapitre E présentera un prototype de ce modèle et des résultats obtenus grâce à ce prototype.

C.1. CAS D'ÉTUDE : GENERALISER LA BD TOPO® AU 1 : 50 000

Avant de rentrer dans la description du modèle CollaGen, nous allons présenter le cas d'étude qui va nous servir à illustrer les capacités du modèle CollaGen. Il s'agit du problème classique de la généralisation cartographique de cartes topographiques au 1 : 50 000 à partir d'une base de données géographique très détaillée, que les agences nationales de cartographie comme l'IGN essaient de résoudre depuis des années. Dans notre cas, la base de données initiale est la BDTopo® de l'IGN dont la résolution est métrique et l'échelle de référence approximativement le 1 : 15 000 (Figure C-1).

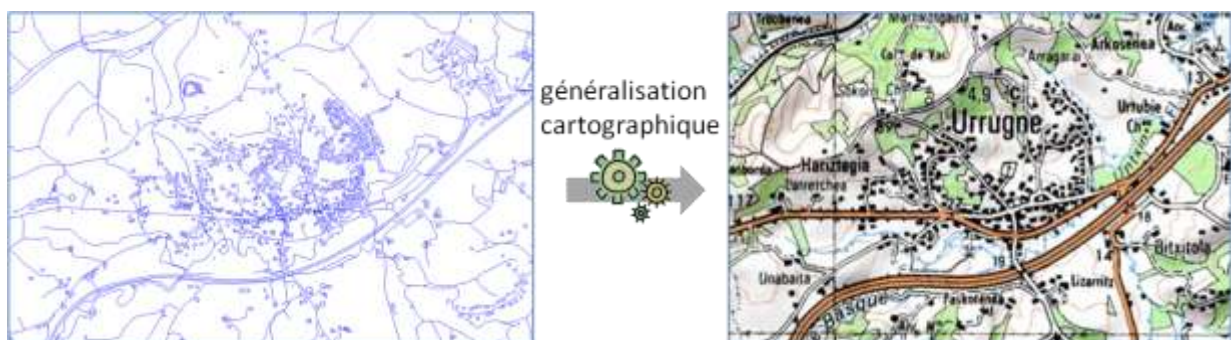


Figure C-1. Extrait de la BDTopo® de l'IGN sur la ville d'Urrugne (64) et carte au 1 : 50 000 de l'IGN sur la même zone.

Ce cas d'étude est un classique de la recherche en généralisation cartographique. Il est donc relativement fertile en processus existants et les comparaisons seront aisées pour évaluer notre modèle CollaGen et son implémentation. De plus, les contraintes classiques présentées en grille de

lecture dans le chapitre A s'appliquent bien à ce cas d'étude et seront donc utilisés pour illustrer notre modèle.

Ce cas d'étude est par ailleurs celui sur lequel s'est concentré notre implémentation d'un prototype pour CollaGen donc nous le présenterons avec plus de précisions au chapitre E.

C.2. QUI FAIT QUOI DANS COLLAGEN ?

Le modèle CollaGen met en jeu des ressources (i.e. les entrées du système) relativement nombreuses et de nature très différente. Les interactions d'un humain utilisateur avec le modèle sont donc multiples et de différentes natures. Nous modélisons ici ces interactions selon le rôle que peut prendre cet humain par rapport au modèle, afin de bien définir qui fait quoi dans l'utilisation et la mise en place d'un système basé sur le modèle CollaGen. Nous définissons quatre rôles : le *modélisateur*, le *développeur*, le *fournisseur* et l'*utilisateur*.

Le premier rôle qu'un humain peut adopter dans l'ordre chronologique de la mise en place d'un système basé sur le modèle CollaGen est celui du *modélisateur*. Il est responsable de la réalisation de l'ontologie du domaine de la généralisation à partir de connaissances précises sur le déroulement et les objectifs de la généralisation cartographique (Figure C-2). Cette ontologie servira de socle à l'expression formelle de toutes les autres ressources dans un vocabulaire partagé. De plus, le modélisateur est responsable de la saisie de règles d'orchestration issues des connaissances afin de permettre à l'utilisateur de choisir celles qui lui conviennent.

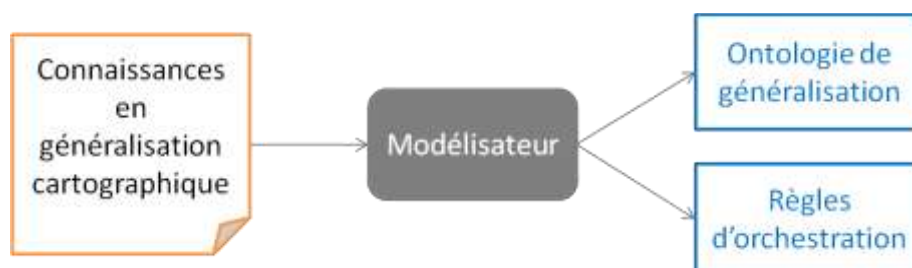


Figure C-2. Le modélisateur remplit l'ontologie du domaine de la généralisation et définit des règles d'orchestration à partir de connaissances en généralisation cartographique.

Le deuxième rôle, *développeur*, est important en termes de travail à exécuter, car il s'agit à la fois d'implémenter le modèle CollaGen présenté dans ce chapitre et de développer des outils pour alimenter une boîte à outils utilisée pour le prototype CollaGen (Figure C-3). Cette boîte à outils contient des outils d'analyse spatiale pour construire les espaces géographiques et réaliser les enrichissements nécessaires à l'utilisation des processus de généralisation, mais aussi des outils de mesure permettant d'évaluer les différentes spécifications de la carte et enfin des outils permettant de corriger au mieux les effets de bord. Le chapitre E qui présente notre implémentation du modèle CollaGen est une bonne illustration du rôle de développeur.

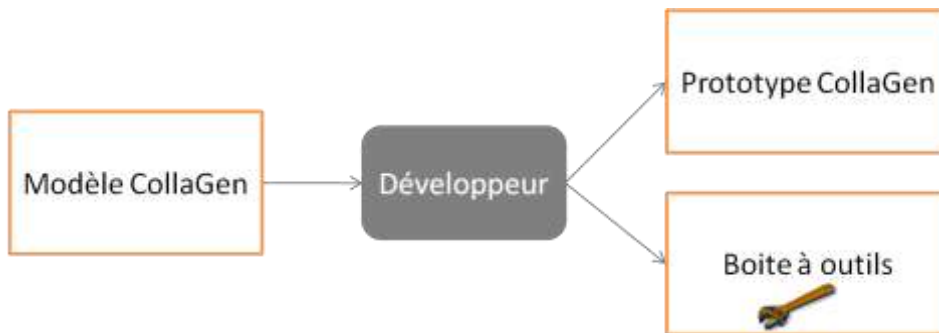


Figure C-3. Le développeur implémente le modèle CollaGen et développe des outils pour la boîte à outils utilisée par le prototype.

Une fois CollaGen implémenté, il ne peut fonctionner efficacement si le (ou les) fournisseur(s) ne lui met(tent) pas à disposition un certain nombre de processus automatique qu'il peut faire collaborer. Le fournisseur est responsable de la mise à disposition, dans CollaGen, des processus qu'il a développés (ou simplement spécialisé à partir d'un modèle de généralisation) (Figure C-4). Cette mise à disposition peut se décomposer en trois actions. Il faut d'abord rendre les processus fournis disponibles dans CollaGen, c'est-à-dire, selon les hypothèses de notre thèse, réaliser une implémentation du processus sur la plateforme logicielle sur laquelle est implémenté CollaGen. Il faut ensuite décrire les processus dans le langage formel de description de processus (voir C.7). Le fournisseur doit enfin fournir pour chaque processus une fonction de traduction donnant les paramètres du processus à partir des spécifications formelles au format CollaGen (voir C.5 et C.6). Le rôle d'un fournisseur peut être assimilé à celui du responsable de la publication d'un traitement sous la forme d'un service Web.

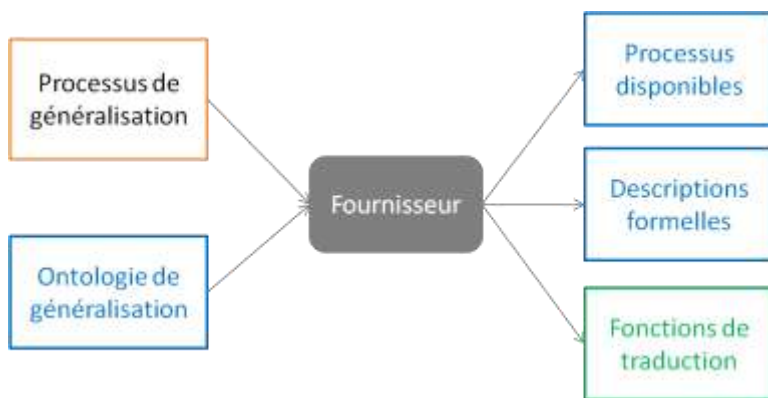


Figure C-4. Le fournisseur rend des processus de généralisation disponibles dans CollaGen, les décrit formellement et fournit des fonctions de traduction des spécifications formelles en paramètres des processus.

Le dernier rôle est celui de l'utilisateur final qui veut utiliser CollaGen pour généraliser des données selon les spécifications d'une carte (Figure C-5). L'utilisateur doit ainsi saisir ses spécifications, qu'il possède souvent sous la forme d'un document de texte libre, en spécifications formelles en s'aidant du vocabulaire de l'ontologie. Il doit également choisir les règles d'orchestration qu'il souhaite utiliser ou en saisir de nouvelles (s'il est un utilisateur expert car cela nécessite de bien connaître la généralisation automatique).

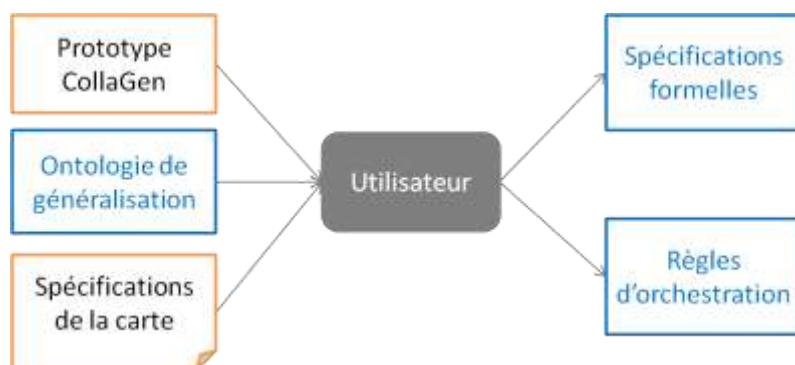


Figure C-5. L'utilisateur cherche à généraliser des données par CollaGen. Il saisit des spécifications formelles et des règles d'orchestration avant d'exécuter le système sur ses données.

Ces rôles peuvent être portés par différentes personnes ou la même : dans le cadre de cette thèse, nous avons tenu tous les rôles afin de réaliser un prototype de CollaGen et de le tester. Cette mise au point permet de clarifier comment sont construites les différentes ressources et quels sont les pré-requis à l'utilisation de CollaGen.

C.3. DYNAMIQUE DU MODELE COLLAGEN

Nous avons vu dans le chapitre précédent que la dynamique pouvait, tant qu'elle respectait une approche pas à pas locale par essai/erreur, prendre différentes formes. Dans CollaGen, nous avons fait le choix d'un processus où toutes les généralisations se font de manière séquentielle. Ce choix est probablement le moins efficace en termes de temps de calcul mais c'est le plus simple à mettre à œuvre tout en garantissant des résultats au moins aussi bons qu'une approche partiellement ou totalement en parallèle. En effet, il permet de s'affranchir des problèmes (effets de bord particuliers) qui se poseraient si nous généralisions deux espaces voisins en même temps, notamment du fait que des objets peuvent faire partie des deux espaces. Ce choix facilite également l'implémentation du modèle ce qui n'est pas négligeable. Enfin, ce choix facilite l'analyse de la séquence du processus : un des objectifs de notre thèse (présentés au chapitre A) étant d'acquérir des connaissances sur l'enchaînement des processus de généralisation, celles-ci pourront être dérivées de l'analyse des séquences automatiquement déterminées par CollaGen.

La Figure C-6 présente le 'workflow' d'une généralisation collaborative dans CollaGen, les couleurs correspondant au composant exécutant l'action concernée (nous gardons le même code couleur dans les parties suivantes pour décrire les composants).

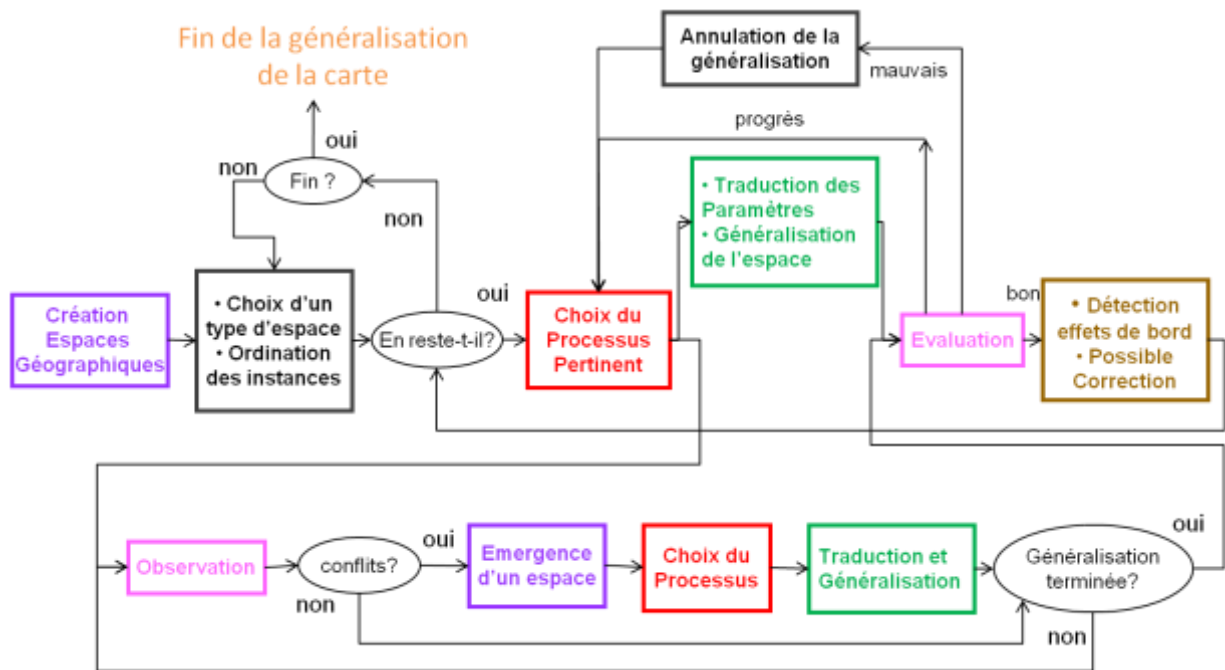


Figure C-6. Diagramme de la dynamique séquentielle de la collaboration dans CollaGen. Seule l'observation se fait en parallèle de la généralisation. Les couleurs des rectangles correspondent aux composants qui effectuent les actions.

Nous faisons le choix de généraliser ensemble toutes les instances d'un même type d'espace géographique (par exemple, tous les espaces urbains puis le réseau routier etc.). Ce choix est à la fois arbitraire et raisonné. En effet, il est plus simple d'orchestrer les généralisations de cette manière. Si nous prenons l'exemple d'un espace rural voisin de deux espaces urbains, l'orchestration est plus simple si le système généralise toujours les espaces urbains ensemble que ce soit avant ou après l'espace rural. Cependant, ce choix est aussi arbitraire car nous n'avons pas testé d'autres solutions par manque de temps et il serait intéressant de comparer notre approche avec une approche plus souple sur les possibilités d'orchestration des espaces.

Nous faisons le choix de détecter les effets de bord après chaque généralisation terminée d'un espace. En effet, en première approximation, il nous semble plus simple de les corriger petit à petit, même si cela représente un coût de temps important. Les compromis nécessaires (voir C.16.4) à la correction des effets de bord nous paraissent plus accessibles généralisation par généralisation que d'un seul coup. Il faut noter qu'un tel choix fait courir le risque de tourner en rond comme le montre la Figure C-7. Toutefois, nous considérons que ce cas n'a quasiment aucune chance de se réaliser. En effet, même si le compromis requis par la correction rétablit des conflits préalablement résolus, ces conflits seront moins sévères et l'évaluation globale de l'espace ne devrait pas être assez affectée pour nécessiter une nouvelle généralisation et entraîner une boucle sans fin. Afin d'être sûrs d'éviter ces problèmes, nous munissons quand même le composant de gestion des effets de bord d'un système de relaxation de conflits comme le proposait (Ruas, 1999), et que nous détaillerons dans la partie C.16.

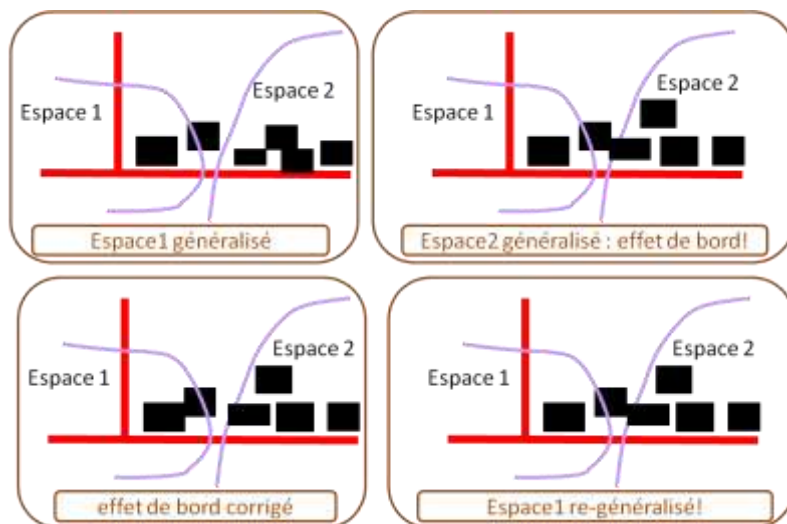


Figure C-7. Exemple où la dynamique de CollaGen pourrait générer une boucle infinie.

Si la dynamique est globalement séquentielle, l'observation des processus est faite en parallèle de la généralisation par ces processus (Figure C-6). Tant que la généralisation d'un espace donné n'est pas terminée, le composant d'observation observe à intervalles réguliers l'état de l'espace. Si le composant d'observation détecte un espace émergent du fait de conflits qui ne se résolvent pas, la généralisation de l'espace est mise en pause le temps que le sous-espace émergent soit généralisé par un autre processus plus adapté, puis elle est reprise là où elle en était. Ce mécanisme de pause/reprise n'est pas représenté sur le diagramme pour alléger sa lecture mais il est indispensable à un tel système d'observation. Cela nécessite des mécanismes particuliers pour éviter des incohérences : par exemple, si un objet est éliminé durant la généralisation de l'espace émergent, le processus généralisant l'espace en entier doit être mis au courant de cette modification du contexte géographique. Ce problème a déjà été identifié dans (Duchêne, 2004, p.110-113) dans le cadre d'une pause dans CartACom pour appliquer un autre processus localement sur un groupe puis reprendre CartACom.

Nous avons vu dans la partie A.1.2 qu'il était indispensable d'enrichir les données par des structures et des relations géographiques initialement implicites afin d'explicitier le contexte géographique pour le communiquer de manière formelle à un processus purement automatique. Ce mécanisme reproduit le raisonnement d'un cartographe qui sait reconnaître ces structures et relations implicites avant de choisir le type de généralisation qu'il va appliquer. Chacun des processus disponibles peut donc requérir qu'un certain nombre de structures et relations soient explicites dans les données afin de fonctionner. Cet enrichissement doit être réalisé automatiquement quelque part dans la chaîne de traitements de la Figure C-6. Dans CollaGen, un processus est choisi pour traiter seulement un espace géographique donné donc les enrichissements nécessaires à ce processus n'ont pas besoin d'être réalisés sur l'ensemble des données. Il s'agit d'information éphémère. L'enrichissement peut donc être réalisé en utilisant la boîte à outils juste après le choix du processus pertinent pour un espace. Cependant, si l'on considère par exemple le cas des 'Strokes' (Thomson, 2006) qui sont des ensembles de tronçons d'un réseau qui se suivent en bonne continuité (i.e. on peut dessiner cet ensemble d'un seul mouvement de crayon sur une feuille), il s'agit d'un enrichissement qui n'a de sens que sur l'intégralité des données (Figure C-8). C'est sa longueur calculée sur l'ensemble du réseau qui fait l'importance du stroke hydrographique de la figure. Calculer de tels enrichissements sur un espace territorial restreint ferait perdre du sens aux structures créées et créer plusieurs fois

ces structures sur l'ensemble des données serait inutile et coûteux. Il faut donc définir un autre moment dans la chaîne de traitement où certains des enrichissements, ceux-ci qui n'ont de sens que calculés sur l'ensemble des données, sont créés une fois pour toute. Cette autre phase d'enrichissement est placée juste après la création des espaces géographiques, au début de la généralisation collaborative.



Figure C-8. Exemple d'un 'stroke' dans un réseau hydrographique. Il tire son importance de sa longueur par rapport à l'ensemble du réseau.

Nous avons vu dans le chapitre précédent que le fonctionnement de la généralisation collaborative et de ses composants reposait sur des ressources : les processus de généralisation disponibles, les espaces géographiques et les connaissances formelles de généralisation (Figure C-9). Les cinq parties suivantes, de C.4 à C.8, décrivent les éléments composants ces connaissances formelles : l'ontologie du domaine de la généralisation cartographique, un modèle formel de contraintes de généralisation, un modèle formel de règles opérationnelles (dont nous présenterons l'intérêt), un modèle formel de description des processus de généralisation et un modèle formel de règles d'orchestration. Le formalisme des espaces géographiques et des processus de généralisation est décrit en C.9. Les moniteurs de spécification sont définis en C.11. Les composants (commençant par 'C_' dans le schéma) seront abordés dans les parties suivantes de C.11 à C.16. Ce schéma sera rappelé au début de chacune de ces parties pour rappeler comment se situe la ressource (ou le composant) par rapport au système global.

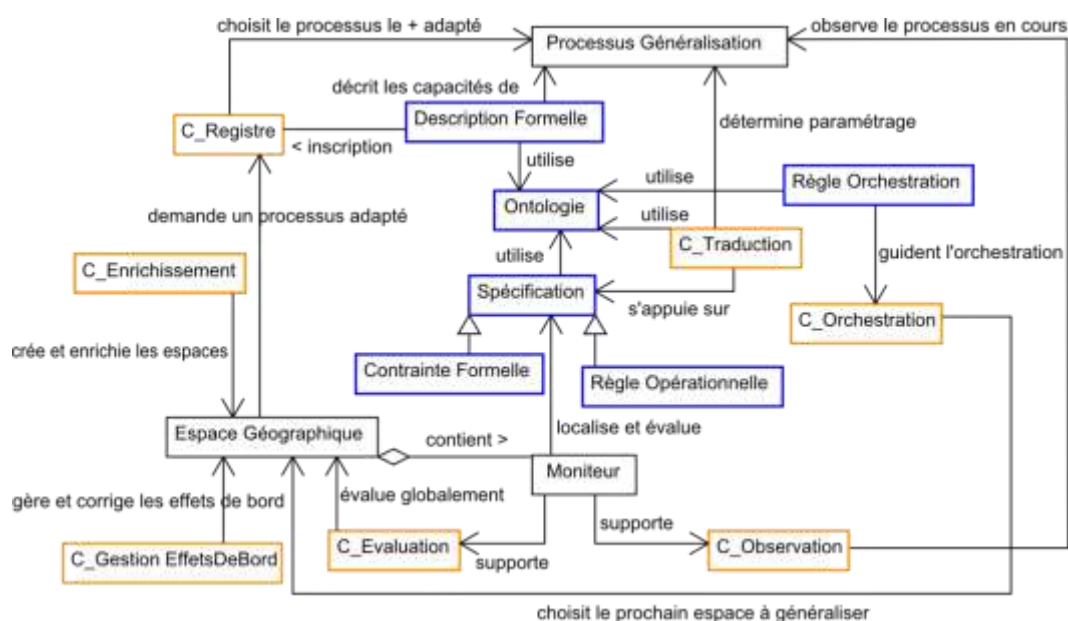


Figure C-9. Schéma UML décrivant les ressources, ressources formelles et les composants de CollaGen.

C.4. UNE ONTOLOGIE DE LA GENERALISATION CARTOGRAPHIQUE

Avant d'exposer nos propositions pour la conception d'une ontologie de la généralisation cartographique, nous allons expliquer ce qu'est pour nous une ontologie, puis faire un point sur la notion de schéma adapté pour la généralisation.

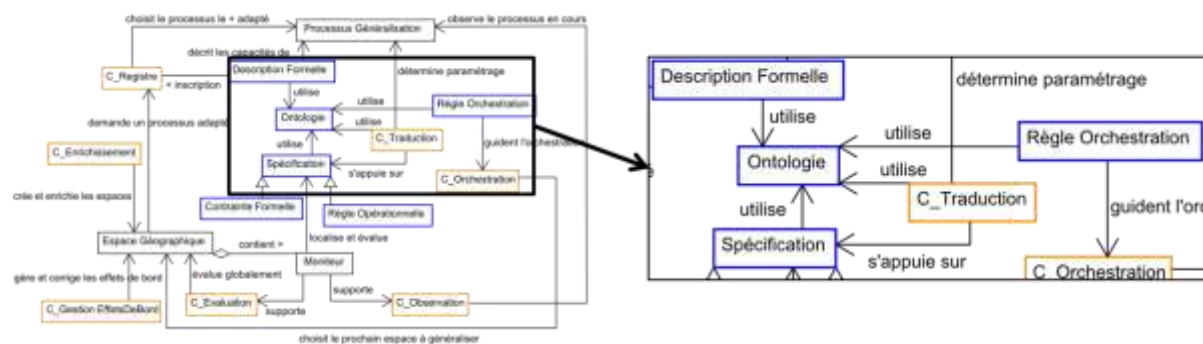


Figure C-10. L'ontologie de généralisation est la ressource formelle sur laquelle s'appuient les autres ressources formelles et le composant de traduction.

C.4.1. Qu'est-ce qu'une ontologie en informatique ?

A l'origine, l'ontologie est un domaine de la philosophie qui étudie l'essence des choses. Il a depuis été repris notamment en informatique sans toutefois être défini de manière consensuelle. Voici quelques définitions trouvées dans la littérature.

- (Grüber, 1995) donne une première définition reprise entre autres dans (Gesbert, 2005; Lemmens, 2006; Chaudhry, 2007) : une ontologie est « la spécification explicite d'une conceptualisation ».
- (Partridge, 2002) donne la définition suivante : « l'ensemble des choses dont l'existence est admise par une théorie ou un système de pensée donné. »
- De manière plus précise, (Uitermark *et al*, 1999) donne la définition suivante : « une ontologie est une collection de concepts partagés. Plus formellement, une ontologie est une collection structurée et limitée de concepts définis sans ambiguïté. »
- Pour (Davis, 1990), une ontologie du domaine, dans un domaine précis comme la généralisation cartographique, est : « ...la définition d'un micromonde : ses objets, ses relations, et les règles les gouvernant. Cette définition est appelée l'ontologie du domaine.»
- Enfin, (Gesbert, 2005) écrit que construire une ontologie revient à "décrire un ensemble de concepts partagés au sein d'un domaine plus ou moins restreint."

(Barde, 2005) propose une vision hiérarchique des outils pour gérer un vocabulaire partagé comme le glossaire, le thésaurus (ou taxonomie) et l'ontologie, du moins formel, le glossaire au plus formel, l'ontologie. La particularité de l'ontologie est de permettre une plus grande diversité de liens entre les concepts ainsi que des propriétés et des instances au contraire des deux autres outils.

Pour nous, une ontologie est un ensemble organisé de concepts, propriétés et relations partagés au sein d'un domaine restreint, ainsi que de règles sur ces concepts, propriétés et relations.

On peut ensuite classer les ontologies en différentes catégories. (Guarino, 1998) les classe selon leur niveau de généralité avec les ontologies de haut niveau (par exemple une ontologie décrivant les concepts de temps et d'espace), les ontologies du domaine (comme les ontologies géographiques), les ontologies de tâches décrivent les concepts relatifs à une tâche générique (comme la construction ou la vente) et enfin les ontologies d'applications qui décrivent les concepts de l'association d'un domaine et d'une tâche (par exemple, la vente de voitures). La structure d'une ontologie peut être simple et se rapprocher d'un thésaurus ou d'un vocabulaire partagé (Gesbert, 2005; Lemmens, 2006) ou se rapprocher d'un schéma conceptuel de base de données avec des relations hiérarchiques et partonomiques entre les concepts comme dans le cas des *ontologies descriptives* de (Cullot *et al*, 2003). Dans ce cas, les ontologies peuvent être (et sont souvent) représentées par des langages à base de logique de description (Baader *et al*, 2003) comme le standard OWL. Les ontologies du domaine géographique sont assez différentes des autres ontologies du fait de l'importance des relations mérologiques (relations de composition) et topologiques entre les concepts géographiques (Smith et Mark, 1998). Il existe donc de nombreuses relations entre les concepts, et le problème de la délimitation de ces concepts (en fonction de l'échelle par exemple) est la principale difficulté de la création d'ontologies géographiques (Smith et Mark, 1998).

L'ontologie répondant à nos besoins est une ontologie d'application car il s'agit d'une ontologie du domaine géographique dans le cadre de la tâche de généralisation cartographique. Une telle ontologie est complexe et peut être modélisée dans un langage adapté comme OWL.

Il existe maintenant de nombreux exemples d'ontologies géographiques construites pour différentes applications. La Figure C-11 montre un extrait d'une ontologie topographique construite à partir des spécifications des bases de données IGN pour intégrer des données topographiques (Abadie et Mustière, 2010).

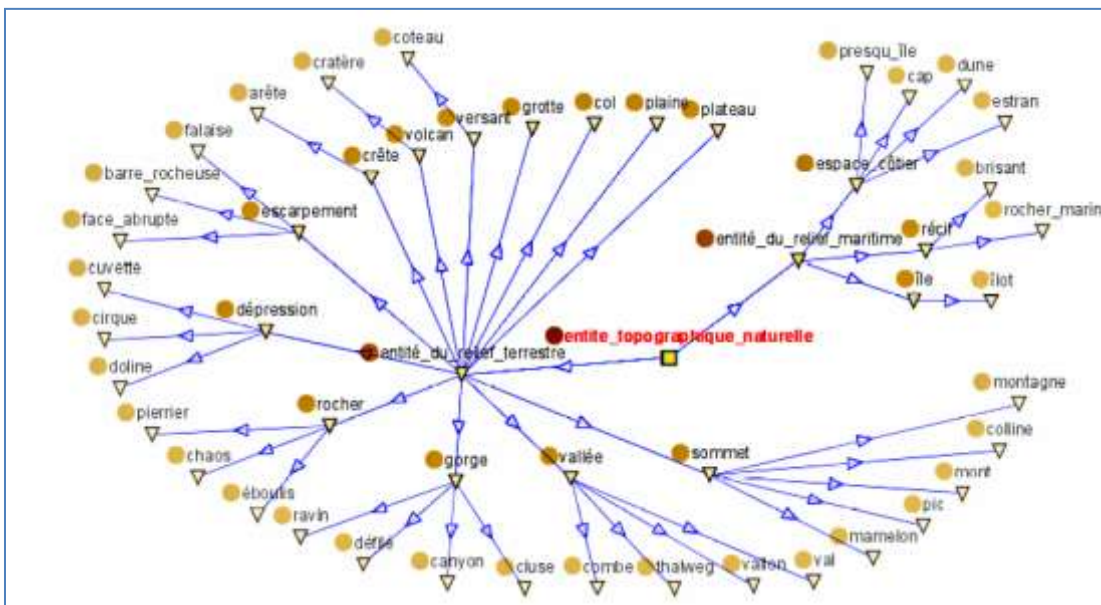


Figure C-11. *Ontologie des entités topographiques naturelles issue de spécifications (Abadie et Mustière, 2010).*

(Horrocks, 2008) affirme qu'il existe trois grands types d'applications possibles d'une ontologie:

- corriger des modèles conceptuels par des raisonnements logiques.

- partager et intégrer des informations.
- être un support structuré à un ensemble d'annotations.

A partir de ces trois grands types d'applications auxquels nous pouvons les rattacher, voici quelques exemples d'utilisation des ontologies géographiques:

- la manipulation de concepts géographiques comme des entités d'un SIG (Smith et Mark, 1998; Chaudhry, 2007).
- l'interopérabilité des données et services géographiques en partageant un vocabulaire et une modélisation des entités géographiques et de leurs relations (Lemmens, 2006; Lutz, 2007).
- la mise en correspondance des schémas de données pour l'intégration de base de données (Uitermark et al, 1999; Gesbert, 2005; Abadie, 2009).
- l'appariement de données géographiques grâce au calcul de distance sémantique entre concepts géographiques (Abadie et al, 2007).
- être un support pour annoter des données géographiques non structurées en BD comme celles d'OpenStreetMap (Girres et Touya, 2010).
- en généralisation (voir partie suivante).

Nous souhaitons définir une ontologie pour gérer de l'interopérabilité entre des données et des processus de généralisation en partageant un vocabulaire et une modélisation des entités géographiques et de leurs relations.

C.4.2. Les Ontologies en généralisation

Plusieurs travaux en généralisation ont déjà eu recours aux ontologies ce qui peut nous donner des pistes sur la modélisation de notre ontologie dans CollaGen et son intégration dans un système de généralisation. Ainsi, (Kulik et al, 2005) propose de guider un algorithme de généralisation en attachant le contexte de l'objet traité à une ontologie. Appliqué au filtrage d'une ligne, le nombre de points filtrés est décidé en fonction des informations présentes dans l'ontologie : les différentes formes de lignes géographiques sont décrites dans une ontologie avec une valeur de paramétrage associée.

Plusieurs travaux proposent d'utiliser une ontologie pour faciliter la phase d'enrichissement des données par des structures et des relations géographiques implicites. En effet, (Dutton et Edwardes, 2006) proposent de décrire toutes ces structures et leurs relations dans une ontologie, et utilisent cette ontologie et ses capacités de raisonnement logique pour pouvoir classer des toponymes en fonction de leurs caractéristiques. (Lüscher et al, 2007) proposent d'aller plus loin avec un système de détection de structures guidé par des ontologies décrivant en détail les caractéristiques des structures (Figure C-12). Puis (Lüscher *et al*, 2008b) recherchent dans les données des occurrences de ces différentes caractéristiques pour identifier la structure globale.

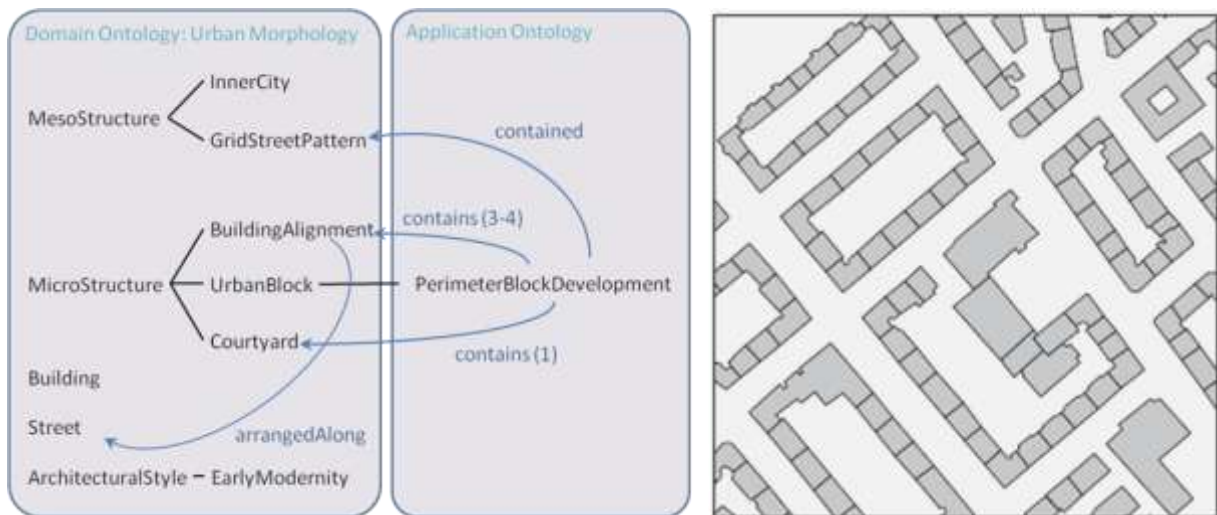


Figure C-12. Description ontologique précise d'un "perimeter block development", une structure urbaine dont on voit quelques exemples à droite (Lüscher et al, 2007).

Cette approche de description des caractéristiques de structures dans une ontologie est également utile pour l'évaluation de la généralisation de ces structures. C'est en effet ce que proposent (Zhang et al, 2008 ; 2010). Ils proposent de décomposer les structures comme une baie ou un alignement de bâtiments en sous-concepts structurés dans une ontologie. Puis ils évaluent la généralisation de la structure en s'assurant que les sous-concepts dont ils ont enrichi les données, sont bien structurés de la même manière dans les données généralisées.

Enfin, nous avons déjà vu dans la partie B.1.3 que (Regnauld, 2007) propose un système basé sur l'utilisation d'ontologies pour garantir l'interopérabilité entre les algorithmes de généralisation qu'il souhaite utiliser à distance à partir de services de généralisation. Il propose l'utilisation de trois ontologies : une ontologie géographique décrivant les données et leurs relations comme dans les exemples précédent, une ontologie des usages géographiques qui permettrait de formaliser les spécifications de l'utilisateur et enfin une ontologie cartographique qui contiendrait les règles de cartographie et de lisibilité ce qui rapproche cette ontologie de l'ontologie des tâches cartographiques proposée par (Iosifescu et Hurni, 2007).

En généralisation, les ontologies ont été exploitées pour décrire les structures et relations complexes entre objets géographiques mis en jeu dans la généralisation cartographique. S'attachant aux problèmes d'interopérabilité entre processus, notre ontologie devrait aussi contenir des informations sur les spécifications et le processus de généralisation cartographique en lui-même. Ceci est en quelque sorte une suite formalisée des travaux de (McMaster et Shea, 1992).

C.4.3. La Problématique du schéma adapté

La généralisation cartographique nécessite, avant toute opération, de modifier le schéma de données initial pour intégrer les enrichissements nécessaires au fonctionnement des processus. Ce schéma de données enrichi est appelé adapté car il est adapté à la tâche de généralisation cartographique (Ruas, 2001). Mais que contient exactement ce schéma adapté par rapport à un schéma initial classique contenant des classes de bâtiments, tronçons de route etc. et des attributs comme la nature des

routes ou le nom des cours d'eau ? Cette information implicite dans les données initiales peut être de différentes natures.

On y retrouve tout d'abord les concepts de type *meso* définis par (Ruas, 1999 ; 2000). Un objet meso est un objet complexe qui est l'agrégation d'objets micros ou meso, qui transcrit un sens géographique de plus haut niveau. La Figure C-13 montre deux exemples d'objets meso assez différents, un alignement de bâtiments autour d'une impasse, un col composé des courbes de niveau bien structurées ou un ensemble d'îles sur un cours d'eau formé par les tronçons hydrographiques. Ces objets meso sont particulièrement utiles dans le processus AGENT car ils sont responsables de la généralisation de tous les objets qui forment ce groupe. Ces groupes d'objets implicites sont exploités dans la plupart des processus de généralisation.



Figure C-13. Trois exemples d'objets meso implicites dans les données. (a) un alignement de bâtiments le long d'une impasse. (b) les courbes de niveau forment un col. (c) ensemble d'île sur un cours d'eau formé par les tronçons hydrographiques.

La construction du schéma adapté prenant en compte ces objets meso nécessite plusieurs opérations avec tout d'abord la création des classes d'objets meso, la création d'associations d'agrégation avec les classes micros et éventuellement des liens représentant des relations spatiales entre les objets meso. La Figure C-14 montre le schéma adapté pour le concept meso de carrefour complexe, utilisé notamment dans les processus de sélection du réseau routier (Touya, 2010).

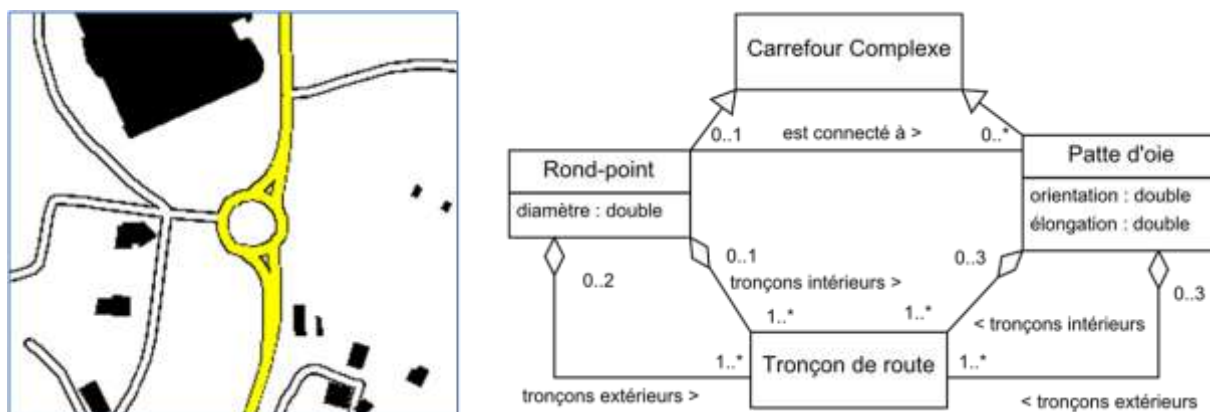


Figure C-14. Un objet meso Carrefour complexe et le diagramme de classe UML de son schéma adapté.

La Figure C-14 présentant un schéma adapté prenant en compte des objets meso fait aussi apparaître un autre type d'élément supplémentaire, les relations géographiques, ici la relation spatiale topologique qui connecte un rond-point et une patte d'oie. Les relations géographiques qui nous intéressent ici sont les relations spatiales qui jouent un rôle dans la lecture et la compréhension de la carte et doivent donc être prises en compte pour la généralisation. Il existe de nombreuses

classifications des relations spatiales, nous pouvons citer (Egenhofer et Franzosa, 1991 ; Ruas et Lagrange, 1995 ; Jones, 1997, p.25 ; Mustière et Moulin, 2002 ; Duchêne, 2004) ou (Mathet, 2000) qui introduit les relations de trajectoire comme « longer » (Figure C-15d) ou « contourner ». (Steiniger et Weibel, 2007) fournit un bon résumé de ces classifications avec une taxonomie assez détaillée distinguant au plus haut niveau les relations verticales (concerne les relations hiérarchiques déjà intégrées ici par les concepts meso), les relations de mise à jour (qui ne nous concernent pas) et les relations horizontales (relations entre deux objets géographiques, existant à un même niveau de détail, et représentant une propriété structurelle commune), dont on peut voir un aperçu en Figure C-15.

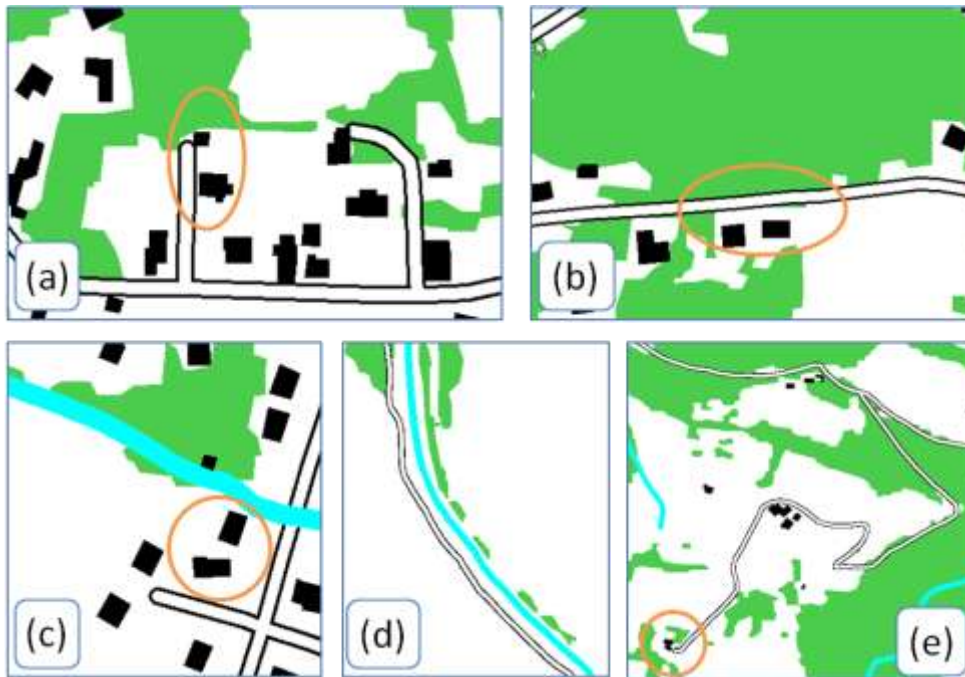


Figure C-15. Exemples de relations géographiques entre deux objets. (a) relation de positionnement relatif du bâtiment par rapport à l'impasse. (b) relation de parallélisme entre le bâtiment et la route. (c) relation d'orientation relative entre les deux bâtiments. (d) la route longe la rivière. (e) relation logique d'accès, l'impasse permet d'accéder au bâtiment.

Dans le cadre de la généralisation qui va chercher à maintenir ou caricaturer ces relations spatiales, le schéma adapté peut même contenir des relations plus complexes comme des relations topologiques entre un objet micro et un objet meso : la Figure C-16 montre la relation de positionnement relatif entre un bâtiment et un sommet que la généralisation va chercher à maintenir, nous ne voulons pas que le bâtiment descende dans la pente suite à un déplacement pour l'éloigner de la route. Ces relations doivent être explicites dans le schéma pour pouvoir être identifiées, prises en compte par un processus de généralisation et évaluées après la généralisation. Ces exemples correspondent aux relations objet-champ définies dans (Gaffuri et al, 2008), des parties du champ étant considérées ici comme des objets meso.

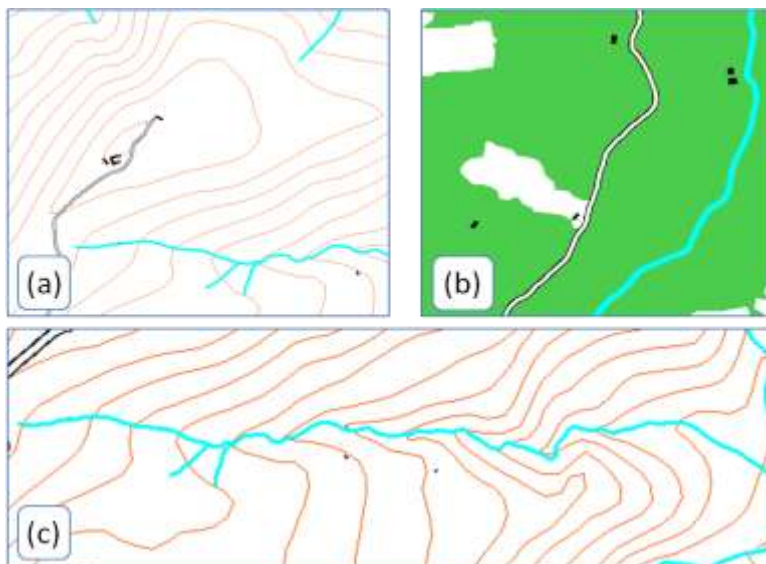


Figure C-16. Des relations plus complexes entre objet et objet meso. (a) un bâtiment est sur un sommet. (b) un bâtiment est dans une clairière. (c) la rivière passe par le talweg.

En dehors des concepts meso et des relations géographiques, le schéma adapté se doit de contenir des classes d'objets géographiques spécifiques au processus de généralisation utilisé. Ces concepts que nous appelons *procéduraux* sont également implicites originellement dans les données mais ont besoin d'être explicités comme prétraitement d'un processus donné. Si l'on prend l'exemple d'un processus issu du modèle CartACom (Duchêne, 2004), il s'agit d'expliciter dans les données les concepts de « petit compact » ou de « linéaire réseau ». Un petit compact est un objet surfacique de petite taille dont la généralisation se fera de manière rigide ou compacte, c'est-à-dire sans déformation, ce qui comprend les bâtiments, les infrastructures sportives ou les cimetières par exemple (Figure C-17). Il s'agit d'une unité de traitement, c'est-à-dire un groupe d'objets traités en partie de la même façon. Un processus issu du modèle GAEL (Gaffuri et al, 2008) nécessite ainsi l'explicitation du concept de « champ », objet continu ayant une valeur en tout point de l'espace comme le relief ou l'occupation du sol. De même, un processus de sélection du réseau routier nécessite de rendre explicites dans le schéma adapté les impasses (Touya, 2010), avec un schéma pouvant gérer les cas d'impasses complexes (Figure C-17).

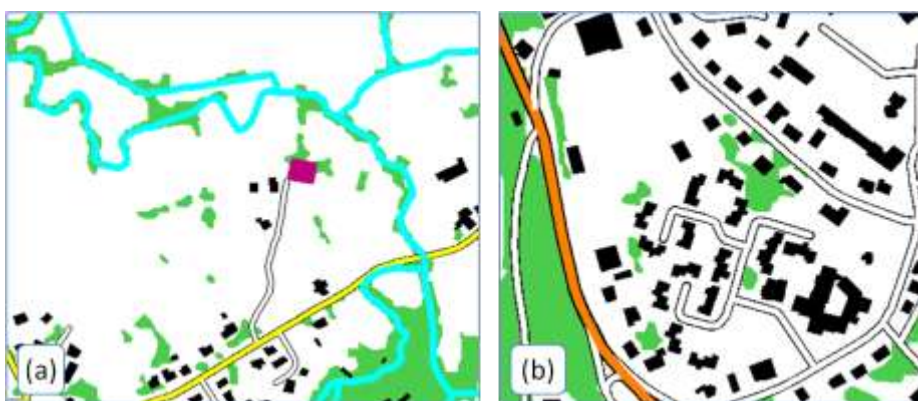


Figure C-17. Deux exemples de concepts procéduraux. (a) le petit compact concerne autant les bâtiments que le cimetière (en magenta) dans CartACom. (b) une impasse complexe est un concept nécessaire à la sélection du réseau routier.

Le schéma de données adapté à la généralisation contient des concepts meso, des relations géographiques spatiales ou sémantiques et des concepts procéduraux.

C.4.4. Que doit-on mettre dans une ontologie de la généralisation cartographique ?

L'ontologie de la généralisation cartographique ayant pour objectif notamment de résoudre l'hétérogénéité sémantique des données géographiques, une telle ontologie doit d'abord être une ontologie des objets géographiques intégrant des concepts comme les bâtiments, les tronçons de route, les forêts. Afin de pallier à l'hétérogénéité sémantique des données, cette ontologie doit également contenir tous les concepts pouvant être inclus dans un schéma adapté, c'est-à-dire les concepts meso, les relations géographiques et les concepts procéduraux utiles à la généralisation. De plus, le recours à l'ontologie ayant pour objectif de permettre une interopérabilité complète entre les composants et les autres ressources du modèle CollaGen, un certain nombre d'autres concepts doivent être ajoutés à l'ontologie :

- les liens du schéma adapté,
- les espaces géographiques,
- les propriétés des concepts géographiques,
- le voisinage thématique fonctionnel,
- une taxonomie des opérateurs,
- une taxonomie des processus de généralisation.

Tout d'abord, nous avons vu précédemment que le schéma adapté possédait de nouvelles classes et propriétés mais aussi de nouveaux **liens entre les nouvelles classes** et les anciennes comme le lien « a pour tronçons incidents » entre un rond point et les tronçons de route qui mènent à ce rond point. Si le développeur de CollaGen souhaite rendre possible la saisie de spécifications sur ces nouveaux concepts en paramètres d'un processus ou s'il souhaite que CollaGen puisse évaluer la généralisation de ces concepts, il devra développer des méthodes utilisant ces liens mais il devra le faire de façon générique (c'est-à-dire sans connaître la manière exacte selon laquelle le lien est implémenté dans le SGBD contenant les données). Pour cela, il devra manipuler uniquement les liens à travers l'ontologie, ce qui implique leur présence dans l'ontologie.

Le modèle CollaGen repose sur le découpage des données en **espaces géographiques**, ceux-ci doivent donc être identifiés dans l'ontologie comme des entités géographiques spécifiques. Dans le cas d'espaces thématiques, nous proposons de faire figurer dans l'ontologie des restrictions sur les thèmes de l'ontologie concernée par cet espace, grâce aux outils de Description Logique présents dans les langages de description d'ontologie comme OWL. Nous proposons également de faire figurer dans l'ontologie les connaissances permettant de déterminer si un espace géographique est un espace clé dans un contexte de carte donné : par exemple, nous faisons figurer dans l'ontologie que les espaces urbains sont clés dans le contexte des cartes topographiques (les espaces urbains doivent être utilisés quels que soient les processus disponibles).

Selon (Barde, 2005), une des différences entre les ontologies et les taxonomies réside dans la présence de **propriétés** sur les concepts dans les ontologies. (Ruas, 1999) explique que les

spécifications d'une carte portent sur des types d'objets (par exemple les bâtiments) et sur un caractère de ces objets (par exemple leur aire ou leur granularité). Si l'on souhaite utiliser l'ontologie comme support de vocabulaire partagé pour formaliser les spécifications, les caractères relatifs aux concepts géographiques de l'ontologie doivent être ajoutés à l'ontologie. Par exemple, la Figure C-18 est un extrait de l'ontologie de notre prototype de CollaGen montrant que le caractère de *granularité* est défini sur les concepts *cours d'eau*, *limite côtière*, *petit compact*, *route*, *surface d'eau* et *végétation*.

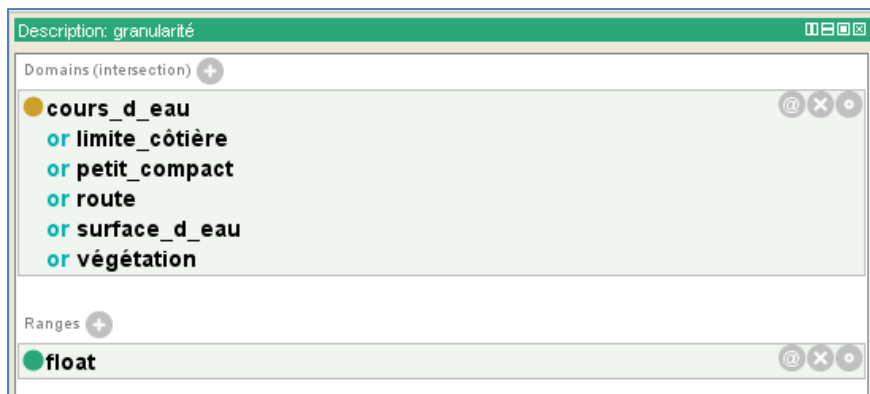


Figure C-18. Extrait de l'ontologie du prototype CollaGen montrant les concepts sur lesquels est défini le caractère de granularité.

Nous avons proposé dans la partie B.4.2 de définir pour les espaces thématiques les concepts possédant un **voisinage thématique fonctionnel** afin de faciliter la prise en compte d'effets de bord : si l'espace est généralisé, CollaGen vérifie directement si des objets des thèmes voisins sont affectés et diffuse la généralisation sur ces objets. Nous proposons d'identifier tous les voisinages thématiques possibles dans l'ontologie, l'utilisateur choisissant ceux qu'il souhaite activer dans sa généralisation collaborative. Nous proposons de réaliser cet ajout en ajoutant une relation « voisinage thématique fonctionnel » dans l'ontologie (« object properties » en OWL) entre le concept « espace géographique » et « entité géographique » (concept parent de toutes les entités géographiques). Pour chaque espace géographique, nous ajoutons alors une restriction logique sur cette relation pour décrire les thèmes voisins de cet espace. Par exemple, le concept d'espace géographique « réseau routier » de l'ontologie a une restriction du type « les voisins thématiques sont tous des forêts ou des stations services » qui signifie que le réseau routier a deux voisins thématiques, les forêts et les stations services, avec lesquels il partage une relation spécifique notamment pour la gestion des effets de bord.

En plus des différentes entités géographiques et des propriétés et relations définies dessus, nous proposons que l'ontologie de la généralisation cartographique soit dotée d'une **taxonomie des opérateurs de généralisation**. Lors du projet européen EuroSDR sur l'état de l'art des logiciels de généralisation cartographique du commerce (Stoter et al, 2009 ; 2010), il a été remarqué que plusieurs des quatre agences nationales de cartographie participant aux tests avaient exprimé certaines de leurs spécifications en termes d'opérations de généralisation à effectuer ou à éviter. Par exemple, certains demandaient de ne jamais agréger des bâtiments et d'autres conseillaient de généraliser les bâtiments par symbolisation hors des zones urbaines. Si l'on souhaite formaliser ces spécifications, il faut intégrer dans l'ontologie une taxonomie des opérateurs de généralisation comme celle présentée en A.1.2.

Enfin, les derniers concepts à intégrer dans notre ontologie de la généralisation cartographique sont ceux issus d'une **taxonomie des processus de généralisation**. Pour décrire simplement les capacités des processus de généralisation automatiques (voir C.7) et pour formaliser les règles d'orchestration (voir C.8) qui servent à guider le composant d'orchestration, nous proposons d'avoir recours à une taxonomie des modèles de généralisation et des processus issus de ces modèles. Dans un premier temps, cette taxonomie servira de vocabulaire partagé mais nous envisageons aussi de l'enrichir avec des propriétés spécifiques aux processus comme leur efficacité ou leur type de résolution (pas à pas ou globale).

Une ontologie de la généralisation cartographique doit contenir l'ensemble des types d'entités géographiques représentées sur une carte mais aussi les entités du schéma adapté et leurs relations, les espaces géographiques, les caractères des entités comme propriétés des concepts, les voisinages thématiques fonctionnels et deux taxonomies des opérateurs et des processus de généralisation. Cette ontologie sert de support à l'interopérabilité de tous les composants et toutes les ressources du modèle CollaGen.

C.4.5. Modélisation de l'ontologie de généralisation

A partir des informations que cette ontologie doit contenir d'après la partie précédente, nous proposons une structure pour modéliser l'ontologie de généralisation dans CollaGen (Figure C-19). Ce modèle est structuré autour d'une taxonomie des concepts topographiques comme celle de l'IGN (Abadie et Mustière, 2010) que nous avons utilisée lors de notre implémentation de CollaGen (voir E.3.2).

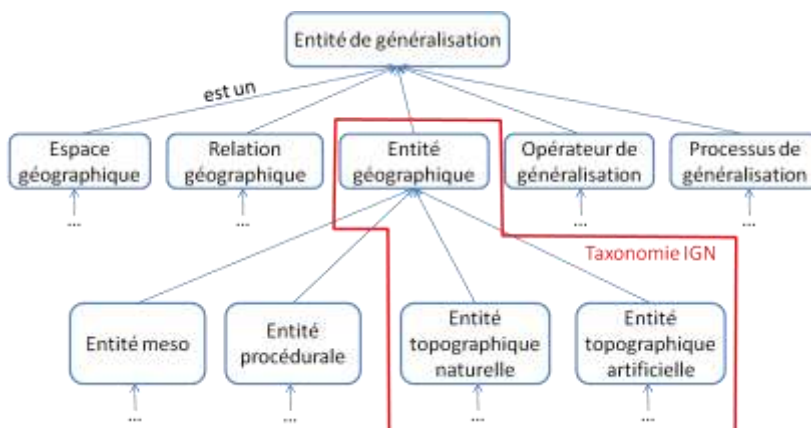


Figure C-19. Modèle de l'ontologie de généralisation proposé dans CollaGen à partir d'une taxonomie des concepts topographiques.

Afin d'intégrer l'information contenue dans cette ontologie dans les modèles formels que nous présentons dans les parties suivantes, nous avons basculé sur un point de vue orienté-objet plutôt qu'ontologique car ces modèles sont eux-mêmes orientés-objet. Des transformations sont nécessaires pour obtenir un modèle objet de l'ontologie (Figure C-20). Nous avons donc décidé de créer des objets intermédiaires dans une classe abstraite appelée *Concept de généralisation* qui correspond d'un point de vue objet au concept *entité de généralisation* de l'ontologie: CollaGen manipule uniquement des objets de cette classe, chacun correspondant à un concept de l'ontologie et donnant accès à sa hiérarchie de concepts, ses propriétés et ses relations. Cette classe, *Concept de*

généralisation, est spécialisée en trois sous-classes : *Concept géographique*, *Concept espace géo* et *Concept processus*. La classe *Concept géographique* contient tous les concepts correspondant à une entité géographique pouvant être représentée dans la base de données (y compris les relations géographiques qui possèdent leur classe spécialisée spécifique, la classe *Relation géographique* et les concepts meso ajoutés après enrichissement). La classe *Relation géographique* permet notamment de stocker les concepts sur lesquels est restreinte la relation (ce lien est généré à partir des axiomes de l'ontologie). La classe *Concept espace géo* contient tous les concepts d'espace géographique de l'ontologie. Enfin, la classe *Concept processus* contient tous les concepts des taxonomies des opérateurs et des processus, qui ne sont pas utilisés de la même manière que les concepts géographiques.

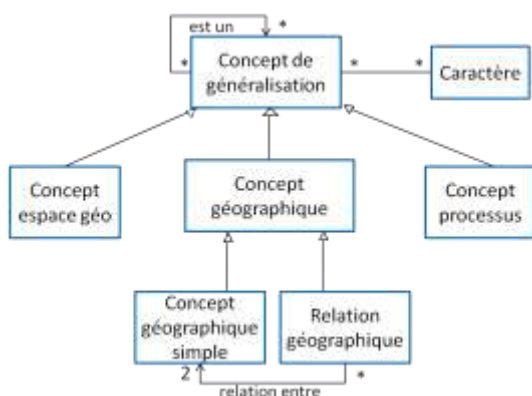


Figure C-20. Diagramme objet (en UML) permettant de manipuler l'ontologie dans CollaGen.

C.4.6. Discussion

Le développement des outils ontologiques à travers le web sémantique mais aussi beaucoup d'autres domaines scientifiques ont poussé la recherche dans le domaine à réfléchir sur des ontologies de haut niveau, selon la classification de Guarino (Guarino, 1998). Il s'agit d'ontologies de concepts très abstraits sur lesquels peuvent s'appuyer des ontologies de bas niveau comme les ontologies du domaine ou d'application comme celle que nous proposons. Il existe ainsi plusieurs propositions d'ontologies de haut niveau qui sont utilisées comme support d'ontologies de bas niveau ou de modèles conceptuels. Nous pouvons citer parmi les plus répandues DOLCE (Gangemi et al, 2002 ; Masolo et al, 2003), BFO (Smith, 1998 ; Smith et Grenon, 2004), SUMO (Niles et Pease, 2001) et OCHRE (Schneider, 2003). Dans le domaine de l'information géographique, un consensus récent propose de s'appuyer sur DOLCE car son parti pris de modéliser les concepts cognitifs et linguistiques correspond bien à la représentation de l'information géographique (Sinha et Mark, 2010). Appuyer une ontologie du domaine sur une ontologie de haut niveau permet de créer des ontologies plus robustes et plus cohérentes sémantiquement (Bittner, 2007) en plus de faciliter l'interopérabilité avec d'autres ontologies s'appuyant sur la même ontologie de haut niveau. Il serait donc intéressant que notre ontologie de la généralisation cartographique spécialise les concepts de DOLCE pour que notre ontologie soit plus robuste et interopérable. Ceci ne fait pas partie de nos propositions, non par choix scientifique mais par manque de temps, car modéliser une ontologie robuste n'est pas le cœur de notre travail de thèse. Ce manque serait donc à combler dans le cadre de travaux futurs pour améliorer le modèle CollaGen.

C.5. UN MODELE FORMEL DE CONTRAINTES DE GENERALISATION

L'analyse fonctionnelle de la généralisation collaborative proposée au chapitre B a fait apparaître le besoin d'un modèle formel pour représenter les spécifications de la carte à généraliser. Ces spécifications formelles doivent servir de format pivot pour les paramètres des processus de généralisation disponibles mais doivent également servir de support unique à l'évaluation pour garantir l'homogénéité de traitement entre les différents espaces géographiques (Figure C-21).

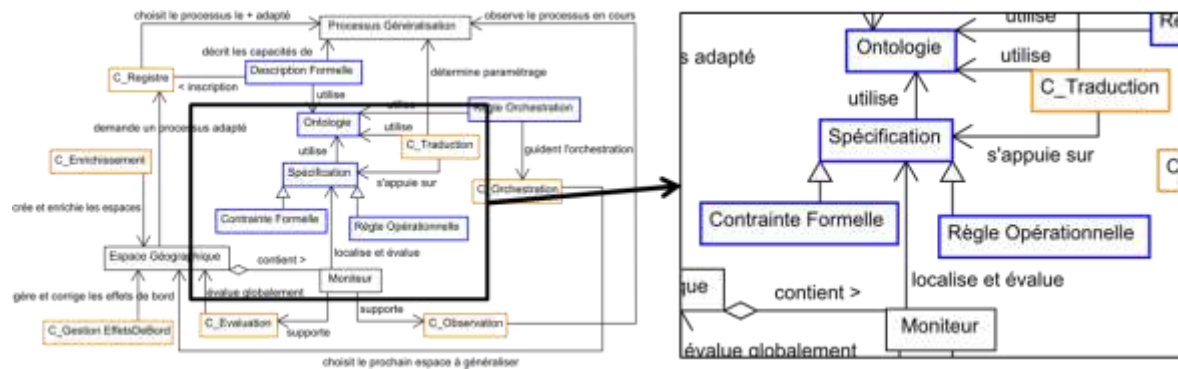


Figure C-21. Les contraintes formelles sont une partie des spécifications formelles qui servent à paramétrer les processus de manière générique et qui garantissent une évaluation homogène grâce aux moniteurs.

Les spécifications d'une carte généralisée peuvent être des consignes en texte libre comme pour la Figure C-22 qui représente un extrait des spécifications de la carte au 1 : 100 000 de l'IGN, la formalisation est donc une étape importante. Nous proposons, pour formaliser les spécifications, deux modèles : un modèle formel de contraintes de généralisation pour la grande majorité des spécifications et un modèle de règles opérationnelles pour certains cas particuliers. Nous présentons d'abord le modèle formel de contraintes de généralisation puis le modèle de règles dans la partie suivante.

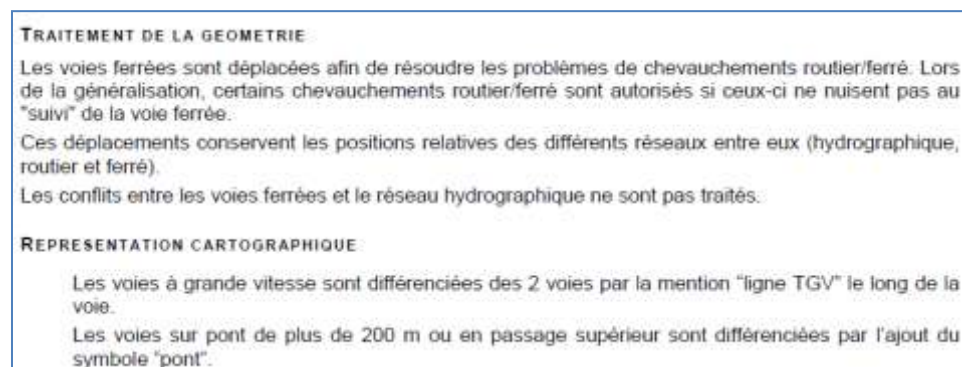


Figure C-22. Extrait des spécifications des tronçons de voie ferrée pour la carte Top100 de l'IGN.

Les premières tentatives pour formaliser les spécifications d'une carte à généraliser visaient à concevoir un système expert composé de règles du type condition/action. (Beard, 1991) a montré que le choix de l'action dépendant fortement du contexte, il était illusoire d'essayer de réaliser un tel système expert, le nombre de règles particulières étant trop important et toujours incomplet. La proposition est alors de basculer sur une approche basée sur les contraintes : la règle « si l'aire du bâtiment est inférieure à 0,2 mm² carte, il faut le grossir » est plutôt représentée par la contrainte « l'aire du bâtiment doit être supérieure à 0,2 mm² carte » (Figure C-22). Cette approche a été suivie

par la majorité des travaux postérieurs ce qui a amené à de nombreuses propositions pour représenter les contraintes.

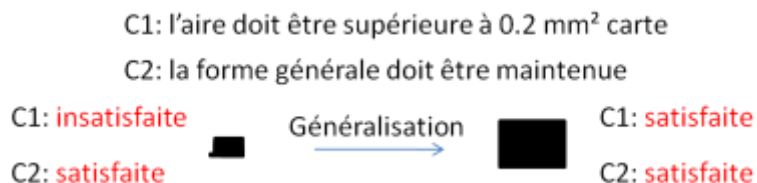


Figure C-23. Deux contraintes définies sur ce bâtiment : elles sont satisfaites après généralisation.

C.5.1. Les Classifications de contraintes de généralisation

Il existe plusieurs méthodes pour classer les contraintes. (Beard, 1991) a été la première à réaliser une typologie des contraintes dont elle distingue **quatre** types :

- les contraintes graphiques qui correspondent aux contraintes de lisibilité des objets cartographiques.
- les contraintes structurales qui permettent de maintenir les informations pertinentes.
- les contraintes d'application qui dépendent de l'objectif du produit dérivé.
- les contraintes procédurales qui contrôlent le guidage du processus.

(Ruas et Plazanet, 1996) propose une typologie de **quatre** types orientée cette fois sur leur nature (ce sur quoi elles portent) :

- les contraintes de lisibilité concernent les seuils de lisibilité des objets géographiques.
- les contraintes de forme concernent la préservation de la forme d'un objet ou d'un groupe.
- les contraintes spatiales concernent les positions absolues et relatives des objets (contient entre autres les contraintes topologiques et les contraintes de proximité).
- les contraintes sémantiques correspondent aux informations sémantiques des spécifications (contient des contraintes de quantité et des contraintes fonctionnelles).

(Weibel et Dutton, 1998) distinguent **cinq** types de contraintes de généralisation :

- les contraintes graphiques qui concernent les tailles et les proximités minimales des objets.
- les contraintes topologiques.
- les contraintes de structure qui correspondent au maintien des structures spatiales et sémantiques.
- les contraintes de Gestalt qui concernent les contraintes d'esthétisme, d'équilibre et de distribution des objets.
- les contraintes de processus qui contrôlent le guidage du processus.

(AGENT, 1998) reprend les quatre premières de la typologie précédente en modifiant quelque peu l'étendue de chaque type. (Ruas, 1999) se place sur un autre plan et distingue **trois** classes de contraintes suivant le type d'entité sur laquelle la contrainte s'applique :

- les contraintes micros s'appliquent sur les objets individuels.
- les contraintes meso s'appliquent sur un groupe d'objets.

- les contraintes macros s'appliquent sur une population d'objets.

(Galanda, 2003) présente une typologie pour les contraintes de généralisation de polygones. Elle se compose de **quatre** types de contraintes :

- les contraintes métriques concernent les problèmes de perception et lisibilité.
- les contraintes topologiques concernent le maintien ou la création de relations topologiques entre les objets.
- les contraintes structurales visent à préserver les structures spatiales et sémantiques.
- les contraintes procédurales concernent le processus de généralisation.

Enfin, (Burghardt et al, 2007) présente une dernière classification des contraintes en **sept** catégories (Figure C-24). Chacune des sept catégories (sur la 3^{ème} ligne du schéma) peut se spécialiser en portant sur un type particulier de géométrie, sur un ou plusieurs objets, à l'intérieur d'une classe ou entre des objets de classes différentes.

Nous proposons d'utiliser conjointement deux classifications des contraintes. Tout d'abord, nous distinguons les contraintes micro (sur un seul objet), meso (sur un groupe d'objets), macro (sur une population d'objets) et relationnelles (sur la relation spatiale entre deux objets, terme issu de (Duchêne, 2004)). Puis nous utilisons une version simplifiée de la classification de (Burghardt et al, 2007) en séparant les contraintes de lisibilité et les contraintes de préservation car leur expression est très différente.

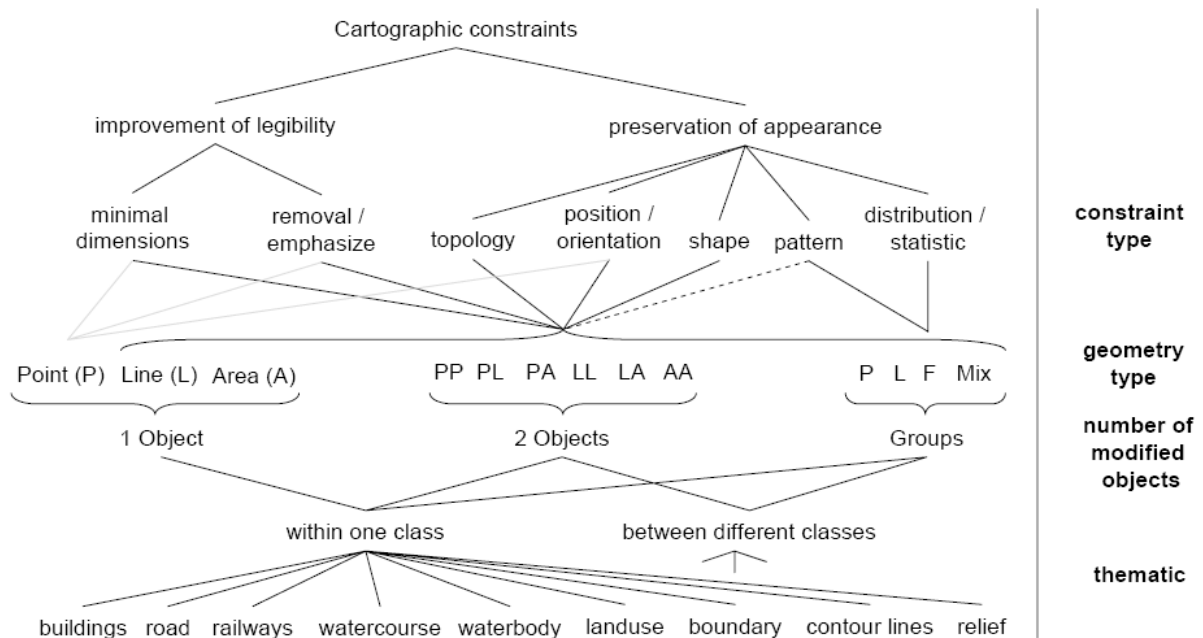


Figure C-24. Typologie des contraintes selon la nature, le type de géométries, le nombre d'objets et la thématique (Burghardt et al, 2007).

C.5.2. Les Formalismes de contraintes existants

Qu'ils suivent l'une ou l'autre des classifications précédentes, plusieurs formalismes ont été proposés pour, soit saisir des contraintes de généralisation, soit les utiliser dans un processus automatique. Un des formalismes les plus importants est issu des travaux d'Anne Ruas et ensuite du projet AGENT (Ruas, 1999 ; Barrault et al, 2001) et les contraintes y sont des objets liés aux agents avec des attributs et des méthodes pour déterminer leur but et leur satisfaction (Figure C-25). Ce modèle objet permet à la fois d'exprimer la contrainte et de l'évaluer par les méthodes intégrées au modèle. Il ne s'agit pas de fait de contraintes mais d'objets décrivant le respect d'une contrainte par un objet géographique. Ce modèle de contrainte a ensuite été enrichi pour gérer des contraintes relationnelles et des contraintes objet/champ (Duchêne, 2004 ; Gaffuri et al, 2008).

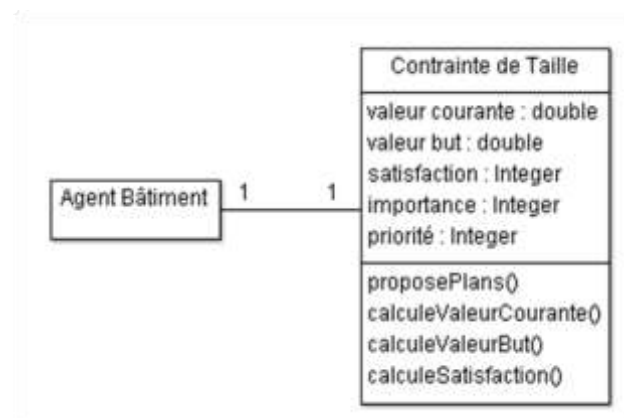


Figure C-25. Modèle objet d'une contrainte micro de taille d'un bâtiment dans le formalisme agent (Ruas et Duchêne, 2007).

Le projet EuroSDR d'état de l'art des logiciels du commerce fournissant des outils de généralisation automatiques, auquel nous avons participé, a également travaillé sur l'expression des contraintes de généralisation de manière formelle (Burghardt et al, 2007 ; Stöter et al, 2010). Un formulaire Excel a été produit, dans lequel les agences de cartographie nationales participant au test (dont l'IGN) devaient saisir leurs spécifications sous forme de contraintes (Figure C-26). Ce formulaire propose de distinguer le caractère contraint, la condition à respecter ou la condition pour être concerné et intègre bien les valeurs numériques permettant de satisfaire la contrainte, comme on le voit sur la troisième contrainte de la figure. Le défaut de ce formulaire est qu'il est plutôt semi-formel, beaucoup de champs étant remplis par du texte libre, ce qui rend l'interprétation de ces contraintes assez floue.

| Constraint ID | Generic Constraint ID | Constraint Type | Geometry | Class | Condition for object being concerned with this constraint | Constrained property | Condition depends on initial value? | Condition to be respected (do not forget the units) | Action | Importance of constraint (1 to 5, 1 is less important) | Exception |
|---------------|-----------------------|----------------------|------------------|----------|---|----------------------|-------------------------------------|---|---|--|-----------|
| IGN-1-1 | EuroSDR-1-29 | Model generalisation | Any | | | Class | yes | target class = initial class | | 5 | |
| IGN-1-3 | EuroSDR-1-16 | Model generalisation | line and polygon | | Initially, no self-intersection | intersection | yes | no self-intersection must be created | | 5 | |
| IGN-1-4 | EuroSDR-1-1 | Minimal dimensions | polygon | Building | | Area | no | target area > 0.2 map mm ² | IF initial area < 0.012 map mm ² THEN {delete} | 5 | |

Figure C-26. Extrait du formulaire Excel permettant d'exprimer les contraintes de l'IGN dans le test EuroSDR.

De plus, concernant la formalisation des contraintes dans d'autres domaines que la généralisation, nous pouvons nous intéresser au langage standard OCL (OMG, 2003). Il s'agit d'un langage qui permet d'exprimer des contraintes sur un diagramme de classes UML. Les invariants qui sont des contraintes définies sur tous les objets d'une classe peuvent ainsi être utilisés pour formaliser une contrainte de généralisation (Zhang et al, 2008 ; Stöter et al, 2008). Des extensions spatiales d'OCL ont notamment été proposées pour faciliter l'expression de contraintes spatiales comme les contraintes d'intégrité dans les bases de données (une contrainte d'intégrité est par exemple « un objet point d'intérêt doit toujours se trouver à l'intérieur d'une zone d'intérêt) (Werder, 2009 ; Pinet et al, 2009). Avec ces extensions spatiales, la contrainte de généralisation suivante « le diamètre des ronds-points doit être supérieur ou égal à 100 m » peut s'exprimer en OCL de la manière suivante :

« context RondPoint inv : self.geometry->perimeter/3.14 >= 100.0 »

(Foerster et al, 2008) propose également d'exploiter OCL pour décrire les capacités des opérateurs de généralisation, pour savoir à quoi s'attendre en sortie d'un opérateur donné comme la simplification ou l'agrégation. L'avantage de ce formalisme est qu'il s'agit d'un standard et qu'il est déjà intégré à de nombreux logiciels. Il est ainsi possible de générer automatiquement du code ou des requêtes SQL à partir du texte des contraintes OCL. Seulement, seules les extensions spatiales nous permettraient d'exprimer les contraintes de généralisation et ces extensions ne font pas partie du standard. De plus, il est parfois très complexe d'exprimer une contrainte en OCL et les contraintes sont alors difficiles à comprendre.

Aucun des formalismes existants ne répond complètement à nos attentes donc nous avons développé notre propre modèle formel de contraintes de généralisation qui synthétise les formalismes issus des projets AGENT et EuroSDR. Le langage OCL n'est finalement pas utilisé du fait de l'absence de standard au niveau de l'extension spatiale mais une perspective serait de réaliser un outil de traduction automatique de notre formalisme vers OCL et vice versa.

C.5.3. Proposition de modèle formel de contraintes de généralisation

Nous proposons donc dans CollaGen de formaliser les contraintes de généralisation dans un modèle orienté objet (comme dans AGENT) qui reprend, formalise et complète le formulaire issu du projet EuroSDR (Figure C-27). Ainsi, une contrainte de généralisation est un objet de la classe *Contrainte de Généralisation*, caractérisé par son nom, son importance, le *concept* de l'ontologie concerné, le *caractère* sur lequel elle porte, un *type d'expression*, un *critère de sélection* et enfin un *espace d'application*. Afin d'expliquer et d'illustrer ce modèle, nous allons utiliser cinq contraintes :

- (C1). L'aire des bâtiments doit être supérieure ou égale à 0.2 mm² carte.
- (C2). Un bâtiment presque parallèle à une route (angle < 15°) doit être rendu parfaitement parallèle.
- (C3). En zone de montagne, les bâtiments doivent garder leur altitude à 20 m près.
- (C4). Les alignements de bâtiments doivent être maintenus.
- (C5). Le réseau hydrographique doit être réduit de 20% en longueur.

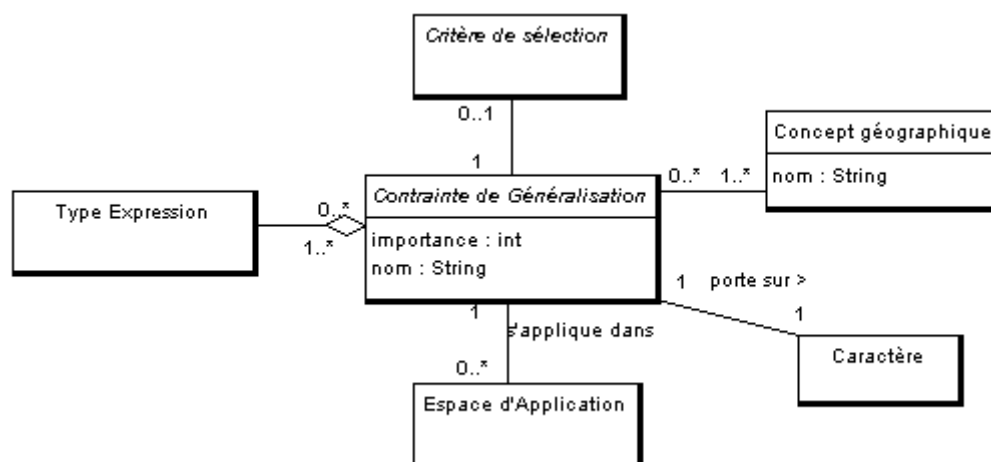


Figure C-27. Vue générale du diagramme de classe UML de notre modèle formel de contraintes de généralisation.

Tout d'abord, *Contrainte de Généralisation* est une classe abstraite et une contrainte est donc un objet d'une des quatre classes qui en héritent, *Contrainte Micro*, *Contrainte Meso*, *Contrainte Relationnelle* et *Contrainte Macro*. Si l'on reprend nos quatre exemples, (C1) est une contrainte micro, (C2) et (C3) sont des contraintes relationnelles (relation d'orientation relative entre bâtiment et route et relation d'altitude entre bâtiment et relief), (C4) est une contrainte meso et (C5) est une contrainte macro. Le *nom* de la contrainte doit désigner si possible sans ambiguïté la contrainte. Par exemple, le nom de (C1) pourrait être « aire minimum des bâtiments ». L'*importance* donne une évaluation de l'importance de la contrainte dans le résultat final (définition reprise de (Ruas, 1999)) : si une contrainte peu importante est non satisfaite après la généralisation, cela n'influencera que peu l'évaluation de la généralisation. L'importance, comme le proposait (Ruas, 1999) est un entier compris entre 1 (peu important) et 5 (très important).

Le *concept géographique* fait directement référence à un concept de l'ontologie. Dans le cas des contraintes relationnelles, ce concept est une relation géographique de l'ontologie à laquelle nous ajoutons les deux concepts concernés par la contrainte. Le *caractère* de la contrainte est également tiré de l'ontologie, il s'agit d'un des caractères définis dans l'ontologie sur le concept. Pour (C1), le concept est « bâtiment » et le caractère « aire ». Pour (C2), le concept est « relation d'orientation relative » entre les concepts « bâtiment » et « route » et le caractère « différence d'orientation principale » (qui traduit l'idée de parallélisme) ; ce cas montre comment le modèle force (et aide) à formaliser des contraintes dont l'intitulé est flou. Pour (C3), le concept est « relation d'altitude » entre les concepts « bâtiment » et « champ relief » et le caractère « altitude ». Pour (C4), le concept est « alignement de bâtiments », concept meso de l'ontologie et il n'y a pas de caractère concerné puisqu'il ne s'agit pas d'une propriété de l'alignement mais de son existence même après la généralisation. Enfin, pour (C5), le concept est « tronçon de cours d'eau » mais la contrainte concerne la population des objets de ce concept et le caractère est une propriété de population, la « quantité de longueur ». Les caractères macro ne sont pas issus de l'ontologie : un choix restreint est proposé en fonction du type de géométrie de la population (par exemple, « quantité en surface » et « quantité en nombre d'objets » pour les populations surfaciques et seulement « quantité en nombre d'objets » pour les populations de géométrie ponctuelle).

Le Type d'expression de la contrainte

Le corps de la contrainte réside principalement dans son *type d'expression*. La Figure C-28 montre plus précisément le modèle que nous avons adopté pour les types d'expression de contrainte de généralisation. Les contraintes de différents types s'expriment à l'aide d'un opérateur, sur une propriété d'un type d'objet, exprimée dans une unité. En nous appuyant sur les contraintes exprimées durant le projet EuroSDR et sur l'expérience des recherches passées au laboratoire, nous avons recensé cinq types d'expression différents pour les contraintes de généralisation :

- **Seuil** : *caractère.valeur()* **motClé** **valeur** **unité** ((C1) : *aire.valeur()* **>=** **0.2** **mm²** **carte**)
- **Marge** : *caractère.valeur()* **motClé** *caractère.valeurInitiale()* à **valeur%** près ((C3) : *altitude.valeur()* **=** *altitude.valeurInitiale()* à **15%** près)
- **Configuration** : *conceptMeso* **motClé** ((C4) : *alignement de bâtiment* **maintenu**)
- **Contrôle** : *caractère.valeur()* (**motClé** **valeur** **unité**) = faux (par exemple pour les routes, *empâtement.valeur()* ne doit pas être **> 0** **sans** **unité**)
- **Réduction** : *caractère.valeur()* **motClé** *caractère.valeurInitiale()* * **valeur** / 100 ((C5) : *quantité longueur.valeur()* = *quantité longueur.valeurInitiale()* * **80** / 100)

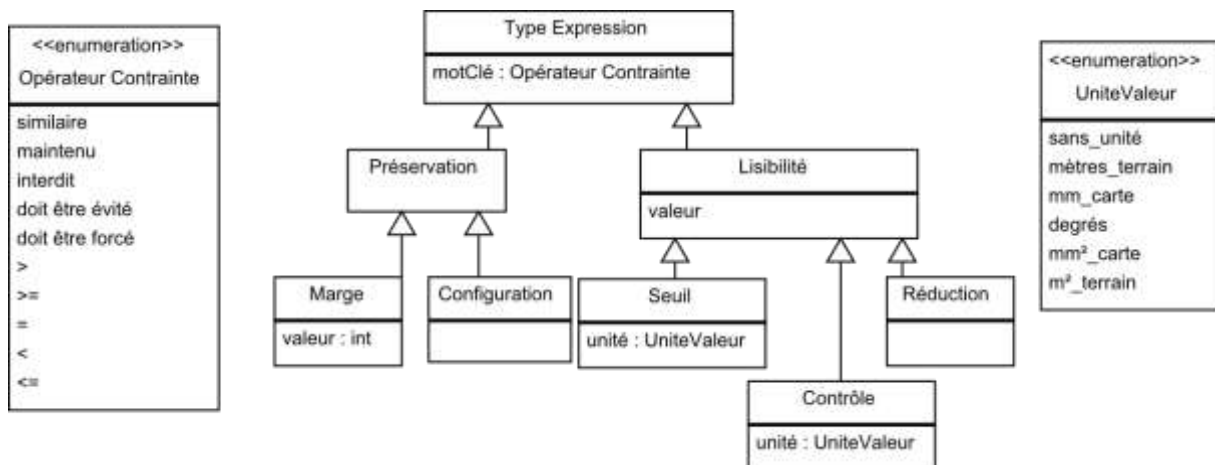


Figure C-28. Diagramme UML plus détaillé du type d'expression d'une contrainte.

On peut noter sur la Figure C-28 que l'énumération des valeurs possibles de mots clés ne se limite pas aux simples =, >, <. D'autres mots clés ont été ajoutés pour plus d'expressivité. Ainsi, *similaire* peut être utilisé dans les expressions de marge pour des contraintes volontairement floues (sans marge précise) comme par exemple « la forme de la forêt doit être similaire à la forme initiale ». L'utilisateur peut ainsi exprimer des contraintes avec deux niveaux de spécialisation suivant ses connaissances du résultat qu'il souhaite. S'il sait précisément comment doit être contraint son résultat, il peut contraindre la forme par des contraintes de type *Marge* sur l'élongation ou la concavité ; s'il n'a qu'une connaissance floue de ce qu'il souhaite, il peut utiliser le mot clé *similaire* sur le caractère de forme (*l'élongation est une propriété* de forme dans l'ontologie). Les quatre autres mots clés sont réservés aux expressions de configuration : *maintenu* signifie que le concept meso est maintenu quand il existe à l'état initial ; *interdit* signifie que le concept meso ne doit pas apparaître dans les données généralisées (par exemple, les carrefours complexes type rond-point ou patte d'oie sont interdits signifie qu'il ne doivent plus être visibles de manière implicite ou explicite sur la carte finale) ; *doit être évité* est une version plus nuancée de *interdit*, car certaines exceptions sont

tolérées ; enfin *doit être forcé* concerne les concepts meso qui n'existent pas à l'état initial et que l'on veut voir apparaître de manière explicite dans la carte finale.

On peut noter également que nous proposons une formalisation des unités pour permettre ensuite des conversions automatiques simples dans le paramétrage des processus. Les valeurs de l'expression peuvent être sans unité, en unité carte (mm ou mm^2), en unité terrain (m ou m^2) ou en unité angulaire (degrés).

Ce modèle de type d'expression permet de couvrir un panel assez exhaustif des contraintes, du moins dans notre cas d'application des cartes topographiques. Une perspective serait d'essayer d'exprimer encore plus de contraintes dans ce modèle pour tester sa flexibilité et d'éventuellement l'enrichir le cas échéant. En effet, (Bard, 2004) propose, dans le cadre de l'évaluation de la généralisation par ses contraintes, d'exprimer le comportement attendu de la contrainte sous la forme d'une fonction d'évaluation. Cette fonction d'évaluation donne la valeur attendue du caractère contraint en fonction de sa valeur initiale. Ces fonctions d'évaluation sont donc un moyen plus fin pour modéliser le type d'expression d'une contrainte. Si nous prenons l'exemple de (C1), une fonction d'évaluation permet d'en exprimer une version plus fine qui va permettre de maintenir au mieux les différences initiales d'aire (Figure C-29). Un enrichissement de ce modèle serait donc de laisser la possibilité à l'utilisateur de saisir une fonction d'évaluation comme type d'expression de sa contrainte quand il souhaite en faire une description avancée.

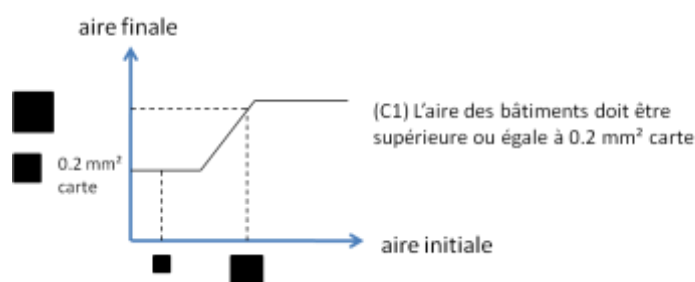


Figure C-29. Exemple de fonction d'évaluation décrivant plus finement l'expression de la contrainte (C1).

Critère de sélection et restriction

Par ailleurs, notre modèle formel de contraintes comprend également un *critère de sélection* associé à chaque contrainte qui correspond au champ « condition for object being concerned by constraint » du formulaire EuroSDR. Ce critère de sélection permet de ne définir une contrainte que sur une partie des objets correspondant à un concept. Nous avons modélisé ce critère de sélection comme une suite de requêtes spatiales et/ou attributaires sur les objets du concept. Par exemple, le critère de sélection de la contrainte « les bâtiments à moins de $0,5 \text{ mm}$ carte d'une route et parallèle à 15° près de cette route doivent être adjacents » est composé de deux requêtes sélectionnant les relations d'orientation relative bâti-route dont le caractère distance est inférieur à $0,5 \text{ mm}$ carte et les relations d'orientation relative bâti-route dont le caractère différence d'orientation est inférieur à 15° . La modélisation de la requête est faite sur le même modèle que les expressions de contrainte de type seuil avec en plus une tolérance possible autour de la valeur. Des cinq exemples de contraintes, seule (C2) présente un critère de sélection qui s'exprime par une requête sur les relations d'orientation relative bâti-route dont le caractère différence d'orientation est inférieur à 15° . Pour

toutes les autres contraintes, tous les objets correspondant au concept de la contrainte sont contraints. Le critère de sélection est utilisé quand CollaGen cherche les objets concernés par la contrainte : ne sont gardés que les objets correspondant au concept de la requête dont le caractère vérifie la requête.

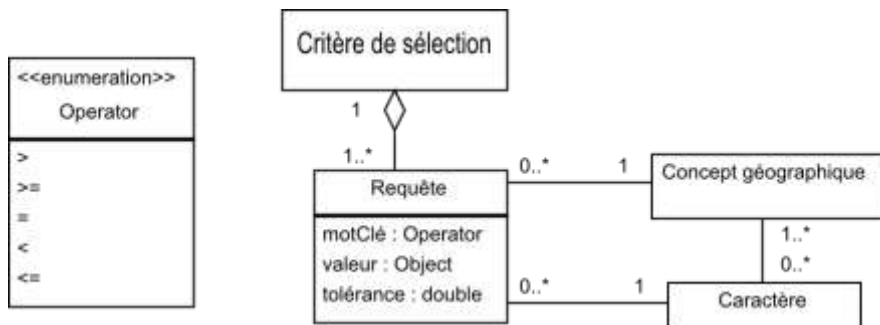


Figure C-30. Diagramme UML plus détaillé des critères de sélection d'une contrainte.

Enfin, une contrainte de généralisation est liée à un dernier élément de modélisation, l'espace d'application. En effet, une contrainte peut n'être définie qu'à l'intérieur de certains espaces géographiques et pas ailleurs. Cela peut être dû au fait que la contrainte n'a pas de sens ailleurs que dans l'espace géographique. Par exemple, la contrainte relationnelle « un bâtiment sur un sommet doit rester sur le sommet » n'a pas de sens en dehors des espaces de montagne car il n'y a pas de sommet en dehors de ces espaces (cette affirmation dépend toutefois de la définition admise d'un sommet et d'un espace de montagne). La notification d'un espace d'application peut aussi être due à une limitation opérationnelle : la contrainte a un sens partout mais chercher à la satisfaire partout n'a pas de sens, car cela rendrait plus difficile de trouver une solution satisfaisant toutes les contraintes, tout en n'améliorant que peu la qualité de la carte. (C3) est la parfaite illustration de ce fait, car maintenir l'altitude d'un bâtiment a un sens partout sur la carte, mais quel est l'intérêt de satisfaire cette contrainte dans une zone plate où les courbes de niveau sont très espacées et peu visibles (d'autant que les procédures mises en œuvre pour le calcul de cette contrainte sont gourmandes en temps et en volume) ? Nous nous contentons donc dans ce cas de restreindre cette contrainte aux espaces pour lesquels sa violation fait baisser la qualité de la carte généralisée. Il peut exister un autre cas pour lequel nous pouvons restreindre l'espace d'application, celui d'une contrainte dont nous savons par avance que le système n'arrivera jamais à la satisfaire dans certains espaces donc elle est appliquée seulement aux autres espaces. Pour illustrer ce cas, nous pouvons prendre l'exemple de la contrainte de positionnement relatif d'un bâtiment par rapport à une impasse. Cette contrainte peut être satisfaite dans les zones moyennement ou peu denses en bâtiments et routes comme les espaces périurbains, ruraux ou de montagne mais elle ne serait satisfaite qu'exceptionnellement en espace urbain où la priorité est la gestion des contraintes de taille et de proximité des objets.

En résumé, nous proposons un modèle formel orienté objet des contraintes de généralisation basé sur la classification micro/meso/macro/relationnel. Une contrainte de généralisation porte sur un concept et un de ses caractères, s'exprime formellement sous une des cinq formes proposées, peut posséder un critère de sélection et peut porter sur un espace d'application restreint. Ce modèle permet de formaliser une grande partie des spécifications existantes d'une carte topographique généralisée. Le formalisme peut être étendu si on se rend compte qu'il n'est pas suffisant pour l'expression de nouvelles contraintes.

C.5.4. Vers des contraintes de généralisation génériques

Les différentes ressources formelles présentées dans cette partie sont relativement interdépendantes. Ainsi, nous avons vu que les contraintes s'appuient sur l'ontologie et nous allons voir en C.7 que la description formelle des processus dépend à la fois de l'ontologie et des contraintes. La création de ces ressources ne peut donc pas se faire dans n'importe quel ordre. La Figure C-31 montre deux scénarios d'enchaînement des créations de ressources en fonction du nombre de personnes différentes jouant les rôles de modélisateur, développeur, fournisseur et utilisateur (voir C.2). Par défaut, l'utilisateur et le fournisseur sont différents. L'utilisateur est alors le dernier à avoir la main sur CollaGen (i.e. il se contente de généraliser ses données à partir de ses spécifications). Or le fournisseur a besoin des contraintes pour décrire formellement les processus qu'il met à disposition. Dans ce cas, il faut que le modélisateur saisisse des contraintes génériques à partir desquelles le fournisseur peut décrire ses données et l'utilisateur créer ses contraintes plus facilement. Le cas particulier où utilisateur et fournisseur sont la même personne (c'est notre cas mais il est aussi répandu dans les IGNs pour la mise en place de chaînes de production automatiques de cartes par généralisation), le problème est plus simple car la personne peut alors saisir ses spécifications avant de décrire les processus.

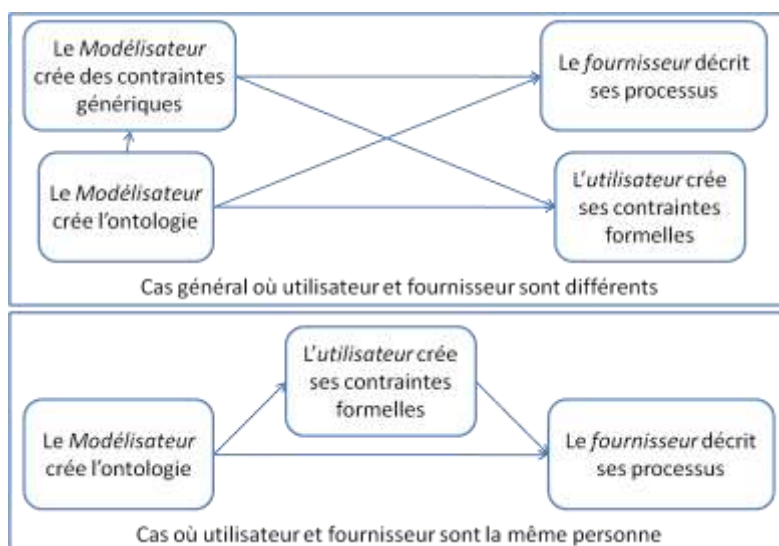


Figure C-31. Interdépendances entre les différentes ressources formelles de CollaGen suivant le nombre de personnes jouant les différents rôles.

Les contraintes génériques reprennent alors le même modèle formel de contraintes de généralisation sauf qu'elles ne traduisent pas directement des spécifications précises. Ainsi, quelques éléments du modèle sont exclus de la partie générique : les valeurs du Type d'expression, le critère de sélection et l'espace d'application. Ainsi, la version générique de (C1) serait « la contrainte d'aire minimum des bâtiments » sans préciser au-dessous de quelle valeur l'aire doit être. C'est le rôle de l'utilisateur de remplir cette valeur en fonction de ses spécifications. Au niveau conceptuel, il paraît logique d'intégrer alors ces contraintes génériques à l'ontologie mais c'est une tâche que nous avons choisi de garder en perspective de notre travail.

C.6. UN MODELE FORMEL DE REGLES OPERATIONNELLES

Nous avons vu dans la partie précédente qu'un des résultats du projet EuroSDR était d'avoir identifié que certaines agences nationales de cartographie utilisaient des spécifications conseillant ou ordonnant l'utilisation d'un opérateur de généralisation spécifique pour traiter un problème. La Figure C-26 montre comment une colonne *Action* a été ajoutée au formulaire pour gérer dans une contrainte ce type minoritaire de spécification. L'exemple de spécification de la Figure C-26 est assez significatif : « si l'aire initiale du bâtiment est inférieure à 0,012 mm carte, alors il faut systématiquement éliminer le bâtiment ». Il s'agit de cas locaux où une opération systématique est appliquée. S'il existe des cas où l'utilisateur souhaite définir des transformations systématiques sous certaines conditions, comme tendent à le prouver cet exemple ainsi que les conclusions de (Stöter et al, 2010), alors, le recours à un système de contrainte est trop lourd et pénalisant. Nous proposons donc d'utiliser dans ces cas particuliers des règles systématiques dites *règles opérationnelles* plutôt que de tout modéliser par des contraintes.

Comme pour les contraintes, nous faisons le choix d'un formalisme objet. Cependant, nous reprenons une manière classique de représenter une règle à l'aide de prémisses et d'une conclusion. Notre modèle objet reprend donc la formulation mathématique suivante d'une règle :

$$\text{Prémisse}_1 \wedge \text{Prémisse}_2 \wedge \dots \Rightarrow \text{Conclusion}$$

La Figure C-32 montre le modèle formel que nous proposons pour représenter ces règles opérationnelles en suivant cette formulation mathématique.

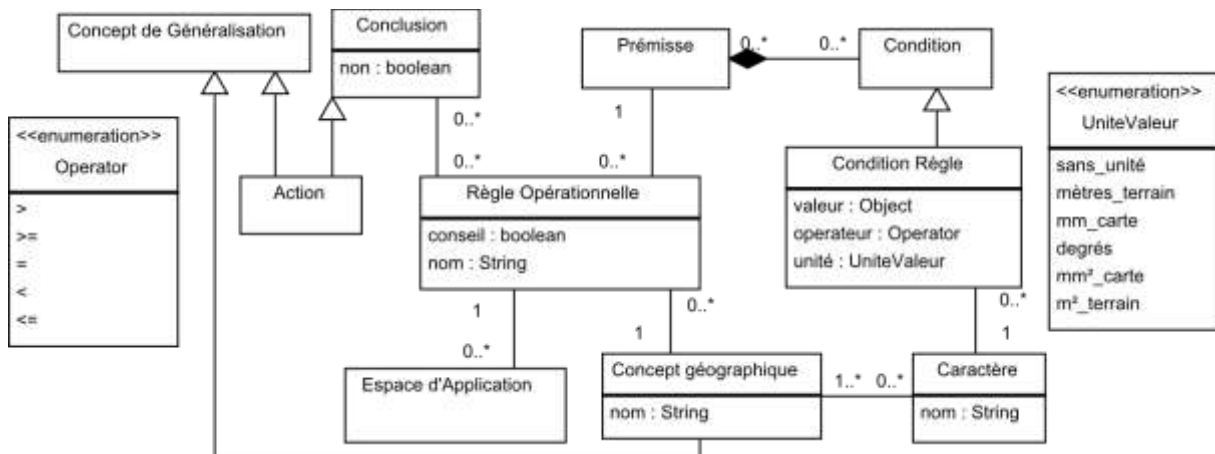


Figure C-32. Diagramme de classes UML représentant le modèle formel des règles opérationnelles.

Dans notre modèle la formulation est un peu différente car il n'y a qu'une prémisse par règle qui est composée de n *Conditions*. Une condition représente une prémisse de la formule ci-dessus. Afin d'illustrer le modèle, nous allons utiliser deux exemples de spécifications à modéliser sous forme de règle opérationnelle :

- (R1). Les bâtiments ne devraient pas être agrégés dans les zones urbaines.
- (R2). Les ronds-points de diamètre inférieur à 100 m doivent être transformés en un simple carrefour ponctuel.

Dans notre modèle formel, une règle opérationnelle est un objet de la classe *Règle Opérationnelle*, identifié par un nom. Une règle opérationnelle peut :

- **Obliger** l'utilisation d'une opération de généralisation (R2 est dans ce cas),
- **Conseiller** l'utilisation d'une opération, ce qui peut être rattaché à la distinction de (Ruas, 1999) entre les contraintes strictes et les contraintes flexibles (comme le maintien de la position initiale),
- **Déconseiller** l'utilisation d'une opération (R1 est dans ce cas),
- **Interdire** une opération.

Dans ce modèle, une condition est l'équivalent d'une requête du critère de sélection des contraintes, c'est-à-dire une condition sur un caractère du concept de la règle. Une *condition de règle* peut donc s'écrire sous la forme « caractère.valeurInitiale() **opérateur** valeur **unité** ». Si l'on reprend nos deux exemples ci-dessus, (R1) n'a pas de condition, la règle étant appliquée sur tous les objets du concept (bâtiment) et (R2) a une condition qui peut s'écrire « *diamètre.valeurInitiale()* < **100 m** **terrain** ».

La conclusion d'une règle peut être dans notre modèle de faire une action ou de ne pas faire une action en fonction de l'attribut booléen *non* de la conclusion. L'action correspond à une opération issue de la taxonomie des opérations intégrée à l'ontologie de la généralisation. La règle ne fait aucun préjugé sur l'algorithme choisi pour réaliser une opération. Pour nos deux exemples, la conclusion de (R1) est l'action *Agréger* (voir l'extrait de la taxonomie de (Mustière, 2001) en Figure A-4) avec l'attribut *non* de la conclusion à *vrai* alors que la conclusion de (R2) est l'action *Squelettiser* (Figure A-4) avec l'attribut *non* de la conclusion à *faux*.

Enfin, comme les contraintes de généralisation, ces règles disposent d'un espace d'application éventuel pour les mêmes raisons : la règle n'a de sens que dans certains espaces ; ou la règle n'est utile que dans certains espaces ; ou enfin il n'est possible de prendre en compte la règle que dans certains espaces. Pour nos deux exemples, seule (R1) dispose d'un espace d'application, l'espace urbain, car ici la règle est fautive ailleurs où une agrégation n'est pas une opération gênante.

Il faut bien se rendre compte que le fait de modéliser ces spécifications sous la forme de règle n'implique pas qu'elles seront gérées par CollaGen dans une approche Condition/Action (Harrie et Weibel, 2007). CollaGen ne modifie pas la manière dont les processus traitent la généralisation mais ne fait que choisir les processus les plus adaptés à la généralisation d'un espace géographique. Ainsi, ces règles ne sont là que pour guider le paramétrage des processus puis évaluer le résultat et sont seulement un moyen élégant et efficace pour modéliser ces spécifications. Par exemple (R2) peut être résolu en zone urbaine par un processus issu d'AGENT et qui va donc traduire cette règle en une contrainte dans son formalisme spécifique pour la prendre en compte.

En résumé, nous proposons un modèle formel orienté objet des règles opérationnelles pour compléter celui des contraintes de généralisation dans le cas de spécifications qui requièrent l'application (ou l'évitement) systématique d'une opération de généralisation particulière. Une règle opérationnelle comporte une prémisse qui est l'équivalent d'une requête sur un caractère et une conclusion qui est d'appliquer ou non une des actions décrites dans l'ontologie.

C.7. DECRIRE LES CAPACITES D'UN PROCESSUS DE GENERALISATION

Dans CollaGen, la généralisation de chacun des espaces géographiques est faite en cherchant le processus automatique de généralisation le plus adapté parmi ceux rendus disponibles par des fournisseurs. Nous avons vu en B.3 qu'il existait plusieurs types d'hétérogénéité et qu'un tel système génère de l'hétérogénéité syntactique (différences de modélisation des concepts) et sémantique (différences sur le sens des concepts). L'hétérogénéité syntactique peut être gérée grâce à l'ontologie de généralisation qui clarifie la modélisation des concepts. Par contre, l'hétérogénéité sémantique pose problème car il faut déterminer sans ambiguïté comment bien utiliser un processus donné et quels sont les résultats attendus. (Bishr, 1998) estime que l'hétérogénéité sémantique est issue de différentes conceptualisations et modélisations (au sens BD) du monde réel : il distingue l'hétérogénéité cognitive (différentes conceptualisations) et l'hétérogénéité de nommage (différents noms donnés au même concept). Afin de gérer cette hétérogénéité sémantique dans l'appariement d'un espace géographique et d'un processus adapté, nous avons besoin de formaliser la description des processus de généralisation (Figure C-33), cette description formelle étant réalisée dans CollaGen par le fournisseur qui connaît bien les capacités de son processus. Dans cette partie, nous allons tout d'abord faire un tour d'horizon rapide de la description de processus traitant des données géographiques puis nous discutons de la définition des capacités d'un processus de généralisation. Enfin, nous présentons le modèle formel que nous proposons pour décrire un processus de généralisation.

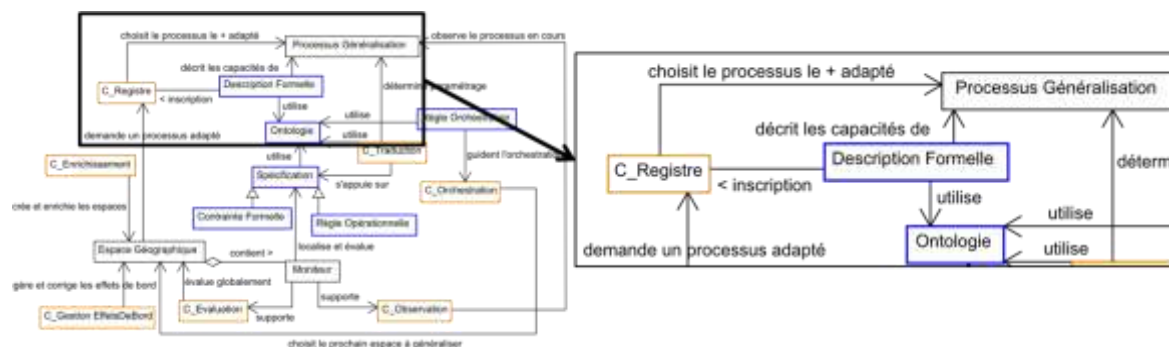


Figure C-33. Les descriptions formelles sont une formalisation des capacités d'un processus de généralisation qui s'appuient sur l'ontologie et sont utilisées par le registre pour permettre le choix du processus le plus adapté à la généralisation d'un espace donné.

C.7.1. La Description des traitements géographiques

Une des étapes essentielles pour permettre l'interopérabilité de traitements géographiques est la description formelle de ces traitements pour qu'un utilisateur ou une machine en comprenne précisément les capacités, c'est-à-dire les objectifs, les caractéristiques, les entrées et les sorties. La modélisation et la description de traitements ou de processus est depuis longtemps un sujet de réflexion dans le domaine de l'entreprise ou de la représentation des connaissances (Chandrasekaran *et al*, 1998).

Dans le domaine des traitements de géographiques dont font partie les processus de généralisation, (Bucher *et al*, 2005; Bucher, 2006) proposent un modèle inspiré de UML2 pour décrire les

traitements sous forme décomposée de *Fonctions* et d'*Activités*. Ce modèle est repris dans le cadre de services de restructuration de données (Bucher et Balley, 2007) et dans le cadre d'acquisition de métadonnées sur les traitements auprès d'experts (Bucher, 2008). Sa simplicité et sa généralité sont intéressants car nous pouvons envisager d'exprimer aisément les processus de généralisation dans ce modèle. Un autre modèle de métadonnées pour le catalogage de traitements géographiques est proposé dans (Abd-el-Kader et Bucher, 2006). Il contient notamment une notion intéressante : l'*évaluation* d'un traitement provient de (Ruas, 2001b). L'évaluation est une appréciation de la réussite du traitement qui peut être renseignée par un utilisateur ou par une évaluation automatique par exemple. Si cette notion n'est pas directement reprise dans le modèle que nous proposons dans cette partie, elle est reprise dans le cadre du composant registre décrit en C.13.

Afin de mieux comprendre, quelles informations doivent être décrites dans les capacités d'un traitement géographique et comment elles peuvent être modélisées, il est intéressant de faire une analogie entre la composition de services Web et notre problème de collaboration de processus de généralisation (comme en B.1.1). En effet, les services Web sont publiés de manière indépendante sans savoir quelles chaînes de services pourraient les utiliser. Les services Web doivent donc être décrits de manière formelle pour qu'un système automatique de composition soit capable d'apparier des besoins de services avec les capacités des services publiés. Pour permettre cet appariement automatique, (Schade et al, 2004) comparent deux méthodes de description des services Web. La première méthode proposée repose sur la description des entrées, sorties des processus à l'aide de concepts ontologiques puis la description des pré-conditions et des effets sous forme de règles logiques. La deuxième repose sur un langage de description algébrique des capacités d'un service, Haskell (Peyton Jones, 1998) qui grâce à ses axiomes, joue le même rôle qu'une ontologie. Ce dernier cas, nous paraît trop complexe pour être utilisé dans notre problème.

(Lutz et al, 2003) énoncent que si l'appariement est fait par un courtier indépendant, la description doit contenir les besoins et les capacités du traitement. Il définit également cinq niveaux de formalisme pour permettre l'appariement. Plus la sémantique des capacités est formelle, plus l'appariement est facile. Le dernier niveau correspond à des métadonnées faisant toutes références à des ontologies. (Lutz, 2007) propose dans le cadre de la composition de services Web de décrire les besoins à l'aide de pré-conditions et les sorties du processus à l'aide de post-conditions. Les pré-conditions sont les conditions à remplir pour pouvoir utiliser le service. Les post-conditions décrivent comment les données en entrée ont été transformées par le service et non directement les transformations que réalise le service. Cela revient à une version simplifiée de la description de services proposée dans (Schade et al, 2004). Toujours dans le contexte de la description de services Web géographiques pour la composition de services complexes, (Lutz et al, 2007) proposent de décrire à la fois le but de la composition et les capacités des services dans une ontologie générale de la description des services, spécialisée ensuite en ontologie du domaine pour un cas particulier. Les pré-conditions et post-conditions sur les services sont formalisées par des règles sur les éléments de l'ontologie du type « si l'entrée est de tel type d'entité de l'ontologie, alors le traitement renvoie tel type d'entité en sortie. » Ce qui est intéressant dans cette dernière approche est que la formalisation du but global aide à formaliser les capacités du processus. Dans notre cas de la généralisation collaborative, le but global est la généralisation complète de la carte, c'est-à-dire la satisfaction optimale de l'ensemble des contraintes et règles formalisant les spécifications de la carte.

Un des besoins particuliers des traitements géographiques par rapport à d'autres services est celui des structures spatiales et sémantiques (comme un réseau structuré, un graphe, ou un alignement de bâtiments) qui doivent être explicites dans les données en entrée pour que le traitement opère. Le modèle à base d'activités permet ainsi de décrire les structures spatiales dont un traitement a besoin (Bucher et Balley, 2007). De plus, (Balley, 2007) propose de créer automatiquement ces structures une fois décrites grâce à des services de restructuration des données pour avoir des données en entrée qui collent bien aux besoins du traitement. (Neun et al, 2008) reprend cette idée en proposant ce qu'il appelle des services de support pour créer les structures spécifiques à la généralisation.

Pour résumer, voici les points des formalismes existants que nous considérons comme particulièrement pertinents. Comme (Bucher, 2006), nous estimons qu'un modèle orienté objet est bien adapté car simple et en cohérence avec les formalismes précédents de contraintes et règles opérationnelles. La recommandation de (Schade et al, 2004) d'utiliser une ontologie pour servir de support à un vocabulaire partagé est à noter (c'est un choix que nous avons fait par ailleurs). La description de nos processus doit formaliser les besoins et capacités (Lutz et al, 2003), pré-conditions et post-conditions (Lutz, 2007) étant un moyen pratique pour y parvenir. Nous notons également qu'il est intéressant de se baser sur l'objectif global (la satisfaction globale des contraintes) pour formaliser les capacités (Lutz et al, 2007). Les traitements géographiques et notamment de généralisation nécessitent aussi de bien formaliser les structures spatiales utilisées par un processus (Bucher et Balley, 2007). Enfin, il est intéressant d'intégrer une composante d'évaluation du processus dans la description (Abd-el-Kader et Bucher, 2006) mais nous nous limitons ici à l'information pouvant être remplie par le fournisseur donc il ne peut s'agir que d'évaluation *a priori*.

C.7.2. Les Capacités d'un processus de généralisation cartographique

Nous avons vu dans la partie précédente quels modèles pourraient être utiles pour décrire formellement les capacités d'un processus de généralisation, afin de permettre l'utilisation de ces descriptions dans le composant registre. Nous décrivons dans cette partie quelles sont les capacités à décrire en s'appuyant sur les particularités des processus de généralisation. Lorsque l'on souhaite décrire un processus de généralisation de manière informelle comme nous l'avons fait dans notre état de l'art, les capacités peuvent se traduire par les points suivants :

- La structure et le schéma de données acceptés en entrée du processus (une classe de bâtiments surfacique, une classe de routes linéaires et une classe d'îlots par exemple). Si la question du nom des classes peut se résoudre en ayant recours à une ontologie, ce point inclut la nécessité de décrire dans les capacités **les enrichissements** nécessaires au fonctionnement du processus (les îlots dans l'exemple précédent).
- Le type de données en entrée sous la forme de contexte géographique (par exemple une ville) ou de thème de données (par exemple les routes). Cela revient à dire, en utilisant le formalisme de la généralisation collaborative, qu'il faut décrire **les espaces géographiques** sur lequel le processus peut être appliqué.

- Le type de spécifications qu’ils peuvent prendre en compte (par exemple, un processus peut généraliser des données en prenant en compte la distance minimum de lisibilité entre deux bâtiments). Cela se traduit dans le formalisme de la généralisation collaborative, en **contraintes** (et règles opérationnelles) que le processus peut satisfaire, plus ou moins bien.
- Les **échelles initiales et finales** pour lesquelles le processus est valide, car différents sauts d’échelles nécessitent des traitements différents, qui ne sont donc tous bien géré par chaque processus.
- Le type de **transformations** que le processus est capable d’effectuer sur les données. Ces transformations pourraient être exprimées à partir de la taxonomie des opérateurs de l’ontologie, ou à l’aide du formalisme OCL comme dans (Foerster et al, 2008). Néanmoins, dans CollaGen, nous ne sommes pas intéressés par la manière utilisée par les processus pour satisfaire les spécifications mais uniquement par leur satisfaction ou leur non satisfaction. Nous choisissons donc de ne pas inclure ce point dans notre description formelle, même si l’information serait théoriquement intéressante pour comprendre les capacités d’un processus.

La description formelle des capacités d’un processus de généralisation doit, dans le cadre de CollaGen, contenir les enrichissements nécessaires, les espaces géographiques pouvant être utilisés en entrée, les contraintes que le processus peut gérer et satisfaire *a priori* et les échelles pour lesquelles le processus est valide.

C.7.3. Un Modèle formel pour décrire un processus de généralisation

Par analogie avec la composition de services Web, nous proposons, pour décrire les processus de généralisation et ses capacités listées dans la partie précédente, un modèle orienté objet (Figure C-34). Ce modèle repose sur la formalisation des besoins sous la forme de pré-conditions et des capacités sous la forme de post-conditions. Dans ce modèle, la plupart des éléments sont décrits par des concepts de l’ontologie de la généralisation, ce qui garantit une homogénéité sémantique dans les descriptions. Nous allons illustrer ce modèle par la description de trois processus de généralisation :

- (P1). Le processus CartACom décrit en E.1.3 issu de (Duchêne, 2004)
- (P2). Les Beams mis en œuvres pour les routes (Bader et Barrault, 2001)
- (P3). Un processus de généralisation de la végétation issu de (Touya et al, 2010)

Une description de processus correspond tout d’abord à un processus de généralisation, désigné parmi ceux de l’ontologie de généralisation (qui contient une taxonomie des processus de généralisation). Pour (P3) par exemple, le concept de processus correspondant doit être un sous concept de « processus par séquences prédéfinies » car ce processus dédié à la végétation est une séquence prédéfinie d’opérations morphologiques de simplification.

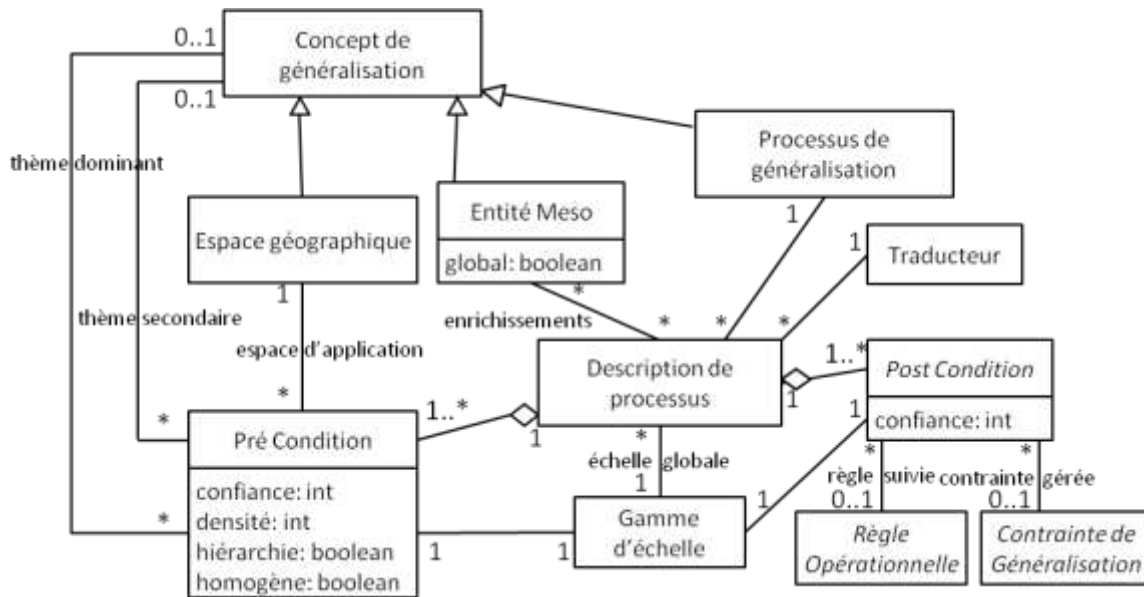


Figure C-34. Diagramme de classes UML représentant notre modèle de description des capacités d'un processus de généralisation.

La description comprend également les structures spatiales nécessaires au fonctionnement du processus de généralisation. Celles-ci sont formalisées par un ensemble de concepts meso et procéduraux de l'ontologie de généralisation (seuls les concepts meso apparaissent sur la figure pour des questions de lisibilité). Par exemple, (P1) va requérir l'explicitation du concept de « Petit Compact » (entité cartographique de petite taille et de contour rigide, e.g. un bâtiment) et (P2) les différentes formes de carrefours complexes (ronds-points, pattes d'oie et échangeurs) sur lesquels le processus ne devrait pas être appliqué. L'attribut booléen *global* de l'enrichissement signifie qu'il faut pratiquer l'enrichissement sur l'ensemble des données s'il vaut *vrai*, et seulement dans l'espace d'application s'il vaut *faux*. Dans les exemples de (P1) et (P2), l'attribut sera à *faux* car il n'est pas utile de réaliser ces enrichissements sur l'ensemble des données.

Suivant l'échelle finale de la carte, les notions géographiques que l'on va vouloir représenter, ainsi que les choix de conceptualisation et de modélisation de ces notions, peuvent varier fortement (Mackaness, 2007). Les transformations que l'on doit effectuer pour généraliser des données géographiques dépendent donc fortement de l'échelle finale. Mais elles dépendent aussi des choix de conceptualisation et de modélisation faits dans les données initiales, et donc de l'échelle de référence de la base de données initiales. (Brewer et Buttenfield, 2009) illustrent ce problème en proposant des choix de modélisation et de transformation suivant les sauts d'échelle et les objectifs de la carte. Les processus utilisés dans le cadre de CollaGen sont donc aussi dépendants du saut d'échelle, et le choix du processus adapté doit aussi prendre en compte l'adaptation du processus au saut d'échelle choisi. Nous définissons un saut d'échelle comme un couple échelle initiale/échelle finale et non simplement comme une différence ou un rapport entre ces deux échelles. Ainsi nous proposons de modéliser les capacités d'un processus à réaliser différents sauts d'échelle sous la forme d'une *Gamme d'échelles*. Cette gamme d'échelle est composée d'un saut d'échelle optimal, puis d'un minimum et un maximum pour l'échelle de référence et l'échelle finale. La description des capacités est complètement valable quand la généralisation est proche du saut d'échelle optimal et elle est relativement juste tant qu'on reste entre le minimum et le maximum. Par exemple, le saut d'échelle optimal de (P1) si on se fie aux exemples proposés est du 1 : 15 000 au 1 : 35 000. Nous

n'avons pas beaucoup d'indications sur la gamme d'échelles initiales possibles mais nous pouvons considérer que le processus est valable entre 1 : 10 000 et 1 : 15 000. D'après les tests réalisés, (P1) rencontre quelques problèmes isolés pour une échelle finale de 1 : 50 000 donc nous pouvons estimer qu'il est raisonnable de ne pas le pousser au-delà du 1 : 60 000. La détermination de telles gammes d'échelles n'est pas une tâche évidente pour le fournisseur donc l'utilisation de ces gammes est assez souple : il n'est pas nécessaire d'être trop précis.

Nous arrivons maintenant au cœur du modèle de description avec les besoins (ou entrées) du processus qui sont décrits par les pré-conditions et les capacités (ou sorties) qui sont décrites par les post-conditions. Les pré-conditions représentent les conditions que doivent remplir les données en entrée du processus pour qu'il fonctionne correctement. Dans le cadre de CollaGen, nous proposons de modéliser ces pré-conditions sous la forme d'espaces géographiques (issus de l'ontologie) pour lesquels le processus est adapté. Une pré-condition est donc principalement un *Espace Géographique* couplé à *Indice de confiance* qui traduit simplement si le processus est peu (indice de 1) ou parfaitement (indice de 5) adapté à cet espace. Cet indice de confiance permet de moduler l'adaptation d'un processus à un espace et de ne pas restreindre les capacités d'un processus aux espaces pour lesquels il est parfaitement adapté. Cela renforce les chances de trouver un processus adapté à chaque espace même s'il y a peu de processus disponibles. Si nous prenons nos trois exemples, nous pouvons considérer que (P1) est parfaitement adapté aux espaces ruraux (5), moyennement adapté aux espaces périurbains (3) et aux espaces de montagne (3). Pour (P2), il est parfaitement adapté à l'espace facilitateur « graphe de flexibilité » (Lemarié, 2003) mais il est également adapté aux espaces thématiques réseau routier et réseau hydrographique s'ils ne sont pas trop grands (donc avec un indice de confiance faible, 2 par exemple). Pour (P3), il est uniquement adapté à l'espace thématique « couche de végétation » mais avec un indice de confiance maximal.



Figure C-35. Exemple d'espaces peu dense et thématiquement hétérogène (bâtiments, routes, végétation, cours d'eau, points d'eau, talus, chemins, cimetière etc.)(Duchêne, 2004).

Il est également possible de décrire une pré-condition autrement que par un simple espace géographique de l'ontologie mais en lui associant certaines valeurs des caractéristiques densité, homogénéité thématique, hiérarchie, thème dominant et thème secondaire. La Figure C-35 montre ainsi un espace peu dense, sans hiérarchie et hétérogène thématiquement qui pourrait être un espace adapté à (P1). Ces caractéristiques peuvent être utilisées, soit pour caractériser un espace territorial sans préciser lequel (par exemple, « ce processus est adapté aux espaces denses et hiérarchiques dont le thème principal est le bâti »), soit pour être plus précis que l'ontologie (par exemple, « ce processus est adapté aux espaces ruraux homogènes thématiquement dont le thème

principal est le bâti et le secondaire le réseau routier »). Les quelques caractéristiques introduites dans ce modèle de description sont une première proposition qui ne demande qu'à être enrichie.

L'autre élément clé de notre modèle réside dans la formalisation des capacités de chaque processus à travers ses post-conditions. Les post-conditions sont des conditions sur les données traitées qui traduisent la manière dont ces données seront *a priori* modifiées. Nous proposons de modéliser ces post-conditions sous la forme de contraintes de généralisation et règles opérationnelles *a priori* satisfaites en sortie du processus. Les contraintes et règles, présentées dans les parties précédentes, traduisent en effet les spécifications attendues de la carte généralisée. Les capacités sont donc ainsi décrites en termes de portions de l'objectif global remplies par le processus, comme le suggérait (Lutz et al, 2007). Comme pour les pré-conditions, les post-conditions couplent un élément formel de spécification avec un indice de confiance allant de 1 à 5, un 5 signifiant qu'*a priori*, ce processus satisfait parfaitement la contrainte ou la règle. Les contraintes et les règles utilisées pour décrire les post-conditions des processus sont les versions génériques de ces contraintes : le fournisseur décrit qu'un processus satisfait *a priori* « la contrainte d'aire minimum des bâtiments » quelle que soit la valeur minimum choisie ensuite par l'utilisateur. Si l'on reprend nos trois exemples de processus, un échantillon des post-conditions de (P1) est : « contrainte d'aire minimum des bâtiments » (5), « contrainte de proximité entre bâtiments » (5), « contrainte de proximité entre bâtiment et route » (5) et « contrainte de maintien de la position relative entre une impasse et un bâtiment » (4). Pour (P2) nous pouvons considérer les contraintes suivantes : « contrainte de superposition entre symboles de routes différentes » (5), « maintien de la forme des virages » (5), « maintien de la forme des routes droites » (4), « contrainte de coalescence » (4) (contrainte pour empêcher plusieurs parties du symbole d'une même route de se superposer, ce qui peut arriver dans les virages serrés). Enfin, pour (P3), nous pouvons considérer l'échantillon suivant : « contrainte de taille minimum des zones arborées » (5), « contrainte de granularité des zones de forêt » (4), « contrainte de taille minimum des trous dans les zones arborées » (5).

Enfin, notre modèle formel de description permet de décrire un aspect spécifique à CollaGen, le nom de la fonction qui permet la traduction des contraintes et règles formelles en paramètres spécifiques de ce processus. Cette fonction fait partie du composant de traduction présenté en C.12 et est intégrée à CollaGen par le fournisseur de processus en même temps que le processus et sa description formelle.

C.8. UN MODELE FORMEL DE REGLES D'ORCHESTRATION

Dans le chapitre B, les ressources formelles requises pour mettre en œuvre un système de généralisation collaborative comme le modèle CollaGen comptaient un dernier élément : un ensemble de connaissances formelles pour guider l'orchestration du système, c'est-à-dire pour savoir à un moment donné du processus collaboratif quel est le prochain espace géographique à généraliser (Figure C-36). Nous proposons que ces connaissances soient divisées en deux types : d'un côté des règles d'orchestration qui guident le composant d'orchestration dans ses choix, et de l'autre côté des connaissances qui permettent une orchestration même quand il n'y a pas de règle pour un cas

particulier. Ces dernières connaissances peuvent se rapprocher des travaux en Systèmes Multi-Agents pour déterminer l'ordonnancement séquentiel de déclenchement des agents (Meurisse, 2001). Le cas de ces connaissances est présenté dans la partie sur le composant d'orchestration (C.15). Dans cette partie, nous nous concentrons uniquement sur les règles d'orchestration, en présentant d'abord un découpage de la généralisation en grandes étapes, puis en nous demandant quels types de règles d'orchestration nous souhaitons formaliser. Enfin, nous présentons le modèle formel proposé.

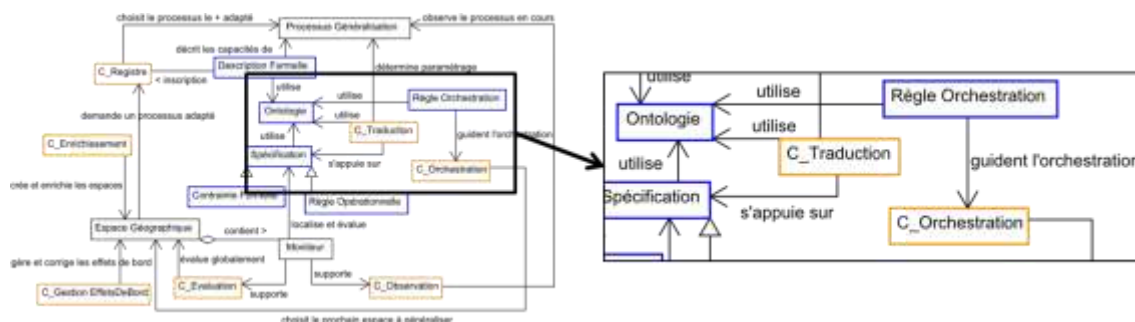


Figure C-36. Les règles d'orchestration s'appuient sur l'ontologie pour guider le composant d'orchestration dans son choix du prochain type d'espace à généraliser à un moment donné.

C.8.1. Les Grandes étapes de la généralisation

Pour orchestrer à haut niveau une approche pas à pas locale, il est tout de même nécessaire de suivre un plan prédéfini qui enchaîne les étapes de généralisation pas à pas comme le *Global Master Plan* de (Ruas et Plazanet, 1996). L'exemple de plan donné pour la généralisation d'une carte topographique au 1 : 50000 dans cet article illustre bien ce qu'il doit être selon les auteurs :

- Tache 1 : Elimination sémantique des objets dans chaque classe
- Tache 2 : Simplification des attributs et reclassification
- Tache 3 : Agrégation des objets de même classe connectés
- Tache 4 : Simplification des réseaux
- Tache 5 : Traitement des zones denses
- Tache 6 : Traitement des objets structurants (voies ferrées, routes importantes)
- Tache 7 : Simplification et amélioration des objets moins importants

Même si cet exemple ne doit pas être pris pour un plan immuable, il fait apparaître qu'il existe des étapes dans la généralisation grâce auxquelles le système est optimal. Par exemple, il est raisonnable comme dans l'exemple de réaliser la sélection des réseaux avant la généralisation des zones denses, car cette sélection va libérer de la place et faciliter la généralisation des zones denses. La Figure C-37 montre un exemple de processus qui change la géométrie des carrefours complexes trop petits en un simple point. Ce type de processus doit logiquement être exécuté avant la sélection du réseau routier et avant la généralisation cartographique de cette zone urbaine. Ces grandes tendances guidées par la logique du processus de généralisation (i.e. on commence par ce qu'il est déjà possible de faire et ce qui facilite la suite), nous poussent à découper le processus de généralisation cartographique en un plan global de haut niveau dans lequel seront répartis les processus de généralisation.

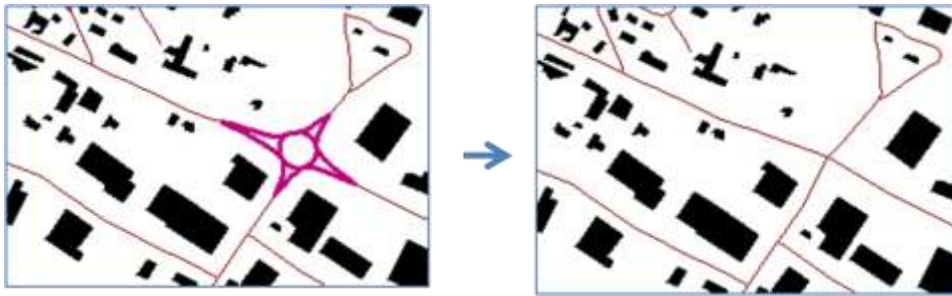


Figure C-37. *Processus de typification pour changer la géométrie des carrefours complexes en un simple point.*

Nous proposons donc de représenter la généralisation cartographique en cinq grandes étapes :

- Etape 1 : Généralisation de schéma (attributs et reclassification/agrégation)
- Etape 2 : Changements d'implantation géométrique
- Etape 3 : Sélection pour chaque thème (même si certains thèmes sont liés donc sélectionnés ensemble)
- Etape 4 : Généralisation prenant en compte les symboles et la lisibilité
- Etape 5 : Généralisation graphique

La première étape regroupe tous les processus de généralisation attributaire (un attribut passe de quatre valeurs à deux ou deux attributs sont fusionnés par exemple) et les processus de reclassification puis agrégation des objets re-classifiés comme le processus de reclassification de l'occupation du sol de (Revell, 2007a).

La deuxième étape concerne les processus qui réalisent des changements systématiques d'implantation géométrique (polygone en polyligne, polygone en point etc.) des concepts géographiques. La Figure C-37 montre un processus qui doit être appliqué à cette étape, un autre exemple étant la transformation de l'hydrographie surfacique représentant les cours d'eau en objets linéaires (Regnauld et Mackaness, 2006).

La troisième étape regroupe tous les processus de sélection pure, qu'ils éliminent les objets en fonction de la sémantique, de la géométrie ou de leur contexte spatial. Les processus de cette étape ne tiennent pas encore compte des contraintes de lisibilité ou de taille de symbole qui pourront entraîner d'autres opérations de sélection dans les étapes suivantes. La partie A.2.5 présente quelques processus qui peuvent être utilisés durant cette étape. A partir de cette étape (inclusive), l'ordre d'application des processus devient fondamental ; avant l'ordre importe moins car les opérations sont plus indépendantes (même si certains processus de sélection dépendent les uns des autres, par exemple, la sélection des routes peut dépendre de la sélection des sites touristiques (si un site est éliminé, la route y menant peut l'être aussi)).

A partir de la quatrième étape, les contraintes de lisibilité et les symboles sont pris en compte, il s'agit de la généralisation cartographique classique dont AGENT, CartACom ou les Beams sont des exemples connus. Il s'agit de l'étape qui, par expérience, regroupe les processus les plus complexes.

La cinquième étape dite de généralisation graphique s'inspire des travaux de (Harrie et Sarjakoski, 2002). Il s'agit d'une sorte de post-traitement de la généralisation cartographique qui permet de corriger par de légères déformations les derniers conflits de lisibilité résiduels. Cette étape ne

contient pas de processus qui vont transformer radicalement les données mais plutôt améliorer la qualité de lisibilité de la carte.

La répartition des processus de généralisation dans ces grandes étapes est relativement difficile à faire comme l'était la distinction entre généralisation de modèle et généralisation graphique donc elle peut rester volontairement perméable. Si un processus est difficile à classer, il peut être marqué comme utilisable dans plusieurs des grandes étapes proposées. La détermination des grandes étapes correspondant à chaque processus est effectuée par le modélisateur car elle est intégrée à l'ontologie pour enrichir la taxonomie des processus de généralisation. Par exemple, le concept correspondant au modèle de généralisation par moindres carrés peut avoir dans l'ontologie comme étapes valides les étapes 4 et 5 car il peut être mis en œuvre dans les deux étapes.

C.8.2. Quelles règles souhaite-t-on formaliser pour guider l'orchestration ?

Avant de savoir comment formaliser des règles d'orchestration, il faut connaître le type de règles que l'on imagine pour guider un processus d'orchestration. La définition des cinq grandes étapes doit nous permettre de mieux guider l'orchestration des processus, donc, les règles devront d'abord nous permettre de nous situer dans ces étapes et nous guider dans le passage d'une étape à une autre : il doit être possible de modéliser des règles qui nous disent à quel moment passer de la généralisation cartographique à la généralisation graphique.

Ces règles d'orchestration doivent aussi nous permettre de formaliser les connaissances attendues dans le Global Master Plan de (Ruas et Plazanet, 1996). Par exemple, l'expérience nous indique qu'il faut commencer par généraliser les routes les plus importantes pour pouvoir les fixer ensuite quand le système généralise les routes mineures en les déformant. Nous savons aussi qu'il vaut mieux d'abord généraliser les routes puis les bâtiments en zone rurale car si les bâtiments sont d'abord éloignés des routes, le système risque en déplaçant les routes de récréer des conflits entre les bâtiments et la route (Figure C-38). L'expérience de la généralisation peut également nous donner des connaissances sur ce qu'il ne faut pas faire, sans indication précise de ce qu'il faut faire. Par exemple, il ne faut pas généraliser les zones de montagne avant les zones rurales : ces zones se chevauchent et si l'on souhaite maintenir les relations entre les bâtiments et le relief, par exemple, il vaut mieux traiter le relief une fois que les bâtiments ont été généralisés.

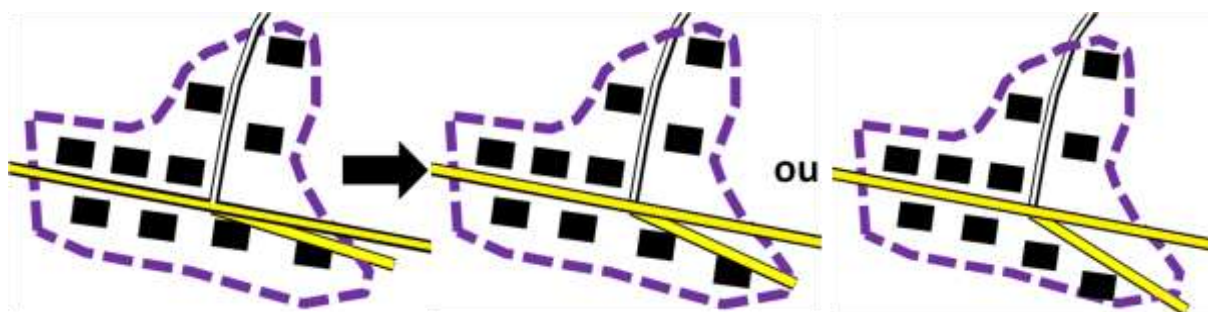


Figure C-38. Intérêt de généraliser les routes avant les zones rurales (2^{ème} résultat) plutôt que l'inverse (1^{er} résultat).

Enfin, nous souhaitons intégrer formellement ces règles dans la généralisation collaborative de CollaGen. Il s'agit donc de donner dans ces règles des indications sur le prochain type d'espace à généraliser, et non sur le type d'opération à effectuer sur les objets. Exceptionnellement, ces règles peuvent désigner un processus à appliquer. En effet, la description des processus comprend les types d'espaces pour lesquels ils sont adaptés (voir partie précédente) et le système peut donc en déduire le prochain espace à généraliser.

Dans les règles d'orchestration, nous souhaitons formaliser l'enchaînement des grandes étapes et les connaissances générales globales sous la forme du prochain type d'espace à généraliser à un moment donné de la collaboration.

C.8.3. Un Modèle formel de règle d'orchestration

En cohérence avec les autres modèles formels présentés dans cette partie et notamment le modèle de règles opérationnelles, nous proposons un modèle formel orienté objet qui repose sur la formalisation de prémisses qui entraînent une conclusion (Figure C-39). Afin d'illustrer et d'expliquer ce modèle formel, nous allons utiliser les exemples de règles d'orchestration suivants :

- (R1). La généralisation cartographique doit commencer par l'espace urbain.
- (R2). Le réseau routier est généralisé après l'espace périurbain.
- (R3). Si toutes les parcelles d'espace ont été couvertes par une généralisation cartographique, il faut passer à la généralisation graphique.
- (R4). Durant l'étape de sélection, il faut utiliser un processus de sélection des routes après la généralisation du réseau hydrographique.

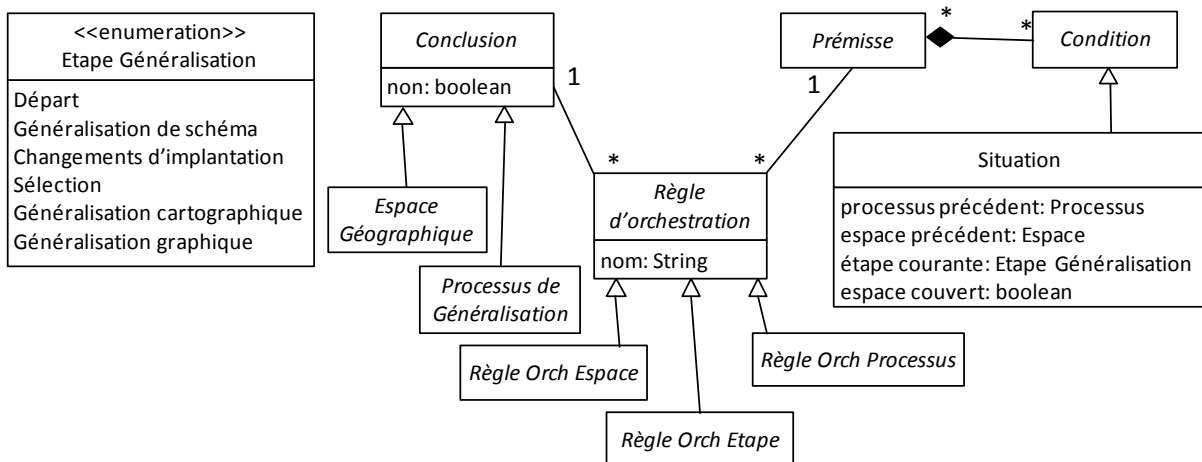


Figure C-39. Diagramme de classe UML du modèle formel de règle d'orchestration.

Comme pour les règles opérationnelles, une règle est constituée d'une prémisse et d'une conclusion, une prémisse étant composée de conditions liées par le connecteur logique *Et*. Nous définissons trois types de règles procédurales : les règles de type « espace » qui proposent la généralisation d'un espace à un moment donné, les règles de type « processus » qui proposent l'utilisation d'un processus à un moment donné, et les règles de type « étape » qui proposent de passer à l'étape globale suivante. Par exemple, (R1) et (R2) sont des règles de type « espace » car elles proposent la généralisation de l'espace urbain et périurbain respectivement ; (R3) est une règle de type « étape »

car elle propose de passer à l'étape de généralisation graphique ; enfin, (R4) est une règle de type « processus » car elle propose d'utiliser un processus de sélection du réseau routier (n'importe quel processus dont la description fait référence au concept de l'ontologie « processus de sélection des routes »).

Pour les règles d'orchestration, une conclusion peut être soit un type d'espace géographique issu de l'ontologie, soit un processus de généralisation. Les conclusions de type espace géographique sont réservées aux règles de type « espace » géographique et le principe est le même pour les processus. Les règles de type étape n'ont simplement pas de conclusion car celle-ci est implicite : la conclusion est de passer à l'étape suivante dans l'ordre de l'énumération. Donc pour (R1), la conclusion est le concept « espace urbain » de l'ontologie, pour (R2) le concept « espace périurbain », pour (R3), il n'y a pas de conclusion et pour (R4), la conclusion est le concept de l'ontologie « processus de sélection des routes ».

Dans ce cas, la condition de la prémisse est simplement le descriptif de la *situation* actuelle dans le processus collaboratif. Cette situation décrit l'étape globale courante, puis l'espace généralisé lors du tour de boucle précédent ainsi que le processus utilisé (exprimés en termes de concepts de l'ontologie). La situation décrit également avec l'attribut booléen *espace_couvert* si toutes les parcelles de l'espace ont bien été traitées dans l'étape courante. Par exemple, si dans la généralisation cartographique, les espaces urbains qui ne couvrent pas tout l'espace à généraliser, sont les seuls à avoir été généralisés, alors l'attribut *espace_couvert* vaut *faux*. Pour (R1), la situation a comme étape courante *Généralisation cartographique* et les espaces et processus précédents sont inconnus car ils n'influent pas sur la règle. Il faut noter que le champ *étape courante* peut ne pas être rempli ce qui signifie que la règle est valable à toutes les étapes. Ainsi, pour (R2), il n'y a pas d'étape courante mais un *espace précédent* qui est l'espace thématique réseau routier. Pour (R3), la situation a son attribut *espace couvert* à *vrai* et comme étape courante *Généralisation cartographique*. Enfin, (R4) a une situation avec un *espace précédent*, le réseau hydrographique, durant l'étape de *sélection*.

Par ailleurs, nous avons inséré une valeur *Départ* dans les Etapes de Généralisation pour permettre d'écrire des règles qui guident l'orchestration au démarrage : une règle dont la prémisse a pour *étape courante* la valeur *Départ* donne dans sa conclusion la première action de l'orchestration à réaliser.

Une règle formelle d'orchestration a pour prémisse une situation dans le processus de collaboration (étape courante, espaces précédemment généralisés...) et pour conclusion, suivant son type, un nouveau type d'espace à généraliser, un processus à utiliser ou bien la proposition de passer à l'étape globale suivante.

C.8.4. Bilan de la formalisation des connaissances en généralisation

Nous avons présenté dans les parties précédentes la manière dont nous proposons de formaliser les différentes connaissances en généralisation cartographique pour pouvoir les utiliser automatiquement et de manière optimale dans les composants de généralisation collaborative mis en œuvre dans CollaGen.

- L'ontologie de la généralisation cartographique contient les concepts nécessaires pour gérer l'interopérabilité sémantique des différents composants et ressources. Les composants et les autres ressources formelles ont ainsi recours aux concepts de l'ontologie pour partager un vocabulaire commun.
- Les modèles formels de contrainte de généralisation et de règle opérationnelle permettent de modéliser les spécifications de la carte. Ils sont utilisés pour la description des processus, mais ils seront surtout utilisés par les composants de traduction (pour traduire les spécifications en paramètres des processus), registre (pour mettre en rapport les conflits actuels d'un espace géographique avec les spécifications *a priori* satisfaites par un processus), observation et évaluation (pour évaluer en temps réel la satisfaction des spécifications) et gestion des effets de bord (pour vérifier si les spécifications sont bien respectées juste au bord des espaces).
- Le modèle formel de description des processus de généralisation permet de modéliser les besoins et les capacités des processus. Ces descriptions seront utilisées par le composant registre qui va les analyser pour déterminer le processus le plus adapté à la généralisation d'un espace donné. Elles seront éventuellement utilisées par le composant d'orchestration si les règles désignent seulement le processus suivant à utiliser. Enfin, ces descriptions seront exploitées par le composant d'enrichissement pour déterminer les enrichissements en concepts meso et procéduraux qu'il doit effectuer dans un espace donné.
- Le modèle formel de règles d'orchestration modélise les connaissances sur l'orchestration globale d'un processus de généralisation cartographique. Les règles seront utilisées uniquement par le composant d'orchestration afin de le guider dans ses choix du prochain type d'espace géographique à généraliser.

C.9. ESPACES GEOGRAPHIQUES ET PROCESSUS DANS COLLAGEN

Après les connaissances formelles présentées dans les parties précédentes, les autres ressources de la Généralisation Collaborative manipulées dans CollaGen sont les espaces géographiques et les processus de généralisation disponibles. Nous présentons d'abord la mise en œuvre générale des espaces géographiques, puis nous détaillons plus le problème du choix des limites des espaces par une étude de sensibilité dans le cas des zones urbaines. Nous présentons enfin notre mise en œuvre des processus de généralisation disponibles dans CollaGen.

C.9.1. Modélisation des espaces géographiques dans CollaGen

Concernant les espaces géographiques, la mise en œuvre dans CollaGen suit exactement la modélisation présentée en B.4, chacun des espaces étant muni d'un voisinage dans lequel les effets de bord peuvent être localisés et traités. Les espaces géographiques sont rendus utilisables dans CollaGen par le développeur qui doit intégrer les espaces clés selon l'ontologie et un certain nombre

d'espaces pouvant être facilitateurs ou spécifiés. Par exemple, les espaces répétitifs (Figure C-40) ne peuvent être utilisés comme espace facilitateur d'un processus comme (Burghardt et Neun, 2006) que si le développeur les a intégrés dans CollaGen avec un algorithme pour sa délimitation et un algorithme pour la construction de son voisinage dans le cas des espaces thématiques.

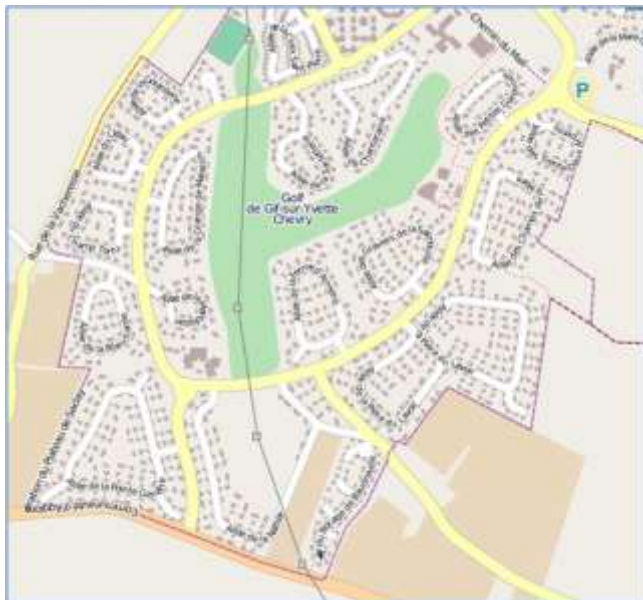


Figure C-40. Exemple d'espace résidentiel répétitif à Gif-sur-Yvette.

Nous avons vu dans la partie B.4.3 que la généralisation collaborative nécessite de choisir le contour des espaces territoriaux le plus adapté (à la généralisation en général voire aux processus de généralisation disponibles). Dans CollaGen, nous proposons de mettre en place une étude de sensibilité de la généralisation au choix du contour des espaces territoriaux. La sensibilité peut être définie de la manière suivante : il s'agit de l'étude de la variabilité du résultat d'une opération en fonction des variations des données en entrée. L'analyse de la sensibilité permet de savoir si une petite perturbation en entrée peut avoir des impacts sur les données finales (Ruas, 1999, p.90). Dans le cas du contour des espaces géographiques, cela revient à étudier l'impact de la variation du contour sur le résultat de la généralisation collaborative, *a priori* seulement autour de ce contour (il nous paraît improbable qu'une variation légère dans le contour d'une ville influence la généralisation d'une zone située à plusieurs kilomètres car cette variation ne modifierait pas le contexte géographique si loin).

Les études de sensibilité sont très utilisées dans des domaines différents, notamment dans l'évaluation et la validation de modèles physiques pour lesquels on peut faire varier les entrées du modèle par des méthodes probabilistes (Saltelli et al, 2000). Même si ces méthodes d'étude de la sensibilité ne peuvent pas être transposées directement quand les entrées sont géométriques comme les objets géographiques, elles peuvent être adaptées comme le prouvent (Lodwick et al, 1990 ; Bonin, 1998) qui introduisent des bruits statistiques dans les géométries pour appliquer des méthodes de sensibilité.

La méthodologie que nous proposons dans notre cas revient à générer un grand nombre de limites pour un type d'espace et de généraliser dans chaque cas, l'espace et son extérieur direct avec les mêmes processus pour estimer les différences provoquées par le choix de la limite de l'espace. La génération des limites est faite en utilisant des méthodes différentes et pour chaque méthode en

faisant varier de manière raisonnée et aléatoires les paramètres. L'utilisation de limites déterminées à la main par différentes personnes permet également d'affiner cette étude de la sensibilité. Nous proposons également un ensemble de mesures géométriques et géographiques pour qualifier un contour d'espace géographique afin d'orienter notre analyse de l'étude de sensibilité. Nous avons mis en œuvre cette méthode sur la détermination de la meilleure méthode pour délimiter les zones urbaines et les résultats de cette analyse sont présentés en E.4.2.

C.9.2. Les Processus disponibles dans CollaGen

Comme cela est précisé dans les hypothèses de cette thèse, nous nous limitons au cas des processus de généralisation disponible sur notre plate-forme logicielle de travail et donc implémentés et exécutables sur cette plate-forme. Le cas des processus de généralisation disponibles via un appel à un service Web n'est pas abordé ici. Pour qu'un processus de généralisation cartographique automatique soit rendu disponible dans CollaGen, il faut que le fournisseur de processus spécialise la classe abstraite Processus de Généralisation dont on peut voir le diagramme en Figure C-41.

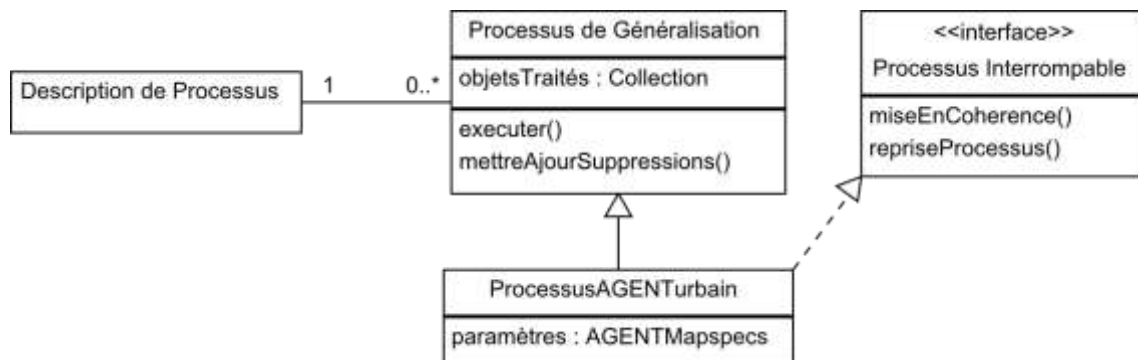


Figure C-41. Diagramme de classe UML du modèle abstrait spécialisé par les processus de généralisation disponibles.

Un processus de généralisation est relié à sa description formelle (voir C.7). Il conserve en mémoire une collection des objets traités (i.e. modifiés) par le processus, qui est alimentée au mieux durant l'exécution (si le processus est « observable » ou « interrompable », c'est-à-dire qu'il peut faire des pauses dans sa généralisation puis la reprendre et dans notre modèle, cela revient à implémenter l'interface *Processus Interrompable*), à la fin sinon. Connaître les objets traités par un processus évite de perdre du temps à évaluer les objets de l'espace géographique qui n'ont pas été traités (et donc dont l'état n'a pas pu s'améliorer ou se dégrader). La méthode `executer()` encapsule l'exécution du processus pour qu'elle puisse être déclenchée de manière générique par CollaGen.

Un processus de généralisation possède également une méthode pour faciliter la gestion des objets éliminés. En effet, si l'on souhaite, comme dans CollaGen, pouvoir revenir en arrière si la généralisation s'est mal passée afin d'essayer un autre processus, il faut accorder une attention particulière aux objets éliminés, de même que pour l'évaluation des contraintes sur ces objets (si le bâtiment est éliminé, la contrainte d'aire minimale d'un bâtiment ne doit pas être évaluée). La méthode `mettreAJourSuppressions()` permet de rendre compte, dans un formalisme pivot (voir C.15), des éliminations d'objets qui peuvent être réalisées de manière différente suivant les processus. Par exemple, le processus AGENT du logiciel Clarity™ (issu du projet AGENT) marque les objets éliminés

avec un attribut spécifique alors que le prototype de CartACom issu de (Duchêne, 2004) les place dans une classe dite « poubelle ».

Enfin, la Figure C-41 montre un exemple de spécialisation pour un processus basé sur AGENT dédié à la généralisation des zones urbaines. Cette spécialisation a pour attributs supplémentaires tous les paramètres spécifiques au processus (qui se résument ici à un objet complexe AGENTMapspec regroupant toutes les assignations de contraintes sur les agents et les paramètres de ces contraintes (Hardy et al, 2003)). Ces paramètres sont ceux utilisés par la méthode *executer()* et sont remplis à partir des contraintes par le composant de traduction.

C.10. LES MONITEURS DE CONTRAINTE

Avant de détailler la mise en œuvre de chacun des six composants de la généralisation collaborative dans CollaGen dans la partie suivante, nous introduisons ici un concept clé, le moniteur de contrainte, qui permet de comprendre le fonctionnement de la plupart de ces composants (Figure C-42).

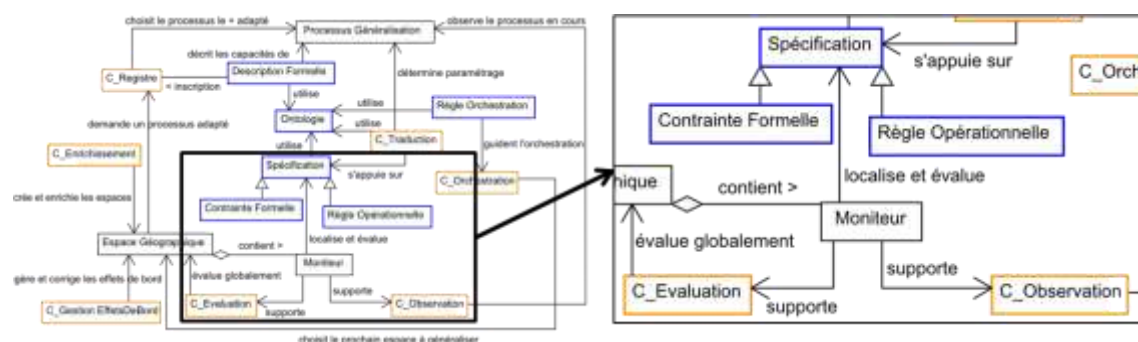


Figure C-42. Les moniteurs de contrainte localisent et évaluent les spécifications formelles comme support du composant d'observation et d'évaluation. Ils permettent de s'assurer que les différents espaces sont généralisés de manière homogène et harmonieuse.

Les premiers modèles théoriques définissant l'approche pas à pas locale (Brassel et Weibel, 1988 ; McMaster et Shea, 1998 ; Ruas et Plazanet, 1996) reposaient sur la localisation des conflits et leur résolution pas à pas. Ainsi, plusieurs processus de généralisation se sont basés sur la recherche des conflits, notamment dans la généralisation des routes ou des linéaires naturels (cours d'eau, trait de côte), avant d'adapter les opérations de généralisation aux conflits identifiés (Plazanet, 1996; Skopeliti et Tsoulos, 1999 ; 2001 ; Mustière, 1998 ; Nakos *et al*, 2008). Les conflits sont la traduction de spécifications de la carte non respectées localement. (Ruas, 1999) propose d'utiliser des objets contraintes pour localiser les conflits, ces objets contraintes étant associés à des objets géographiques. Dans (Ruas, 1999), les contraintes servent à la fois à localiser et observer les conflits mais aussi à agir pour les résoudre (elles proposent des algorithmes à appliquer). (Duchêne et Gaffuri, 2008) introduisent une différenciation entre les fonctions d'observation des contraintes et les fonctions de décision. En suivant cette idée, nous proposons d'utiliser des contraintes, qui traduisent les spécifications, pour localiser les conflits, mais seulement avec leurs fonctions

d'observation des conflits. Ainsi, nous appelons *moniteur de contrainte* l'objet intermédiaire que nous introduisons dans cette partie, qui permet de localiser et évaluer les conflits relatifs à une contrainte donnée.

Nous allons d'abord définir ce moniteur de contrainte et présenter la modélisation que nous en proposons. Puis nous allons expliquer de quelle manière CollaGen va manipuler ces moniteurs de contrainte.

C.10.1. Définition et modélisation

Nous définissons un moniteur de contrainte comme un élément, situé dans la base de données géographique, qui identifie, localise et évalue en temps réel un conflit relatif à une contrainte de généralisation et un objet géographique concerné par cette contrainte.

Nous avons volontairement dissocié sémantiquement la contrainte (i.e. une instance du modèle formel de contrainte présenté précédemment) de son moniteur car conceptuellement, le moniteur ne contraint rien mais se contente de localiser les endroits de la carte où la contrainte est active, et d'évaluer si la contrainte est bien respectée à cet endroit. Si l'on considère deux contraintes « aire des bâtiments > 0.2 mm² carte » et « granularité des bâtiments > 0.1 mm carte », chacun des bâtiments va être lié à deux moniteurs de contrainte responsables de la localisation d'un éventuel conflit avec ces contraintes et de l'évaluation de la satisfaction de ces contraintes sur le bâtiment en question.

La Figure C-43 montre le modèle que nous proposons pour représenter les moniteurs de contrainte ; celui-ci étend une partie des modèles de contraintes instanciées proposés dans (Ruas, 1999 ; Barrault et al, 2001). Un moniteur possède ainsi une *valeur initiale*, une *valeur courante* et une *valeur but* à partir desquelles il calcule sa *satisfaction*. La satisfaction est à choisir dans une énumération de huit valeurs de *non satisfait* à *parfait*. Dans ce choix de valeurs de satisfaction non numériques et peu nombreuses, nous rejoignons le point de vue de (Ruas, 1999) plutôt que par exemple (Barrault et al, 2001 ; Gaffuri, 2008) qui proposaient des valeurs réelles ou de 0 à 100. En effet, (Ruas, 1999, p.89) montre que la mesure des contraintes est très sensible au choix de l'algorithme de mesure et à la méthode d'interprétation de cette mesure donc il ne nous semble pas pertinent de donner une valeur numérique trop variable à la satisfaction car nous aurions du mal à l'interpréter : des satisfactions de 6.4 et 6.9 pour une contrainte de taille par exemple sont elles vraiment différentes ? Ce choix de valeurs de satisfaction est un compromis entre la perception des différences pour permettre l'interprétation et la perception des évolutions (il faut qu'il y ait assez de valeurs pour formaliser une petite progression ou régression de la satisfaction de la contrainte).

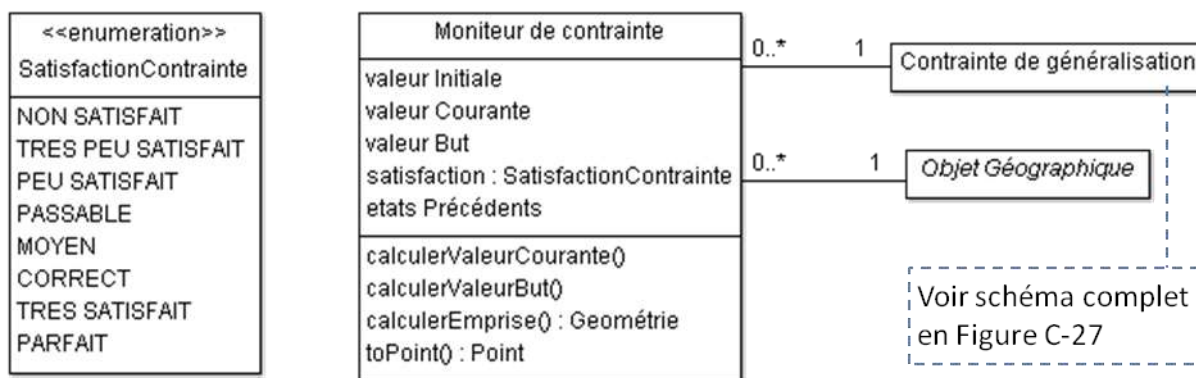


Figure C-43. Diagramme de classe UML d'un moniteur de contrainte.

On peut noter également dans la Figure C-43 que le moniteur de contrainte stocke ses différents états (à chaque fois qu'il s'évalue dans la généralisation collaborative) pour pouvoir estimer l'évolution positive ou négative de sa satisfaction tout au long du processus. Ce point est particulièrement utile pour le composant d'observation et d'évaluation.

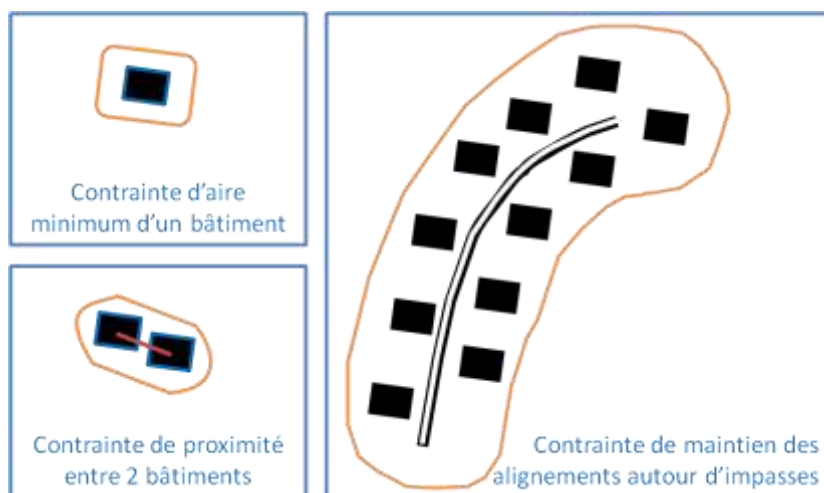


Figure C-44. Trois exemples (micro, relationnel et meso) de calculs de l'emprise (contour orange) d'un moniteur de contraintes.

Enfin, le moniteur de contrainte peut être localisé de deux manières, en calculant son centre ou son emprise. La définition de l'emprise est importante car, outre la commodité pour visualiser les résultats, elle permet de facilement déterminer les moniteurs situés à l'intérieur d'un espace géographique (et donc déterminer si l'espace est bien généralisé). Le centre du moniteur est un point de l'espace qui correspond à une localisation approximative. Par exemple, si la contrainte relative au moniteur est une contrainte micro comme la contrainte d'aire minimum d'un bâtiment, le centre du moniteur est le centre de gravité du bâtiment concerné. Dans le cas d'une contrainte relationnelle, le moniteur va être centré au milieu du segment reliant les centres de gravité des deux objets concernés par la relation contrainte. Dans le cas d'une contrainte meso, le moniteur sera placé au centre de gravité du groupe d'objets formant l'objet meso. L'emprise du moniteur représente de son côté l'emprise sur la carte du conflit formalisé par le moniteur. La Figure C-44 montre trois exemples de calculs d'emprise pour des moniteurs de contraintes micro, meso ou relationnelles. Les moniteurs de contrainte étant issus de classes spécialisant la classe *Moniteur de contrainte*, cette méthode de

calcul d'emprise peut être redéfinie par polymorphisme pour déterminer des emprises plus adaptées à la contrainte concernée.

C.10.2. Fonctionnement des moniteurs de contrainte

Les moniteurs de contrainte sont utilisés dans CollaGen pour les trois fonctions : identifier, localiser et évaluer les conflits. La Figure C-45 illustre comment la combinaison des trois fonctions d'un moniteur de contrainte en fait un outil très pratique pour guider les composants garantissant la collaboration des processus.

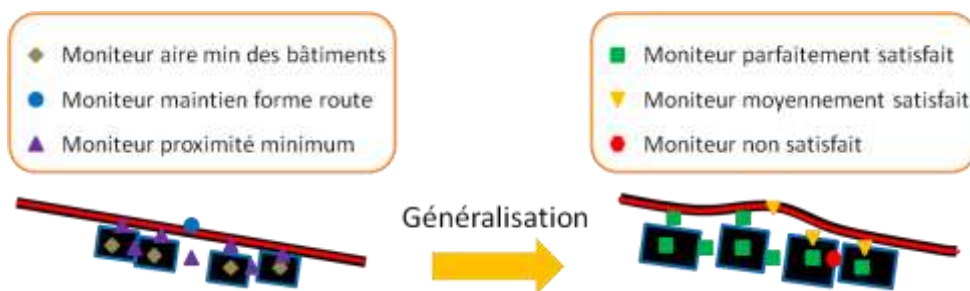


Figure C-45. Principe de la localisation des moniteurs de contrainte : ils identifient géographiquement les conflits.

Les moniteurs de contrainte permettent ainsi de localiser, avec une précision suffisante, les zones mal généralisées ou nécessitant une généralisation (suivant la satisfaction des états précédents des moniteurs). Les états précédents sont importants, car, pour atteindre un bon état, il faut souvent appliquer plusieurs opérations voire processus, donc, l'analyse de l'évolution aide à l'évaluation. Ils sont ainsi un support pratique pour l'évaluation globale de l'état de satisfaction général d'un espace géographique (s'il a besoin d'être généralisé ou pas). Les moniteurs de contrainte sont utilisés par les composants registre, d'observation et d'évaluation, d'orchestration et de gestion des effets de bord et sont créés par les deux autres composants de traduction (qui met en rapport les fonctions d'évaluation du moniteur avec les bonnes contraintes formelles) et d'enrichissement qui place les moniteurs sur la carte.

La conception d'un système de Généralisation Collaborative comme le modèle CollaGen nécessite la mise en œuvre de six composants indispensables comme le rappelle la Figure C-46. Les parties suivantes, de C.11 à C.16, décrivent en détail comment chacun de ces composants est modélisé avec dans l'ordre le composant d'enrichissement géographique, le composant de traduction, le composant registre, le composant d'observation et d'évaluation, le composant d'orchestration et enfin le composant de gestion des effets de bord.

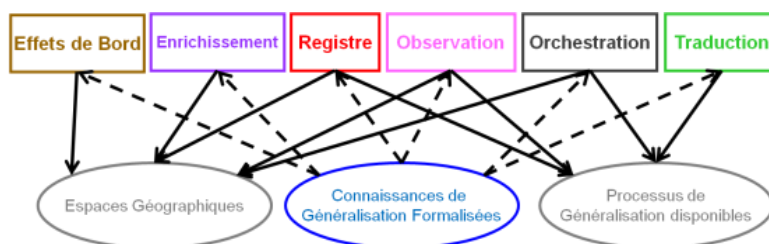


Figure C-46. Les composants (rectangles) et les ressources (ellipses) de la Généralisation Collaborative mis en œuvre dans CollaGen.

C.11. COMPOSANT D'ENRICHISSEMENT GEOGRAPHIQUE

Le composant d'enrichissement géographique est le premier à être mis en action lors d'une généralisation collaborative et il est responsable de tous les enrichissements nécessaires (Figure C-47). Les espaces géographiques représentent la plus grande partie de cet enrichissement. Dans un premier temps, nous allons décrire comment le composant prend en charge l'enrichissement en concepts meso et procéduraux puis l'enrichissement en relations géographiques. Nous présentons ensuite les mécanismes permettant au composant de construire les espaces géographiques et enfin, nous expliquons le principe de la boîte à outils utilisée par le composant pour réaliser ces enrichissements.

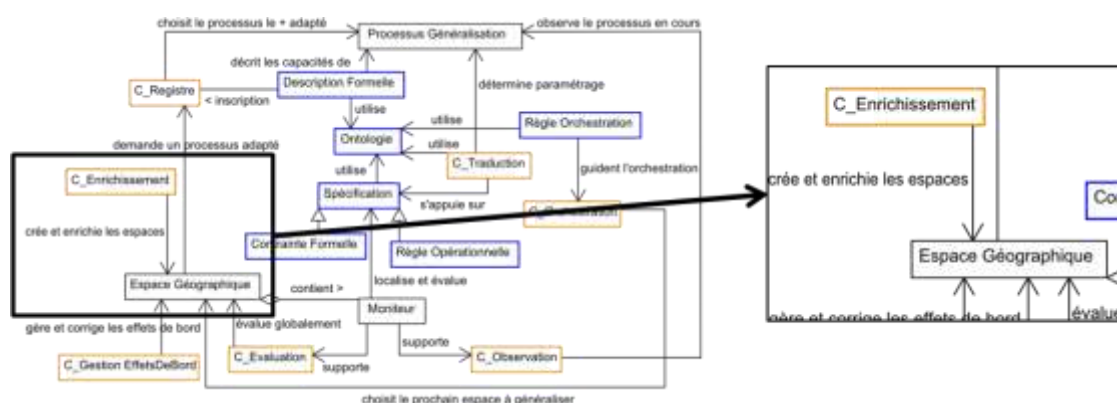


Figure C-47. Le composant d'enrichissement permet principalement de créer et enrichir les espaces géographiques.

C.11.1. Enrichir les données par des concepts meso et procéduraux

L'enrichissement des données consiste à rendre explicites dans la base de données géographique initiale, les concepts implicites nécessaires à la compréhension du contexte géographique et donc nécessaires aux processus de généralisation (Mackness and Edwardes, 2002). Ces concepts implicites, qu'ils soient meso ou procéduraux, sont décrits dans l'ontologie de la généralisation et pour chacun des processus, la description formelle permet d'identifier ceux de ces concepts qui sont nécessaires au processus. La base de données initiale doit donc être modifiée en ajoutant les classes nécessaires à ces enrichissements meso et procéduraux comme le montre la Figure C-48.

Dans un premier temps, nous ajoutons une classe chapeau de toutes les classes initiales d'objets géographiques, que l'on va appeler *ObjetCollaGen* sur laquelle est défini un attribut *estSupprimé* qui vaut vrai quand l'objet est actuellement considéré comme supprimé par CollaGen (un des processus précédemment utilisés l'a éliminé). Un *objet procédural* est un *ObjetCollaGen* particulier en relation avec un des processus décrits dans l'ontologie (son explicitation n'est utile que si ce processus peut être utilisé dans CollaGen). Un *objet meso* est lui un objet *CollaGen* composé d'objets *CollaGen*. La figure montre un exemple de spécialisation de ces classes. Si nous prenons le cas de *l'îlot Urbain*, nous remarquons que l'enrichissement concerne aussi les propriétés (ici la densité de l'îlot) et les relations (ici avec les tronçons de route entourant l'îlot) définies dans l'ontologie. Ceci permet de

faciliter la réalisation de moniteurs de contrainte qui peuvent s'appuyer sur les propriétés et relations présentes dans l'ontologie pour mesurer l'état des contraintes.

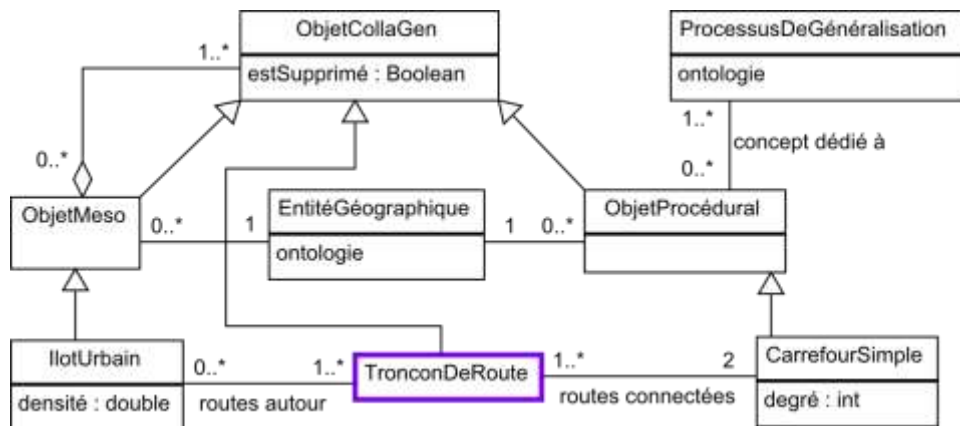


Figure C-48. Diagramme de classe UML représentant les enrichissements de type meso ou procédural apportés aux données initiales par le composant d'enrichissement avec un exemple de spécialisation. Seule la classe *TronconDeRoute* fait partie des données initiales dans cet exemple.

Une fois le schéma de données enrichi avec les classes, propriétés et relations relatives aux concepts meso et procéduraux, le composant d'enrichissement est responsable de la création de ces nouveaux objets. Pour cela, le composant fait appel à des algorithmes dédiés, ajoutés par le développeur, qui s'appuient uniquement sur les concepts de l'ontologie : ils peuvent ainsi être génériques et indépendants du schéma de données initial. Par exemple, enrichir les données avec le concept procédural de carrefour simple implique le développement d'un algorithme capable (1) d'identifier ces carrefours à partir des objets de la base de données correspondant au concept de route et (2) d'instancier la relation entre les carrefours et les routes. De même, pour que la propriété degré des carrefours puisse être mesurée à tout moment durant la généralisation collaborative, par des moniteurs de contrainte notamment, les propriétés doivent être munies d'algorithmes de mesure (ici, il se contente de compter les routes connectées qui ne sont pas éliminées). La Figure C-49 résume les responsabilités réparties aux différents rôles pour chacun des enrichissements meso et procédural dans CollaGen. Le lien entre concept décrit dans l'ontologie nécessitant un enrichissement et algorithme de création du concept est réalisé par le composant de traduction (C.12.2).

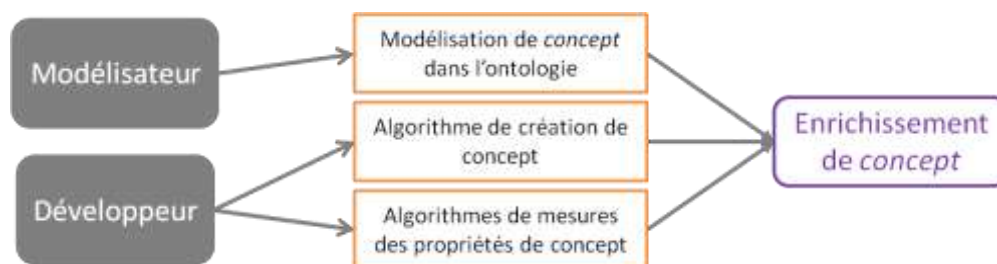


Figure C-49. Schéma résumant les besoins pour l'enrichissement d'un concept meso ou procédural quelconque dans CollaGen.

Il reste maintenant à déterminer comment sont choisis les enrichissements à réaliser et à quel moment du processus collaboratif ils sont réalisés par le composant d'enrichissement géographique. Les enrichissements nécessaires au fonctionnement d'un processus de généralisation donné sont décrits dans la description formelle de chacun des processus. Nous avons vu que certains de ces

enrichissements comme les « strokes » (Thomson, 2006) n'avaient de sens que s'ils étaient réalisés sur l'ensemble des données. Donc, dans un premier temps, la première action réalisée lors d'une généralisation collaborative est la consultation des descriptions des processus disponibles par le composant d'enrichissement. S'il identifie dans ces descriptions des concepts décrits comme globaux, il les construit un par un sur l'ensemble de l'espace généralisé. Les autres enrichissements n'interviennent que plus tard dans le déroulement d'une généralisation par CollaGen : quand un espace géographique a été choisi pour être généralisé et qu'on a trouvé le processus le plus adapté, le composant d'enrichissement consulte à nouveau la description de ce processus et crée, uniquement dans l'espace concerné, les enrichissements en concepts meso et procéduraux nécessaires.

C.11.2. Expliciter les relations géographiques

Nous avons vu dans les parties précédentes que certaines des spécifications de la carte pouvaient porter non pas sur des objets géographiques mais sur une relation entre deux de ces objets. Les contraintes relationnelles sont des contraintes généralement sur le maintien, la caricature ou l'élimination d'une relation géographique. Afin d'évaluer ces contraintes, nous proposons d'explicitement ces relations géographiques contraintes par des objets *relation* dans la base de données. Les moniteurs de contrainte relationnelle sont donc liés à un objet relation plutôt qu'à un objet géographique. La Figure C-50 montre le schéma que nous proposons pour enrichir les données en relations, en nous inspirant de l'instanciation des relations dans CartACom (Duchêne, 2004).

Une *Relation Géographique* est un *Objet CollaGen* en relation avec deux objets CollaGen et possédant une géométrie linéaire reliant ces deux objets pour permettre un contrôle visuel des relations instanciées. Une relation correspond toujours à une des relations géographiques décrites dans l'ontologie de la généralisation. Une relation géographique est dotée d'une méthode pour chercher les voisins candidats à former une relation avec un objet donné et d'une méthode permettant de valider que deux objets sont bien en relation selon cette relation. Ce lien avec les méthodes est fait à travers l'ontologie car les méthodes sont bien liées au concept de relation et non à un objet relation qui n'existe pas encore (CollaGen ne sait pas si une relation existe quand il la cherche). Ces méthodes sont représentées sous la forme d'objets méthodes de classes 'singletons' : il n'existe qu'un objet-méthode par classe spécialisée de *Recherche De Relation* (et de *Validation de Relation*), objet qui permet d'appliquer la méthode qui a été spécialisée par polymorphisme.

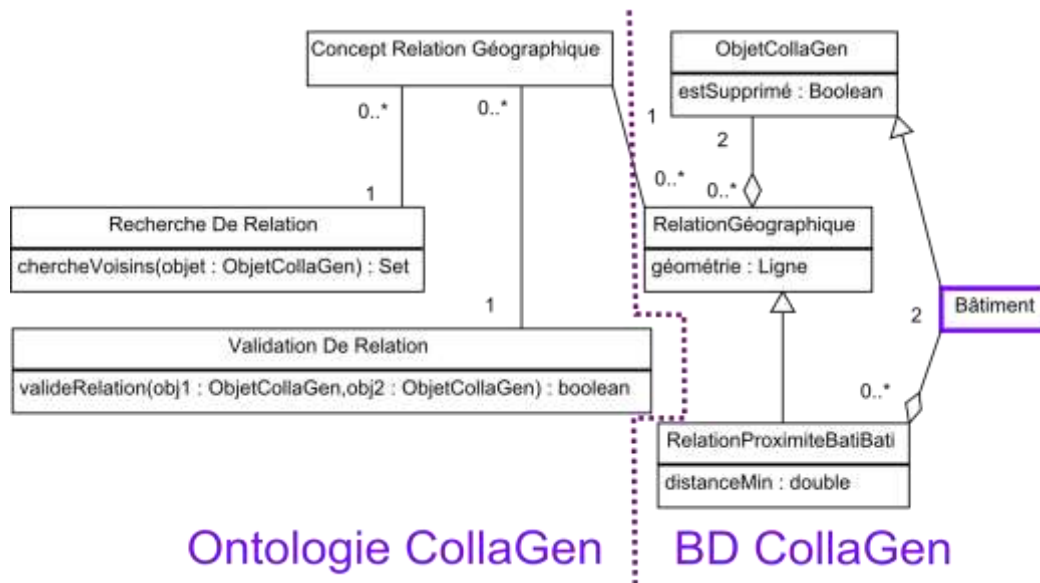


Figure C-50. Diagramme de classe UML représentant les enrichissements en relations spatiales avec un exemple de spécialisation pour la relation de proximité entre bâtiments.

Dans l'exemple spécialisé pour la relation de proximité entre bâtiments, une propriété est ajoutée comme dans le cas des enrichissements meso et le lien avec les objets composant la relation est aussi spécialisé (avec des bâtiments dans ce cas). Nous pouvons noter que contrairement au modèle de relation dans CartACom (Duchêne, 2004, p.65), la relation n'est pas directement liée à une contrainte et n'a donc pas de valeur but ou de satisfaction et peut même posséder plusieurs propriétés suivant la modélisation dans l'ontologie. Cela permet de modéliser les relations en tant qu'entité du schéma adapté de la généralisation collaborative et non plus comme un simple support des contraintes.

Les enrichissements en relations géographiques sont construits à partir des méthodes de recherche et de validation des relations. Quand CollaGen veut construire les instances d'une relation donnée, l'algorithme suivant est utilisé :

```

ConstruireRelation(nomRelation)
  Initialisation Concept relation ← nomRelation
  Initialisation Concept concept1 ← relation.concept1
  Initialisation Ensemble d'objets objets ← {objets de type concept1}
  Pour tous les objets obj de objets faire
    Initialisation Ensemble voisins ← relation.chercheVoisins(obj)
    Pour tous les objets vois de voisins faire
      Si (relation.valideRelation(obj, vois))
        Construire relation de type relation entre obj et vois
    Fin pour
  Fin pour

```

Les enrichissements en relation sont réalisés en début de processus collaboratif, en même temps que les enrichissements globaux par le composant d'enrichissement. Le composant va consulter les contraintes et les règles opérationnelles, pour voir quelles sont les relations géographiques de l'ontologie dont doit tenir compte la généralisation collaborative. Les relations ainsi contraintes sont ensuite créées avec l'algorithme précédent.

C.11.3. Construire les espaces géographiques

La tâche centrale assignée au composant d'enrichissement est de découper les données en espaces géographiques territoriaux et thématiques. L'enrichissement consiste ici en une spécialisation du modèle d'espace géographique présenté en partie B.4.2. Si les espaces thématiques sont construits simplement en déterminant les classes d'objets correspondant au thème de l'espace, les espaces territoriaux et mixtes sont, eux, construits en utilisant des algorithmes d'analyse spatiale adaptés. Le développeur du composant d'enrichissement doit donc associer chaque espace décrit dans l'ontologie avec un algorithme choisi comme le plus pertinent (voir E.4.2). Il est possible que la construction de certains espaces s'appuie sur la construction d'autres espaces, auquel cas, ces dépendances doivent être gérées par le composant qui va commencer par construire ceux qui sont indépendants des autres. Un exemple de dépendance est l'algorithme de construction des espaces ruraux présenté dans le dernier chapitre (E.1.5) qui dépend de la construction des espaces urbains et périurbains.

Suivant si les espaces sont clés ou facilitateurs, spécifiés ou émergents (B.4.2), ils ne sont pas construits par le composant d'enrichissement au même moment. Dans un premier temps, tous les espaces considérés comme clé dans l'ontologie sont construits par le composant en utilisant leur algorithme associé, dans l'ordre des dépendances s'il en existe.

Pour la construction des espaces facilitateurs (i.e. les espaces construits pour optimiser l'utilisation d'un des processus disponibles), le composant d'enrichissement lit les descriptions des processus disponibles. Si certains de ces processus ont comme seule pré-condition avec un indice de confiance élevé, un espace géographique non clé, il est construit par le composant comme espace facilitateur (car l'indice de confiance traduit la fiabilité du processus sur ce type d'espace). Par exemple, si un processus a dans sa description deux pré-conditions avec un espace non-clé de confiance 5 et un espace clé de confiance 1, l'espace non-clé est considéré comme facilitateur (et il est donc construit par le composant). Si par contre les deux pré-conditions sont un espace non-clé et un espace clé avec la même confiance de 4, l'espace non-clé ne sera pas considéré comme facilitateur, car il n'est pas jugé nécessaire à une utilisation optimale du processus mis à disposition.

Pour la construction des espaces spécifiés (i.e. ajoutés d'après les spécifications), le composant d'enrichissement prend connaissance de l'ensemble des contraintes de généralisation et des règles opérationnelles. Si l'une de ces contraintes ou règles est formalisée avec un espace d'application non clé, l'espace est construit par le composant. Ces trois actions de construction d'espace sont effectuées l'une après l'autre et si un espace a été construit comme facilitateur, il ne va pas être considéré dans la recherche d'espaces spécifiés.

Enfin, pour la construction d'espaces émergents, le processus est complètement différent. Les espaces émergents sont des sous espaces qui correspondent à un groupe de conflits non résolus par le processus généralisant l'espace, observé par le composant d'observation qui fait alors appel au composant d'enrichissement pour délimiter l'espace autour de ce conflit. Cette construction, quand elle a lieu, est donc faite au cours de la généralisation d'un espace. Le mécanisme complet d'observation de conflits et de construction de l'espace émergent est décrit d'un seul bloc dans la partie concernant le composant d'observation et d'évaluation (C.14).

Une fois que tous les espaces sont construits (sans prendre en compte les espaces émergents), le voisinage de chaque espace avec les autres espaces est construit automatiquement par le composant d'enrichissement. Ce voisinage sert à la gestion des effets de bord.

C.11.4. Création des moniteurs de contrainte

Le composant d'enrichissement est enfin responsable d'un dernier type d'enrichissement des données initiales : la création des moniteurs de contrainte. Ainsi, il va construire pour chacun des objets géographiques de la base, qu'ils soient présents initialement ou issus des enrichissements précédents, un moniteur par contrainte ou règle opérationnelle concernant cet objet, ce qui représente une quantité très importante d'objets nouveaux dans la base. Cette création des moniteurs ne peut se faire qu'avant toute généralisation car les moniteurs doivent connaître l'état initial des situations qu'ils évaluent pour savoir si elles évoluent dans le bon sens au cours de la généralisation.

C.11.5. La Boîte à outils

Nous avons vu dans la partie C.2 que le rôle de développeur de CollaGen consistait en partie à alimenter une boîte à outils dont les composants peuvent se servir. Le composant d'enrichissement est ainsi le composant le plus gourmand en outils car il a besoin :

- d'algorithmes et de mesures pour l'enrichissement en objets meso et procéduraux.
- d'une méthode de recherche de voisins pour chaque relation.
- d'une méthode de validation pour chaque relation.
- d'au moins un algorithme pour la construction de chacun des types d'espaces territoriaux et mixtes.
- d'un ensemble de méthodes pour calculer les valeurs, courante, but et satisfaction des moniteurs pour chaque contrainte et règle.

L'absence d'outil pour un de ces cas entraîne sa non utilisation dans CollaGen. De même, s'il n'existe pas dans la boîte à outil de méthodes pour mesurer la valeur courante, la valeur but ou la satisfaction de la contrainte de maintien la forme générale des zones arborées, cette contrainte ne sera pas prise en compte dans CollaGen, ce qui réduit la qualité de la généralisation finale. Ce premier composant illustre le fait que la qualité des résultats de CollaGen dépend fortement de la quantité et de la qualité des outils qui lui sont mis à disposition. Néanmoins, l'avantage de se reposer sur des connaissances formelles et une ontologie est que la conception de ces outils est relativement générique : les outils pourront être utilisés pour plusieurs applications.

Le composant d'enrichissement géographique permet de réaliser automatiquement tous les enrichissements de la base de données initiale nécessaire au déroulement de la généralisation collaborative de CollaGen. Il s'agit des espaces géographiques construits en plusieurs temps mais aussi des concepts meso requis par les processus, des relations géographiques sujettes à des contraintes et des moniteurs de contrainte. La construction de ces enrichissements repose sur une boîte à outils contenant des algorithmes d'analyse spatiale. La qualité générale de la généralisation collaborative est très influencée par la qualité et l'exhaustivité des outils dans la boîte.

C.12. COMPOSANT DE TRADUCTION

Dans CollaGen, le composant de traduction a pour rôle principal de déterminer le paramétrage spécifique de chaque processus de généralisation en fonction des spécifications formelles de la carte et de l'ontologie (Figure C-51).

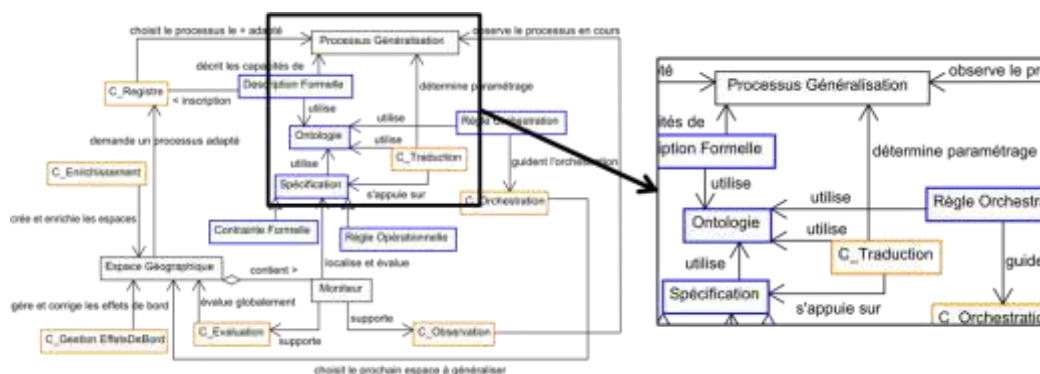


Figure C-51. Le composant de traduction permet de déterminer le paramétrage des processus de généralisation à partir des spécifications formelles et de l'ontologie.

De manière plus générale, le composant de traduction a pour fonction d'assurer l'interface entre les éléments hétérogènes (i.e. ne suivent pas de standard pour éviter l'hétérogénéité syntaxique, syntactique et sémantique) qui alimentent le système (schéma de données initial, les algorithmes de la boîte à outils, les processus de généralisation et les algorithmes des moniteurs de contrainte) et les éléments formels qui servent de format pivot garantissant l'interopérabilité (l'ontologie de la généralisation, les contraintes formelles et les descriptions formelles des processus) (Figure C-52). Les autres composants peuvent ainsi se contenter de manipuler ces éléments génériques sans se soucier des problèmes d'interopérabilité causés par les éléments hétérogènes.

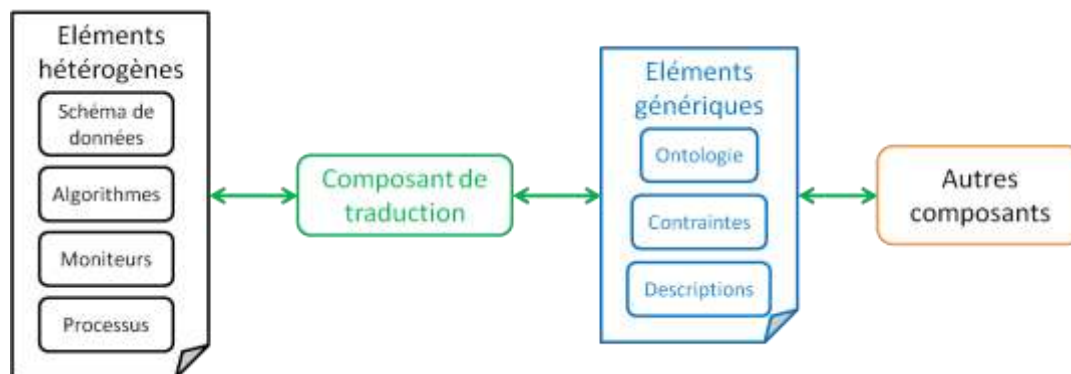


Figure C-52. Le composant de traduction sert d'interface entre les connaissances formelles génériques utilisées par les autres composants et les éléments hétérogènes présents dans CollaGen.

Plusieurs outils sont ainsi regroupés dans ce composant de traduction pour assurer cette interface. Tout d'abord, le composant assure l'annotation sémantique du schéma de données initial par les concepts de l'ontologie, afin d'éviter aux composants de manipuler cette source d'hétérogénéité que peut être le schéma de données : quand ils veulent accéder aux données, les autres composants manipulent directement les concepts de l'ontologie (en fait, ils manipulent notre schéma objet représentant l'ontologie, voir p. 85). Le composant de traduction assure également un appariement entre l'ontologie et la boîte à outils pour désigner les outils qui permettent au composant

d'enrichissement de construire par enrichissement chaque concept (meso, relation géographique etc.) de l'ontologie. Le composant de traduction comporte aussi un système d'instanciation des contraintes formelles et des règles opérationnelles en associant celles-ci avec un type de moniteur de contrainte chargé de les évaluer. Puis, la fonction principale du composant de traduction est d'assurer l'interopérabilité au niveau des paramètres des processus de généralisation disponibles. Ceci est assuré grâce à des fonctions de traduction associées à chaque processus dont la forme est décrite dans cette partie. Enfin, la dernière petite mais essentielle fonction du composant de traduction est d'assurer la traduction des unités quand les différents composants manipulent par exemple des mesures en mètres terrain ou mm carte.

C.12.1. Annotation sémantique de la base de données initiale

La mise en relation de ce que l'on appelle les ressources d'information dans le domaine du Web sémantique (soit dans notre cas les bases de données géographiques) et d'une ontologie est un procédé appelé *annotation sémantique*. Il s'agit de déterminer à quel concept de l'ontologie correspond une classe d'objets géographiques de la base de données. (Lemmens, 2008) distingue trois types de méthodes pour stocker les annotations sémantiques :

- **Balisage sémantique** : on encode les ressources d'information avec une référence directe à l'ontologie. Dans le cas des données géographiques, on ajoute à chaque classe un attribut « balise sémantique » qui contient un lien vers un concept de l'ontologie.
- **Recensement** : l'annotation est stockée dans l'ontologie. Dans le cas des données géographiques, on inclut le nom des classes d'objets correspondant à un concept dans la description de ce concept.
- **Appariement sémantique** (registration mapping en anglais) : l'annotation est faite dans une source indépendante de l'ontologie et des ressources, sous la forme d'un registre qui contient une liste de référence de l'ontologie vers la ressource et vice versa. Dans le cas des données géographiques, l'appariement sémantique contient un ensemble de relations entre classes d'objets et concepts de l'ontologie.

CollaGen a pour objectif de permettre des généralisations collaboratives génériques du point de vue du schéma de la base de données initiale, donc les modélisations et les noms de classes doivent pouvoir varier fortement d'un cas à l'autre. Pour illustrer ce besoin, nous pouvons noter que pour deux des zones tests utilisées pour tester CollaGen (voir E.3.1), le schéma est déjà très différent : par exemple, le concept de l'ontologie « bâtiment » correspond aux classes « Building », « Industrial_Building » dans une zone ; il correspond aux classes « Public_Building » et « Bâtiment » dans l'autre zone. La généralité viserait à gérer encore plus de cas de schémas possibles, ce qui écarte l'hypothèse de stockage par recensement : le recensement des ressources pour chaque concept de notre ontologie ne serait jamais exhaustif pour associer à chaque concept les noms de classes possibles.

En outre, en plus des classes du schéma de données initial que nous souhaitons annoter, CollaGen doit pouvoir utiliser de façon interopérable les caractères des concepts géographiques concernés par des contraintes. Cela implique l'annotation sémantique non seulement, des classes mais aussi des

attributs de la base de données initiale. Il paraît donc difficilement envisageable de stocker l'annotation sémantique sous la forme d'un balisage sémantique des données : annoter toutes les classes et tous les attributs du schéma avec les concepts ontologiques correspondant va alourdir fortement le schéma et poser des problèmes de modélisation.

Pour ces deux raisons, nous optons pour une annotation sémantique de nos données par appariement sémantique dans une ressource indépendante intégrée au composant de traduction.

L'annotation sémantique en général est une tâche fastidieuse lorsqu'elle est effectuée manuellement, a fortiori dans le cas de l'annotation de toute une base de données géographique. Il existe assez peu de propositions pour automatiser efficacement cette tâche d'annotation (Handsuh et Staab, 2003). De plus, par leur complexité, les données géographiques sont plus difficiles à annoter automatiquement qu'une simple ressource HTML. (Klien et Lutz, 2005) propose toutefois une approche semi-automatique basée sur la description des relations spatiales et la recherche de ces relations spatiales dans les données pour annoter le schéma. Il existe toutefois de nombreux outils pour annoter des ressources, notamment des ressources Web, de manière semi-automatique et interactive, rendant l'annotation un peu moins fastidieuse (Popov et al, 2003).

L'objectif de notre thèse n'étant pas de développer une méthode d'annotation sémantique automatique, nous optons dans CollaGen pour une annotation sémantique interactive, réalisée par l'utilisateur avant d'exécuter sa généralisation.

Nous proposons de mettre en relation tous les aspects du schéma de données avec des éléments (concept, propriété ou lien) de l'ontologie grâce au registre d'appariement sémantique du composant de traduction (Figure C-53). Tout d'abord, les **classes** sont annotées avec le concept de l'ontologie qui leur correspond. Par exemple, le concept « route » de l'ontologie est associé à la classe « Tronçon de route » dans le cas d'un schéma semblable à celui de la BDTopo® de l'IGN. Ensuite, les **attributs** de ces classes sont annotés par des propriétés de l'ontologie quand ils correspondent à des caractères susceptibles d'être contraints. Par exemple, l'attribut donnant l'aire d'un bâtiment est annoté avec la propriété ontologique correspondante mais le toponyme des cours d'eau ne l'est pas, car il ne risque pas d'être concerné par une contrainte de généralisation. Ensuite, si certains concepts de l'ontologie ne correspondent pas à une classe du schéma mais plutôt à une valeur d'attribut, ils sont associés dans un ensemble particulier de l'appariement sémantique avec la classe, l'attribut et la valeur d'attribut correspondant. Par exemple, le concept d'autoroute correspond dans le schéma de la BDTopo® à la valeur « Autoroute » de l'attribut « Classement » sur la classe « Tronçon de route ». Les **liens** entre concepts dans l'ontologie sont également appariés aux liens entre classes dans le schéma de données. Par exemple, le lien « est_carrefour_de_routes » entre le concept « carrefour simple » et le concept « route » (voir Figure C-48) est apparié au lien « Carrefour_simple_concerne_troncons » entre les classes équivalentes dans le schéma (qui a été créé par le composant d'enrichissement). Enfin, les **relations** géographiques sont annotées de manière spécifique car l'appariement concerne à la fois la relation en elle-même et les concepts associés. Si par exemple nous cherchons à annoter deux classes (dans le schéma) de relations d'orientation relative, l'une entre un bâtiment et une route et l'autre entre deux bâtiments (il y a deux classes au niveau du schéma, car les méthodes de mesure de ces deux spécialisations de la relation ne sont pas les mêmes), nous pouvons alors les annoter avec le même concept de relation

d'orientation relative avec, dans un cas les concepts de « bâtiment » et de « route » et seulement le concept de « bâtiment » dans l'autre cas.

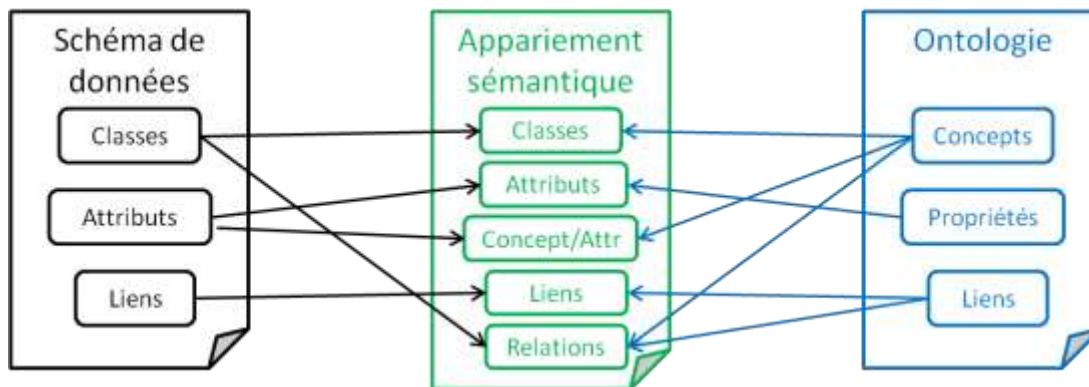


Figure C-53. L'appariement sémantique et ses différents ensembles mettent en relation le schéma de données et l'ontologie de la généralisation. Les flèches désignent à quoi sont reliés dans cet appariement les éléments du schéma et l'ontologie.

Il faut noter que le registre d'appariement sémantique est doté de méthodes pour l'interroger dans les deux sens (en partant du concept ou de la classe du schéma) quel que soit le choix d'implémentation réalisé pour le stockage. Si le sens d'utilisation privilégié dans CollaGen est clairement de chercher la classe du schéma qui correspond à un concept de l'ontologie donné, il peut être pratique pour certaines manipulations de pouvoir chercher dans l'autre sens.

C.12.2. Appariement entre ontologie et boîte à outils

Nous avons vu dans la partie décrivant le fonctionnement du composant d'enrichissement que la création des enrichissements en concepts meso, procéduraux ou en relations géographiques dépendait de l'utilisation d'algorithmes d'analyse spatiale présents dans la boîte à outil de CollaGen. Pour que les concepts soient construits avec les bons algorithmes, ils doivent être appariés à ces algorithmes par le composant de traduction.

Dans un premier temps, les algorithmes de la boîte à outils doivent être modélisés de manière à ce que leur exécution puisse être réalisée de manière générique, à partir du concept de l'ontologie correspondant. La Figure C-54 montre le modèle de la boîte à outils qui permet l'accès générique à ses algorithmes par la méthode *executer()*. Un algorithme (objet de la classe *Algorithme*) est ainsi toujours relié à l'appariement sémantique (présenté dans la partie précédente), qui lui permet de faire correspondre aux concepts de l'ontologie mis en jeu, les classes du schéma de données. Dans ce modèle, les paramètres et les résultats de l'algorithme sont des attributs des classes d'algorithmes. Par exemple, un algorithme de création d'espace géographique (un objet de la classe *AlgoEspace* de notre schéma) a un seul paramètre (la base de données initiale) et ne renvoie pas de résultat (il crée simplement les espaces géographiques dans la classe identifiée par l'appariement sémantique) ; une méthode de validation de relation, c'est-à-dire un objet de la classe *Validation Relation Géographique* (déjà présentée p. 120, tout comme la classe *Recherche De Relation*), a deux paramètres (les deux objets éventuellement concernés par la relation) et renvoie un booléen valant vrai si une relation doit être créée entre ces deux objets.

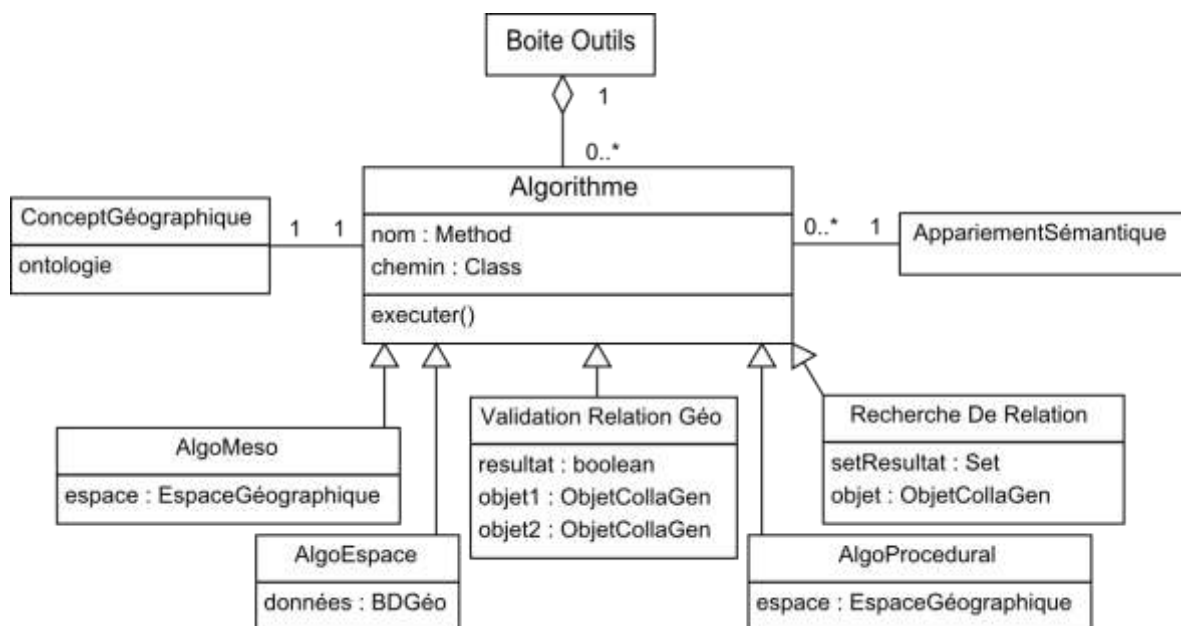


Figure C-54. Diagramme de classe UML représentant la formalisation des algorithmes de la boîte à outils pour leur exécution générique.

Dans un deuxième temps, l'appariement entre les algorithmes de la boîte à outils et les concepts géographiques créés par ces algorithmes revient à instancier le lien apparaissant sur le diagramme entre l'algorithme et le concept géographique. Comme dans le cas de l'appariement sémantique, nous ne cherchons pas à développer une méthode d'annotation automatique des algorithmes donc, cet appariement est réalisé interactivement. Le même mécanisme est assuré par le composant de traduction pour les mesures d'un caractère comme le montre la Figure C-55 à la différence qu'ils sont appariés à des propriétés de l'ontologie.

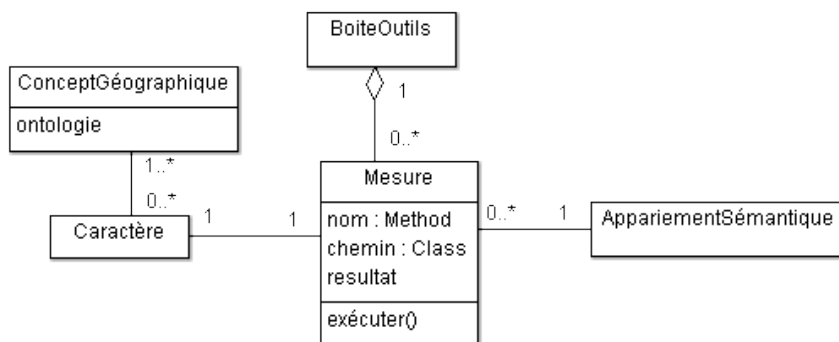


Figure C-55. Diagramme de classe UML représentant la formalisation des mesures de caractère de la boîte à outils.

C.12.3. Appariement entre contraintes formelles et moniteurs

Dans CollaGen, des moniteurs de contrainte permettent d'identifier et d'évaluer les contraintes formelles sur chaque objet concerné par ces contraintes (voir C.10). Ces moniteurs fonctionnent à partir d'une contrainte formelle (l'évaluation de la satisfaction du moniteur d'aire minimum se réfère à la valeur contenue dans la contrainte formelle d'aire minimum liée, 0.2 mm² carte par exemple)

mais il faut un système pour associer les contraintes formelles avec les moniteurs souhaités. Nous pouvons imaginer par exemple que la boîte à outils soit dotée de plusieurs moniteurs différents pouvant gérer une même contrainte formelle. Le choix du moniteur associé à chaque contrainte est donc d'autant plus important.



Figure C-56. Outil d'appariement entre les contraintes d'AGENT et leurs méthodes de monitoring dans le logiciel Clarity™.

La mise en œuvre du modèle de généralisation AGENT dans le logiciel Clarity™ de la société 1Spatial™ fournit un exemple opérationnel s'apparentant à ce type d'appariement. Le logiciel propose un formalisme et un outil pour lier la contrainte définie par ailleurs dans le système avec l'équivalent de notre moniteur qui est dans ce cas un ensemble de méthodes comme « `calculeValeurBut()` » (Figure C-56). Clarity™ peut ainsi associer à chacune de ces méthodes essentielles à l'évaluation en ligne d'une contrainte, une méthode programmée en Java, désignée ici par son emplacement dans l'arborescence du code utilisé par le logiciel.

Dans CollaGen, les moniteurs de contrainte sont un peu plus qu'un ensemble de méthodes d'évaluation des contraintes et nous choisissons plutôt d'apparier chacune des contraintes formelles et chacune des règles opérationnelles à l'un des types spécialisés de moniteurs de contrainte contenus dans la boîte à outils (Figure C-57). Ainsi, comme dans l'exemple de la figure, plusieurs moniteurs peuvent être disponibles pour une contrainte et plusieurs contraintes peuvent pointer sur le même moniteur. Dans le premier cas, il existe deux moniteurs d'aire minimum, car celui dédié aux bâtiments utilise l'aire initiale pour évaluer l'aire courante (si le bâtiment initial était légèrement plus grand que l'aire minimale, il doit aussi avoir une aire plus grande que son aire initiale). Donc la contrainte d'aire minimum des bâtiments est appariée au moniteur le plus approprié. Dans le deuxième cas, l'évaluation et la localisation des contraintes d'aire minimum des forêts et des lacs ont des comportements similaires donc les deux contraintes sont appariées à la même classe de moniteur bien que les seuils ne soient pas nécessairement les mêmes : une instance moniteur d'aire minimum liée à la contrainte formelle sur les forêts utilise le seuil décrit dans cette contrainte formelle, idem pour un moniteur d'aire minimum correspondant à un lac.

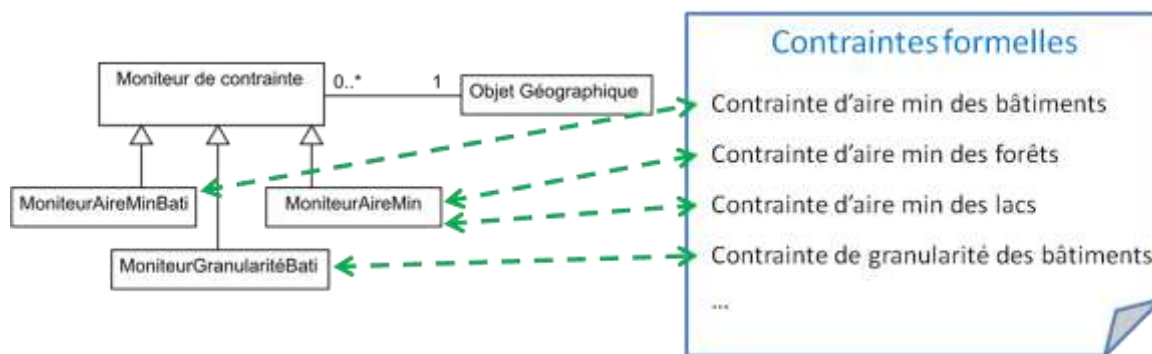


Figure C-57. Appariement des contraintes formelles avec un type de moniteur de contrainte assuré par le composant de traduction.

C.12.4. Les Fonctions de traduction

La fonction qui a donné son nom au composant de traduction est d'assurer la traduction des spécifications formelles (contraintes et règles opérationnelles) en paramètres spécifiques de chacun des processus de généralisation disponibles. Dans cette partie, nous effectuons d'abord un rappel sur les divers paramètres que l'on peut trouver dans des processus automatiques de généralisation, puis nous décrivons un modèle pour remplir sa fonction de traduction et enfin nous détaillons les difficultés que l'on peut rencontrer lors de cette opération.

Diversité des paramètres

Afin de bien mettre en valeur la diversité dans les paramètres rencontrés pour exécuter des processus automatiques de généralisation, nous présentons quatre processus implémentés sur notre plate forme de recherche pour être des processus disponibles du prototype de CollaGen que nous avons développé (voir CHAPITRE E). Le premier exemple provient de l'implémentation du modèle AGENT dans le logiciel Clarity™. Pour effectuer une généralisation à l'aide d'AGENT, il faut définir à la fois, le modèle de données qui associe les types d'agents avec les classes de données géographiques, et surtout l'entité regroupant les spécifications de la généralisation dite 'AgentMapspec'. Cette entité liste pour chaque type d'agent les contraintes que l'on fait porter sur lui, avec une valeur d'importance et une valeur de priorité (i.e. urgence de traitement *a priori*). Elle regroupe également tous les paramètres utilisés par les contraintes. Par exemple, le paramètre « Building Minimum Size » est utilisé notamment par la contrainte dite « Building Size » qui contraint l'aire minimum d'un agent bâtiment. L'échelle finale de la carte doit également être saisie dans cette entité 'AgentMapspec'.

L'implémentation des Beams dans le logiciel Clarity™ nécessite un type de paramétrage assez différent. Il nécessite d'abord de définir un certain nombre de valeurs numériques propres au fonctionnement du processus comme le nombre maximum d'itérations ou le facteur de force de réaction, ou en rapport avec les spécifications de la carte comme le seuil de séparation entre les symboles. Il faut également définir le nom des classes d'objets concernés comme dans le cas précédent. Il faut enfin définir les propriétés de résistance à l'étirement et la compression pour chaque classe traitée. Nous pouvons ainsi définir sur les classes de cours d'eau et de routes mineures moins de résistance à la compression que sur les classes de routes importantes (les cours peuvent être plus tordus que les routes sans perdre leur sens géographique).

L'implémentation de la généralisation par Moindres Carrés que nous avons développée dans le cadre de CollaGen (voir E.1.2) repose sur des paramètres que nous avons encapsulés sous forme de contraintes. Il s'agit dans ce cas de définir un système d'équation sur-contraint (plus d'équations que d'inconnues) sur les coordonnées finales de chaque point des objets traités. Ainsi, une contrainte sur le maintien de la position initiale d'un bâtiment se traduit par deux équations pour chacun des points dans la géométrie du bâtiment : une équation $\Delta x = 0$ et une équation $\Delta y = 0$. Paramétrer le processus par moindres carrés revient donc à ajouter des équations dans le système global en fonction des spécifications qui s'appliquent sur chacun des objets. Un autre point clé de ce paramétrage, car difficile à optimiser (Harrie, 2003), est la construction d'une matrice de pondération des équations (les équations dont nous voulons qu'elles soient complètement satisfaites doivent avoir un poids élevé).

Enfin, nous avons développé un processus simple à base de séquences prédéfinies qui permet de généraliser une couche de végétation. Il contient quatre opérations dont un utilisateur peut modifier la séquence : une élimination des petites zones, une dilatation/agrégation, une simplification du contour et une simplification des trous dans les géométries des zones de végétation. Le paramétrage de ce processus consiste à choisir les opérations que l'utilisateur souhaite utiliser, dans quel ordre et avec quels paramètres numériques.

Modèle de fonction de traduction

La fonction de traduction doit être fournie par le fournisseur de processus de généralisation en même temps qu'il rend le processus qu'il a développé disponible dans CollaGen. Nous proposons donc de définir la signature de telles fonctions pour que le composant de traduction puisse les déclencher de manière générique, quel que soit le processus à paramétrer. Puis, nous détaillons un modèle pour remplir cette fonction, que nous illustrons par les exemples utilisés précédemment.

La signature que nous définissons pour les fonctions de traduction est la suivante :

```
fonctionDeTraduction(echelleFinale : double,  
                    contraintes : Collection<ContrainteFormelle>,  
                    regles : Collection<RegleOperationnelle>,  
                    annotation : AnnotationSémantique,  
                    processus : ProcessusDeGeneralisation)
```

- L'échelle finale est tout d'abord un paramètre d'entrée indispensable pour beaucoup de processus comme AGENT car les transformations que l'on veut paramétrer dépendent de l'échelle de la carte.
- Une fonction de traduction utilise la formalisation des spécifications pour récupérer les valeurs des paramètres contenues dans les types d'expression du modèle formel. Cela correspond dans la signature aux deux collections de contraintes formelles et des règles opérationnelles.
- Une fonction de traduction nécessite de faire appel à l'annotation sémantique du schéma de données géographique initial car les spécifications formelles sont exprimées en termes de concepts ontologiques. Cette annotation sémantique est également gérée par le composant de traduction mais n'a pas besoin d'être réalisée au moment de la rédaction d'une fonction de traduction.

- Une fonction de traduction prend en entrée le processus de généralisation qu'elle va paramétrer (voir la modélisation des processus de généralisation en C.9.2).

Pour écrire le contenu de ces fonctions de traduction, le fournisseur de processus peut s'appuyer sur le modèle présenté ci-dessous, même si toutes les étapes décrites ne sont pas toujours nécessaires :

```

fonctionDeTraduction(...)
  récupération des noms de classe utiles dans l'annotation
  Initialisation des paramètres de processus relatifs au schéma
  Pour spec de processus.paramètres Faire
    Initialisation concept ← concept lié à spec
    Initialisation caractère ← caractère lié à spec
    Initialisation typeExpression ← type d'expression lié à spec
    Initialisation contrainte ← contraintes.select(concept,
caractère, typeExpression)
    Si (contrainte existe)
      Initialisation de spec à partir de contrainte
    Sinon
      Initialisation règle ← règles.select(concept, caractère)
      Initialisation de spec à partir de règle
    Fin si
  Fin pour
  Initialisation des paramètres restant avec des valeurs par défaut

```

La première étape revient à récupérer par l'annotation sémantique les noms des classes d'objets relatives aux concepts utilisés par le processus à paramétrer. Par exemple, pour AGENT Urbain (processus AGENT spécialisé pour les paysages urbains), la fonction récupère les classes relatives aux concepts de bâtiment, route et îlot urbain ; pour le processus de généralisation de la végétation, elle récupère la ou les classes relatives au concept de végétation.

Puis, la fonction initialise l'ensemble des paramètres relatifs à ce schéma de données : par exemple dans AGENT Urbain, elle relie les classes de bâtiment obtenues précédemment aux « agents bâtiments » connus du processus AGENT Urbain.

Puis, pour chacun des paramètres pouvant correspondre à une des spécifications formelles, la fonction cherche s'il existe une contrainte ou une règle correspondant (à l'aide du concept et du caractère concerné) et elle initialise le paramètre à partir de la contrainte (ou règle) récupérée. Par exemple, pour le seuil de séparation entre symboles de routes des Beams, la fonction cherche dans sa collection de contraintes formelles s'il y a une contrainte de proximité entre routes qui porte sur le caractère de distance minimum entre les symboles. Si elle trouve une contrainte qui correspond, elle récupère la valeur de distance minimum et initialise avec le paramètre « seuil de séparation » des Beams. Pour AGENT, si la fonction trouve une contrainte sur les bâtiments et sur leur aire avec une expression de type seuil, elle ajoute dans l'entité 'AgentMapSpec' une contrainte de type « Building Size » sur les agents bâtiments et initialise le paramètre « Building Min Size » avec la valeur contenue dans la contrainte formelle. Dans le cas des moindres carrés, la fonction peut même initialiser la matrice de poids à partir de la valeur d'importance contenue dans les contraintes et les règles formelles. Il convient de noter que la recherche de la contrainte correspondant à un paramètre est effectuée en exploitant l'arborescence de réification de l'ontologie : dans l'ontologie, un bâtiment

« est un » petit compact donc la fonction commence par chercher les contraintes correspondantes sur les bâtiments et si elle n'en trouve pas, cherche s'il y en a sur les concepts de plus haut niveau comme petit compact puis au-dessus si nécessaire.

Enfin, dans un dernier temps, la fonction initialise avec des valeurs par défauts tous les paramètres qui n'ont pu être remplis à l'aide des spécifications comme le nombre d'itérations maximum pour les Beams. Ce dernier point permet de mettre en lumière une des difficultés possibles de la traduction, difficultés décrites dans la partie suivante.

Difficultés de traduction possibles

Même en suivant le modèle proposé, le fournisseur peut rencontrer des difficultés pour écrire le contenu de la fonction de traduction. Nous avons listé trois types de difficultés dans la traduction : des contraintes sans correspondance dans les paramètres, des paramètres sans équivalent dans les contraintes et des différences de conceptualisation des paramètres.

En réalisant des fonctions de traduction qui suivent le modèle proposé, toutes les contraintes ne seront pas prises en compte par le processus. D'une part, cela ne signifie pas que le processus ne tiendra pas compte du problème formalisé par cette contrainte, car il peut le faire sans avoir recours à un paramètre. Par exemple, l'algorithme de simplification du bâti intégré à AGENT (Ruas, 1999), prend en compte une contrainte de maintien de la forme en L de certains bâtiments sans que ce soit paramétrable. Pour prendre un autre exemple, les Beams tiennent compte de la contrainte qu'une route doit garder le plus possible sa position initiale sans que cela soit paramétrable. D'autre part, si nous avons recours à un système de généralisation collaborative, c'est bien parce que les processus disponibles ne sont pas capables de prendre en compte l'ensemble des spécifications d'une carte généralisée complète. La fonction de traduction se contente donc de traduire des spécifications pour les seuls paramètres attendus d'un processus.

Une deuxième difficulté réside dans le cas inverse, quand certains des paramètres ne correspondent à aucune contrainte ni règle. Cela peut être dû à des paramètres purement procéduraux et indépendants des spécifications de la carte comme le nombre d'itérations maximum des Beams. Cela peut être dû aussi à l'absence de spécifications concernant un concept et un caractère donné. La plupart des processus ne fonctionnent pas si l'on ne remplit pas l'ensemble des paramètres donc dans un cas comme dans l'autre, il faut trouver une solution pour remplir ces paramètres. Pour le cas des spécifications manquantes, la solution que nous proposons est de fournir dans la fonction de traduction des paramètres par défaut en fonction de l'échelle finale : si aucune contrainte ni règle n'a été trouvée pour remplir un paramètre dans la collection des contraintes formelles passée en entrée, la fonction de traduction est conclue par l'initialisation du paramètre avec la valeur par défaut (cette valeur par défaut est trouvée par expérimentations du fournisseur sur le processus qu'il rend disponible). Une autre possibilité est de chercher une contrainte proche (la même sur un autre concept par exemple) et d'appliquer la même valeur de paramètre. Pour le cas des paramètres procéduraux, la seule solution est de les initialiser avec des valeurs par défaut. Si le choix de ces paramètres peut conduire à des fortes variations de résultats, le fournisseur de processus peut fournir plusieurs fonctions de traduction qui pourront être essayées dans le système d'essai/erreur de CollaGen comme deux processus concurrents pour être les plus adaptés : CollaGen essaie le

processus paramétré par la première fonction de traduction, et si le résultat est évalué comme mauvais, revient en arrière et essaie avec l'autre fonction de traduction.

Enfin, la dernière difficulté peut être engendrée par des différences de conceptualisation du problème entre la contrainte formelle et le paramètre du processus. Par exemple, le processus de généralisation de la végétation possède un seuil de simplification qui correspond à l'algorithme de Whirlpool (Dougenik, 1980), c'est-à-dire un seuil de proximité pour former des amas de points consécutifs du polygone qui vont être remplacés par leur centre de gravité (Figure C-58). La contrainte pouvant servir à déterminer ce paramètre est la contrainte de granularité des zones arborées qui porte sur le caractère « longueur minimum d'un côté du polygone » (Figure C-58). Il n'y a pas de relation directe entre les deux seuils mais ils sont corrélés : si on se base sur l'un, on peut résoudre l'autre à peu près. Nous proposons dans ces cas au fournisseur de faire une adaptation de la valeur contenue dans la contrainte pour son paramètre. Dans notre exemple, si nous prenons un seuil de Whirlpool valant deux fois le seuil de la contrainte de granularité, nous maximisons nos chances que notre contrainte soit satisfaite après la généralisation.

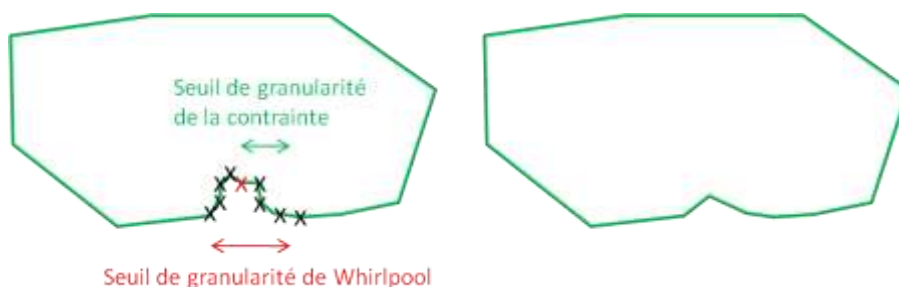


Figure C-58. Différence de conceptualisation de la granularité entre une contrainte (longueur minimum pour un côté du polygone) et un paramètre du processus de généralisation de la végétation (seuil de proximité entre points pour former des amas). A droite la version généralisée avec la conceptualisation de l'algorithme.

C.12.5. La Gestion des unités

Les valeurs numériques présentes dans les contraintes formelles, valeurs utilisées entre autres par les fonctions de traduction et les fonctions d'évaluation des moniteurs de contrainte, peuvent être exprimées avec différentes unités, notamment en unité carte ou en unité terrain. Lorsqu'elles sont utilisées pour initialiser un paramètre de processus, ces valeurs doivent être exprimées dans la bonne unité. Par exemple, si la contrainte de proximité entre deux bâtiments est exprimée en mm carte, il faut la convertir en mètres terrain pour l'utiliser dans les Beams, pour que le processus s'y retrouve. Le composant de traduction est donc doté de méthodes de traduction des unités qui permettent, à partir de la contrainte formelle, de récupérer la valeur numérique dans l'unité souhaitée.

En résumé, le composant de traduction est composé principalement de fonctions de traduction rédigées par les fournisseurs de processus dans une forme générique. Le composant de traduction permet également de relier les concepts géographiques de l'ontologie avec les éléments du schéma de données correspondant, ce qui fait office d'annotation sémantique pour CollaGen. Le composant de traduction permet enfin de rendre tous les autres composants complètement interopérables en faisant l'intermédiaire entre les éléments génériques du modèle CollaGen

(contraintes formelles, ontologie, descriptions formelles), les outils réellement implémentés et utilisés par CollaGen, et les données réellement dans la base de données.

C.13. COMPOSANT *REGISTRE*

Un système de généralisation collaborative doit choisir le meilleur processus pour généraliser chacun des espaces géographiques qu'il traite à un moment donné du processus, ce qui peut être ramené par analogie au problème dit d'appariement (matchmaking) bien connu des communautés de recherche en Systèmes Multi-Agents (SMA) et en composition de services Web. Nous pouvons définir ce problème d'appariement comme la médiation entre des demandeurs et des fournisseurs de services pour une coopération avec bénéfice mutuel (Sycara et al, 2002). Dans le cas de CollaGen, les demandeurs peuvent être assimilés aux espaces géographiques et les fournisseurs aux processus de généralisation. Quand l'appariement est fait par un courtier (i.e. une entité extérieure aux demandeurs et aux fournisseurs), ce courtier peut être assimilé à un service d'appariement (Lutz et al, 2003) rendu par un outil de type registre qui enregistre les capacités des services fournis, les analyse ainsi que les contenus des demandes et donne la réponse la plus adaptée aux demandeurs. CollaGen est donc doté d'un composant registre (nom tiré de la norme UDDI, voir partie suivante) pour jouer le rôle de courtier entre les espaces géographiques et les processus de généralisation.

Dans un premier temps, nous allons brièvement revenir sur les fonctionnalités que peut posséder un registre en faisant référence à la littérature en SMA et en composition de services Web. Puis, nous détaillons le modèle de registre que nous proposons dans CollaGen. Nous décrivons ensuite comment notre registre répond à une demande d'un espace géographique quelconque. Puis, nous présentons pourquoi et comment nous avons doté le registre de capacités d'apprentissage. Enfin, nous décrivons les autres capacités dont nous avons doté le composant registre de CollaGen.

C.13.1. Fonctionnalités d'un registre

UDDI (pour Universal Description Discovery and Integration) est un standard fondé sur XML pour réaliser des annuaires de services Web (UDDI, 2004). Un annuaire UDDI permet de publier des services, de chercher des services à partir de descriptions techniques et de localiser ces services pour les utiliser. Un annuaire UDDI a trois composants :

- Des *pages blanches* qui regroupent les coordonnées et la description des fournisseurs de services
- Des *pages jaunes* qui recensent l'ensemble des services disponibles
- Des *pages vertes* qui contiennent les descriptions techniques des capacités des services disponibles.

Ces trois composants sont indispensables pour permettre à l'annuaire UDDI de remplir toutes les fonctions d'appariement entre demandeurs et services. Dans la norme UDDI, ces trois composants

ont chacun leur standard : par exemple, les pages jaunes contiennent les descriptions WSDL (Web Service Description Language) des services. S'il doit suivre la norme UDDI, notre composant registre doit donc être capable de stocker l'équivalent des pages blanches, jaunes et vertes pour les processus de généralisation.

Dans le domaine des SMA, la question des registres pour réaliser un appariement entre un agent demandeur et un agent fournisseur de service a depuis longtemps été abordée. Ainsi, la norme FIPA qui standardise notamment les communications entre les agents comporte une spécification consacrée aux registres dits 'Directory Facilitators' qui sont vus comme des agents courtiers capables de stocker des capacités d'autres agents et de mettre en relation des demandeurs et des fournisseurs (FIPA, 2004). Ces agents registres possèdent une liste des agents fournisseurs de services qu'un autre agent peut interroger de manière standardisée. Dans ce cas, les fonctions de base d'un registre sont l'inscription au registre d'un agent et la recherche d'un agent fournisseur de service. (Wong et Sycara, 2000) proposent, toujours dans le domaine des SMA, une classification des agents chargés de réaliser des appariements entre demande et services (appelés agents intermédiaires), selon plusieurs critères que l'on peut rattacher à notre registre : la nature des informations envoyées à l'agent intermédiaire, le contenu d'une requête ou la méthode utilisée pour y répondre par exemple. Ces caractéristiques permettent de bien décrire notre proposition dans CollaGen.

(Schade et al, 2004) décrivent, dans le contexte de la composition de services Web, les fonctionnalités que peuvent avoir des services fournissant l'appariement entre une demande et un service, comme peuvent l'être les registres. Un registre peut ainsi proposer des appariements *exacts*, *relâchés* ou de type *plug-in*. Un appariement est *exact* quand les descriptions de la demande et des capacités sont équivalentes. L'appariement est *relâché* quand les descriptions sont plus ou moins similaires. Un appariement *plug-in* est entre les deux car il signifie que l'une des spécifications est plus spécifique ou équivalente à l'autre. Ce type d'appariement est intéressant pour nous car un processus qui peut traiter plus d'espaces que l'espace demandeur et plus de contraintes que les contraintes actuellement localisées dans l'espace (la description de la demande est donc plus spécifique que celle des capacités du processus) est un processus adapté à la généralisation de l'espace pour nous. (Sycara et al, 2002) propose des méthodes pour réaliser ces appariements exact, relâché et plug-in à partir de différents filtres qui éliminent et ordonnent petit à petit les services pouvant potentiellement répondre à la requête.

Le site internet des Pages Jaunes est sûrement le registre le plus utilisé en France et il a une particularité qui nous intéresse particulièrement : il mélange les fonctions classiques de registre comme celle de la norme UDDI et les fonctions d'un moteur de recherche internet. En effet, le registre peut être interrogé avec une requête contenant une information incomplète et il répond par une liste ordonnée par pertinence de services disponibles.

Nous retenons plusieurs points de cet état de l'art rapide sur les registres et les agents intermédiaires. Tout d'abord un registre dans CollaGen suivant la norme UDDI doit posséder un composant pages blanches pour stocker les « coordonnées » des processus, un composant pages jaunes listant les processus et un composant pages vertes stockant les descriptions formelles des processus. Le contenu d'une requête et la manière d'y répondre doivent être modélisés et nous allons chercher à proposer des appariements de type plug-in ou relâché pour faire correspondre un

espace géographique et un processus adapté. Enfin, comme le cas des pages jaunes internet, nous cherchons plutôt une liste de processus ordonnés par pertinence comme réponse du registre.

C.13.2. Modélisation du composant registre

Le composant registre de CollaGen a pour mission de réaliser l'appariement entre un espace géographique qui a besoin d'être généralisé et le processus de généralisation le plus adapté parmi les processus disponibles dont les capacités sont décrites formellement. En s'appuyant sur l'état de l'art précédent, nous proposons une modélisation du registre de généralisation comme un ensemble de services disponibles (partie pages jaunes) permettant l'accès aux processus correspondants (partie pages blanches) et produisant ses réponses à partir des descriptions formelles des processus (partie pages vertes) (Figure C-59).

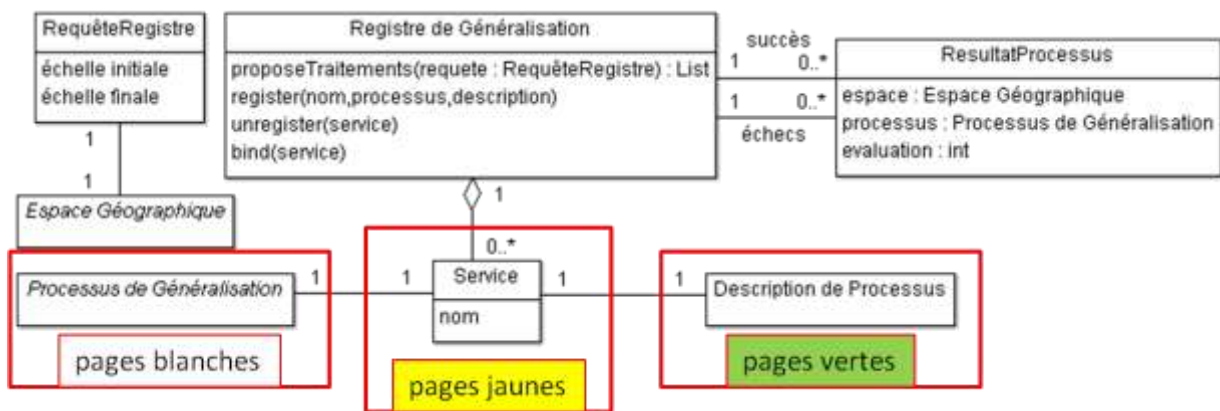


Figure C-59. Diagramme de classe UML du modèle de registre proposé dans CollaGen.

Nous proposons qu'un registre de généralisation implémente les méthodes des deux normes UDDI et DF-FIPA pour lui donner un fonctionnement générique :

- *proposeTraitements* (request dans UDDI et FIPA), qui permet de récupérer la liste des processus adaptés à une requête, ordonnés par pertinence. Le fonctionnement de cette méthode est décrit en détail dans la partie suivante.
- *register* (subscribe dans UDDI et register dans FIPA), qui permet de créer un nouveau service dans le registre à partir d'un nom, d'un processus et de sa description.
- *unregister* (seulement dans la norme FIPA), qui permet d'enlever un service du registre.
- *bind* (seulement dans UDDI) qui permet d'accéder au chemin vers le service, pour son exécution.

Nous modélisons une requête au registre comme une instance d'espace géographique et un saut d'échelle, avec l'échelle initiale des données et l'échelle finale de la carte. A partir de l'instance d'espace géographique, le registre peut avoir accès à ses caractéristiques générales comme sa densité ou son thème principal (voir C.7) mais aussi aux conflits cartographiques présents dans l'espace géographique grâce aux moniteurs de contraintes contenus dans l'espace.

Enfin, un registre de généralisation stocke les résultats des différentes généralisations qu'il a facilitées. Un résultat de processus est triplet contenant un espace géographique, un processus de généralisation qui a été utilisé sur cet espace et l'évaluation rendue par le composant d'évaluation de

la généralisation de cet espace par le processus. Seuls les résultats comprenant une bonne ou une mauvaise évaluation sont stockés sous la forme de succès ou d'échec : les résultats moyens ne sont pas stockés. Les succès permettent de favoriser le processus sur des espaces similaires alors que les échecs ont l'effet inverse. Ce système d'apprentissage par renforcement est décrit plus loin dans cette partie C.13.

C.13.3. Comment le registre répond-il à une requête de généralisation ?

Une requête au registre de généralisation de CollaGen est un triplet (espace géographique, saut d'échelle, étape courante (sélection, etc.)). La réponse attendue est une liste de processus dont la description est appariée par un appariement de type *plug-in* à l'espace géographique et ordonnée par pertinence décroissante. Comme dans (Sycara et al, 2002), la méthode proposée pour déterminer la réponse est une séquence de filtrage des services du registre. La séquence des filtres successifs appliqués est présentée en Figure C-60 : le registre filtre d'abord par les pré-conditions des descriptions formelles des processus, en tenant compte du type d'espace puis de la compatibilité de la gamme d'échelle, il filtre par les post-conditions des descriptions formelles, et enfin il filtre en fonction des succès et échecs rencontrés précédemment dans le processus collaboratif. Seul le filtrage par les pré-conditions élimine des processus de la liste, les deux filtres suivants se contentent de réordonner la liste par pertinence.

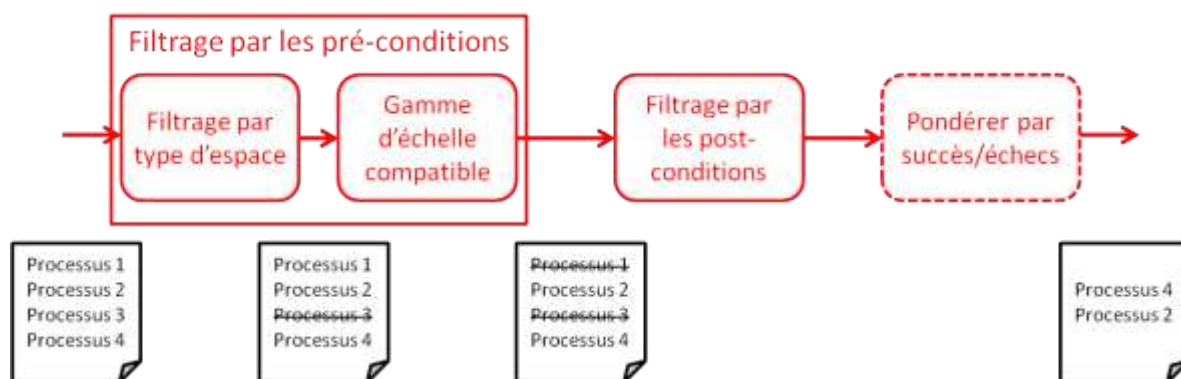


Figure C-60. La séquence des filtres effectués par le registre pour répondre à une requête de généralisation. L'étape de pondération est en pointillés car elle n'est pas systématique (il faut que des succès ou échecs soient déjà stockés).

Filtrage par type d'espace

Le filtrage par les types d'espace décrits dans les pré-conditions réalise dans un premier temps des appariements *plug-in* : si les pré-conditions sont équivalentes ou plus générales que l'espace géographique sujet de la requête, le processus est conservé. Pour analyser les espaces, le composant se réfère aux relations taxonomiques décrites dans l'ontologie : dans l'ontologie l'espace thématique *réseau routier* « est un » *espace thématique de type réseau* donc, si le sujet de la requête est l'espace *réseau routier* (les espaces thématiques n'ont qu'une instance), tous les processus qui possèdent une pré-condition *réseau routier* ou une pré-condition *espace thématique de type réseau* sont conservés par le filtrage de type d'espace.

Le registre réalise dans un deuxième temps un appariement relâché par le type d'espace à partir des descriptions générales des espaces. Nous avons vu qu'une pré-condition pouvait aussi être, plutôt qu'un type d'espace, un espace quelconque dont les caractéristiques générales en termes de densité d'objets, hiérarchisation des objets, homogénéité thématique, thème dominant et thème secondaire sont précisées. Par exemple, « ce processus est adapté aux espaces territoriaux peu denses dont le thème dominant est le bâti et le thème secondaire le routier ». Dans ce cas d'un espace quelconque, le registre mesure ces caractéristiques dans l'espace sujet de la requête et réalise un appariement relâché si au moins la moitié des caractéristiques correspond. Un appariement relâché à l'issue de ce filtrage est ensuite toujours mis en queue de liste par les autres filtrages, car assez imprécis. Seuls les succès ou échecs précédents peuvent faire passer un appariement relâché devant un appariement plug-in car nous considérons ces appariements relâchés comme moins fiables que les appariements plug-in.

Gamme d'échelle compatible

Le deuxième filtrage effectué par le registre vérifie la compatibilité des gammes d'échelles (la gamme d'échelle globale du processus et la gamme d'échelle de la pré-condition appariée dans l'étape précédente). Nous avons vu que la description formelle d'un processus de généralisation comprend une gamme d'échelle avec un saut d'échelle optimal, des échelles initiales minimum et maximum et des échelles finales minimum et maximum. Cette gamme d'échelle peut être spécialisée (en d'autres termes, réduite) pour chacune des pré-conditions. Ce filtrage prend en compte la gamme d'échelle réduite (correspondant à la pré-condition utilisée) de chacun des processus déjà filtrés. Si l'échelle initiale ou l'échelle finale, décrites dans la requête de généralisation faite au registre, est hors des limites de la gamme d'échelle du processus, celui-ci est éliminé. De plus, si ces échelles sont assez éloignées du saut d'échelle optimal de la gamme, l'appariement est considéré comme relâché, et le processus est placé en queue de liste avec les autres appariements relâchés.

Filtrage par les post-conditions

Les deux premiers filtrages doivent avoir éliminé les processus de généralisation disponibles non pertinents et cette étape de filtrage par les post-conditions permet de réordonner les processus restant en fonction de leur pertinence calculée principalement en ayant recours aux post-conditions des descriptions formelles. Trois critères sont utilisés pour calculer la pertinence d'un processus : la confiance dans la pré-condition, le saut d'échelle et surtout l'adéquation des post-conditions avec les conflits effectivement identifiés par des moniteurs dans l'espace à généraliser. La Figure C-61 illustre le fonctionnement du filtrage (l'échelle n'est pas prise en compte pour simplifier). Avec une confiance dans la pré-condition identique de 4, la pertinence est décidée en faveur du processus 2 car ses post-conditions sont mieux en adéquation avec les conflits présents dans l'espace rural.

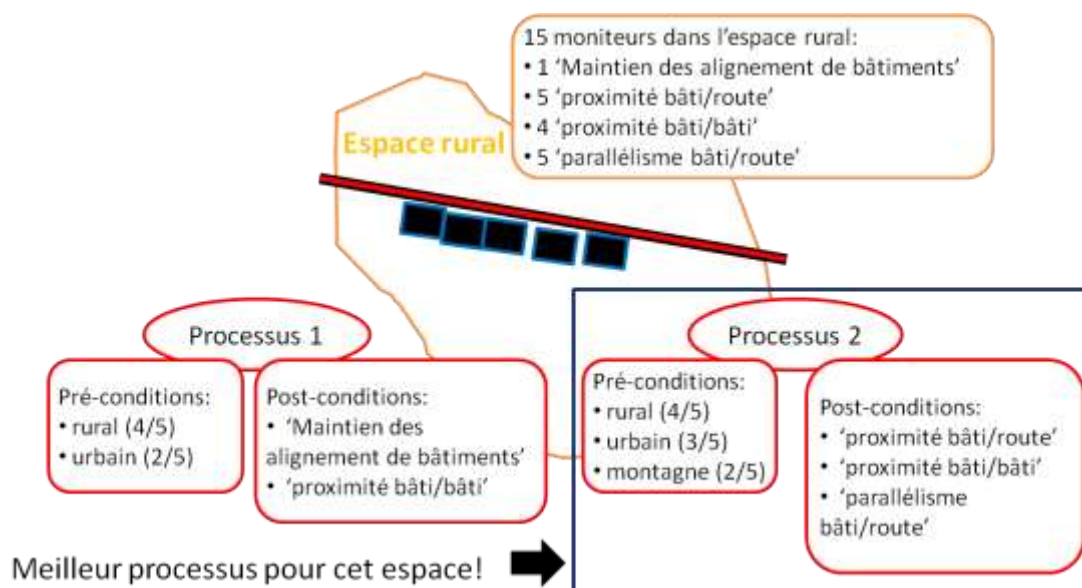


Figure C-61. Illustration du filtrage par les post-conditions par mise en adéquations avec les moniteurs contenus dans l'espace.

La pertinence d'un processus par rapport à un espace géographique est une valeur numérique comprise entre 0 et 5 (5 étant la pertinence maximum). Les processus sont classés du plus pertinent (celui dont la pertinence est la plus grande) au moins pertinent. Cette valeur de pertinence est calculée de la manière suivante :

Les coûts dus aux contraintes et à l'échelle sont des réels compris entre 0 et 1 qui pondèrent la pertinence *a priori* exprimée dans la pré-condition. Le coût dû aux contraintes est le plus complexe à calculer car il nécessite de trouver un indicateur pour quantifier les conflits à partir des moniteurs de contraintes localisés à l'intérieur de l'espace géographique demandeur et de faire le rapport avec les moniteurs dont la contrainte correspondante est prise en compte par le processus d'après les post-conditions. Le tout est pondéré par l'importance des contraintes et l'indice de confiance des post-conditions :

Où

Notre fonction f est donc un moyen de remédier à la difficulté de quantifier les conflits à partir des contraintes. Il faut noter que cette fonction donne plus d'importance aux contraintes relationnelles car elles impliquent deux objets et aux contraintes meso dont l'importance est multipliée par le nombre d'objets composant le meso. Si nous reprenons l'exemple de la Figure C-61, le coût dû aux contraintes est différent car le moniteur de la contrainte de maintien d'alignement de bâtiments concerne cinq objets ce qui donne des coûts de contrainte de 13/33 pour le processus 1 et 28/33 pour le processus 2 qui est bien toujours le processus le plus pertinent (on considère ici que les indices de confiance des post-conditions sont tous de 5).

Le coût d'échelle essaie d'évaluer la distance du saut d'échelle de la généralisation demandée avec le saut d'échelle optimal de la description formelle du processus dont le composant évalue la pertinence. Le registre évalue donc l'écart avec l'échelle initiale et l'écart avec l'échelle finale. L'échelle finale et l'échelle initiale contribuent chacune pour moitié au coût d'échelle. Par exemple, si le saut d'échelle optimal d'un processus est de 1 : 15000 à 1 : 50000, son coût d'échelle pour un saut d'échelle assez proche de 1 : 10000 à 1 : 55000 est de 0,92. Ainsi, tant que le saut d'échelle est proche du saut optimal, le coût d'échelle contribue bien moins que le coût de contrainte à la pertinence du processus.

Faire contribuer les contraintes à la pertinence de manière quantitative plutôt que qualitative est relativement dangereux car cela peut amener à de mauvaises interprétations de la pertinence d'un processus. Ce n'est pas parce qu'il y a une grande quantité de contraintes non satisfaites qu'il y a une grande quantité de conflits. Ainsi, l'expérience du projet EuroSDR d'état de l'art des logiciels de généralisation du commerce (Stöter et al, 2010) montre que l'évaluation quantitative des contraintes peut être trompeuse. En effet, les cartes généralisées obtenues avec les différents logiciels participant au test ont été évaluées de différentes manières et notamment par des experts qui ont noté les cartes pour chacune des contraintes que les cartes devaient respecter (Stöter et al, 2009). Ces travaux montrent que cumuler ces chiffres d'évaluation est trompeur car certaines contraintes peu nombreuses dans les données (comme les contraintes meso) ont beaucoup plus d'importance sur la qualité et la lisibilité de la carte que d'autres contraintes bien plus fréquentes dans les données. Pour se prémunir contre ces problèmes, nous avons introduit dans nos formules de pertinence une forte pondération par l'importance ainsi qu'une pondération par le nombre d'objets concernés par chaque contrainte. Nous considérons que ces précautions nous protègent de mauvaises évaluations de la pertinence mais des tests sur un plus grand nombre de cas sont nécessaires pour bien valider cette proposition (voir E.4).

Exemple de requête

Reprenons maintenant de manière plus complète l'exemple précédent de généralisation d'un espace rural pour illustrer le déroulement complet de la méthode de réponse à une requête au registre de généralisation. La Figure C-62 présente la situation initiale avec une requête pour généraliser cet

espace rural du 1 : 15 000 au 1 : 50 000 avec quatre processus enregistrés. Le premier filtre sur le type d'espace permet d'éliminer le processus 3 qui n'a pas de pré-condition avec *espace rural* ou un concept parent de l'espace rural. Le deuxième filtre permet d'éliminer le processus 4, car son échelle finale maximum (1 : 25000) est supérieure à l'échelle finale de la requête (1 : 50000). Le troisième filtre permet d'ordonner les processus 1 et 2 par pertinence. Le processus 1 a un indice de confiance de 4 pour les espaces ruraux, un coût d'échelle de 0,88 et un coût de contrainte de 0,57 (16+30+18+25/155) ce qui donne une pertinence de 2,02. Le processus 2 a un indice de confiance de 4 également, un coût d'échelle de 1 et un coût de contrainte de 0,46, ce qui donne une pertinence de 1,83. Le processus 1 est donc le premier dans la liste de processus servant de réponse à la requête.

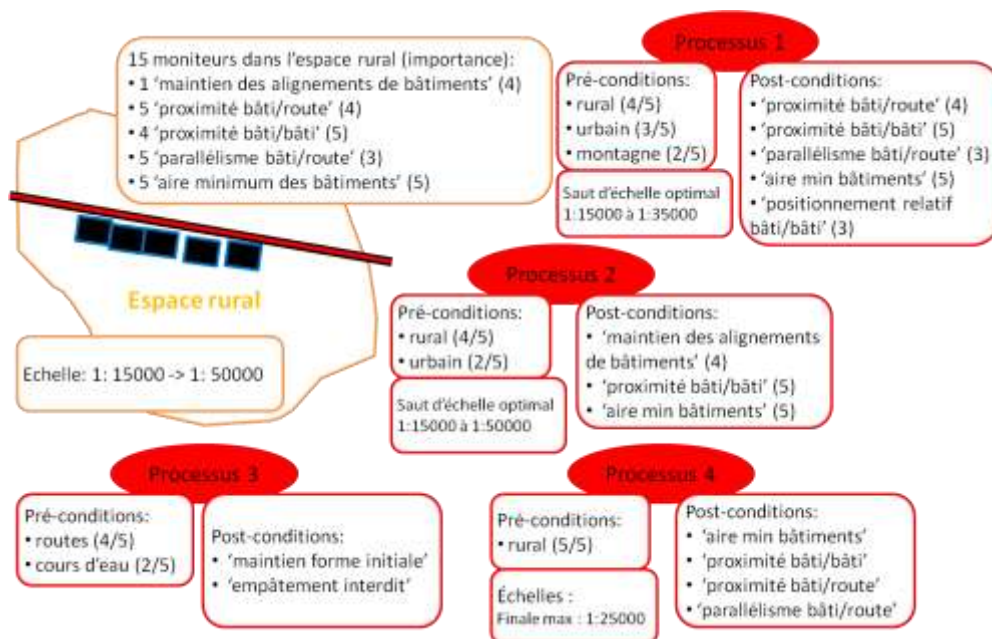


Figure C-62. Exemple de requête au registre de généralisation : parmi les 4 processus disponibles, quels sont les plus adaptés pour généraliser cet espace rural ?

C.13.4. Un Registre qui apprend de ses erreurs

La Figure C-60 qui présente les filtres appliqués pour répondre à une requête de demande de service propose aussi en pointillés un filtrage par les succès et les échecs précédents dans le processus collaboratif. Le registre ne prend ses décisions qu'en comparant l'espace à généraliser à des descriptions formelles de capacités qui peuvent s'avérer inexactes ou imprécises. Pour gommer légèrement l'impact d'imprécisions dans ces descriptions formelles, nous proposons de doter le registre de capacités d'apprentissage. Ce type d'apprentissage peut s'assimiler à une des trois familles d'apprentissage artificiel, l'apprentissage par renforcement (les autres étant l'apprentissage supervisé et l'apprentissage non supervisé).

L'Apprentissage par renforcement

L'apprentissage par renforcement est une méthode permettant d'apprendre par essai-erreur à choisir la meilleure action selon la situation dans laquelle on se trouve (Sutton et Barto, 1998).

L'apprentissage par renforcement se fait en général en attribuant une récompense en cas d'action évaluée comme bonne pour apprendre petit à petit comment prendre la bonne décision dans chaque situation. Le registre de généralisation est un cas d'apprentissage par renforcement actif car le registre décide lui-même comment répondre à une requête (Russell et Norvig, 2006, p.858). Il existe de nombreuses méthodes d'apprentissage par renforcement en fonction de l'information antérieure stockée et de la modélisation des capacités de décision. Nous pouvons citer notamment (Hoët et Sabouret, 2010) qui proposent d'analyser tout l'historique des actions précédentes sous forme d'arbre, pour apprendre la décision à prendre à un instant donné. Mais des modélisations plus simplistes conviennent aussi bien dans le cas de notre registre.

Capacités d'apprentissage du registre

On a vu dans le modèle du registre présenté en Figure C-59 qu'il était capable de stocker les succès et échecs passés. De plus, nous avons vu que la réponse à une requête pouvait être pondérée par un filtrage en fonction des résultats passés stockés. Nous sommes dans le cas simple où le composant d'évaluation va fournir l'évaluation des résultats au registre après chaque généralisation. En plus des récompenses qui vont favoriser un processus s'il s'est déjà avéré efficace sur un espace, nous proposons de pénaliser les processus ayant entraîné une mauvaise généralisation. Une généralisation moyenne n'a pas d'impact sur la décision du registre.

La première étape de l'apprentissage repose sur la récupération des résultats stockés qui correspondent à la situation traitée actuellement. Il s'agit donc de retrouver un espace géographique qui ressemble à celui qui est le sujet de la requête au registre. Pour déterminer la ressemblance, trois critères vont être utilisés : la ressemblance sémantique, la ressemblance de caractéristiques (celles d'un espace géographique, comme la densité d'objets le thème dominant par exemple) et la ressemblance de conflits. Et quatre niveaux de ressemblances sont définis :

- Dissemblance : quand il n'y a similarité selon aucun des trois critères.
- Point commun : un seul des critères présente une similarité.
- Ressemblance : deux des critères sont similaires.
- Similarité : les trois critères sont similaires.

La ressemblance sémantique est calculée en fonction des concepts ontologiques : si un des deux concepts d'espace (celui de l'espace faisant une requête et celui ayant déjà été généralisé) partage une relation « est un » avec l'autre, ils se ressemblent sémantiquement (par exemple, « réseau routier » ressemble sémantiquement à « réseau de transport »). La ressemblance de caractéristiques est calculée à l'aide des caractéristiques générales des espaces géographiques : si un certain nombre des caractères sont identiques, les deux espaces se ressemblent en termes de caractéristiques. Enfin, la ressemblance de conflits est calculée en comparant le nombre et la nature des moniteurs de contrainte (même problème d'adéquation que précédemment car une quantité de contraintes n'est pas forcément une quantité de conflits) : si les deux espaces contiennent à peu près les mêmes moniteurs dans les mêmes proportions, ils se ressemblent. Suivant le niveau de ressemblance, l'intensité des récompenses et des pénalités va varier.

La deuxième étape de l'apprentissage repose sur l'attribution de récompenses et de pénalités en fonction des résultats passés sur des espaces ressemblants. Puisque nous utilisons une valeur de

pertinence pour classer les processus, nous proposons d'exprimer les récompenses et les pénalités en termes de pondération de la valeur de pertinence : le composant ajoute (enlève) 0,2 pour chaque succès (échec) similaire recensé. Quand le niveau de ressemblance baisse (e.g. on passe de *similarité* à *ressemblance*), le composant divise par deux la récompense (pénalité) sachant qu'un succès (échec) dissemblable ne modifie pas la pertinence. Ce chiffre de 0,2 a été déterminé empiriquement à partir de valeurs constatées de pertinence en testant le registre. Il faudrait toutefois s'assurer de la sensibilité de cette valeur au nombre d'espaces géographiques présents dans la zone à généraliser.

C.13.5. Autres capacités du registre

Le registre de généralisation de CollaGen est capable de répondre à d'autres types de requêtes durant la généralisation collaborative. En effet, à l'émergence d'un espace géographique lors de l'observation d'une généralisation (voir C.14), il est demandé au registre de trouver un processus adapté à la généralisation de cet espace émergent qui possède des caractéristiques différentes d'un espace classique (plus petit, plus homogène en conflits). Le registre fait dans ce cas une réponse simplifiée dans laquelle seules l'échelle et les contraintes sont prises en compte pour évaluer la pertinence, les deux premiers filtrages étant évités.

De plus, le registre de généralisation est mis à contribution pour déterminer le meilleur processus de correction d'effets de bord constatés au voisinage d'un espace après sa généralisation. Cette capacité n'ayant pas été implémentée dans notre prototype, nous n'irons pas plus loin dans la modélisation de son fonctionnement qui reste une des perspectives (proches) de notre travail de thèse.

Pour résumer cette partie décrivant le composant registre de CollaGen, il s'agit d'un annuaire de processus de généralisation inspiré des travaux en SMA et services Web. Il détermine les processus adaptés à la généralisation d'un espace en fonction de leurs pré-conditions et les classe par pertinence en fonction de l'échelle et surtout de l'adéquation entre les contraintes gérées par le processus selon les post-conditions et les moniteurs effectivement présents dans l'espace géographique. Le registre est également doté de capacités d'apprentissage par renforcement qui pondèrent la pertinence par les résultats obtenus précédemment sur des espaces ressemblants.

C.14. COMPOSANT D'OBSERVATION ET EVALUATION

CollaGen est fondé sur l'approche pas à pas locale de généralisation qui sous-tend le principe d'essai-erreur : le système essaie une solution locale, l'évalue, si c'est bon, il traite une autre zone, sinon il essaie autre chose. Une des étapes clés de ce fonctionnement par essai-erreur est l'évaluation de la généralisation. Le composant d'observation et d'évaluation est chargé de mettre en place l'étape d'évaluation dans le cadre de la généralisation collaborative. Nous présentons d'abord un très rapide état de l'art de l'évaluation en généralisation pour identifier ce qui se rapporte à notre problème. Puis nous décrivons le mécanisme support de l'évaluation dans CollaGen, c'est-à-dire l'évaluation des

conflits locaux par les moniteurs. Nous décrivons ensuite le principe d'observation en parallèle d'une généralisation et comment des espaces émergents (i.e. des zones de conflits résiduels) peuvent être identifiés par cette observation. Enfin, nous présentons nos propositions pour évaluer globalement un espace géographique après sa généralisation.

C.14.1. L'Évaluation en généralisation

Selon (Mackaness et Ruas, 2007), on peut distinguer trois grands types d'évaluation à propos de la généralisation cartographique : l'évaluation pour *contrôler*, pour *paramétrer* et pour *valider*. L'évaluation pour *contrôler* permet l'enchaînement et la validation des opérations de généralisation dans le processus automatique. L'évaluation pour *paramétrer* consiste à évaluer les différents paramétrages d'un processus de généralisation pour faciliter le choix à l'utilisateur. Enfin, l'évaluation pour *valider* peut servir à identifier les erreurs pour les corriger après coup, de manière interactive ou automatique, à décrire la qualité de la généralisation en fournissant des métadonnées sur le résultat généralisé et à noter qualitativement le résultat généralisé. Dans CollaGen, nous nous intéressons à l'évaluation pour contrôler dans le cadre du mécanisme par essai-erreur et à l'évaluation pour estimer, à la fois pour corriger des erreurs locales et pour noter qualitativement une généralisation afin de permettre le mécanisme d'apprentissages des succès/échecs du registre. Nous nous intéressons donc aux contributions dans ces deux domaines de l'évaluation.

La plupart des processus de généralisation automatique présentés en A.2 reposent sur une ou plusieurs étapes d'évaluation pour contrôler. Parmi ces processus, nous pouvons citer (Harrie et Sarjakoski, 2002 ; Ware *et al*, 2003) mais surtout (Ruas, 1999; Barrault *et al*, 2001; Ruas et Duchêne, 2007) qui proposent après l'application de chaque opération de généralisation au niveau meso ou micro d'évaluer la satisfaction de toutes les contraintes de l'objet (ou des objets) traité(s) pour valider la progression ou l'annuler et essayer autre chose. Cette évaluation pour contrôler repose sur une fonction d'évaluation globale (qui est par défaut la moyenne pondérée par l'importance des satisfactions des contraintes) et sur une fonction de validation de la progression. (Taillandier, 2008) propose ainsi des méthodes de révision automatique des connaissances pour optimiser la fonction de validation alors que (Taillandier et Gaffuri, 2010) proposent une fonction d'évaluation plus complexe qui peut être modifiée par un utilisateur suite à un dialogue basé sur des échantillons de résultats. (Brazile, 2000) propose dans le cadre d'une approche pas à pas locale qui étend celle de (Weibel et Dutton, 1998) de nombreuses étapes de contrôle, avec notamment plusieurs opérations réalisées en parallèle sur la même situation qui sont ensuite évaluées pour garder la meilleure. (Filippovska *et al*, 2008) proposent tout un jeu d'algorithmes de mesures pour réaliser un tel processus sur les bâtiments.

Parmi les travaux sur l'évaluation pour estimer, certains se concentrent sur l'évaluation de la généralisation à partir de la satisfaction des contraintes. Ainsi, les projets AGENT (Barrault *et al*, 2001) et EuroSDR (Stöter *et al*, 2009) ont fait appel à des experts cartographes pour évaluer des résultats de généralisation automatique notamment en fonction de certaines contraintes. (Bard, 2004) propose une méthode automatique d'évaluation des données généralisées en fonction des données initiales qui, dans le processus de généralisation, doivent servir de référence à l'évaluation (Mackaness et Ruas, 2007). L'auteur propose de déterminer des fonctions de qualité qui évaluent la généralisation d'un caractère d'un objet ou groupe d'objets soumis à une contrainte. Cette fonction

donne par exemple le résultat suivant : "la généralisation de ce bâtiment est bonne en ce qui concerne la taille". Une fonction de référence donne la valeur idéale que le caractère devrait avoir dans une généralisation parfaite et la qualité de la généralisation sur le caractère est évaluée par la distance à cette fonction de référence (Figure C-63). (Burghardt et al, 2008) a repris cette approche directement sur les contraintes (et non les caractères) et propose de déterminer les fonctions de référence par reverse engineering (en utilisant des résultats généralisés à la main). La problématique principale en perspective de ces travaux réside dans l'agrégation de ces évaluations individuelles pour obtenir des généralisations plus globales sur des objets reliés à plusieurs contraintes ou sur un ensemble d'objets comme un espace géographique.

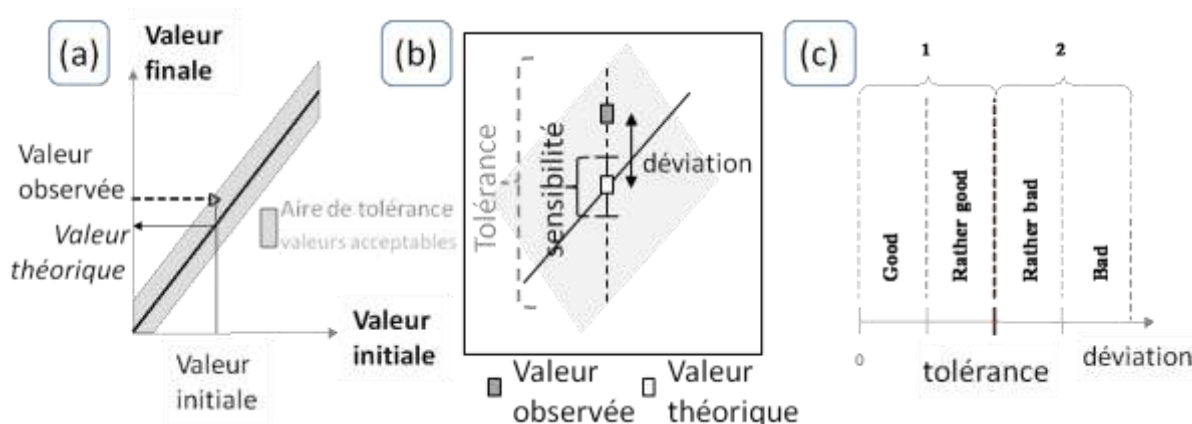


Figure C-63. Détermination de la qualité de la généralisation pour une contrainte en fonction du positionnement de la valeur finale par rapport à la courbe de référence. L'image b) montre l'introduction d'une valeur de sensibilité pour laquelle le système ne fait pas de différence avec un résultat idéal (Bard, 2004).

Un autre point problématique relevé par (Bard, 2004 ; Burghardt et al, 2008) réside dans l'évaluation des objets de type meso car leur généralisation relève souvent de la typification et il est difficile quand des objets ont été éliminés de déterminer si la structure du meso a bien été conservée. Comme début de réponse à ce problème, (Zhang et al, 2008) propose de décomposer les meso en parties élémentaires dans une ontologie et de décrire formellement la structure du meso dans l'ontologie afin d'évaluer individuellement les éléments de cette structure (il est à noter que notre proposition reprend ce principe de modélisation des objets meso dans l'ontologie de généralisation). Enfin, (Zhang et al, 2010b ; Mao et al, 2010) proposent respectivement un algorithme pour mesurer le maintien de structures meso pour des contraintes de répartition spatiales dans un îlot et le maintien des alignements.

Que ce soit pour contrôler ou pour estimer, évaluer la généralisation en se fondant sur les contraintes paraît être une bonne idée. Mais plusieurs difficultés apparaissent : l'évaluation des contraintes meso et l'agrégation de satisfactions de contraintes en une évaluation globale.

C.14.2. Evaluation des conflits par les moniteurs

En tenant compte de l'état de l'art précédent, nous proposons de fonder notre système d'évaluation sur l'évaluation individuelle des contraintes de généralisation. L'évaluation individuelle des contraintes repose sur les moniteurs de contrainte qui identifient les endroits où les contraintes (et les règles opérationnelles) sont effectives. Nous avons vu dans la partie C.10 que le moniteur est

doté d'une méthode pour calculer la satisfaction de la contrainte sur l'objet suivi par le moniteur. Et chaque type de moniteur est associé par le composant de traduction à des algorithmes qui permettent entre autres de calculer la satisfaction courante de la contrainte. Nous proposons que cette méthode de calcul de la satisfaction repose sur les mêmes principes que (Bard, 2004), c'est-à-dire sur une distance à une valeur idéale de la contrainte en fonction de la valeur initiale. La méthode de calcul de la satisfaction doit donc prendre en compte une sensibilité autour de la valeur idéale qui dépende de la méthode pour calculer la valeur observée de la contrainte. L'exemple ci-dessous est l'algorithme permettant de déterminer la satisfaction d'un moniteur de contrainte d'aire minimum d'un bâtiment. La sensibilité y est introduite par la valeur epsilon de 5 m².

```

CalculerSatisfactionAireMinBâtiment(moniteur, bâtiment)
  Si(bâtiment est éliminé)
    Retourner satisfaction ← TRES_SATISFAIT

  Initialisation epsilon = 0.002 * échelleFinale m2 (50m2 au 1 : 50000)
  Initialisation but = moniteur.getValeurBut()
  Initialisation courante = moniteur.getValeurCourante()

  Si (|but - courante| < epsilon) satisfaction ← PARFAIT
  Sinon si (courante > but) satisfaction ← TRES_SATISFAIT
  Sinon si (courante > 0.8 x but) satisfaction ← CORRECT
  Sinon si (courante > 0.75 x but) satisfaction ← MOYEN
  Sinon si (courante > 0.7 x but) satisfaction ← PASSABLE
  Sinon si (courante > 0.6 x but) satisfaction ← PEU_SATISFAIT
  Sinon si (courante > but/2) satisfaction ← TRES_PEU_SATISFAIT
  Sinon satisfaction ← NON_SATISFAIT
  Fin Si
  
```

Tous les aspects de la méthode de (Bard, 2004) ne sont pas couverts dans une telle méthode de calcul de satisfaction. Nous devons également ajouter une méthode calculant le but du moniteur en fonction de sa valeur initiale à laquelle le calcul de la satisfaction va faire appel. Cette méthode de calcul de la valeur but doit encapsuler l'équivalent de la fonction de référence comme le montre la Figure C-64 sur le même exemple de moniteur.

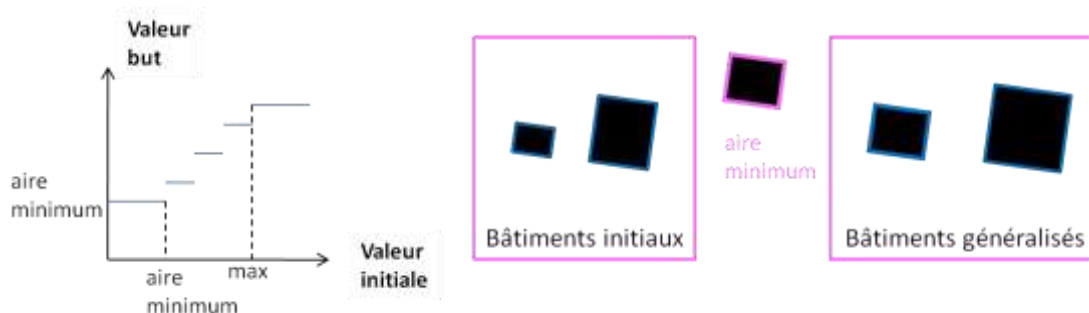


Figure C-64. Illustration du calcul de la valeur but pour un moniteur de contrainte d'aire minimum des bâtiments : quand la valeur initiale est légèrement supérieure à la valeur minimum de la contrainte, la valeur finale doit être un peu supérieure pour faciliter la différenciation entre les bâtiments.

Dans CollaGen, des évaluations plus globales (sur tout un espace géographique par exemple) sont nécessaires. Nous proposons de fonder les évaluations globales sur des distributions de satisfaction de moniteurs de contraintes. Une distribution permet de répartir les moniteurs selon leur

satisfaction (Figure C-65) ou selon leur type, par exemple pour créer des sous-distributions des moniteurs de contraintes micros ou relationnelles (pour évaluer la satisfaction globale des contraintes relationnelles par exemple).

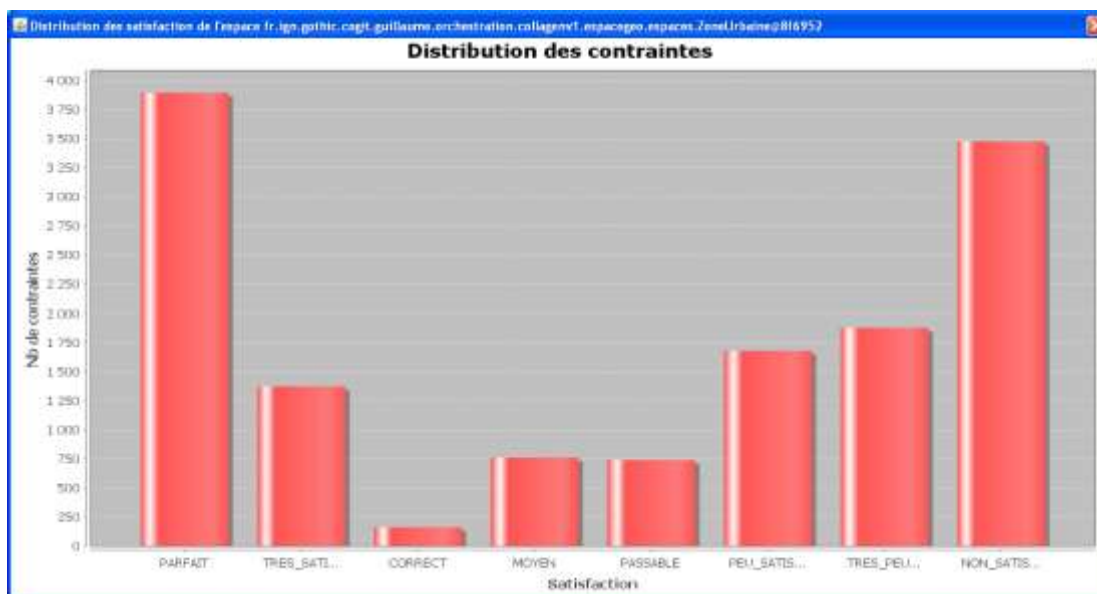


Figure C-65. Exemple de distribution de moniteurs de contraintes dans un espace urbain représentée par un histogramme de leur satisfaction.

La manière d'analyser les distributions de moniteurs dépend de l'objectif de l'évaluation dans CollaGen (il y a plusieurs étapes d'évaluation avec des objectifs différents) et les parties suivantes décrivent comment ces distributions sont analysées dans le cas de l'observation en ligne d'une généralisation et dans le cas d'une évaluation *a posteriori* d'une généralisation.

C.14.3. Observation en ligne d'une généralisation

Objectifs

(Ruas, 1999) et (Duchêne, 2004) notamment ont montré que dans le déroulement d'une généralisation pas à pas locale (mais c'est aussi valable pour d'autres approches), des conflits proches pouvaient rester non résolus et en plus gêner la généralisation des objets voisins. L'objectif de l'observation en ligne d'une généralisation est donc de vérifier que des zones de conflits persistants n'apparaissent pas et d'alerter le système en cas de suspicion d'apparition.

De plus, CollaGen fonctionne par essai-erreur et quand l'application d'un processus de généralisation est évaluée comme mauvaise, il l'annule et essaie autre chose à la place. Les processus de généralisation étant souvent gourmands en temps de calcul du fait de leur complexité, un objectif secondaire de l'observation est de déclencher un deuxième type d'alerte quand le système observe que la généralisation a très mal commencé et que cela ne mènera pas à une bonne généralisation. Le système peut ainsi gagner du temps en annulant tout de suite la généralisation pour en essayer une autre.

Nous présentons d'abord le problème de la taille de l'espace des observables, commun aux deux objectifs et notre réponse à ce problème. Puis nous décrivons comment le composant d'observation de CollaGen répond aux deux objectifs énoncés plus haut. Enfin, nous comparons notre proposition avec une méthode d'observation en ligne du modèle de généralisation CartACom.

Problème de la taille de l'espace des observables

En considérant que la généralisation d'une carte topographique peut être spécifiée par une centaine de contraintes (Stöter et al, 2010) et qu'un petit espace géographique comme une ville de 15 000 habitants comporte plusieurs centaines d'objets, nous voyons que le nombre de moniteurs de contraintes présents dans la distribution d'un espace géographique peut être très important (on peut compter déjà des milliers de moniteurs de contraintes de proximité par exemple). Pour éviter de trop allonger le temps de calcul d'un processus de généralisation, nous devons donc écarter les hypothèses d'une observation en temps réel des contraintes (toutes les microsecondes par exemple) et d'une observation de temps en temps de l'ensemble des moniteurs de contraintes. Cette dernière hypothèse est écartée car le coût en temps de calcul serait trop important, l'ensemble de la généralisation collaborative étant déjà un processus long qu'il faut éviter le plus possible d'alourdir. Afin de gérer la taille de l'espace des observables nous proposons de définir un pas de temps assez large pour ne pas observer trop longtemps. Ce pas de temps est défini par le fournisseur de processus lorsqu'il implémente la classe spécialisée de Processus de Généralisation nécessaire à la publication de processus dans CollaGen (voir C.9.2). La précision du pas de temps dépend fortement de l'observabilité du processus (voir partie suivante).

En nous inspirant des travaux de (Morvan et al, 2009) qui travaille sur l'observation en temps réel de simulation multi-agents de phénomènes naturels, nous proposons enfin de n'observer qu'un échantillon bien choisi des moniteurs de contraintes pour alléger la phase d'observation. Nous proposons de faire dépendre la méthode pour déterminer un échantillon représentatif de moniteurs du type d'espace à observer : une méthode par défaut est implémentée et peut être spécialisée pour chacun des types d'espaces géographiques. La méthode par défaut adopte un échantillon avec une répartition spatiale et sémantique équitable : elle découpe l'espace par une grille régulière et choisit au hasard 20% des contraintes sur chacun des types d'entités géographiques présents (par exemple, 20% des moniteurs de contrainte sur les bâtiments et 20% des moniteurs de contrainte sur les routes). Cette valeur peut être modifiée si nous souhaitons bouger le curseur entre la vitesse et la précision de l'observation. Une perspective d'évolution de cette proposition serait d'exploiter la théorie des sondages (Dusseix et Grosbras, 1996) pour déterminer le nombre représentatif de moniteurs à observer en fonction d'une erreur maximale dans la précision de l'observation.

Toutefois, afin de rendre notre composant d'observation plus efficace, nous proposons une observation en deux temps. L'idée est d'avoir un premier niveau d'observation très général et rapide avec un échantillon plus restreint. Si durant l'une des périodes d'observation, une grande proportion des moniteurs observés dans une zone pose problème, le composant le note et réalise un zoom lors des observations suivantes pour observer un échantillon plus important dans cette zone. La Figure C-66 illustre cette observation en deux temps : l'espace rural est découpé en trois zones d'observations par les routes (c'est une spécialisation de la méthode par défaut) et un échantillon de deux moniteurs est choisi dans chaque zone ; à la première observation, les moniteurs sont satisfaits

dans la zone du bas, non satisfaits à droite et à l'état initial à gauche (les objets n'ayant pas été encore traité par le processus, ils ne sont pas observés) ; la zone de droite est observée ensuite à la loupe en prenant un plus gros échantillon (ici 5 moniteurs) ce qui permet à la deuxième observation d'identifier un conflit local important non résolu par le processus de généralisation.

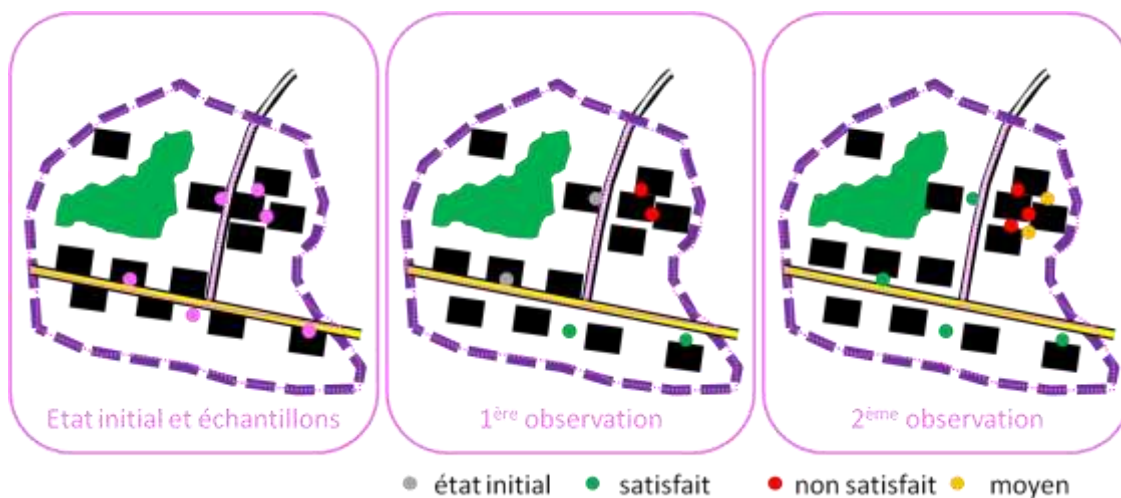


Figure C-66. Illustration de la méthode d'observation par zoom : le composant commence par des petits échantillons dans chacune des 3 zones découpées par les routes et à la 2^{ème} observation, le composant zoome (plus gros échantillon) suite aux problèmes observés dans la 1^{ère} observation.

Processus observables ou « boîte noire »

Nous avons vu dans les chapitres précédents que les processus de généralisation utilisables dans CollaGen pouvaient suivre des approches très diverses et donc se dérouler de manière différente suivant l'approche. Par exemple, un processus issu du modèle CartACom va généraliser séquentiellement chaque agent (bâtiment, route etc.), l'ordre d'activation dépendant du contexte. Il peut donc être interrompu théoriquement pour lui faire reprendre un cours normal plus tard (la possibilité d'interruption dépend aussi de l'implémentation du processus et, sur notre plateforme, l'implémentation permet l'interruption). Par contre, un processus de généralisation par moindres carrés fonctionne en une seule passe, il calcule son système d'équations, le résout et transforme les géométries en une fois : CollaGen ne peut pas le mettre en pause et regarder l'état des objets généralisés jusqu'à présent, tout est généralisé ou rien ne l'est. Nous appelons les processus comme CartACom des *processus observables* ou *interrompables* alors que ceux fonctionnant comme les moindres carrés sont appelés *boîtes noires*. Parmi les mécanismes que nous présentons ensuite (l'interruption anticipée d'un processus et l'émergence d'un sous-espace), seule l'émergence d'un sous-espace peut être observée lors de la généralisation par un processus de type boîte noire mais seulement à la fin de la généralisation et non pendant, comme pour les processus observables. Lorsque le fournisseur rend disponible un processus, il doit ainsi préciser si son processus est une boîte noire ou un processus observable (cela correspond à l'ajout de méthodes 'mettre en pause' et 'reprendre').

Par ailleurs, lorsque la généralisation est mise en pause, le composant d'observation s'assure qu'il n'y a pas d'incohérence créée avec l'état des connaissances du système en pause : il s'assure que le

processus de généralisation connaît bien les dernières versions des géométries des objets et les objets qui sont éliminés.

Interruption anticipée d'un processus de généralisation

Afin d'optimiser au maximum le temps de calcul d'une généralisation collaborative, nous proposons d'interrompre de manière anticipée un processus observable dont l'observation mettrait en lumière d'importants problèmes qui conduiraient sans doute à un état non valide, à l'étape d'évaluation finale de l'espace géographique généralisé. Si le nombre de moniteurs observés comme insatisfaits après le passage à l'observation à la loupe est supérieur à un seuil et si ces moniteurs n'ont pas vu leur satisfaction évoluer positivement entre les dernières observations, alors une alerte d'interruption est déclenchée et la généralisation est mise en pause. Nous proposons un premier seuil empirique de deux tiers des moniteurs en utilisant la quantité pondérée définie en C.13.3 mais ce seuil devra être validé par plus de tests.

Une fois la généralisation en pause, le composant vérifie plus précisément si la généralisation se déroule très mal ou s'il est encore possible qu'elle s'améliore. Pour cela, il commence par regarder l'ensemble des moniteurs de contraintes pour vérifier que l'échantillon est bien représentatif de l'état de la généralisation de l'espace. Si c'est le cas, il met en rapport les moniteurs non satisfaits avec la description formelle du processus utilisé : si les contraintes non satisfaites étaient annoncées comme gérées par le processus avec un fort indice de confiance (au moins 4/5), alors il laisse une chance au processus de s'améliorer ; si les contraintes non satisfaites ne sont pas gérées par le processus d'après la description, il le laisse continuer car il n'était de toute façon pas supposé gérer ces contraintes ; enfin, si les contraintes non satisfaites sont gérées avec un petit indice de confiance, le composant considère que le processus ne généralisera pas bien l'espace géographique et il l'interrompt de manière anticipée. Comme pour certains autres des nombreux problèmes que CollaGen doit traiter, il s'agit ici d'une première proposition qui ne demande qu'à être améliorée à la suite de tests, pour déterminer si certains processus interrompus n'auraient pas dû l'être et inversement.

Comparaison avec d'autres méthodes d'observation

Pour évaluer la méthode d'observation que nous proposons, il est intéressant d'analyser si d'autres modèles ou processus de généralisations sont dotés de tels modules d'observation et comment ils fonctionnent. Tout d'abord, dans le cadre du modèle AGENT, (Taillandier, 2008b ; Taillandier et al, 2010) proposent un mécanisme d'observation et de diagnostic de la qualité des connaissances procédurales qui guident le modèle. Les deux méthodes proposées utilisent comme critère l'efficacité de la généralisation à un moment donné du processus, calculée en fonction du premier quartile et de la moyenne de la distribution des satisfactions de contrainte. (Taillandier, 2008b) utilise d'autres critères issus de l'analyse des traces du processus et les combine par une méthode d'analyse multi-critères. (Taillandier et al, 2010) utilisent aussi des critères issus de l'analyse des traces mais les combinent par une méthode fondée sur la théorie des fonctions de croyance (Shafer, 1976). La différence principale avec notre méthode est l'utilisation des traces du processus AGENT car la méthode est intégrée au processus ce qui rend le diagnostic d'une mauvaise généralisation

plus facile à effectuer. Dans notre cas, l'observation doit être générique vis-à-vis du processus observé, donc, le composant ne peut pas compter sur l'analyse de traces qui peuvent ne pas exister pour certains processus et sont de toute façon hétérogènes dans leur forme. Nous ne pouvons donc pas utiliser une telle méthode pour l'observation dans CollaGen.

Le modèle CartACom est également doté d'un module d'observation qui a le même objectif que le nôtre à savoir identifier à temps les zones dans lesquelles le processus n'arrive pas à résoudre la généralisation (du fait d'une trop grande densité pour CartACom). Ce module observe la satisfaction des contraintes toutes les n activations d'agents mais écoute également les communications entre les agents pour déterminer si un problème existe (Duchêne et Touya, 2010). Cette méthode est donc *a priori* plus précise dans ses observations et les conclusions tirées des observations, mais moins générique que notre proposition car seulement adaptée à CartACom. Nous proposons dans le dernier chapitre des résultats de tests pour comparer effectivement les deux méthodes.

Pour résumer, dans CollaGen, l'observation est faite pendant la généralisation, tous les pas de temps, sur un échantillon représentatif spatialement et sémantiquement de moniteurs de contrainte. Un système de loupe permet d'observer plus en détail une zone ayant posé problème à l'observation précédente. Si le composant observe des conflits (qui sont censés être résolus par le processus) trop nombreux, le processus peut être arrêté de manière anticipée pour en essayer un autre.

C.14.4. Emergence d'un espace géographique

En plus de l'alerte déclenchée pour une éventuelle interruption anticipée, l'observation de la généralisation peut déclencher une alerte d'émergence quand le composant soupçonne l'émergence d'une zone de conflits non résolus. Le composant crée alors un sous espace géographique émergent à partir de cette zone de conflits.

Définition

L'émergence est un concept issu de l'informatique et très courant en SMA qui est en rapport avec l'observation de phénomènes collectifs non prévus (Duchêne et Touya, 2010). Nous retenons de (Forrest, 1991 ; Beni et Hackwood, 1992 ; Drogoul, 1993 ; Bonabeau et al, 1995 ; 1995b ; Jean, 1997 ; Servat et al, 1998 ; Müller, 2004) définissant l'émergence dans différents contextes les points suivants :

- L'émergence est liée à la présence d'au moins deux niveaux d'entités : celui des entités interagissant (causes de l'émergence) et celui, de plus haut niveau du phénomène émergent. Pour nous il s'agit du niveau des objets géographiques et du niveau des sous-espaces nécessitant une généralisation par un processus différent.
- L'émergence est liée à la présence d'un observateur pouvant identifier le phénomène émergent, donc d'un point de vue. Pour nous le point de vue est celui de l'observation d'un processus de généralisation par le composant d'observation de CollaGen.
- L'émergence est liée à l'apparition de phénomènes ayant une certaine stabilité dans le temps
 - la notion d'émergence ne concerne d'ailleurs que les systèmes évoluant dans le temps.

Dans notre cas, les différentes étapes et déformations nécessaires à la généralisation finale peuvent être vues comme une évolution du système dans le temps, et les conflits qui nous intéressent sont ceux qui perdurent sans pouvoir être résolus.

- L'émergence révèle un changement dans le comportement global d'un système. Dans notre cas, il s'agit d'une partie des données que le processus de généralisation n'arrive pas à traiter correctement.

Mécanisme d'alerte

Durant l'observation, une alerte est déclenchée pour que le composant d'observation arrête la généralisation et se mette à détecter un espace géographique émergent. Le système d'alerte est basé sur l'analyse de la répartition spatiale des moniteurs de l'échantillon en conflit persistant. Nous considérons ici comme moniteur en conflit persistant un moniteur dont l'objet rattaché a déjà été traité par le processus de généralisation (mais pas forcément modifié), dont la satisfaction est faible et n'a pas augmenté ou dont la satisfaction a diminué de manière notable (ce que nous avons fixé à deux niveaux dans notre échelle des satisfactions). La Figure C-67 montre des exemples de moniteurs dont certains en conflit persistant et d'autres non : le moniteur de maintien de la position initiale est par exemple en conflit persistant car sa satisfaction baisse fortement et le moniteur de proximité entre bâtiment et route n'est pas en conflit persistant malgré sa mauvaise satisfaction stationnaire, car ni la route ni le bâtiment concernés n'ont été traités par la généralisation.

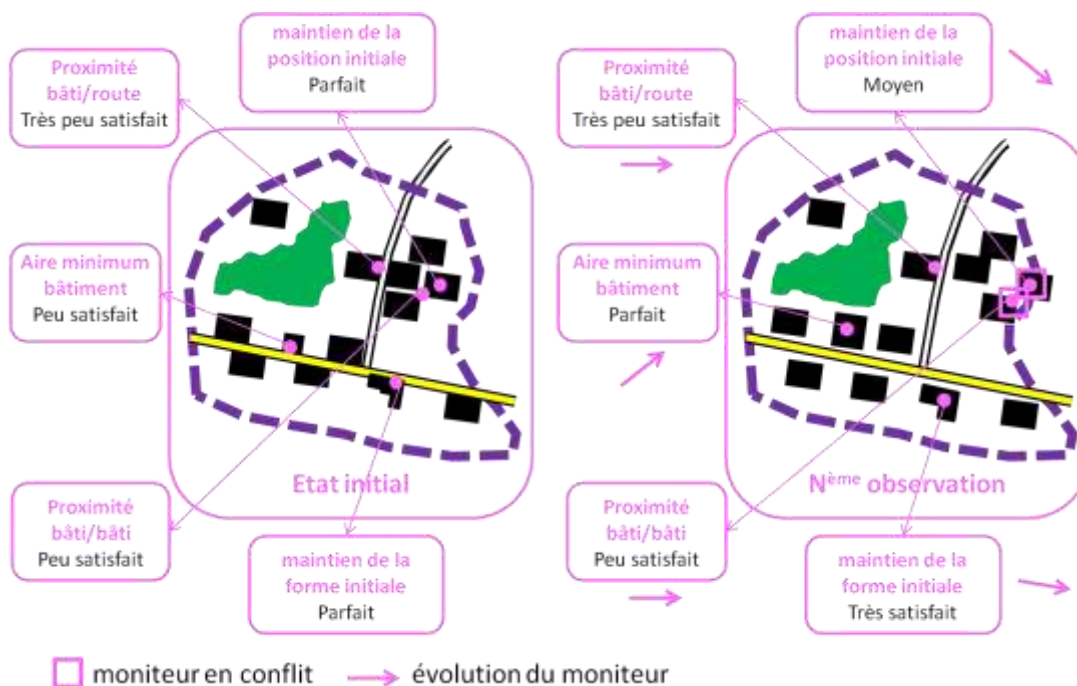


Figure C-67. Détermination des moniteurs en conflit persistant lors d'une observation.

Une fois les moniteurs de l'échantillon en conflit persistant identifiés, leur répartition spatiale est analysée pour déterminer s'il existe des groupes de moniteurs en conflit proches. Pour cela, comme pour la détermination de proximités spatiales entre bâtiments dans (Ruas, 1998 ; Harrie, 1999 ; Burghardt et Cecconi, 2007), nous utilisons une triangulation de Delaunay entre les géométries des moniteurs. La détection de ces groupes se fait en deux temps : le composant réalise une

triangulation entre les moniteurs en conflit, puis élimine les arêtes de la triangulation dont la longueur est supérieure à un seuil (Figure C-68). Les moniteurs en conflit encore reliés par des arêtes de la triangulation forment des groupes de moniteurs, points de départ des espaces émergents.

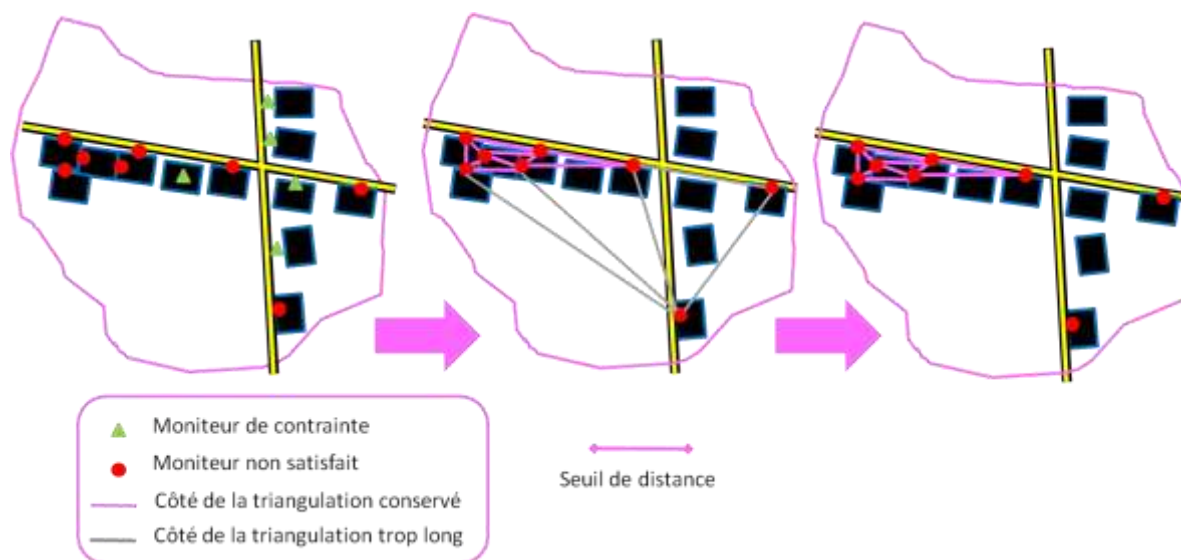


Figure C-68. Analyse de la répartition spatiale des moniteurs en conflits pour localiser des espaces émergents en deux étapes : le composant triangule les moniteurs en conflits puis enlève les arêtes dont la longueur est supérieure à un seuil. Un seul groupe de moniteurs est obtenu dans cet exemple.

Mécanisme de détection de l'émergence

Si des groupes de conflits proches sont détectés à l'étape précédente, le composant met la généralisation en pause pour tenter d'identifier des espaces émergents à partir de ces groupes. Le composant commence par filtrer les groupes en fonction de leur taille : il élimine les petits qui ne sont pas significatifs. Pour cela, il utilise le nombre d'arêtes de triangulation du groupe plutôt que le nombre de moniteurs, car, par exemple, trois moniteurs peuvent être groupés par deux arêtes ou trois ce qui est synonyme de plus de proximité entre les conflits. Jusqu'à présent, le raisonnement est uniquement basé sur les moniteurs de l'échantillon représentatif, donc, le composant élargit ensuite le groupe à l'ensemble des moniteurs en conflit contenus dans l'enveloppe convexe du groupe légèrement dilatée. Nous obtenons alors la vision orientée moniteurs de l'espace émergent. Pour obtenir l'étendue définitive de l'espace émergent, nous basculons vers une vision orientée objets géographiques en récupérant l'ensemble des objets attachés aux moniteurs du groupe de conflits. Par exemple, si un moniteur correspond à une contrainte de proximité entre un bâtiment et une route, le composant ajoute ces deux objets au groupe. Il calcule ensuite le contour de l'espace émergent formé par ces objets en prenant une enveloppe concave de l'ensemble d'objets calculée par la méthode du « bras ballant » (Galton et Duckham, 2006). Puis cette enveloppe est dilatée pour laisser la place aux objets d'être généralisés par un autre processus. Ce mécanisme est illustré dans la Figure C-69. Une fois que l'espace émergent a été construit, il est aussitôt généralisé comme un autre espace (si ce n'est qu'aucune détection d'espaces émergents dans cet espace ne sera réalisée) en consultant le registre, puis le processus choisi est paramétré par le traducteur et la généralisation est exécutée et évaluée.

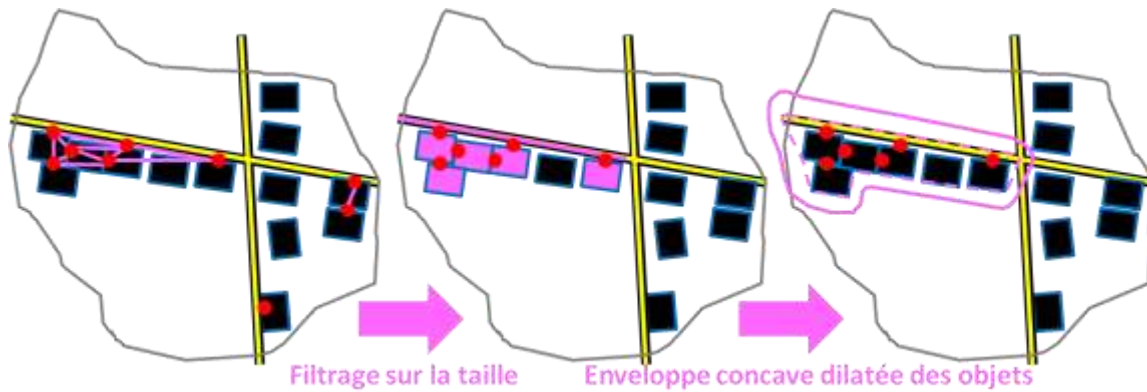


Figure C-69. Mécanisme de détection des espaces émergents : le composant filtre les petits groupes, récupère les objets concernés puis construit l'enveloppe concave et enfin dilate l'enveloppe. L'étape de récupération de l'ensemble des moniteurs est éludée.

Paramètres

Le composant d'observation peut détecter plus ou moins d'espaces émergents si l'on fait varier les trois paramètres de la méthode : le choix des moniteurs en conflit, le seuil de longueur des arêtes et le nombre minimum d'arêtes pour conserver un groupe de moniteurs. Des valeurs empiriques ont été déterminées pour obtenir des résultats équilibrés, et ces valeurs sont évaluées dans la partie E.5, en les comparant avec d'autres valeurs pour ces trois paramètres. Il faut porter une attention particulière aux deux paramètres qui conditionnent l'arrêt du processus de généralisation en vue de l'émergence : la méthode de choix des moniteurs en conflit et la longueur minimum d'arête de triangulation. En effet, arrêter le processus rapidement (avec des paramètres permettant facilement la formation de groupes) permet de corriger les conflits qui gênent le bon déroulement du processus, soit parce que celui-ci essaie de les corriger sans espoir, soit parce que leur présence brouille les prises de décisions sur les objets voisins. Mais en arrêtant trop tôt, le système s'expose à l'émergence d'espaces trop petits qui ne permettent pas de résoudre correctement l'ensemble des conflits.

Comparaison avec CartACom

Parmi les modèles de généralisation dont nous avons connaissance, seul CartACom propose un mécanisme équivalent d'émergence de zones de conflits destinées à être généralisées autrement (Duchêne et Touya, 2010). Il y a deux différences notoires avec la méthode du modèle CartACom. Premièrement, dans CartACom, la détection des objets concernés par le groupe émergent n'est pas limitée à l'analyse des proximités entre conflits persistants mais utilise avantageusement l'analyse des communications entre les agents (CartACom fonctionne par communications transversales entre agents micros) : les agents ayant communiqué avec les agents en conflit sont inclus car ils ont pris part à la mauvaise décision de généralisation. Deuxièmement, l'émergence de groupes est directement guidée par l'opération de groupe (déplacement, élimination, typification etc.) candidate à une application. Dans CartACom, les groupes destinés à une élimination et ceux destinés à une typification ne sont pas construits de la même manière. Comme pour le mécanisme d'observation,

notre méthode de détection de l'émergence est moins précise que dans CartACom mais plus générique car elle peut fonctionner en observant n'importe quel processus de généralisation.

En résumé, le composant d'observation stoppe la généralisation quand il observe des ensembles de moniteurs de contrainte en conflit, regroupés par proximité par un graphe de Delaunay. Les espaces émergents sont alors construits à partir des plus importants de ces groupes en englobant tous les objets concernés par les moniteurs de contrainte. Trois paramètres permettent de détecter plus ou moins d'espaces émergents. Il convient de bien les régler afin de ne pas arrêter inutilement et trop fréquemment le processus.

C.14.5. Evaluation finale de la généralisation d'un espace

Après avoir présenté les capacités d'observation du composant d'observation et d'évaluation, dans cette partie, nous allons décrire la problématique de l'évaluation de la généralisation d'un espace géographique par le processus de généralisation choisi par le registre.

Que souhaitons-nous évaluer ?

Nous avons vu que nous disposons, grâce aux moniteurs de contrainte, d'une évaluation qualitative individuelle du respect des spécifications formalisées de la carte généralisée. Dans CollaGen, l'évaluation finale de la généralisation d'un espace géographique est à la fois une évaluation pour contrôler et une évaluation pour valider si l'on reprend la classification de (Mackaness et Ruas, 2007). Il s'agit d'une évaluation pour contrôler car cette évaluation est le support de la décision de continuer ou de revenir en arrière dans le système en essai-erreur de CollaGen. Il s'agit également d'une évaluation pour valider car il s'agit de qualifier la généralisation de succès ou d'échec afin d'enregistrer ce résultat dans le registre pour alimenter ses capacités d'apprentissage.

Le composant d'évaluation doit réaliser ces évaluations en agrégeant toutes les informations individuelles des moniteurs. De plus, comme CollaGen est un système dans lequel un processus a été choisi automatiquement pour généraliser un espace, nous proposons de distinguer la qualité interne de la généralisation, c'est-à-dire le respect des capacités prévues du processus, et la qualité globale de la généralisation, c'est-à-dire le respect global des spécifications de la carte. Nous allons d'abord expliquer comment fonctionne le mécanisme d'évaluation à partir de ces qualités interne et globale. Puis nous détaillons nos propositions pour déterminer ces deux qualités en analysant les distributions de satisfaction des moniteurs de contrainte. Enfin, nous expliquons notre méthode d'agrégation des évaluations pour noter une généralisation.

Mécanisme d'évaluation

La Figure C-70 montre le mécanisme que nous proposons d'adopter pour interpréter les informations de qualité interne et globale. Le composant agrège les deux informations de qualité pour noter grossièrement la généralisation et stocker la notation dans le registre s'il s'agit d'une bonne note ou d'une mauvaise. Ces informations de qualité permettent de choisir l'une des trois évolutions possibles dans la dynamique de CollaGen :

- Soit le composant considère que l'espace est mieux généralisé que dans l'état précédent mais qu'il ne l'est pas encore parfaitement, auquel cas il **valide le nouvel état** de l'espace (et de ses objets) et fait une **nouvelle requête au registre** pour un nouveau processus qui peut traiter les conflits restants.
- Soit le composant considère que la généralisation de l'espace est terminée et passe au **choix d'un nouvel espace** (les effets de bord éventuellement générés par la généralisation sont vérifiés entre temps).
- Soit le composant considère que la généralisation de l'espace est vraiment mauvaise auquel cas il **annule la généralisation** et remet l'espace (et ses objets) dans leur état précédent. Il essaye alors le processus suivant dans la liste renvoyée par la précédente requête au registre.

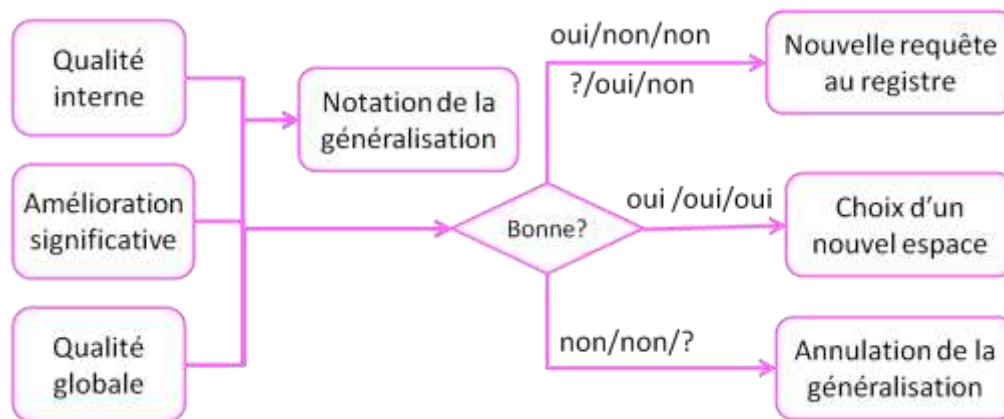


Figure C-70. Description du mécanisme d'évaluation qui détermine la qualité interne et externe à partir de quoi il note la généralisation et prend la décision de valider l'état et passer à un nouvel espace si les deux sont bonnes, valider l'état et demander une généralisation supplémentaire au registre si seule la qualité interne est bonne et enfin annuler la généralisation et essayer le processus suivant dans la liste si la qualité interne est mauvaise.

La première des trois options est choisie uniquement quand chacune des deux qualités interne et globale est évaluée comme bonne et que le processus a amené une amélioration significative de la généralisation de l'espace géographique. Cela signifie que la généralisation par le processus choisi a bien répondu aux attentes et a bien traité les conflits identifiés dans l'espace. La deuxième option est choisie quand l'état de l'espace s'est amélioré significativement mais avec une qualité globale insuffisante, quelle que soit la qualité interne. Pour faire simple, le processus a bien satisfait tout ou partie des contraintes qu'il était censé satisfaire mais n'a pu gérer d'autres conflits présents dans l'espace. Cette option est aussi choisie quand il n'y a pas d'amélioration significative mais que la qualité interne est bonne car cela signifie que le manque d'amélioration est surtout dû à des capacités limitées du processus mais les améliorations mineures (bonne qualité interne) sont quand même bonnes à prendre. CollaGen cherche alors logiquement un autre processus pour compléter le premier et résoudre les conflits restants. Enfin, la troisième option est choisie quand il n'y a aucune amélioration significative et que la qualité interne est mauvaise quelle que soit la qualité globale car cela signifie que le processus n'est pas arrivé à faire ce qu'il était censé faire. Cela peut arriver avec les processus de généralisation car ils ne sont pas infallibles et la diversité des contextes géographiques peut générer une infinité de cas particuliers qu'un processus donné ne peut pas tous gérer. Il est donc logique de revenir à l'état précédent de l'espace (avant la généralisation

défectueuse) et d'essayer le processus suivant dans la liste des processus adaptés renvoyée par le registre.

Nous allons maintenant détailler comment nous déterminons une amélioration significative, la qualité interne et la qualité globale à partir des distributions de satisfactions des moniteurs de contrainte.

Amélioration significative de la qualité

Déterminer si la généralisation d'un espace géographique par un processus de généralisation a provoqué une amélioration significative de la qualité de l'espace se fonde sur la comparaison des distributions des satisfactions avant et après la généralisation. Plus précisément, nous proposons de nous fonder sur la variation de moniteurs non satisfaits (y a-t-il plus ou moins de moniteurs insatisfaits ?) et la variation de moniteurs très satisfaits. A partir de ces deux variations, nous proposons l'algorithme suivant où la fonction `getPourcentage(seuil)` donne le pourcentage de moniteurs de la distribution dont la satisfaction est inférieure ou égale au seuil de satisfaction :

```
ameliorationSignificative(distribution, distributionAvant) : boolean
  Réel nonPct ← distribution.getPourcentage(PEU_SATISFAIT)
  Réel nonPctAv ← distributionAvant.getPourcentage(PEU_SATISFAIT)
  Réel tresPct ← distribution.getPourcentage(CORRECT)
  Réel tresPctAv ← distributionAvant.getPourcentage(CORRECT)
  Booléen amelioration ← faux
  // S'il y a un peu plus de satisfaits et un peu moins de non
  // satisfaits, c'est une amélioration significative
  Si (nonPctAv - nonPct > 0.33*nonPctAv et tresPct - tresPctAv > 0.05)
    amelioration ← vrai
  Fin Si
  // S'il y a une diminution drastique des contraintes non satisfaites,
  // c'est une amélioration significative
  Si (nonPctAv - nonPct > 0.50* nonPctAv)
    amelioration ← vrai
  Fin Si
  // S'il y a une augmentation drastique des contraintes satisfaites,
  // c'est une amélioration significative
  Si (tresPct - tresPctAv > 0.66* tresPctAv)
    amelioration ← vrai
  Fin Si
```

Comme pour la qualité interne et la qualité globale, la méthode que nous présentons est une première proposition qui nécessite des tests de validation et de comparaison dont une partie seulement est effectuée dans cette thèse, les autres sont une des premières perspectives de notre travail.

Qualité interne de la généralisation

La qualité interne d'un traitement se définit en général comme le respect des spécifications définies par le producteur du traitement (Vauglin, 1997). Dans le cas de la généralisation d'un espace géographique dans CollaGen, la qualité interne est la satisfaction globale des contraintes de généralisation que le processus appliqué est censé satisfaire d'après les post-conditions de sa description formelle. Nous proposons de définir la qualité interne d'un processus de généralisation dans CollaGen comme une valeur entre 0 et 1, 1 correspondant à une qualité interne parfaite, c'est-à-dire la bonne gestion de l'ensemble des spécifications formelles présentes dans les post-conditions par rapport à l'ensemble des moniteurs correspondant contenus dans l'espace. Comme le montre l'algorithme ci-dessous, nous proposons de diminuer la qualité interne chaque fois que la satisfaction d'un moniteur (correspondant à une des post-conditions) est inférieure à la moyenne des satisfactions de la distribution. La diminution est proportionnelle à l'indice de confiance de la post-condition et à la distance entre la satisfaction du moniteur et la moyenne. Nous proposons de travailler en relatif (utilisation de la moyenne), car ainsi, la qualité interne acceptée dépend du contexte de satisfaction (on est moins tolérant quand les contraintes de l'espace sont plutôt bien satisfaites).

```

evaluationQualiteInterne(distribution, processus) : double
  Initialisation Réel seuil ← moyenne(distribution)
  Initialisation Réel qualite ← 1.0
  Pour toutes les post-conditions post de processus faire
    Initialisation Réel confiance ← post.getIndiceConfiance()/5
    Pour tous les moniteurs moniteur de distribution faire
      Si (moniteur.contrainte != post.contrainte) suivant
      Si (moniteur.satisfaction <= seuil) suivant
        Initialisation Réel dist ← seuil-moniteur.satisfaction
        Initialisation Réel ratio ← dist*confiance/|distribution|
        qualite = qualite - ratio
    Fin Pour
  Fin Pour

```

Un seuil est ensuite utilisé pour déterminer les généralisations évaluées avec une bonne qualité interne et celles avec une mauvaise qualité interne. Ce seuil doit être représentatif de la confiance que fait le développeur de CollaGen au fournisseur de processus car si la description des post-conditions est très approximative, il est difficile d'obtenir une forte valeur de qualité interne. Par exemple, dans notre implémentation de CollaGen, nous jouons tous les rôles donc nous avons confiance dans nos descriptions et utilisons un seuil de 0,8 ce qui est plutôt élevé.

Qualité globale de la généralisation

La qualité globale d'un traitement ou de données se définit en général comme son adaptation aux besoins de leur utilisateur. Dans le cas de la généralisation, les besoins de l'utilisateur sont une carte bien généralisée, ce qui se traduit dans CollaGen par les contraintes formelles et les règles opérationnelles. Si nous nous ramenons au cas d'un espace géographique généralisé par un processus, la qualité globale de l'espace généralisé correspond à la satisfaction globale de l'ensemble

des contraintes contenues dans l'espace géographique. Seulement, calculer la satisfaction globale d'un ensemble de contraintes de généralisation est un problème scientifique qui n'a pas actuellement de réponse pleinement satisfaisante. En effet, comme le problème de la généralisation est sur-contraint, nous n'obtiendrons jamais une satisfaction maximum pour l'ensemble des contraintes évaluées. Ce problème n'étant pas le cœur de notre travail de thèse, nous ne proposons qu'une première méthode très basique mais également une piste intéressante pour aborder de manière nouvelle ce problème.

Nous proposons donc dans un premier temps d'utiliser la moyenne pondérée par l'importance des satisfactions des contraintes de la distribution, comme c'est le cas notamment dans le modèle AGENT pour évaluer la satisfaction globale d'un objet. Nous utilisons aussi l'écart-type de satisfaction conjointement à la moyenne, car un résultat avec des contraintes très satisfaites et d'autres très peu satisfaites n'est pas aussi convenable que si toutes les satisfactions sont regroupées autour de la moyenne. Et encore, ceci dépend du contexte de la carte : par exemple, s'il est envisagé de réaliser des corrections interactives après généralisation automatique (comme cela se fait dans les chaînes de production de carte actuelles), un résultat global avec 10 moniteurs 'très insatisfaits' et 2 'passables' sera meilleur qu'un résultat avec 1 seul moniteur très insatisfait mais 20 moniteurs passables.

Pour aller plus loin dans l'estimation de la qualité globale d'une généralisation, nous pensons qu'il serait intéressant de faire une analogie avec le problème de bien-être global d'une société qui a été abordé par les économistes et les sociologues (Moulin, 2004). Par exemple, le bien-être de Nash, utilisé dans les problèmes d'allocation de ressources dans les SMA (Nongaillard et al, 2010 ; Ramezani et Endriss, 2010) repose sur le produit de l'ensemble des satisfactions donc de très mauvaises satisfactions font beaucoup baisser le bien-être global.

Notation de la généralisation

Nous avons montré comment étaient évaluées les qualités internes et globales de la généralisation, ainsi que l'amélioration significative d'un espace géographique dans CollaGen pour permettre le contrôle du système. Nous présentons maintenant les deux aspects de l'évaluation « pour valider » présents dans CollaGen. Noter la généralisation dans le but de stocker les succès et les échecs petit à petit dans le registre revient à agréger les évaluations des qualités interne et globale et de l'amélioration en une seule et unique note. Pour permettre l'apprentissage des succès et échecs par le registre, nous proposons que le composant d'évaluation donne trois types de notes : succès, échec ou moyen. Seules les généralisations notées comme des succès et des échecs sont alors stockées pour apprentissage.

- La note **succès** est attribuée uniquement quand la qualité interne est jugée comme bonne et qu'au moins un des deux autres critères (qualité globale et amélioration significative) est positif. Un succès est donc un processus qui fait très bien ce pour quoi il est conçu tout en améliorant l'état général de l'espace géographique.
- La note **échec** est attribuée dans un seul cas, quand les trois critères sont négatifs.
- Aucune note n'est attribuée dans tous les autres cas qui représentent, soit des demi-succès soit des demi-échecs (amélioration peu significative, qualité interne moyenne...). Dans ces

cas assez difficiles à évaluer, nous choisissons donc de ne pas influencer sur les connaissances du registre.

Décrire la qualité de la généralisation

Dans la classification des évaluations en généralisation de (Mackaness et Ruas, 2007), l'évaluation pour décrire signifie la production de métadonnées sur le résultat décrivant ce qui a bien fonctionné et ce qui n'a pas fonctionné. Nous proposons ainsi de décrire dans des traces des statistiques sur les moniteurs non satisfaits et les moniteurs très satisfaits. Par exemple, la Figure C-71 montre la description des moniteurs non satisfaits après la généralisation d'une zone urbaine par un processus issu du modèle AGENT : les trois concepts, types de contraintes, caractères et contraintes dont la satisfaction moyenne des moniteurs est la plus faible sont décrits dans le tableau avec leur représentativité, c'est-à-dire le pourcentage de ces moniteurs parmi les moniteurs non satisfaits. Dans l'exemple de la figure, il s'agit principalement de contraintes sur les objets de type végétation (non traités par le processus) qui sont non satisfaites. De telles métadonnées sur l'ensemble des généralisations effectuées permettent après une généralisation par CollaGen d'analyser en détail ce qui s'est passé.

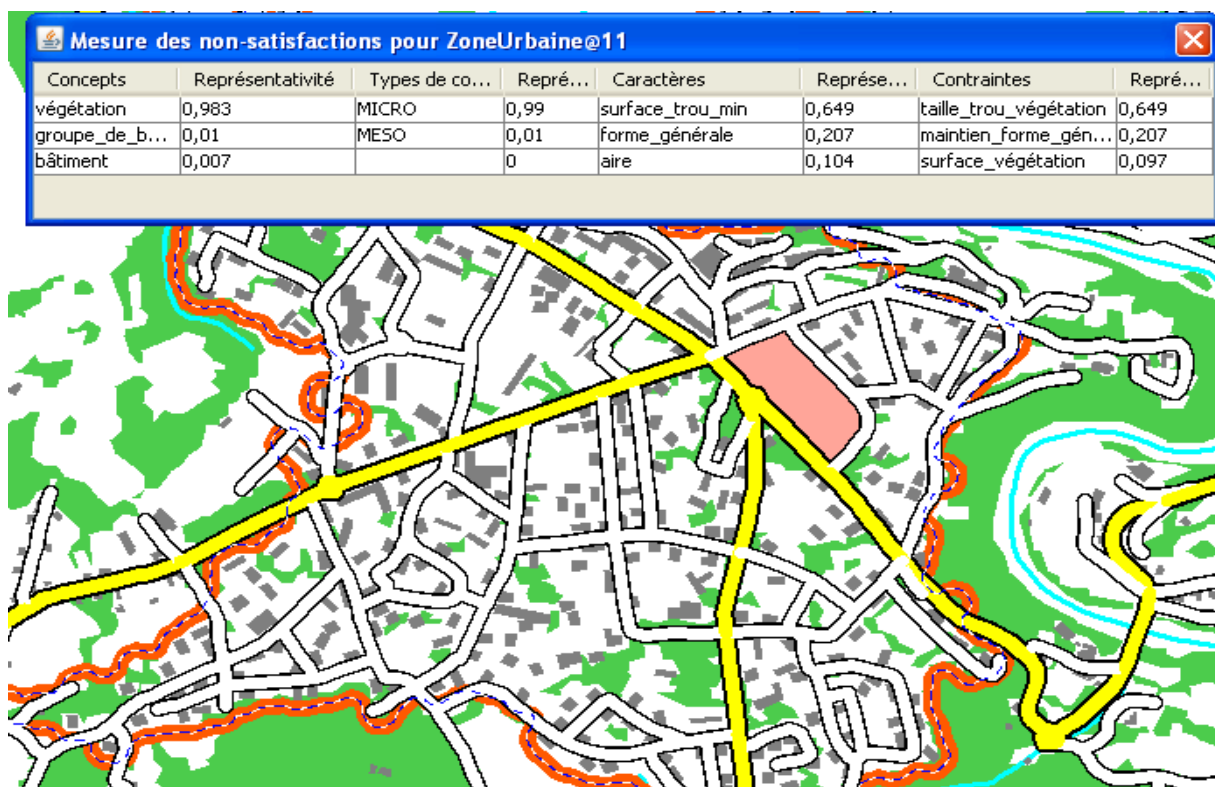


Figure C-71. Exemple de la description de la généralisation d'une ville (contour épais orange) généralisé par un processus issu du modèle AGENT.

En résumé, plusieurs types d'évaluations sont réalisés sur le résultat d'une généralisation d'un espace géographique par un processus de généralisation : le composant évalue la qualité interne du processus en comparant la satisfaction des moniteurs avec la description formelle, évalue la qualité globale pour l'instant de manière simplifiée en calculant moyenne et écart type des satisfactions et évalue si l'état des satisfactions s'est amélioré significativement après la

généralisation. Ces trois critères sont utilisés pour le contrôle du mécanisme par essai-erreur et pour la notation de la généralisation. De plus, une évaluation plus descriptive des moniteurs satisfaits et non satisfaits est produite sous forme de traces de CollaGen.

C.14.6. Interactions entre émergence et évaluation finale

Nous avons détaillé dans cette partie C.14 comment le composant d'observation et d'évaluation pouvait réaliser deux types d'évaluation de la généralisation : l'une en direct pendant le processus de généralisation qui peut mener à son interruption pour mieux généraliser des sous espaces émergents formés de conflits proches ; l'autre après la généralisation qui évalue globalement sur l'espace la satisfaction des contraintes pour valider le nouvel état et l'annuler pour essayer autre chose. Ces deux formes d'évaluations à deux moments différents peuvent donc apparaître redondantes et/ou concurrentes.

Dans CollaGen, nous considérons qu'elles ne le sont pas car elles travaillent à deux niveaux de granularité différents. La généralisation d'un espace rural comprenant plusieurs centaines d'objets bâtiments, routes, chemins ou cours d'eau ne sera pas mal évaluée par l'évaluation globale si cinq ou six de ces objets dans une zone trop dense sont encore en conflit alors que tous les autres ou presque sont bien généralisés. Mais il serait dommage de ne pas pouvoir intervenir localement pour corriger ce groupe de conflits et améliorer encore la qualité de la généralisation de cet espace.

L'évaluation faite à travers l'émergence de sous espaces géographiques a donc pour objectif de s'occuper de ces petits groupes de conflits localisés et non de traiter une grande quantité de conflits partout dans l'espace car il s'agit plutôt de la responsabilité de l'évaluation globale. Afin d'éviter que ces deux modes d'évaluation entrent effectivement en concurrence alors qu'ils ne sont pas conçus pour, deux garde-fous sont ajoutés au fonctionnement de l'observation des phénomènes émergents. Ainsi, les espaces émergents ne peuvent être trop grands sans quoi l'espace émergent n'est pas traité. Et le nombre total d'espaces émergents dans un espace donné est également limité : quand trop de sous-espaces de conflits émergent, le mécanisme d'émergence n'est plus le meilleur moyen de traiter l'espace géographique, il vaut mieux laisser la main à l'évaluation globale qui va permettre de revenir en arrière pour essayer autre chose.

C.15. COMPOSANT D'ORCHESTRATION

Parmi les six composants du cadre formel de généralisation collaborative mis en œuvre dans CollaGen, le composant d'orchestration a un rôle central car il dirige l'orchestration des généralisations d'espaces géographiques par des processus de généralisation adaptés (Figure C-72). Dans le cas de la généralisation cartographique, l'ordre des différentes opérations peut modifier fortement le résultat (Ruas, 1998), a fortiori quand il s'agit de l'ordre de processus de plus haut niveau. Le composant d'orchestration a donc un rôle clé dans le bon fonctionnement de CollaGen. Dans un premier temps, nous allons récapituler l'ensemble des fonctions que doit assurer ce

d'espace suivante et donc au choix du prochain espace. Deux fonctions sont nécessaires pour effectuer ce choix : choisir le prochain type d'espace à traiter (les espaces urbains ou le réseau routier par exemple) et ordonner les instances du type d'espace choisi, car CollaGen repose sur une dynamique séquentielle, une instance d'espace après l'autre (une ville après l'autre par exemple).

Les parties suivantes décrivent précisément ces trois fonctions du composant d'orchestration.

C.15.2. Gestion des états dans CollaGen

Le mécanisme par essai-erreur de CollaGen doit permettre de revenir en arrière si le processus proposé en premier par le registre n'a pas réalisé une bonne généralisation selon le composant d'évaluation. Ce retour en arrière est effectué par le composant d'orchestration quand l'évaluation de la généralisation précédente fournie au composant le réclame. Ce retour en arrière repose sur le stockage, après chaque généralisation, des états précédents de l'ensemble des objets contenus dans l'espace généralisé. Les modifications apportées aux objets durant une généralisation peuvent être de quatre types : élimination, changement de la géométrie, de la sémantique et agrégation. Nous proposons donc un stockage allégé de chacune des étapes en ne conservant que la géométrie de l'état intermédiaire ainsi que l'attribut booléen *est éliminé*, tout en gardant l'objet initial complet (Figure C-74). Quand un objet de la classe *X* est généralisé par un processus, une classe *Etat Précédent X* est créée si elle n'existait pas et un nouvel objet de cette classe est créé avec la géométrie initiale de la généralisation. Cet objet état précédent est relié au nouvel état courant ainsi qu'aux états précédents qu'il possédait quand il était objet courant. Cet objet est également relié à l'espace géographique et au processus qui ont entraîné sa généralisation et son passage en état précédent.

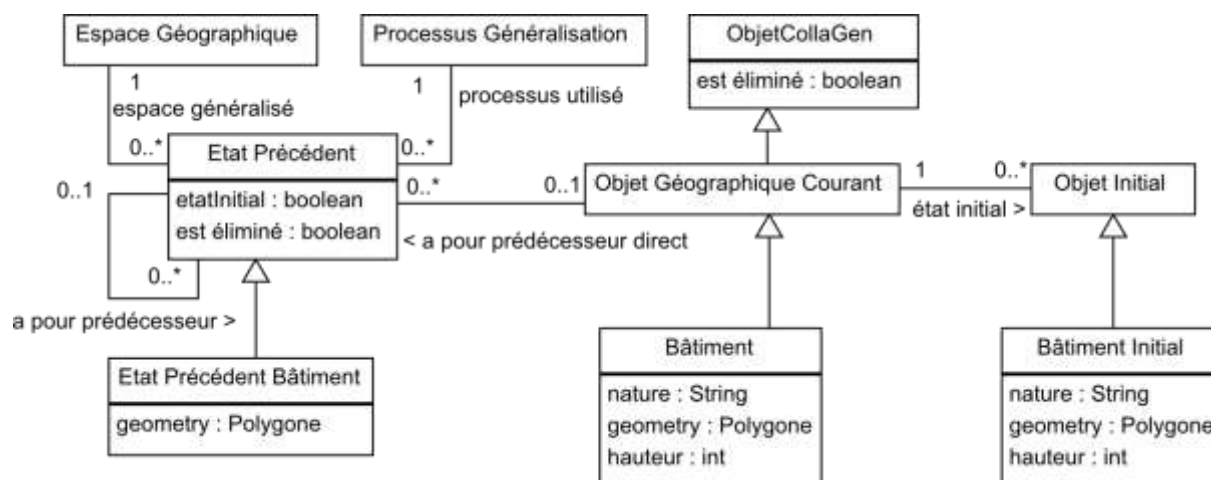


Figure C-74. Diagramme de classe UML du schéma de données nécessaire à la gestion des états des objets dans CollaGen : un nouvel état est créé après chaque généralisation par un processus.

Pour permettre le stockage, le composant d'orchestration réalise une copie des objets contenus dans un espace géographique quand cet espace est sur le point d'être généralisé par un processus. Après chaque généralisation validée par l'évaluation, le schéma de stockage (et la partie correspondante de la base) est mis à jour par le composant d'orchestration : les objets préalablement copiés sont stockés dans les classes d'objets précédents et reliés aux nouveaux objets courants issus de la généralisation ; le lien est réalisé automatiquement grâce à l'identifiant unique hérité de la classe

Objet CollaGen ; les objets éliminés par le processus sont marqués comme éliminés mais sont également stockés. Il convient donc de noter qu'un certain nombre d'états intermédiaires ne sont pas stockés, notamment ceux des généralisations non validées. Si l'on souhaite tracer l'ensemble des étapes du processus collaboratif, il serait préférable d'ajouter leur stockage et donc de modifier légèrement le schéma de la Figure C-74.

Dans le cas d'opérations de reclassification (les objets changent de classe suite à la transformation engendrée par la généralisation), ce schéma de stockage est toujours valable en faisant simplement hériter les nouvelles classes d'objets de *Objet Géographique Courant* et en liant les objets de la nouvelle classe avec des objets d'anciennes classes par le lien *a pour prédécesseur direct*. Par exemple, si un processus de généralisation construit des objets d'une classe *Zone Bâtie* dans les parties très denses des villes comme peut le faire AGENT (Barrault et al, 2001), les objets de cette classe auront chacun pour prédécesseur l'ensemble des bâtiments formant la zone bâtie ajoutés dans la classe *Etat Précédent Bâtiment*.

C.15.3. Choisir un type d'espace géographique

Après avoir vu comment le composant d'orchestration gère les différents états que prennent les objets géographiques durant leur généralisation dans CollaGen, nous détaillons notre proposition pour choisir le prochain type d'espace géographique à généraliser à un moment donné. De manière complémentaire à notre proposition de règles d'orchestration, nous proposons que ce choix soit fait par un système expert intégré au composant d'orchestration capable d'inférences à partir des règles d'orchestration telles que nous les avons modélisées en C.8. Un système expert possède trois composants principaux : une base de règles, des faits et un moteur d'inférence permettant de prendre une décision sur un fait à partir des règles (Jackson, 1998). Dans notre cas, les règles sont les règles d'orchestration et les faits la position courante dans le processus CollaGen. La position dans le processus est modélisée de la manière suivante (Figure C-75) : elle contient l'étape de généralisation courante (par exemple, l'étape de Sélection), l'espace géographique généralisé juste avant (par exemple, l'unique instance de type Réseau hydrographique) et le processus utilisé juste avant pour généraliser l'espace (par exemple, le processus issu de (Touya, 2007)) ; un attribut booléen signale également si l'on se situe au début d'une étape (i.e. la décision précédente du moteur d'inférence a été de changer d'étape plutôt que de généraliser un type d'espace).

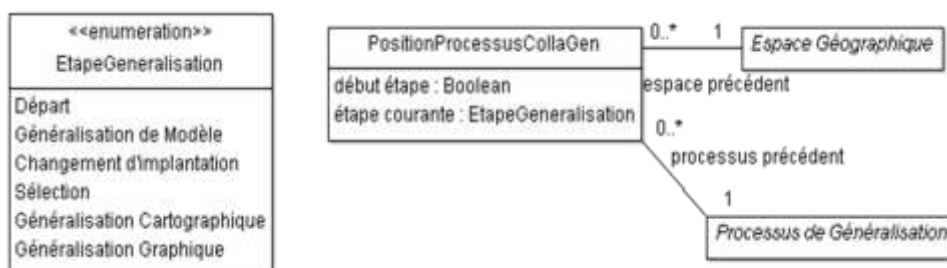


Figure C-75. Diagramme de classe UML de la représentation de la position dans le processus de CollaGen.

Le moteur d'inférence que nous proposons doit gérer trois contraintes : il doit pouvoir donner un type d'espace à généraliser à tout moment du processus, il doit pouvoir gérer un jeu incomplet de

règles et il doit éviter de rester coincé dans une étape. Le moteur fonctionne avec l'algorithme suivant :

- Le moteur cherche d'abord la prémisse la plus proche du fait (la position dans le processus) par chaînage avant (Russell et Norvig, 2006).
 - Si le moteur trouve une règle avec une prémisse assez proche, il applique directement la conclusion de la règle (qui est le prochain type d'espace à généraliser).
 - Sinon (cas d'un jeu de règles incomplet), la priorité du moteur est de sortir de l'étape courante pour passer à la suivante. Il cherche donc par chaînage arrière la règle qui permet de sortir de l'étape et remonte la chaîne (Si les règles sont et C est le passage à l'étape suivante, la chaîne trouvée sera A, B, C). Afin d'éviter l'oubli d'espaces à généraliser, le moteur cherche alors les espaces clés, non traités dans cet espace et non présents dans la chaîne arrière.
 - S'il y en a, il choisit parmi ces espaces celui qui a le plus besoin d'être généralisé.
 - Sinon, le moteur choisit l'espace qui correspond à la première prémisse de la chaîne arrière, ce qui permet de se raccrocher aux règles qui aboutissent à la fin de l'étape.

Nous détaillons maintenant le fonctionnement de chacun de ces points. Tout d'abord, le moteur cherche la prémisse la plus proche par chaînage, la proximité étant calculée par une distance utilisant notamment la hiérarchie des concepts de l'ontologie. Pour établir la distance entre fait et prémisse d'une règle, nous appliquons la méthode suivante :

```

Distance(fait, règle) : double
  // on vérifie qu'ils concernent bien la même étape courante
  Si (regle.etape = ∅) Alors
    Initialisation distance ← 1
  Sinon Si (fait.etape != règle.etape) Alors
    Initialisation distance ← +∞
  Fin Si
  Initialisation PremsseRegleOrchestration premsse ← règle.premsse()
  // on compare prémisse et fait par rapport à l'espace précédent
  Si (fait.espacePrécédent est un premsse.espacePrécédent) Alors
    distance ← distance + nb de concepts entre les deux
  Sinon
    Initialisation distance ← +∞
  Fin Si
  // on compare prémisse et fait par rapport au processus précédent
  Si (fait.processusPrécédent est un premsse.processusPrécédent) Alors
    distance ← distance + nb de concepts entre les deux
  Sinon
    Initialisation distance ← +∞
  Fin Si
  // on traite le cas d'une prémisse de début d'étape
  Si (premsse.espacePrécédent = ∅ et premsse.processusPrécédent = ∅
  et fait.debutEtape = faux) Alors
    Initialisation distance ← +∞

```

Fin Si

Pour résumer, le concept de l'ontologie décrit dans le fait (que ce soit l'espace précédent ou le processus précédent) doit être le même que dans la prémisse, ou plus spécifique, auquel cas la distance est augmentée. Par exemple, si nous avons dans l'ontologie les concepts suivants *réseau routier* est un *réseau de transport* est un *réseau* est un *espace thématique*, alors le fait ayant pour espace précédent *réseau routier* est compatible avec une règle de prémisse sur *réseau* avec une distance de 2.

Si aucune règle n'est à une distance finie de la situation courante dans le processus, le moteur cherche la règle qui termine l'étape courante pour passer à la suivante et réalise le chainage arrière en utilisant cette même distance pour trouver de manière itérative (en partant de cette dernière règle) la règle dont la conclusion est la plus proche de la prémisse courante. Le résultat de ce chainage arrière est un ensemble de règles qui mène d'une prémisse à la fin de l'étape courante, en passant par la généralisation d'un certain nombre de types d'espaces géographiques.

Mais avant de réaliser cet enchaînement de règles, le moteur s'assure qu'il n'a pas oublié d'espace clé à généraliser. Pour cela, il cherche si des types d'espaces clés sont absents de la liste des espaces déjà traités dans cette étape et de la séquence de chainage arrière. S'il y en a, le moteur ne dispose d'aucune connaissance dans les règles pour guider son choix, il doit utiliser d'autres critères. Ce problème se rapproche beaucoup de celui de la stratégie d'activation des agents dans un système multi-agent, notamment ceux dédiés à la généralisation : quel est le prochain agent à activer à un moment donné du processus ? (Ruas, 1999) pose le problème de la stratégie d'activation à un niveau de granularité différent (au niveau des objets meso et micro) dans le modèle AGENT mais ses propositions restent valables dans notre cas. Deux approches complémentaires sont distinguées, la planification prédéfinie ou le choix à partir de l'état des contraintes. La planification prédéfinie correspond à notre cas général d'utilisation des règles d'orchestration donc l'état des contraintes peut être une bonne stratégie simple pour le choix du prochain espace quand aucune ne correspond à la position dans le processus CollaGen. Dans le modèle CartACom, le problème de l'ordre d'activation des agents à généraliser se pose aussi et deux stratégies sont proposées, une activation aléatoire par cycle et par meilleur prochain (Duchêne, 2004). La stratégie par meilleur prochain (guidée par la réception des messages de demande d'action échangés entre les agents) s'avère être la plus efficace et efficiente. (Altay, 2010) propose même une amélioration ad-hoc de la stratégie par meilleur prochain avec des résultats intéressants. Néanmoins, ces stratégies ne sont pas directement applicables à notre problème, donc, nous proposons d'utiliser simplement une méthode guidée par la satisfaction des moniteurs de contraintes à l'intérieur des espaces géographiques : le type d'espace, parmi les espaces clés restant, dont les instances sont les moins satisfaites en moyenne, est choisi par le moteur pour être le prochain à être généralisé.

Dernier cas, si le moteur ne trouve pas de règle permettant de terminer l'étape courante, il choisit simplement un des types d'espaces clés restant, les triant encore en fonction de la satisfaction des moniteurs dans les instances du type. Quand il ne reste plus de type d'espace clé à traiter, le moteur fait passer la généralisation collaborative à l'étape suivante.

Etant donné la difficulté relative de décrire des règles d'orchestration, d'autant plus dans un cadre formel comme CollaGen, nous avons doté le moteur d'inférence de notre composant d'orchestration

de capacités de vérification automatique de la base de règles. Ainsi, il est possible de vérifier si la base de règles d'orchestration contient les incohérences suivantes :

- Incompatibilité de deux règles (R1: Si A et B alors C; R2: Si A et B alors D;)
- Redondance de deux règles (R1: Si A alors B; R2 : Si C alors B)
- Bouclage entre trois règles (R1 : Si A alors B ; R2 : Si B alors C ; R3 : Si C alors A ;)

Dans notre cas, la redondance n'est pas forcément un défaut de cohérence car la généralisation d'un espace donné peut être la conclusion de plusieurs règles avec des prémisses différentes (des étapes courantes différentes par exemple). Donc une base de règle redondante n'est pas jugée comme mauvaise. Mais la présence des deux autres types d'incohérences fausserait fortement les résultats du moteur d'inférence donc la vérification est systématique au lancement d'une généralisation par CollaGen.

Prenons maintenant un exemple simplifié pour illustrer le choix du prochain type d'espace à généraliser à partir des règles d'orchestration. Nous disposons dans la base de règles des règles d'orchestration suivantes :

- (R1). On commence la généralisation cartographique par les espaces urbains.
- (R2). Si le type d'espace précédent est urbain alors on généralise le réseau routier.
- (R3). Si le type d'espace précédent est le réseau routier alors on généralise les espaces ruraux.
- (R4). Durant l'étape de sélection, si le type d'espace précédent est le réseau routier alors on généralise le réseau ferré.
- (R5). Durant l'étape de généralisation cartographique, si le type d'espace précédent est un espace de montagne, on passe à l'étape suivante.

Dans notre exemple, le fait, c'est-à-dire la position dans le processus CollaGen, est le suivant : durant l'étape de généralisation cartographique, le système vient de généraliser les graphes de flexibilité par un processus spécialisant le modèle des Beams (Bader et al, 2005), sachant que dans l'ontologie, le concept graphe de flexibilité est un réseau routier. La distance de ce fait est infinie avec les règles R1 (car le fait n'est pas le début de l'étape de généralisation cartographique), R2, R5 (car l'espace précédent de la prémisse ne correspond pas, le graphe de flexibilité n'étant ni un espace urbain, ni un espace de montagne) et R4 (car l'étape courante n'est pas la même bien que la prémisse corresponde). Par contre la distance avec la règle R3 est de 2 (1 pour l'étape de la règle non précisée et 1 pour la distance dans l'ontologie entre graphe de flexibilité et réseau routier). La règle R3 est donc celle qui est choisie par le moteur d'inférence. Le prochain espace à généraliser est donc l'espace rural. Ensuite, une fois les espaces ruraux généralisés, le moteur est de nouveau interrogé avec 'espace rural' comme espace précédent mais il ne trouve aucune règle qui correspond. Il cherche donc la règle terminant l'étape courante et trouve la règle R5. Et par chaînage arrière, le moteur infère que le prochain type d'espace à généraliser est 'espace de montagne' (en supposant qu'il ne reste plus d'autre espace clé à généraliser à cette étape).

Pour aller plus loin dans la conception d'une stratégie d'activation quand aucune règle n'est disponible, (Meurisse, 2001) propose dans le cadre de systèmes multi-agents, un modèle pour décrire les dépendances entre agents (tel agent doit être déclenché avant tel autre agent). Dans notre cas, une perspective serait de traduire l'ensemble des règles d'orchestration sous la forme de

dépendances pour que le choix d'activation fait en fonction des contraintes ne soit pas en contradiction avec les dépendances (règles d'orchestration).

C.15.4. Ordonner les instances d'espaces géographiques

Une fois que le composant a choisi le type d'espace géographique à généraliser à un moment donné, il n'a pas encore choisi complètement quel sera le prochain espace géographique à généraliser car il faut d'abord trier les différentes instances de ce type d'espace s'il y en a plusieurs (une seule instance pour les espaces thématiques).

Dans le cas général, le problème se rapproche du choix d'un type d'espace en l'absence de règle présentée dans la partie précédente et peut donc se ramener à un problème de stratégie d'activation comme dans un système multi-agent. (Meurisse, 2001) démontre que ce problème d'ordonnement des activations ne présente pas les mêmes difficultés quand les agents dépendent les uns des autres ou quand ils sont indépendants, ce qui est un cas beaucoup plus simple pour lequel une activation aléatoire est idéale. Dans le cas de l'ordonnement des espaces géographiques, la dépendance peut se traduire sous la forme d'effet de la généralisation d'une des instances sur les autres instances. La Figure C-76 montre deux exemples d'espaces géographiques dont les instances n'ont pas les mêmes caractéristiques de ce point de vue : les espaces urbains sont très séparés les uns des autres et la généralisation d'une des instances n'a aucune incidence sur la généralisation des autres (on peut considérer ces espaces comme indépendants) ; les espaces ruraux, découpés par le réseau routier ont de nombreuses instances voisines et la généralisation d'une instance peut avoir un impact significatif sur les autres, par exemple en déplaçant la route servant de frontière (on peut considérer ces espaces comme dépendants les uns des autres).

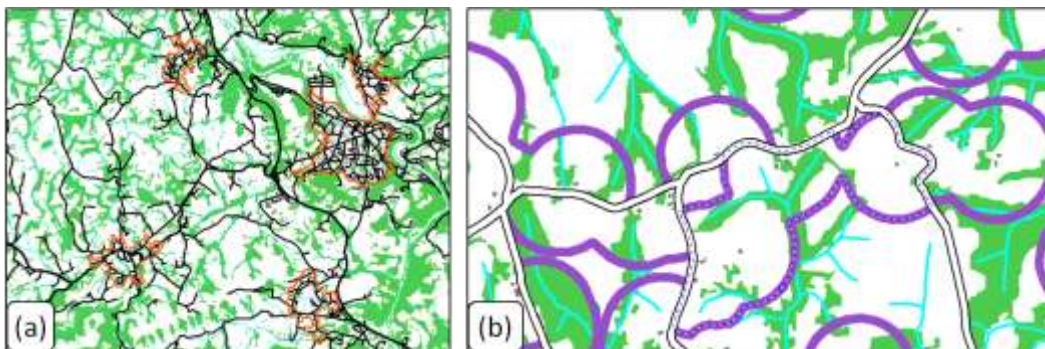


Figure C-76. Les espaces urbains (a) sont indépendants car la généralisation ne peut en affecter un autre alors que les espaces ruraux (b) sont dépendants de leurs voisins.

Nous avons défini en B.4.2 le voisinage d'un espace géographique afin de modéliser l'étendue de l'impact que pouvait avoir la généralisation de l'espace sur les entités voisines. Nous proposons donc d'utiliser ces voisinages pour mesurer le degré d'interdépendance des instances d'un type d'espace lorsque celui-ci est choisi par le composant d'orchestration. Donc, si plus de 10% des instances possèdent au moins un voisin du même type par intersection ou adjacence, les instances sont jugées interdépendantes.

A partir de cette mesure d'interdépendance, l'ordonnement des instances peut être réalisé. Si les instances sont mesurées comme indépendantes, l'ordonnement sera fait aléatoirement comme

suggéré par (Meurisse, 2001). Par contre, si les instances sont mesurées comme interdépendantes, l'ordonnement sera fait en fonction du nombre de moniteurs de contrainte très peu satisfaits contenus dans les instances, car il vaut mieux résoudre d'abord les conflits les plus importants quitte à gêner légèrement les voisins si l'on souhaite une réussite de la généralisation sur toutes les instances. Nous ne proposons pas de comparaison de différentes méthodes par manque de temps, mais nous sommes conscients qu'il faudrait le faire pour valider complètement ce choix.

En conclusion, le composant d'orchestration remplit trois fonctions pour permettre le choix du prochain espace géographique à généraliser :

- **Il gère et stocke les états intermédiaires de tous les objets géographiques généralisés après chaque généralisation d'un espace géographique.**
- **Il choisit le prochain type d'espace à généraliser en réalisant une inférence logique sur les règles formelles d'orchestration. En l'absence de règle correspondant à la position courante dans le processus, un choix est fait, fondé sur la satisfaction des moniteurs de contraintes.**
- **Il ordonne enfin les instances de l'espace géographique choisi en fonction de la satisfaction des contraintes ou de manière aléatoire en fonction de l'interdépendance des instances, mesurée grâce au voisinage des espaces.**

C.16. COMPOSANT DE GESTION DES EFFETS DE BORD

Le dernier des composants de CollaGen que nous présentons est celui qui est utilisé en dernier dans la dynamique d'une généralisation collaborative : le composant de gestion des effets de bord (Figure C-77). L'application d'un processus de généralisation uniquement à l'intérieur d'un espace géographique, qu'il soit territorial ou thématique, peut générer de nouveaux conflits à l'extérieur de cet espace avec des objets extérieurs, donc non pris en compte par le processus. Nous allons d'abord proposer une taxonomie des effets de bord possibles avec un système comme CollaGen. Puis nous présentons un modèle de contraintes de cohérence permettant d'identifier les effets de bord et la méthode de détection fondée sur ces contraintes de cohérence. Nous présentons enfin comment nous proposons de corriger les effets de bord, une fois détectés.

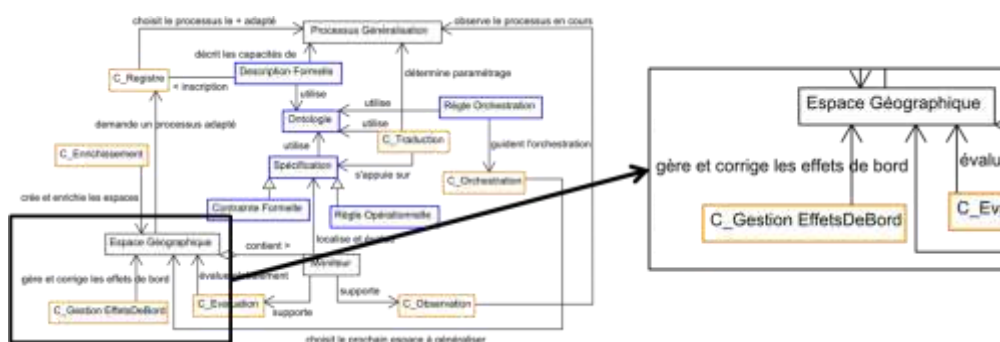


Figure C-77. Le composant gère les effets de bord engendrés par la généralisation d'un espace géographique.

C.16.1. Taxonomie des effets de bord

La généralisation cartographique est souvent comme un processus holistique (Bundy et al, 1991), c'est-à-dire un processus dont tous les éléments doivent être pris en compte pour prendre une décision, même locale. Il est donc évident que ne généraliser qu'une partie seulement des données, qu'elle soit délimitée géométriquement ou thématiquement, comme nous le proposons dans CollaGen, ne peut que générer des problèmes que nous appelons effets de bord autour des points d'interaction entre les entités prises en compte et les entités non prises en compte. La taxonomie que nous proposons dans cette partie, récapitulée en Figure C-82, est issue de tests d'application de processus de généralisation sur différentes situations d'espaces géographiques ainsi que de notre expérience dans la généralisation de cartes topographiques. Son extension à la généralisation d'autres types de cartes (cartes d'occupation du sol, cartes touristiques) nécessite peut-être l'ajout de types spécifiques à ces types de carte. Nous distinguons quatre grands types d'effets de bord dans le cadre de la généralisation collaborative : les effets d'*adhérence*, les effets d'*instabilité d'états*, les effets d'*ignorance* et les effets d'harmonie générale.

Les effets d'*adhérence* sont dus à la présence d'objets géographiques dans l'adhérence de l'espace géographique généralisé (voir définition précise p. 62). En topologie, l'adhérence d'un ensemble est le plus petit ensemble fermé contenant l'ensemble des éléments de l'ensemble. Dans notre cas, l'adhérence d'un espace est un polygone qui englobe l'ensemble des composants de l'espace que l'on peut voir comme un buffer autour des espaces territoriaux par exemple. Nous distinguons deux types d'effets d'adhérence, les effets d'*adhérence thématique* et les effets d'*adhérence géométrique*.



Figure C-78. (a) un effet de bord de type adhérence thématique : le cours d'eau ne passe plus au milieu de la forêt et le voisinage thématique fonctionnel est perdu. (b) effet de bord de type adhérence géométrique : le cours d'eau coulant le long de la route passe sous la route suite à sa généralisation.

Les effets d'*adhérence thématique* proviennent directement des voisinages thématiques fonctionnels que nous avons définis en B.4.2 : deux thèmes de données peuvent partager un voisinage fonctionnel, leurs objets partagent souvent une relation topologique (comme l'adjacence). Par exemple, les stations services sont le long des routes, les limites administratives partagent souvent la même géométrie que les routes et les cours d'eau. La Figure C-78 montre une forêt qui entoure un cours d'eau et la généralisation de ce cours d'eau entraîne par adhérence thématique la perte de cette relation.

Les effets d'*adhérence géométrique* correspondent aux effets de bord sur les objets situés dans l'adhérence de l'espace généralisé sans être présents de manière fonctionnelle et donc de manière moins prévisible. Si nous considérons, par défaut, que le plus petit fermé contenant l'ensemble des composants d'un espace géographique (son adhérence) est l'union des zones tampons des voisinages de l'espace, l'adhérence peut contenir des objets non pris en compte par le processus

généralisant l'espace, avec lesquels des conflits peuvent apparaître. La Figure C-78 montre un exemple dans lequel la généralisation des routes qui sont éloignées les unes des autres par les Beams entraîne un conflit avec un cours d'eau présent dans l'adhérence des routes.

Les effets d'*instabilité d'états* sont spécifiques à un voisinage de type intersection entre deux espaces, une fois que les deux espaces ont été généralisés. Ce sont des effets qui traduisent des décisions incohérentes sur un même objet des deux processus de généralisation qui ont traité cet objet. La particularité de ces effets de bord est que leur correction n'est pas toujours réclamée car ils peuvent déjà être issus d'un compromis et il peut être difficile de trouver une meilleure solution que cette incohérence. Nous distinguons deux types d'effets d'instabilité, les effets d'*opérations opposées* et les effets de *rechute de satisfaction*.

Les effets d'*opérations opposées* correspondent à une situation dans laquelle une opération (grossissement, rotation, déplacement par exemple) a été appliquée à un objet par le premier processus alors que le deuxième processus a appliqué à l'objet une opération à effet opposé (déplacement ou rotation dans le sens contraire, réduction après un grossissement par exemple). La Figure C-79 montre un exemple de déplacement par la généralisation de la zone rurale avec un déplacement contraire par le processus généralisant l'espace de montagne (pour maintenir les relations de présence des bâtiments sur un sommet) dans un deuxième temps.

Les effets de *rechute de satisfaction* correspondent à des moniteurs de contrainte dont la satisfaction augmente avec la première généralisation mais diminue après la deuxième. Même si la rechute est issue d'un compromis, considérer ces cas comme des effets de bord permet de valider ce compromis avec un contexte plus large que celui du deuxième espace géographique concerné. La Figure C-79 montre ainsi un exemple dans lequel le processus traitant l'espace de montagne déforme les courbes de niveau et diminue un peu la taille du bâtiment faisant baisser la satisfaction du moniteur de contrainte d'aire minimum du bâtiment. Cette solution est donc incohérente avec la précédente et il faut s'assurer que le compromis ayant abouti à cette solution est bon (par le processus de correction d'effets de bord présenté plus loin dans cette partie).

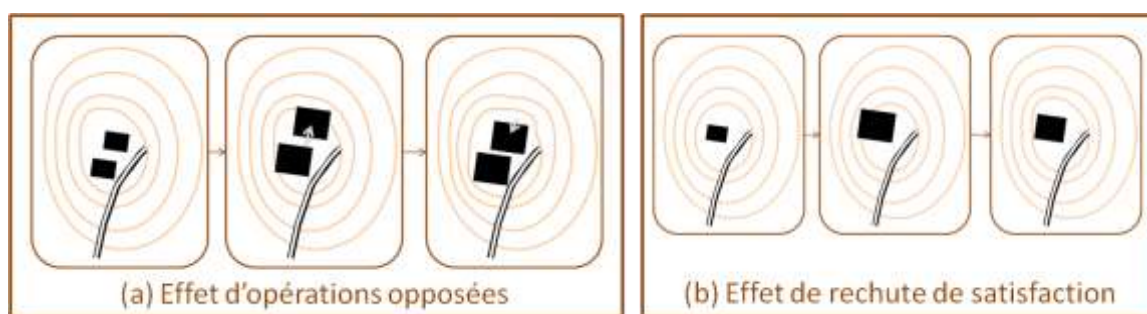


Figure C-79. Exemples d'effets d'incohérence dans une zone rurale et de montagne. (a) un effet de bord de type opérations opposées : le bâtiment est déplacé vers le haut de la figure par le processus de l'espace rural et vers le bas par le processus de l'espace de montagne. (b) un effet de bord de type rechute de satisfaction : le bâtiment est grossi de manière satisfaisante par le processus de la zone rurale mais rétréci sous le seuil de satisfaction par le processus de l'espace de montagne.

Les effets d'*ignorance* correspondent au fait qu'un processus de généralisation est aveugle aux objets qu'il ne traite pas, c'est-à-dire les objets non contenus dans l'espace, et peut donc créer des conflits avec eux. Les effets d'ignorance concernent principalement des relations géographiques entre un objet de l'espace généralisé et un objet extérieur à cet espace. Nous distinguons trois types

d'effets d'ignorance, les effets de *construction de relation*, les effets de *destruction de relation* et les effets de *baisse de qualité de relation*.

Les effets de *construction de relation* correspondent à la création d'une relation géographique qui n'existait pas initialement entre un des objets de l'espace et un objet extérieur. Ce type de relation entraîne de fausses interprétations des données de la carte et est donc à éviter le plus possible. L'utilisateur ne souhaite pas par exemple faire apparaître d'alignement de bâtiments quand il n'y en a pas. La Figure C-80 montre un exemple de deux espaces ruraux séparés par une route. Initialement, il n'existe pas de relation de symétrie par rapport à la route entre deux bâtiments, relation utile lorsqu'on souhaite se repérer sur une carte. La généralisation, en déplaçant les bâtiments de manière indépendante en replace deux en position symétrique, créant une relation inexistante dans les données initiales et surtout sur le terrain.

Les effets de *destruction de relation* sont plus simples à conceptualiser, il s'agit de la disparition d'une relation géographique présente dans les données initiales et sur le terrain entre un des objets de l'espace et un objet extérieur. La Figure C-80 montre un autre exemple de deux espaces ruraux séparés par une route puis généralisés par des processus différents (CartACom en bas, Moindres Carrés en haut). La relation de symétrie des bâtiments par rapport à la route est perdue après la généralisation.

Enfin, les effets de *baisse de qualité de relation* ressemblent aux effets de destruction mais cette fois la relation n'est pas complètement détruite mais simplement détériorée. La relation est toujours présente dans les données mais moins nette donc plus difficile à interpréter alors que la généralisation a plutôt tendance à caricaturer ces relations pour les mettre en valeur. La Figure C-80 montre un exemple de deux espaces ruraux séparés par un cours d'eau et généralisés par des processus différents. La relation de différenciation de taille entre le très petit bâtiment et le gros bâtiment est largement atténuée par la généralisation indépendante. Un processus pouvant tenir compte de cette relation aurait moins grossi le petit bâtiment et plus grossi le gros.

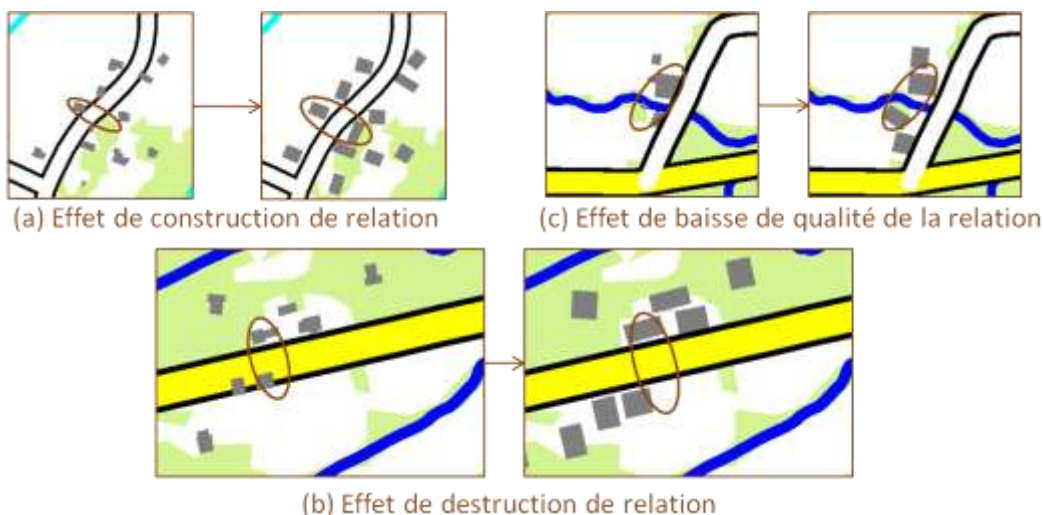


Figure C-80. (a) effet de bord de type construction de relation : la généralisation de chacun des espaces ruraux séparés par la route crée une relation de symétrie par rapport à la route. (b) effet de bord de type destruction de relation : la relation de symétrie des bâtiments par rapport à la route est perdue après la généralisation des deux espaces ruraux. (c) effet de bord de baisse de qualité de relation : la relation de différenciation des tailles entre le tout petit bâtiment et le gros bâtiment est fortement réduite après la généralisation des deux espaces ruraux séparés par la rivière.

Enfin, les effets d'*harmonie générale* se placent à un niveau plutôt macro alors que les effets précédents étaient plutôt d'un niveau micro. Ces effets concernent l'impact de la généralisation sur l'équilibre et l'harmonie des espaces géographiques entre eux. Si deux espaces géographiques, qu'ils soient du même type (deux espaces urbains) ou d'un type différent (un espace périurbain et un espace rural), qu'ils soient voisins ou non, peuvent se ressembler par leurs caractéristiques (ils présentent la même densité de bâtiments par exemple), alors la généralisation de l'un ou de l'autre ne doit pas avoir d'impact sur cette ressemblance. De même, si deux espaces sont très différents, cette différence structure la géographie de la zone et ne doit pas être atténuée par la généralisation de ces espaces. La Figure C-81 illustre ce type d'effet de bord avec deux espaces ruraux se ressemblant avant généralisation (densité importante en bâtiments le long des routes) et dont la ressemblance est atténuée après généralisation par deux processus différents.



Figure C-81. Illustration d'un effet de bord de type harmonie générale. Avant généralisation, les deux espaces se ressemblent mais une fois leur généralisation (l'un par AGENT et l'autre par CartACom), cette ressemblance est atténuée par la baisse de densité de bâtiment dans l'espace (a) généralisé par AGENT.

La Figure C-82 récapitule cette taxonomie des effets de bord que nous proposons. Il faut noter une fois encore que cette taxonomie s'attache principalement aux effets de bord identifiés dans notre cas d'étude et que son extension à d'autres cas d'études pourrait faire apparaître d'autres types d'effets de bord.

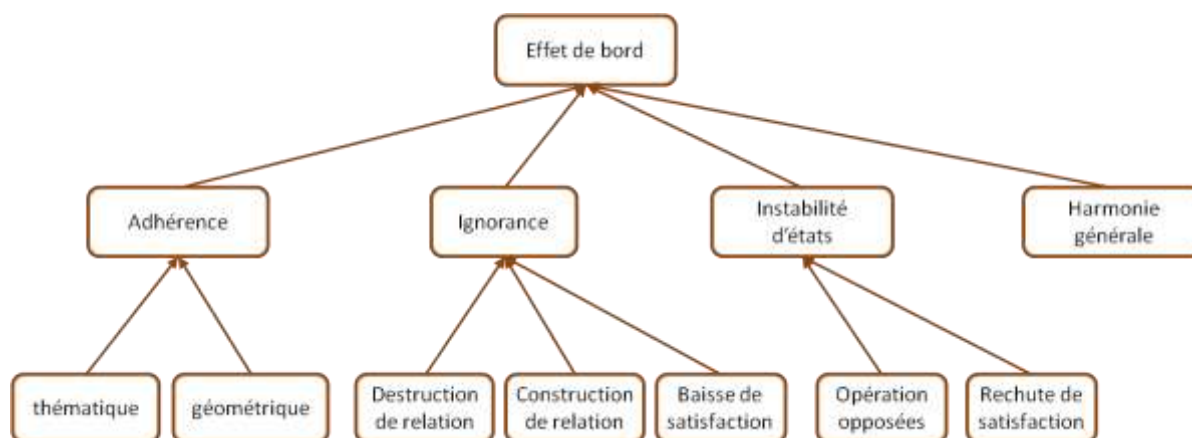


Figure C-82. Taxonomie des effets de bord dans le cadre de la généralisation cartographique.

C.16.2. Un Modèle de contraintes de cohérence

Après avoir identifié quels types d'effets de bord pouvaient se produire dans une généralisation collaborative, nous proposons dans cette partie un modèle pour repérer ces effets de bord lors de la généralisation d'un espace géographique dans CollaGen. En suivant le paradigme de représentation des conflits par contrainte et moniteurs, nous proposons d'utiliser des *moniteurs de cohérence* pour repérer et évaluer les effets de bord de la taxonomie. Un moniteur de cohérence a les mêmes caractéristiques qu'un moniteur de contrainte avec une satisfaction, une valeur initiale et une valeur courante, sans valeur but car le but est toujours d'éviter les effets de bord. Nous proposons d'utiliser un moniteur pour gérer les effets de bord d'un espace géographique dit central par rapport à un espace voisin (les moniteurs sont utilisés après sa (ou ses) généralisation(s)). Un moniteur concerne alors un objet central et un objet voisin ou un simple objet en partage. Nous définissons cinq types de contraintes de cohérence pour couvrir les sept sous-types de la taxonomie : les contraintes *fonctionnelles*, *inter-espace*, de *construction de relation*, d'*incohérence* et d'*opération cohérente* et nous définissons un type de moniteur pour chacune de ces contraintes de cohérence.

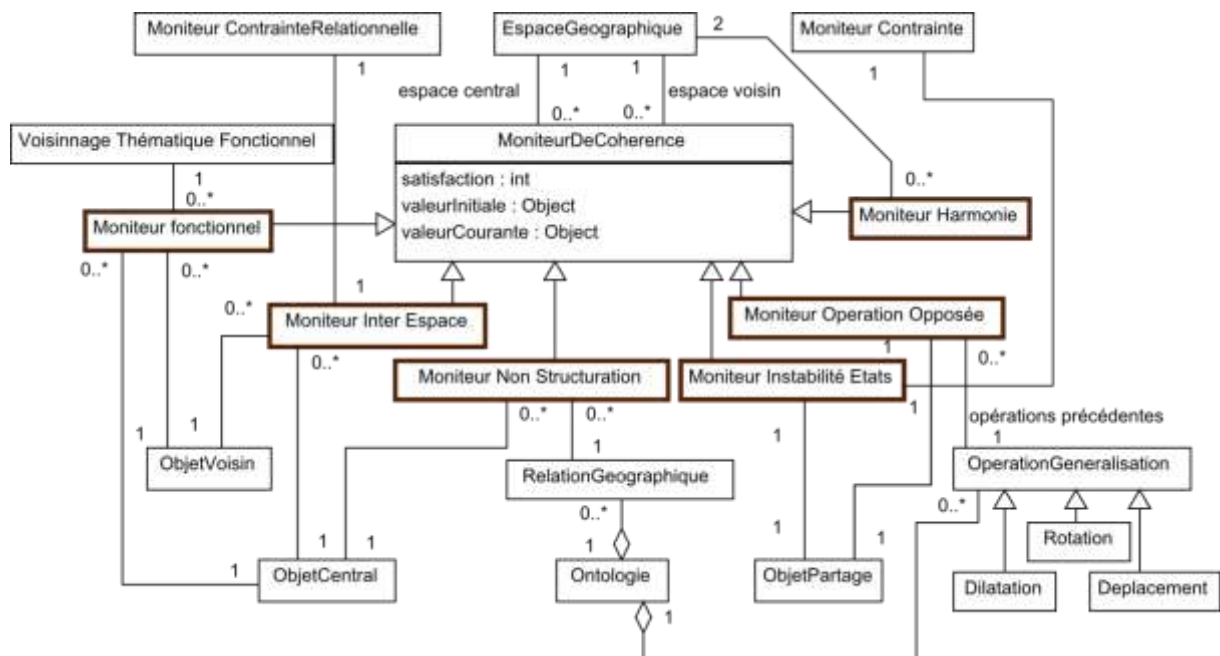


Figure C-83. Diagramme de classe UML du modèle de moniteurs de contrainte de cohérence permettant de vérifier les effets de bord.

Un moniteur *fonctionnel* permet de gérer les effets de bord de type *adhérence thématique* de notre taxonomie. Un moniteur fonctionnel est lié à un voisinage thématique fonctionnel précis, par exemple, le voisinage entre cours d'eau et forêts. Nous avons vu que les voisinages fonctionnels étaient décrits dans l'ontologie. Le type de relation topologique entre des instances des deux thèmes (la forêt entoure le cours d'eau dans l'exemple) doit également être décrit dans l'ontologie pour que le moniteur puisse vérifier si un tel effet de bord a lieu. Il existe un moniteur fonctionnel pour chaque voisinage thématique existant et pour chaque couple (objet central, objet voisin) vérifiant la relation topologique du voisinage.

Un moniteur *inter-espace* permet de gérer tous les effets de bord mettant en jeu une relation existante entre un objet central et un objet voisin. Cela concerne les effets de bord de type

destruction de relation, baisse de qualité et une partie des adhérences géométriques. Un objet moniteur inter-espace est lié à un moniteur de contrainte relationnelle dont les valeurs courante et de satisfaction permettent de dériver les siennes: si le moniteur de contrainte relationnelle est satisfait, le moniteur inter-espace également et il n'y a pas d'effet de bord pour ce moniteur. Un moniteur inter-espace est lié à un objet central et un objet voisin, qui sont les deux objets concernés par le moniteur de contrainte relationnelle. Par exemple, il existe une relation de proximité entre un bâtiment central et un cours d'eau voisin, donc, un moniteur pour gérer la contrainte de proximité minimum entre bâtiments et symbole de cours d'eau. Tant que le moniteur de cette contrainte relationnelle est satisfait, le moniteur inter-espace aussi, même s'ils se sont rapprochés l'un de l'autre.

Un moniteur de *construction de relation* permet de gérer les effets de bord de type construction de fausse relation à partir d'un objet central et de voisins potentiels ainsi que la partie restante des effets de bord de type adhérence géométrique. En effet, les conflits générés dans l'adhérence géométrique d'un espace se traduisent dans CollaGen par la création de relations de proximité dont la distance minimum est trop faible. Un moniteur de non structuration est donc lié à une relation géographique de l'ontologie et un objet central situé à proximité du bord de l'espace. Il existe donc pour chacun de ces objets centraux au bord autant de moniteurs de non structuration que de relations géographiques auquel ce type d'objet peut prendre part (une route ne peut pas prendre part à une relation de différenciation entre bâtiments). Nous avons vu dans les parties précédentes que le composant de traduction permettait de rattacher chaque relation géographique avec des algorithmes de la boîte à outils (composant d'enrichissement) qui permettent de déterminer si une telle relation existait entre un objet et ses voisins. Ces algorithmes sont donc utilisés pour déterminer la valeur courante et la satisfaction des moniteurs de non structuration (s'il n'y a pas de nouvelle relation, le moniteur est très satisfait, s'il y en a plusieurs du même type, le moniteur est très insatisfait).

Un moniteur d'*instabilité d'états* permet de gérer les effets de bord du type *rechute de satisfaction* à partir d'un objet partagé, c'est-à-dire un objet appartenant à plusieurs espaces géographiques. Un moniteur d'incohérence est donc lié à un moniteur de contrainte quelconque attaché à l'objet partagé pour vérifier si la contrainte gérée par le moniteur connaît une rechute de satisfaction durant les multiples généralisations de l'objet. Comme dans le cas du moniteur inter-espace, le moniteur d'incohérence se base sur les satisfactions du moniteur de contrainte pour déterminer la sienne. Il existe, pour chaque objet partagé par plusieurs espaces, un seul moniteur d'incohérence qui gère l'ensemble des moniteurs de contrainte attachés à l'objet. Ainsi, si un seul moniteur de contrainte rechute alors que les autres continuent de progresser, le moniteur d'incohérence reste satisfait car il s'agit simplement d'une dégradation due à la nature sur-contrainte du problème de généralisation.

Un moniteur d'*opérations opposées* permet de gérer les effets de bord de type *opérations opposées* sur un objet partagé. Un moniteur d'opération cohérente conserve, durant les généralisations successives de l'objet partagé, le compte des opérations géométriques qui lui sont appliquées sous la forme d'un objet opération de généralisation. Un tel objet est relié à l'opération de généralisation correspondant dans la taxonomie des opérateurs intégrée à l'ontologie mais aussi à la quantité de modification appliquée avec l'opération. Nous proposons de nous limiter dans ce cas à trois types d'opérations, la rotation, la dilatation et le déplacement. Dans le cas de la rotation, la quantité est

l'angle de rotation, dans le cas de la dilatation, la quantité est le ratio aire d'arrivée par aire de départ et dans le cas du déplacement, la quantité est le vecteur de déplacement. CollaGen n'ayant aucune prise sur le contenu des processus qu'il utilise pour généralisation, le composant doit mesurer après coup pour déterminer quelle(s) opération(s) a été effectuée et avec quelle quantité en comparant les objets avant et après la généralisation.

Enfin, un moniteur d'*harmonie* permet de gérer les effets de bord de type *harmonie générale* entre deux espaces dont les caractéristiques sont, soit très ressemblantes, soit très dissemblables. Le fonctionnement de ces moniteurs d'harmonie est calqué sur le modèle des contraintes macro de (Ruas, 1999), appliqué sur la population des espaces géographiques. Si l'espace généralisé est semblable et dissemblable à un autre espace géographique, le moniteur est satisfait si ces ressemblances et dissemblances n'ont pas été atténuées par la généralisation.

C.16.3. Détection des effets de bord

La détection des effets de bord indésirables, un des deux objectifs du composant de gestion des effets de bord (avec leur correction) est fondée sur la satisfaction des contraintes de cohérence que nous avons introduites dans la partie précédente. Nous allons voir dans cette partie quand et comment sont créés les moniteurs de cohérence d'un espace géographique, comment le composant interprète leur satisfaction pour identifier et quantifier des effets de bord et comment l'utilisateur de CollaGen peut jouer sur des paramètres pour faire varier les effets de bord détectés.

La création des moniteurs de cohérence est la première étape de la détection des effets de bord et elle s'effectue avant toute généralisation lors de la création de chacun des espaces géographiques. Ainsi, les moniteurs de contrainte peuvent comparer les états généralisés avec l'état initial. Pour chacun des cinq types de moniteurs de cohérence, le composant crée le nombre adapté de moniteurs par objet : par exemple, pour un objet central, il crée un moniteur inter-espace par relation géographique avec un objet voisin et il crée autant de moniteurs de non structuration que de types de relation géographique pris en compte moins les types pour lesquels l'objet fait déjà partie d'une relation (ce choix est toutefois à valider sur un jeu de données très large pour s'assurer que le composant ne fait pas trop de vérifications) ; pour un objet partagé, le composant crée un unique moniteur d'incohérence et un unique moniteur d'opération cohérente.

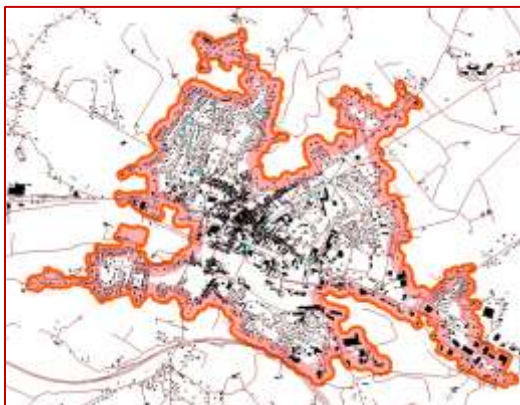


Figure C-84. *Etendue de la zone (fond coloré) dans laquelle sont construits les moniteurs de cohérence pour un espace urbain.*

Cependant tous les objets d'un espace ne sont pas concernés par la création de moniteurs de cohérence. Seuls ceux qui sont potentiellement concernés par des effets de bord créés par la généralisation de cet espace doivent être surveillés pour optimiser le traitement. Les objets potentiellement concernés sont ceux qui se situent près du bord de l'espace. Nous définissons ainsi une bande de terrain, à l'intérieur des espaces géographiques, dans laquelle les objets sont concernés par les moniteurs de cohérence, la largeur de cette bande dépendant de l'échelle finale de la carte (une distance de 2 mm carte est utilisée car l'expérience prouve que les objets ne sont pas déplacés au-delà de cette distance). La Figure C-84 montre un exemple de cette zone sujette aux effets de bord pour un espace urbain. Le composant de gestion des effets de bord crée donc l'ensemble de ces moniteurs selon le modèle présenté en Figure C-83, calcule leur valeur initiale et les stocke pour une utilisation après chaque généralisation de l'espace. Certains moniteurs sont toutefois 'symétriques' et ne sont pas créés en double quand ils concernent deux espaces géographiques : les moniteurs concernant des objets partagés ne sont pas créés pour chaque espace partageant l'objet et les moniteurs inter-espace sont également créés une seule fois car le calcul satisfaction ne varie pas si l'objet central et l'objet voisin sont inversés (pour une relation de parallélisme par exemple, ils sont chacun parallèles à l'autre).

Nous avons vu comment un moniteur de cohérence est construit dans CollaGen et voyons maintenant comment le composant de gestion des effets de bord les utilise pour identifier et quantifier des effets de bord après la généralisation d'un espace géographique. Nous pouvons distinguer deux comportements différents pour identifier des effets de bord : les moniteurs liés à des objets centraux sont utilisés après chaque généralisation de l'espace alors que les moniteurs liés à des objets partagés ne sont pris en compte qu'après plusieurs généralisations de l'objet (le nombre de processus de généralisation appliqués à l'objet est stocké sur les moniteurs de cohérence en question). La phase d'identification des effets de bord prend donc en compte tout ou partie des moniteurs de cohérence associés à l'espace qui vient d'être généralisé et lit leur satisfaction : un moniteur de cohérence non satisfait devient un effet de bord. Contrairement à la phase d'observation qui ne propose d'intervenir que de manière ponctuelle quand le nombre de conflits identifiés à un endroit est important, tous les moniteurs non satisfaits sont ici pris en compte. Toutefois le même principe de classification des moniteurs par proximité est utilisé en réalisant une triangulation des positions des moniteurs de cohérence (Figure C-85). Cela permet aux effets de bord proches comme dans la Figure C-85b d'être traités de manière groupée pour gagner en efficacité. Mais contrairement à l'observation en ligne, même les groupes singletons de moniteurs sont considérés comme des effets de bord. Les effets de bord sont ainsi quantifiés en termes de taille de groupes et de valeur de satisfaction (les moniteurs très peu satisfaits correspondent à des effets de bord plus graves que des moniteurs moyennement satisfaits). Cette quantification sera utilisée pour déterminer le processus de correction adéquat.

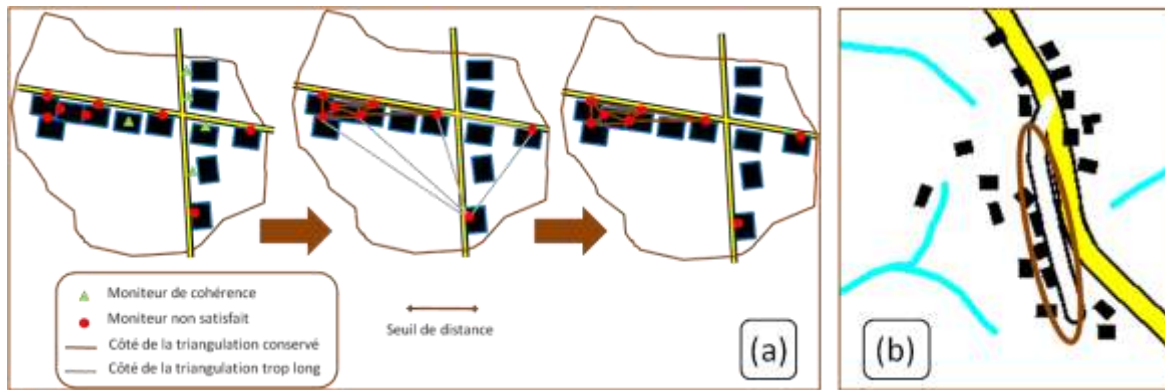


Figure C-85. (a) Reprise du système de classification des moniteurs par triangulation de leur position utilisé par le composant d'observation pour les moniteurs de cohérence. (b) La généralisation des routes (écartées) entraîne plusieurs effets de bord proches que la classification par proximité des moniteurs permet de traiter ensemble.

L'utilisateur de CollaGen peut paramétrer l'étendue de la recherche des effets de bord en la limitant ou pas à certaines relations notamment. Pour chacun des moniteurs de cohérence, il peut choisir les effets de bord qu'il veut voir corriger selon le même principe que la saisie des contraintes formelles, il traduit ses spécifications. Nous pouvons considérer qu'il s'agit de contraintes du point de vue de l'utilisateur : il contraint le résultat d'éviter certains effets de bord. Pour les moniteurs fonctionnels, l'utilisateur peut ainsi restreindre les effets de bord à certains des voisinages thématiques fonctionnels définis dans l'ontologie. Pour les moniteurs inter-espace comme pour les moniteurs de non structuration, l'utilisateur peut restreindre les effets à certaines des relations géographiques définies dans l'ontologie. Pour les moniteurs d'incohérence, l'utilisateur peut restreindre les effets à certaines des contraintes qu'il a définies par ailleurs et non toutes : il choisit, comme quand il définit ses contraintes, pour quelles relations les effets de bord sont vérifiés et corrigés (e.g. les relations de proximité entre bâtiment...). Enfin, pour les moniteurs d'opérations opposées, l'utilisateur peut choisir lesquelles parmi les trois opérations définies sont traitées par les moniteurs.

C.16.4. Correction des effets de bord

Nous avons vu dans la partie précédente comment étaient identifiés les effets de bord, à partir des moniteurs de cohérence et nous détaillons dans cette partie comment ces effets de bord peuvent être corrigés au mieux sans 'défaire' la généralisation. Nous proposons quatre méthodes de correction suivant les cas : la *diffusion*, la *déformation équilibrée*, l'*arbitrage* et la *relaxation de contrainte*.

La méthode que nous appelons *diffusion* consiste en une propagation de la généralisation qui a entraîné l'effet de bord sur les objets voisins en conflit. Afin d'éviter de propager la généralisation sur toute la carte, cette propagation est amortie à mesure qu'on s'éloigne du cœur de l'objet central (ou des objets centraux) responsable(s) de l'effet de bord. La diffusion est un problème qui a déjà été abordé en généralisation, notamment dans le cas de la généralisation des routes. En effet, certaines opérations de simplification ou de caricature ont pour résultat de bouger les bords des routes, et les reconnecter à l'ensemble du réseau nécessite une diffusion pour éviter de fortes déformations (Figure C-86a). Il existe plusieurs modèles pour réaliser des diffusions parmi lesquelles nous pouvons citer l'adaptation des 'snakes' (Burghardt et Meier, 1997 ; Bader et Barrault, 2000), la diffusion à vol

d'oiseau (Legrand et al, 2005), l'étirement élastique (Anders et al, 2007) ou l'adaptation de modèles de généralisation comme les Beams (Bader et al, 2005) et GAEL (Gaffuri, 2009) (Figure C-86b).



Figure C-86. (a) diffusion du réseau routier après déplacement de la route de droite suivant la flèche (Legrand et al, 2005). (b) diffusion des bâtiments par un processus basé sur le modèle GAEL après la généralisation des routes (Gaffuri, 2008).

La méthode que nous appelons *déformation équilibrée* consiste à déformer la situation causant l'effet de bord vers une position d'équilibre moyen entre la situation avant (sans effet de bord mais mal généralisée) et après généralisation (donc avec effet de bord mais bien généralisée du point de vue d'un espace géographique) (Figure C-87). Tous les modèles de généralisation cherchant une solution globale optimale peuvent être adaptés pour réaliser cette déformation équilibrée. Parmi, ces modèles de généralisation adaptables, nous pouvons citer les Beams en construisant un graphe reliant les objets comme dans (Bader et al, 2005), les Moindres Carrés (Harrie et Sarjakoski, 2002), GAEL (Gaffuri, 2008) en construisant une triangulation des objets à déformer ou encore le modèle par analogie avec une colonie de fourmis de (Richards et Ware, 2010). Si ces processus de déformations peuvent en plus prendre en compte la contrainte violée qui a conduit à l'effet de bord (proximité entre bâtiment dans la Figure C-87, l'effet de bord sur la contrainte de maintien des différences de taille n'est pas pris en compte), le résultat n'en est que meilleur. Comme pour la méthode de diffusion, plus la boîte à outils de CollaGen contient des processus capables de réaliser une déformation équilibrée, mieux les effets de bord peuvent être corrigés.

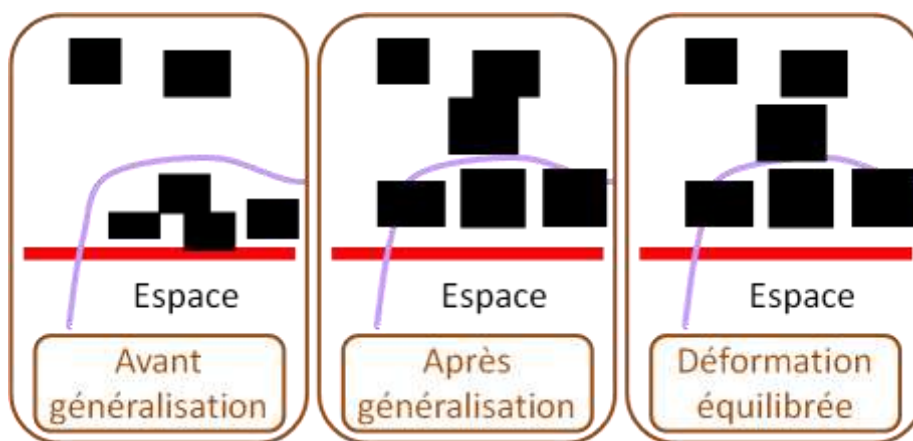


Figure C-87. Illustration du principe de la déformation équilibrée sur un effet de bord de type adhérence géométrique : le bâtiment est mis dans une position moyenne entre sa position initiale et sa position généralisée.

Le défaut principal de cette méthode est que, pour la généralisation cartographique, le 'tout ou rien' est souvent une meilleure solution qu'une moyenne équilibrée, mais cette méthode peut s'avérer efficace pour certains des effets de bord identifiés dans notre taxonomie.

Une autre méthode pour corriger les effets de bord est de s'inspirer des travaux pour résoudre les conflits à l'intérieur d'un objet meso (i.e. un groupe d'objets), qui traitent la résolution de conflits dans une zone assez réduite et sur-contrainte. (Ruas, 1999) propose notamment, dans le cas des objets meso, de réaliser des arbitrages si aucune situation ne convient parfaitement. A ce moment là, le composant de gestion des effets de bord adopte un comportement de *contrôleur* (Ruas, 2000) et détermine quelle situation garder : la situation initiale, la situation après la généralisation de l'espace, voire la situation après plusieurs généralisations dans le cas d'objets partagés par plusieurs espaces. Il s'agit d'une méthode à appliquer plutôt en dernier recours car elle ne fournira pas de résultat complètement satisfaisant.

La dernière méthode que nous proposons, la *relaxation d'une contrainte* est également issue de l'analogie avec le comportement des objets meso dans le contrôle de la généralisation de leur composant. Il s'agit, dans ce cas encore, d'adopter le comportement de contrôleur de (Ruas, 2000) et de déterminer une des contraintes violées (la moins importante *a priori*) et re-déterminer la satisfaction des moniteurs de cohérence en ne prenant plus en compte cette contrainte. Par exemple, dans un cas où des effets de bord sont des relations de proximité et d'orientation relative sujettes à des contraintes, la relaxation des contraintes sur la relation d'orientation relative (moins importante que la proximité minimum) peut permettre d'augmenter la satisfaction globale de la situation au-dessus du seuil de déclenchement des effets de bord. La situation n'est pas corrigée mais devient acceptable en modifiant nos critères d'acceptabilité. Relâcher une contrainte permet de trouver des solutions qui auraient été normalement refusées.

Afin de corriger les effets de bord, le composant dédié doit choisir laquelle de ces quatre méthodes (et lequel des outils associés) il doit utiliser pour chaque cas d'effet de bord. Le problème se rapproche de celui du choix de l'algorithme à appliquer pour généraliser un groupe d'objet ou celui du choix du processus à appliquer sur un espace. N'ayant pas approfondi ce sujet, nous nous contentons de proposer des solutions simples:

- la diffusion est choisie pour corriger les effets de bord issus des moniteurs fonctionnels et des moniteurs de non structuration liés à la relation de proximité. Le choix de la méthode de diffusion dépend alors de la nature des objets à diffuser (par exemple les 'snakes' pour diffuser des éléments de réseau comme des routes ou des cours d'eau).
- La déformation équilibrée est utilisée pour corriger les effets de bord issus des autres types de moniteurs. L'équilibre est plus ou moins pondéré entre la situation avant et après effets de bord : par exemple, pour la correction d'effets de bord du type destruction de relation, un poids plus fort va être mis sur la position 'avant' dans laquelle la relation existait.
- La relaxation d'une contrainte est choisie si la méthode de correction n'a pas amélioré significativement la situation. Dans certaines situations, une solution moyenne ou équilibrée est pire qu'une solution tranchée (plusieurs contraintes insatisfaites au lieu d'une seule par exemple) et il vaut alors mieux relâcher la contrainte la moins importante pour trouver une solution satisfaisante.
- L'arbitrage d'une des solutions est choisi si même la relaxation n'a pas amélioré significativement la situation. En effet, dans certaines situations, il n'y a pas de contrainte moins importante à relâcher, ou alors, n'en relâcher qu'une ne rend pas la situation globalement plus satisfaisante. Dans ce cas, nous proposons de trancher pour une des deux situations (avant ou après) comme solution de dernier recours.

En conclusion, la généralisation par CollaGen peut entraîner différents types d'effets de bord au voisinage des espaces géographiques généralisés : des effets d'adhérence, d'ignorance et d'incohérence. Ces effets de bord sont identifiés et quantifiés grâce à des moniteurs de cohérence stockés sur les espaces géographiques dont la satisfaction est calculée de manière différente suivant les effets de bord surveillés. Nous proposons quatre méthodes complémentaires pour corriger les effets de bord détectés : la diffusion, la déformation équilibrée, la relaxation de contrainte et l'arbitrage.

Cependant, nous n'avons pas testé de manière approfondie les propositions que nous faisons pour modéliser le composant de gestion d'effets de bord, contrairement aux autres composants de CollaGen. Il reste donc indispensable de valider ce modèle par des tests et éventuellement de le modifier pour gérer les effets de bord dans CollaGen.

C.17. BILAN

Dans ce chapitre, nous avons détaillé notre proposition pour la généralisation collaborative, le modèle CollaGen. Ce modèle est complexe car il remplit des fonctions très différentes par ses six composants principaux et les ressources que ces composants produisent et manipulent (Figure C-88). Voici les idées importantes à retenir de ce modèle :

- Un humain peut interagir avec CollaGen de quatre manières : en temps que modélisateur, développeur, fournisseur et utilisateur.
- Des connaissances formelles (une ontologie de la généralisation cartographique, des contraintes formelles, des règles d'orchestration, des descriptions de processus) servent de support aux interactions interopérables des six composants.
- CollaGen fonctionne de manière séquentielle, les espaces géographiques étant généralisés les uns après les autres.
- Le choix de l'espace géographique à généraliser est inféré à partir des règles d'orchestration.
- Le registre s'appuie sur l'adéquation entre la description formelle des processus et les conflits identifiés dans l'espace pour déterminer une liste des meilleurs processus à appliquer.
- Le composant de traduction permet de paramétrer automatiquement à partir des contraintes formelles les processus de généralisation disponibles dans CollaGen.
- Une observation en ligne de la généralisation permet de nous arrêter pour généraliser autrement une zone de conflit émergente puis reprendre la généralisation en cours.
- La généralisation d'un espace est évaluée de manière globale pour déterminer si elle est de bonne qualité ou si CollaGen doit essayer autre chose.
- Les effets de bord sont identifiés au voisinage des espaces géographiques par des moniteurs de cohérence et corrigés par des méthodes automatiques ad-hoc.
- Une boîte à outil contient tous les algorithmes utilisés par les composants : plus elle est riche, plus CollaGen a de possibilités pour réaliser des bonnes généralisations.

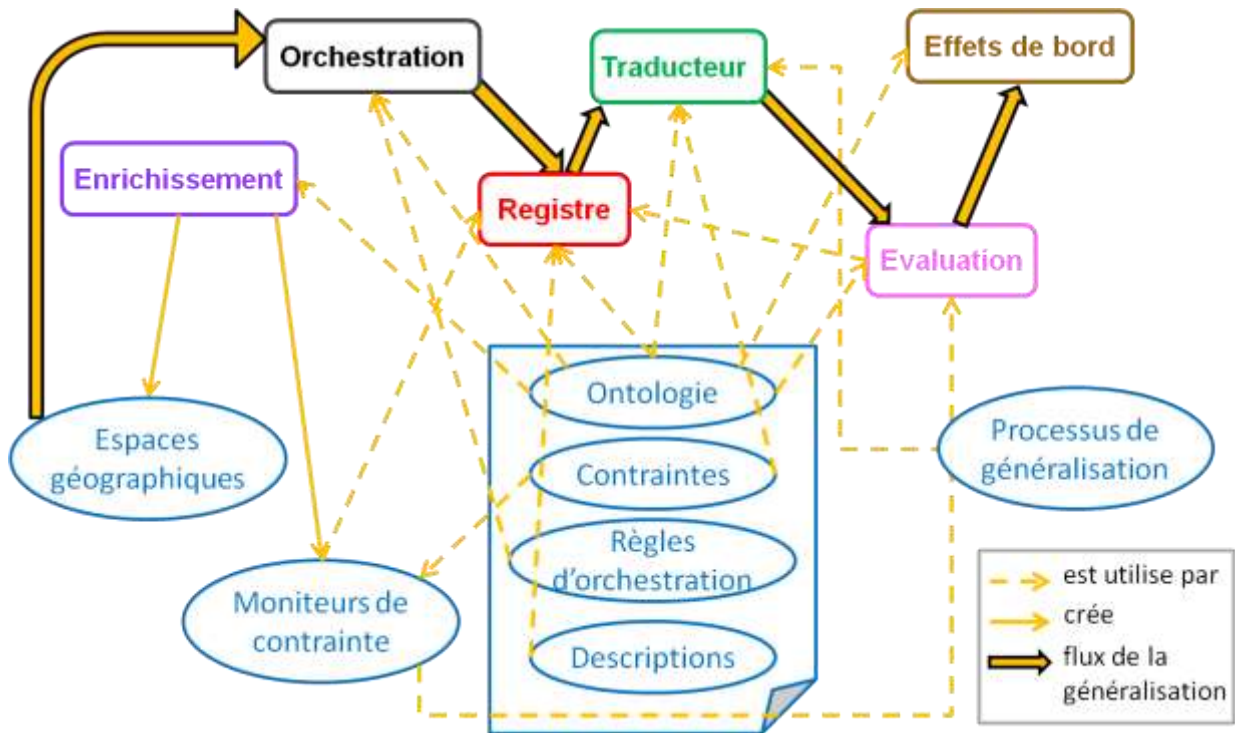


Figure C-88. Diagramme bilan des interactions entre composants et ressources dans CollaGen.

CHAPITRE D. MISE EN ŒUVRE DE COLLAGEN PAR UN SYSTEME MULTI-AGENTS

Dans le chapitre précédent, nous avons présenté le modèle CollaGen que nous proposons pour réaliser des généralisations collaboratives. Dans ce chapitre, nous allons faire un pas de plus vers le prototype de CollaGen tout en restant dans le domaine de la modélisation, en détaillant comment CollaGen peut être mis en œuvre comme un Système Multi-Agent. Dans une première partie, nous expliquons pourquoi nous considérons qu'une perspective SMA pour CollaGen est une évolution naturelle par rapport aux mécanismes décrits dans le chapitre précédent. Puis, nous détaillons quelles entités de CollaGen deviennent des agents dans cette perspective. Nous présentons ensuite la répartition des capacités des six composants entre les agents (introduits juste avant). Enfin, dans une dernière partie, nous discutons de la pertinence d'une telle approche du point de vue généralisation et SMA : nos agents sont-ils bien des agents et si oui, leur contribution est-elle significative ?

D.1. VERS UNE PERSPECTIVE SMA POUR COLLAGEN

Dans cette partie, nous expliquons pourquoi nous considérons naturel d'adopter une perspective SMA pour mettre en œuvre le modèle CollaGen. Il faut noter tout d'abord que, dans les chapitres précédents définissant la généralisation collaborative et CollaGen, nous avons beaucoup emprunté, par analogie, au domaine des systèmes multi-agents. Le modèle CollaGen possède donc déjà des caractéristiques provenant des SMA. La notion même de collaboration de processus de généralisation est issue d'une analogie avec la collaboration entre agents dans les SMA (B.1.2). Nous avons également emprunté des caractéristiques issues des SMA et de la norme 'Directory Facilitator' (FIPA, 2004) pour la modélisation du registre de CollaGen chargé de mettre en relation un espace géographique et un processus adapté à cet espace (C.13). Le module d'observation de CollaGen est également inspiré des modules d'observation dans les SMA (C.14). Enfin, notre proposition pour l'évaluation globale d'un espace généralisé repose sur une analogie avec la satisfaction globale d'une population d'agents (C.14). Ces nombreuses analogies nous incitent à nous questionner sur l'adaptation d'une modélisation SMA au problème de la généralisation collaborative.

Dans un premier temps, nous remarquons que les agents sont particulièrement adaptés à l'automatisation de la généralisation cartographique. En effet, le problème de l'automatisation de la généralisation cartographique a les caractéristiques suivantes :

- Il s'agit d'un problème qui s'exprime et se résout de manière **fortement distribuée** (Lamy et al, 1999). En effet, depuis les premiers modèles pas à pas locaux, la stratégie générale consiste à identifier une situation locale posant problème, à la traiter, puis, à passer à la suivante.
- Il s'agit d'un problème **holistique** dans lequel une vue plus générale est nécessaire à la résolution des situations locales (Bundy et al, 1995). Il est en effet indispensable d'analyser le contexte d'une situation (les relations géographiques, les voisinages, les structures meso...) pour établir la bonne stratégie de généralisation, ce que font, au moins en partie, la plupart des processus de généralisation.
- La généralisation se résout principalement en exprimant les spécifications de la carte sous la forme de **contraintes**. Mais elle n'est pas adaptée aux techniques classiques de CSP (constraint solving problem) utilisées en intelligence artificielle pour optimiser la résolution d'un problème contraint (Bellicha et al, 1995). Le coloriage d'un graphe planaire avec quatre couleurs sans utiliser deux fois la même couleur pour des faces adjacentes est un exemple classique de CSP. Dans de tels problèmes, on optimise les valeurs affectées à des variables pour satisfaire les contraintes (dans l'exemple du coloriage, les variables sont les faces du graphe et sa valeur affectée la couleur). Pour la généralisation, la valeur affectée aux variables serait la géométrie mais les transformations de géométries sont obtenues par des algorithmes dont on ne peut pas anticiper la valeur, donc, il est difficile de l'exprimer *a priori* par des équations. De plus, la nécessité d'éliminer des objets limite encore plus l'utilisation de techniques issues des CSP (Duchêne, 2004, p.48-50).
- Dans le cadre du paradigme pas à pas local adopté dans CollaGen, la généralisation cartographique propose un **environnement évolutif** dans le temps de la généralisation, car le fonctionnement pas à pas fait que les objets géographiques sont transformés petit à petit.
- Il existe des **heuristiques** pour décrire des stratégies locales efficaces de généralisation. (Ruas et Mackaness, 1997 ; Lamy et al, 1999) proposent ainsi des heuristiques pour guider efficacement la généralisation des zones urbaines de manière pas à pas locale. Ces heuristiques ont été reprises dans la conception du modèle AGENT.

Ces caractéristiques font de la généralisation cartographique un champ d'application idéal des SMA situés qui visent à la résolution de problèmes. En effet, les agents sont adaptés aux problèmes distribués, car ils agissent de manière autonome pour atteindre leur objectif individuel. De même, les capacités de perception de leur environnement immédiat et de communication des agents avec leurs voisins sont un bon moyen pour mettre en place un processus holistique tenant compte du contexte géographique : les prises de décisions peuvent tenir compte de l'avis des agents voisins et de la configuration locale des objets géographiques comme la disposition des autres objets autour de l'agent. De plus, les SMA sont un bon moyen pour tenter de résoudre des problèmes de satisfaction de contrainte qui ne correspondent pas à la conception classique d'un CSP, notamment lorsque des heuristiques existent pour guider les agents. Enfin, les capacités de raisonnement des agents pour prendre des décisions nécessitent des heuristiques pour obtenir des agents un comportement efficace et efficient. Modéliser la généralisation cartographique comme un SMA dans lequel les agents situés sont les objets géographiques qui cherchent à se généraliser, est donc assez naturel et s'est avéré très efficace (Duchêne et al, 2011). Du fait de son évolution dans le temps (les agents se généralisent petit à petit), les techniques de simulation dans les SMA peuvent aussi apporter à la modélisation de la généralisation cartographique par analogie. Par exemple, le modèle de simulation

multi-niveaux PADAWAN (Picault et al, 2010) pourrait nous aider à modéliser les liens entre un agent, l'environnement dans lequel il est situé et éventuellement l'environnement qu'il encapsule : nous pourrions alors modéliser un effet de bord (un bâtiment qui sort de l'environnement encapsulé par l'espace géographique) comme une molécule qui sort d'une cellule dans (Picault et al, 2010).

Outre que les agents se montrent particulièrement adaptés à la généralisation cartographique, notre approche de la généralisation collaborative dans CollaGen s'inspire des problématiques de la composition automatique de services Web et plus généralement du problème dit de 'matchmaking' dans la littérature anglo-saxonne c'est-à-dire d'appariement entre un besoin et un fournisseur de service correspondant à ce besoin (Figure D-1). Dans notre cas, le besoin est la généralisation d'un espace géographique donné et les services sont les processus de généralisation.



Figure D-1. Schéma classique du problème de 'matchmaking' ou d'appariement entre un demandeur et un fournisseur par service grâce à un intermédiaire.

Or, il s'avère que les systèmes multi-agents sont aussi particulièrement adaptés à la résolution de ce type de problème, que ce soit entre plusieurs agents ou entre un humain et des agents. (Decker et al, 1997) proposent des modèles d'agents intermédiaires ('middle-agents') pour le commerce sur internet, modèles qui sont étendus au problème de l'appariement entre agents logiciels dans (Sycara et al, 2002). La composition automatique de services Web est un sous problème de l'appariement qui est plus proche de notre problématique. Dans ce domaine également, (Sabouret, 2003) propose l'utilisation d'un modèle d'agent service web sémantique pour adapter les interactions du service avec un utilisateur humain d'Internet. (Charif et Sabouret, 2004 ; Charif et Sabouret, 2009) vont plus loin en proposant un modèle d'interactions entre agents service web, puis, un mécanisme de coordination pour permettre la chorégraphie des agents (prise de décision collégiale entre les services contrairement à une orchestration centralisée, voir B.1.1). Si nous transposons ceci à notre problème, la mise en correspondance d'un espace à généraliser avec un processus de généralisation adapté est un problème qui se prête bien à une modélisation SMA. Mais contrairement aux modèles SMA de généralisation cartographique, les agents ne sont pas, dans les approches présentées dans ce paragraphe, situés dans un environnement métrique comme notre base de données géographique : ce sont des agents dits logiciels.

Par ailleurs, bien que ce ne soit pas une justification scientifique, le laboratoire COGIT a adopté dans son approche de la généralisation cartographique, une approche pas à pas locale résolument orientée vers les SMA, initiée par (Ruas, 1999) et présentée dans (Duchêne et al, 2011), comme le prouvent les modèles AGENT (Ruas, 1999), CartACom (Duchêne, 2004) et GAEL (Gaffuri, 2008). Adopter une perspective SMA pour CollaGen permet donc à notre modèle de s'insérer plus facilement dans le travail collectif du laboratoire et sa plate-forme de travail (Renard et al, 2010).

Nous faisons donc le choix d'adopter une perspective SMA pour mettre en œuvre CollaGen : nous allons intégrer l'approche de généralisation par des agents situés se généralisant eux-mêmes, et le domaine des agents logiciels, pour ses techniques permettant l'appariement entre un demandeur et un agent fournissant un service par un agent intermédiaire.

D.2. QUI SONT LES AGENTS DANS COLLAGEN ?

Une fois admis que nous souhaitons modéliser CollaGen comme un système multi-agents, il est nécessaire de définir les composantes de ce SMA : les agents qui le composent, l'environnement des agents, le système de communication et de perception des agents et leur organisation (Demazeau, 1997). Dans un premier temps, nous présentons globalement notre vision SMA de CollaGen et l'organisation des agents, notamment en comparaison des autres modélisations SMA dans le domaine de la généralisation. Nous décrivons l'environnement des agents. Nous décrivons ensuite les agents situés de notre système puis les agents logiciels, et enfin leur méthode de communication et de perception de l'environnement.

D.2.1. Schéma général du Système Multi-Agents CollaGen

La Figure D-2 résume l'organisation et les agents qui composent le système multi-agent que nous proposons pour mettre en œuvre CollaGen. Cette organisation fait cohabiter des agents placés dans des mondes et des environnements différents : le monde géographique et le monde logiciel. Le monde géographique contient des agents espaces géographiques, situés et dont le comportement est proche de celui des agents géographiques des modèles AGENT et CartACom. Le monde logiciel contient des agents logiciels processus de généralisation (et processus de correction d'effets de bord). L'agent registre réalise l'interface entre les deux mondes en mettant en rapport un agent espace et un agent processus. L'agent chef d'orchestre est chargé d'orchestrer la collaboration des processus et pilote le comportement des agents dans les deux mondes. L'agent observateur est un peu à part : il se contente d'observer les objets géographiques au cours de leur généralisation et de créer de nouveaux agents espaces lors de l'émergence de zones de conflits.

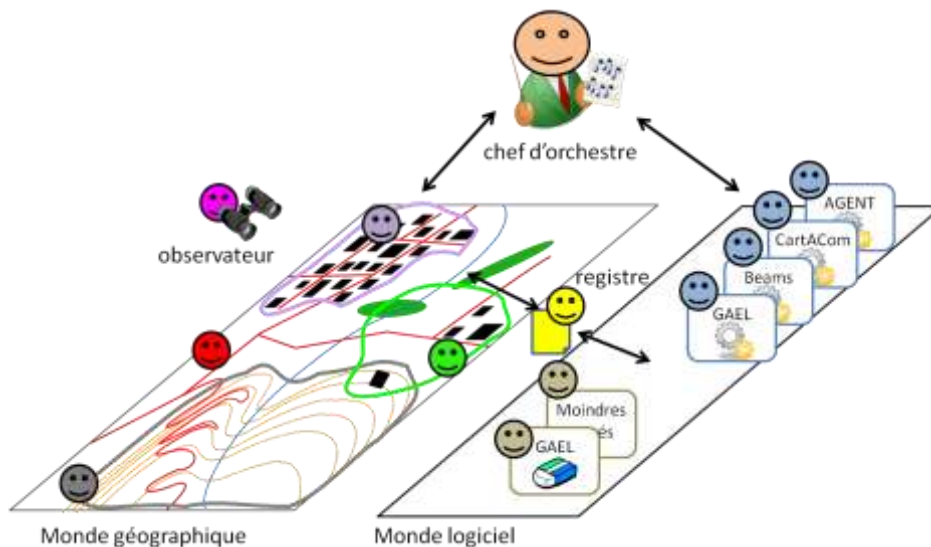


Figure D-2. Organisation de CollaGen dans sa perspective SMA avec la cohabitation d'agents espaces situés et d'agents processus logiciels. Le registre réalise l'interface entre les deux mondes pilotés par le chef d'orchestre. L'observateur observe la partie géographique de l'environnement c'est-à-dire les objets géographiques.

Nous n'avons pas le choix de modéliser l'organisation de notre SMA à partir des différents modèles de description et de spécification d'organisation qui existent dans la littérature SMA. Parmi ces modèles, nous pouvons citer MOISE (Hannoun et al, 2000) qui permet de décrire l'organisation sous

la forme de rôles, IODA (Kubera et al, 2011) qui permet de décrire l'organisation sous la forme d'interactions ou AGR (Ferber et Gutknecht, 1998). Comme le montre la Figure D-2, l'organisation que nous proposons est très spécifique de la généralisation collaborative : en se référant à la taxonomie de (Lacomme et al, 2009), notre organisation est un mélange d'organisation de type *activation* (l'orchestration des généralisations est faite par le chef d'orchestre), *découverte* (le registre permet de découvrir les agents espaces et processus pouvant interagir) et *situationnelle* (les agents peuvent interagir avec les autres agents rencontrés dans l'environnement). Nous avons donc opté pour une organisation ad-hoc, plus proche de notre modèle CollaGen, pour faciliter cette mise en œuvre comme SMA.

Dans les parties suivantes, nous présentons les différents environnements dans lesquels évoluent les agents. En outre, nous allons décrire plus précisément chaque groupe d'agents : les agents situés, les agents logiciels et les agents que nous appelons coordinateurs (chef d'orchestre, registre et observateur) du fait de leur rôle dans l'orchestration et leur capacité à interagir avec les deux mondes. Nous allons également décrire les interactions de chaque type d'agent avec les autres agents du système. Nous adoptons la définition de (Weiss, 1999) d'une interaction qui est un évènement mettant en jeu deux agents ou un agent et son environnement.

D.2.2. L'Environnement du SMA

L'environnement d'un SMA est l'ensemble des éléments non agents qui entourent les agents et interagissent avec eux. Dans CollaGen, l'environnement est composé de deux parties différentes : l'environnement géographique, qui est l'environnement des agents situés, et l'environnement logiciel qui est l'environnement des agents processus. Les agents espaces évoluent dans l'environnement géographique, les agents processus dans l'environnement logiciel et les agents coordinateurs dans les deux (Figure D-3). L'environnement géographique représente l'extrait de la base de données géographique que l'on souhaite généraliser. Il s'agit d'un environnement qui est situé dans l'espace euclidien de la projection cartographique choisie dans la base de données. Les éléments de l'environnement (les bâtiments, les routes, etc.) comme les agents placés dans cet environnement ont des coordonnées pour les situer précisément. Ainsi les agents espaces qui évoluent dans cet environnement géographique sont situés par leur géométrie et leurs voisinages définis dans les chapitres précédents. Les entités que les agents peuvent percevoir dans cet environnement sont les objets géographiques de la base de données (bâtiments, route, groupes de bâtiments etc.) et les moniteurs de contrainte qui possèdent eux aussi des coordonnées pour les situer précisément. Nous verrons dans les parties suivantes que certains agents ont une perception complète de l'environnement géographique alors que d'autres en ont une perception partielle.

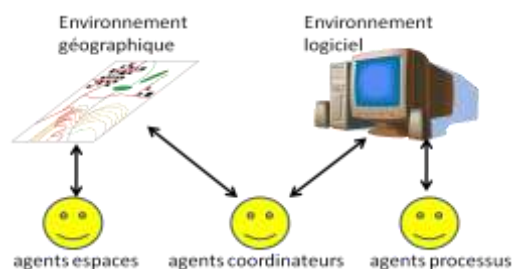


Figure D-3. Les trois types d'agents de CollaGen et les portions de l'environnement dans lesquelles ils évoluent.

Dans CollaGen, contrairement aux précédents modèles de généralisation fondés sur des SMA, les objets géographiques à généraliser (e. g. les bâtiments, les routes etc.) ne sont pas considérés comme des agents mais comme des éléments de l'environnement. Plusieurs raisons nous ont poussés à faire ce choix. Tout d'abord les objets géographiques micro ou meso n'ont pas de comportement proactif du point de vue de CollaGen en termes de traitement de généralisation : ils ne sont que des objets contenus dans des espaces géographiques. Dans les modèles précédents, les objets géographiques analysaient eux-mêmes leurs contraintes de généralisation et choisissaient les algorithmes à s'appliquer. Si certains processus dans lesquels les objets géographiques sont des agents peuvent être déclenchés durant une généralisation par CollaGen, le système n'ayant pas accès au modèle sous-jacent d'un processus, nous ne pouvons pas considérer que les objets sont des agents du point de vue de CollaGen. D'autre part, nous pourrions imaginer que les objets géographiques soient tout de même des agents proactifs gardant des capacités d'auto-évaluation et de décision sur la validation de la généralisation qui leur a été appliquée. Nous ne faisons pas ce choix car nous considérons que la généralisation par un processus s'applique sur un espace dans son ensemble et l'évaluation pour contrôler l'efficacité et valider la progression a plus de sens au niveau global de l'espace. L'évaluation d'un objet géographique seul n'est pas un indicateur significatif de la validation de la généralisation de l'espace. En outre, ces agents n'auraient pas d'interaction entre eux ni avec les autres agents du système. Nous considérons donc qu'il est inutile de modéliser les objets géographiques comme des agents.

L'environnement logiciel est celui dans lequel évoluent les agents processus. Dans cet environnement, les entités, agents ou autres ne sont pas localisées par des coordonnées mais par un emplacement sur la machine accueillant le système collaboratif. Ainsi, les agents processus sont localisés par l'emplacement du code qui permet leur exécution et par leur adresse mémoire une fois qu'ils sont actifs. Nous proposons toutefois un environnement logiciel minimaliste uniquement adapté à CollaGen, les agents n'évoluant que dans le cadre de la plateforme logicielle sur laquelle CollaGen a été mise en œuvre. Ainsi, les agents sont statiques (ils ne se déplacent pas dans un réseau) et n'ont pas accès à un réseau car nous rappelons qu'une des hypothèses de notre travail est que tous les processus soient implémentés sur la plateforme accueillant CollaGen. L'environnement logiciel contient alors les emplacements des processus et des différents algorithmes de la boîte à outils mais également les informations de connexion effective à la base de données, et des ressources comme l'ontologie de généralisation, les contraintes et règles formelles traduisant les spécifications de la carte ou encore les fichiers de traduction permettant l'interopérabilité des processus (voir C.12).

CollaGen repose sur une approche pas à pas de la généralisation, donc, les objets sont généralisés petit à petit. Nous choisissons de stocker les états intermédiaires des objets géographiques (c'est-à-dire de l'environnement géographique) pour permettre des retours en arrière. Nous considérons ici pour des questions de modélisation, que l'environnement géographique n'est constitué que des objets géographiques dans leur état courant de généralisation. Les différents états des objets sont conceptuellement de l'ordre de la base de données permettant le stockage de l'information, donc, nous considérons que ces états font partie, avec la base de données, de l'environnement logiciel de notre système multi-agent. Ce choix permet de bien séparer les environnements en termes d'agents pouvant interagir avec eux : les agents espaces n'ont besoin de percevoir que l'état courant des objets géographiques pour raisonner et les agents processus peuvent transformer (modifier,

éliminer, agréger) ces objets géographiques dans la base de données même sans percevoir l'environnement géographique.

D.2.3. Les Agents situés

Les agents situés dans CollaGen sont les espaces géographiques, espaces émergents inclus. Nous avons vu dans la partie précédente que les espaces géographiques évoluent dans l'environnement géographique de CollaGen. La Figure D-4 résume les interactions d'un agent espace géographique avec l'environnement et les autres agents du système. Les agents espaces géographiques sont créés par le chef d'orchestre. Un agent espace n'évolue que dans l'environnement géographique, donc, il ne perçoit que son environnement géographique, qu'il soit composé d'objets géographiques ou d'autres agents espaces. L'agent espace est capable de percevoir tous les objets géographiques qui sont contenus topologiquement et thématiquement dans la portion de l'espace qu'il encapsule. Par exemple, un espace de type territorial comme un espace de montagne perçoit les objets géographiques, quels que soit leur thème, qui sont contenus dans la délimitation de l'espace ; de son côté, un espace mixte comme un espace de routes de montagne ne perçoit que les objets du thème concerné situés dans la délimitation de l'espace. De la même manière qu'il perçoit les objets qu'il contient, un agent espace perçoit les moniteurs de contrainte qu'il contient. A partir des objets géographiques, l'agent espace est capable de se caractériser, pour se décrire si nécessaire auprès du registre. A partir des moniteurs de contrainte, l'agent évalue à plusieurs moments de son cycle de vie son niveau de satisfaction. Par exemple, sur la Figure D-4, l'agent n'est pas satisfait car ses moniteurs sont globalement insatisfaits donc il va demander au registre de lui trouver un processus adapté pour améliorer la généralisation des objets contenus.

Les agents espaces ont également des interactions entre eux. Avant le début des cycles de généralisation, chaque agent espace cherche dans son voisinage topologique les agents qui sont ses voisins. Ses capacités de perception des autres agents sont donc limitées à ses voisinages géométriques et topologiques. Chaque espace identifie et réifie des relations d'adjacence et d'intersection avec d'autres espaces. De plus, après une généralisation dont la progression a été validée par l'agent espace, celui-ci communique avec les agents espaces avec lesquels il a établi des relations de voisinage pour déterminer si la généralisation a provoqué des effets de bord avec ces espaces voisins. Si c'est le cas, les deux agents font une requête conjointe au registre pour corriger les effets de bord. Afin de gérer des effets de bord éventuels avec des parties de son contour qui ne font pas partie d'un voisinage avec un autre espace, l'agent espace analyse aussi son environnement directement extérieur. En effet, cela a été sous-entendu jusque là mais CollaGen gère la possibilité que certaines parties des données ne soient pas contenues dans un espace. Plusieurs causes peuvent amener cette situation : soit il y a un vide de données et cette partie n'a été incluse dans aucun agent espace, soit le nombre d'algorithmes disponibles pour découper les données en espaces est limité et certaines parties ne sont pas incluses dans les types d'agents espaces créés.

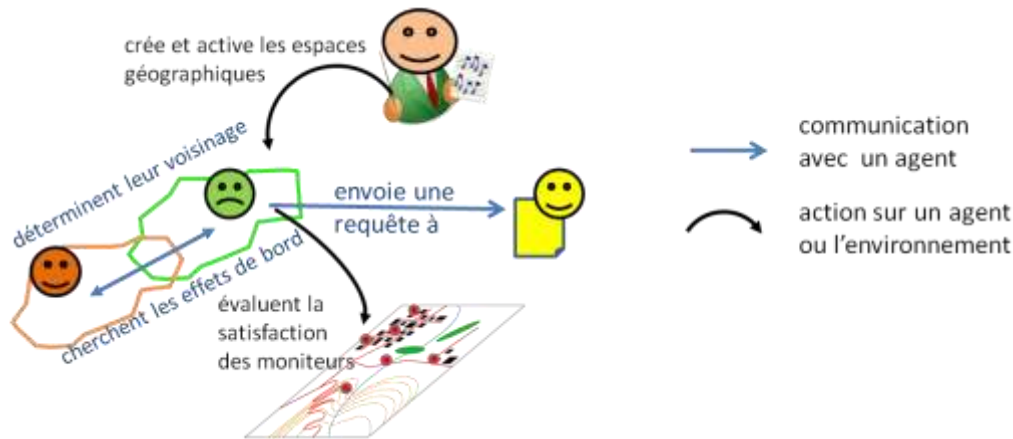


Figure D-4. Interactions d'un agent espace géographique avec l'environnement géographique et les autres agents.

Les agents espaces émergents ont des capacités de perception et de communication plus limitées : en effet, ils ne sont pas capables d'établir des relations de voisinage. Ils ne peuvent que se caractériser, s'évaluer et contacter le registre en cas de non satisfaction des moniteurs qu'ils contiennent.

D.2.4. Les Agents logiciels

Les agents logiciels sont très différents des agents situés dans CollaGen. Cette population d'agents est composée des processus de généralisation et des processus de correction d'effets de bord. La Figure D-5 illustre les interactions d'un agent processus avec les autres agents du système et avec son environnement logiciel. Nous pouvons tout d'abord remarquer que, contrairement aux agents espaces, les agents processus n'ont pas d'interactions entre eux : en effet, nous avons fait le choix dans le chapitre B de proposer une collaboration des processus par orchestration (donc une collaboration déléguée à une entité centralisée, l'agent chef d'orchestre) plutôt que par chorégraphie qui aurait nécessité que les agents processus communiquent entre eux pour prendre les décisions de coordination. Dans CollaGen, les agents processus ne communiquent qu'avec l'agent registre et le chef d'orchestre. La première interaction d'un agent processus est de communiquer à l'agent registre sa description formelle pour s'inscrire comme fournisseur de généralisation. Quand un couple espace/processus a été proposé par le registre, le chef d'orchestre demande à l'agent processus de réaliser une généralisation sur un espace. L'agent processus interagit alors avec son environnement logiciel. Pour se paramétrer correctement, il consulte l'ontologie et les spécifications représentées par les contraintes formelles et les règles opérationnelles. Une fois paramétré, l'agent processus s'exécute sur les objets de la base de données géographiques correspondant aux objets contenus dans l'espace à généraliser : l'agent processus ne perçoit pas l'environnement géographique et ses objets localisés dans l'espace, mais il perçoit l'adresse logicielle de la base de données géographique dont est issu l'environnement géographique et il s'exécute sur ces objets.

Un agent correcteur d'effets de bord adopte le même comportement qu'un agent processus, bien qu'il ne soit pas activé au même moment du processus collaboratif de CollaGen : il inscrit ses capacités dans le registre, et quand il est choisi par le chef d'orchestre, il détermine ses paramètres

en fonction des spécifications et de l'ontologie et il s'exécute sur les données présentant des effets de bord dans le voisinage d'un espace.

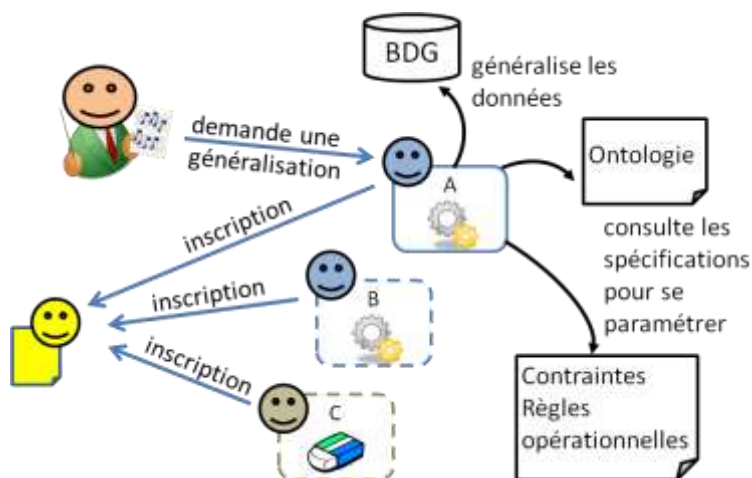


Figure D-5. Interactions d'un agent processus (ou correcteur) avec l'environnement logiciel et les autres agents.

D.2.5. Les Agents coordinateurs

L'Agent observateur

Les trois agents coordinateurs de CollaGen ont des comportements et des objectifs assez différents. Commençons par présenter l'agent observateur. Son objectif est d'observer le bon déroulement de la généralisation réalisée par un agent processus sur un agent espace géographique. Mais plutôt que d'analyser les traces de l'agent processus, il observe la partie de l'environnement géographique située à l'intérieur de l'agent espace en cours de généralisation. L'observateur n'a pas accès directement à l'environnement géographique et aux objets géographiques qui le composent : il ne peut percevoir l'environnement géographique que par l'intermédiaire d'artefacts comme le proposent (Piunti et al, 2009 ; Ricci et al, 2010). L'intérêt d'utiliser des artefacts est d'encapsuler un certain nombre de fonctionnalités sur des entités facilement manipulables par les agents. Dans notre cas, les artefacts sont les moniteurs de contrainte. Ainsi, l'observateur n'a pas besoin de savoir comment évaluer les contraintes de granularité d'un bâtiment et de densité d'un îlot, il doit juste savoir manipuler le moniteur pour connaître ses satisfactions courante et passée. L'autre objectif de l'observateur est de lancer la création de nouveaux agents espaces émergents lorsqu'il observe l'émergence de zones de conflits résiduels. Pour simplifier la modélisation, il remplit cet objectif en communiquant ses observations de zones émergentes au chef d'orchestre qui possède par ailleurs les capacités pour créer et activer de nouveaux agents espaces. L'observateur communique avec trois types d'agents :

- l'agent processus en train de réaliser sa généralisation autorise l'observateur à observer les moniteurs (il se met en pause pendant l'observation)
- l'observateur communique des zones émergentes au chef d'orchestre
- les agents espaces émergents communiquent les modifications apportées aux objets généralisés dans l'espace émergent (qui a été généralisé entre temps) ;

- l'observateur communique ces modifications à l'agent processus en pause afin que celui-ci redémarre sans incohérence (il sait quels objets ont été modifiés et éliminés entre temps).

La Figure D-6 résume les interactions entre l'agent observateur, son environnement et les autres agents.

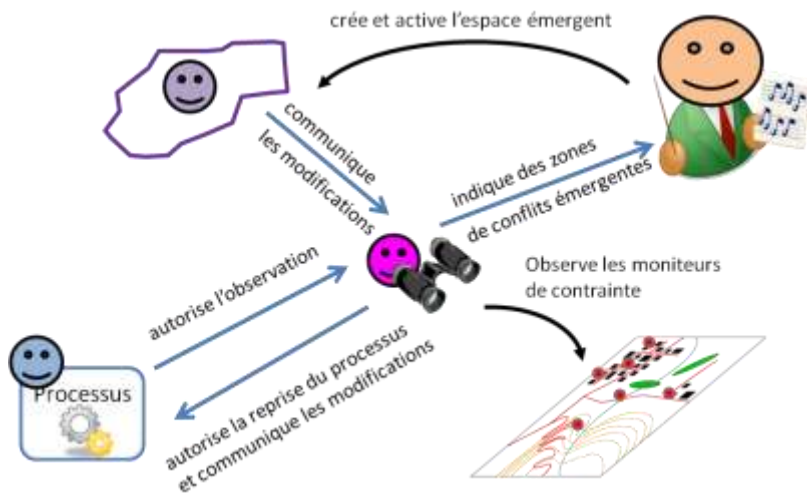


Figure D-6. Interactions de l'agent observateur avec l'environnement géographique et les autres agents.

L'Agent registre

L'agent registre est un agent intermédiaire ('middle agent') au sens de (Sycara et al, 2002). Nous faisons le choix de modéliser le registre comme un agent intermédiaire, car la norme FIPA-ACL sur les registres (FIPA, 2004) ainsi que les travaux sur l'appariement besoin/service (Decker et al, 1997) suggèrent que le registre doit être un agent proactif plutôt qu'un élément consultable de l'environnement. La Figure D-7 montre les interactions de notre agent registre avec les autres agents de CollaGen. Les agents processus s'inscrivent au registre pour que l'agent les prenne en compte dans ses calculs. L'agent registre reçoit des requêtes de la part d'agents espaces. Il analyse alors les moniteurs contenus dans l'espace, calcule sa réponse et recherche les emplacements des processus constituant cette réponse dans l'environnement logiciel. Finalement, il renvoie au chef d'orchestre une liste des processus avec leurs emplacements pour que le chef d'orchestre puisse communiquer avec eux. Selon la classification de (Decker et al, 1997), un agent intermédiaire de type courtier met en relation un client qui ne connaît aucune des capacités des fournisseurs mais dont il connaît déjà les besoins. Par contre, un agent apparieur ne connaît pas *a priori* les besoins du client mais les capacités des fournisseurs sont connues de tous. Notre agent registre est donc plutôt du type apparieur (matchmaker) que courtier car seul le « client » connaît ses préférences (seul l'espace géographique connaît ses caractéristiques) et tous les intervenants connaissent les capacités des processus via le registre qui rend ses informations publiques. Cette analyse est confirmée par l'utilisation de la taxonomie de (Wong et Sycara, 2000) qui classe les agents intermédiaires en fonction de six critères (« Quel type d'information est envoyée dans la requête ? » par exemple). Selon cette classification, les critères de notre agent registre correspondent à une variante d'apparieur. Nous pourrions donc utiliser les nombreux travaux de la littérature, par exemple (Badica

et Badica, 2008), qui formalisent le fonctionnement d'un appairier pour modéliser notre agent et ses protocoles de communications.

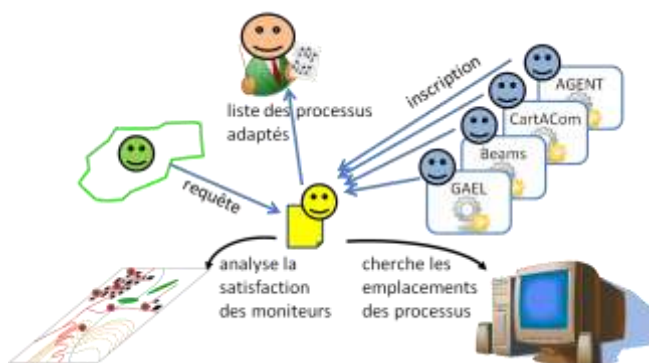


Figure D-7. Interactions de l'agent registre avec les environnements et les autres agents.

L'Agent chef d'orchestre

Enfin, l'agent chef d'orchestre est l'entité centralisée qui réalise l'orchestration des agents processus. L'agent chef orchestre a donc des interactions avec tous les autres agents du système, comme nous l'avons vu en présentant les autres agents dans les paragraphes précédents (Figure D-8). Les agents espaces sont les seuls avec qui le chef d'orchestre a une interaction bidirectionnelle : le chef d'orchestre choisit à chaque cycle de généralisation un espace à généraliser et lui indique qu'il a été choisi, et cet espace lui communique son évaluation une fois la généralisation effectuée. Le registre et l'observateur ne font que communiquer des informations au chef d'orchestre, respectivement la liste des processus adaptés à l'espace choisi et les éventuelles zones émergentes durant une généralisation. Les agents processus se contentent de recevoir des instructions du chef d'orchestre, celles de généraliser un espace donné.

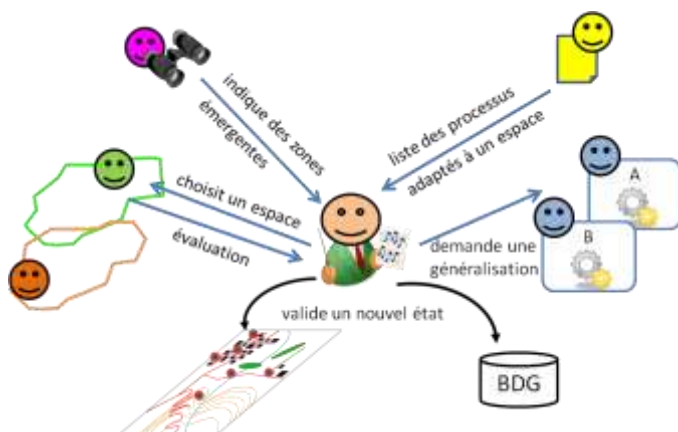


Figure D-8. Interactions de l'agent chef d'orchestre avec tous les autres agents et les environnements.

Le chef d'orchestre interagit également avec les deux environnements du système (géographique et logiciel). C'est le chef d'orchestre qui valide ou invalide les résultats d'une généralisation à partir de l'évaluation communiquée par l'espace. Il modifie l'environnement géographique en validant l'état courant des objets géographiques modifiés par le processus, mais aussi l'environnement logiciel en stockant dans la base de données l'état précédant la généralisation.

D.2.6. Communication et perception

En décrivant dans les parties précédentes l'objectif de chacun des agents et ses interactions avec les autres agents du système, nous avons mis en avant le besoin de communication entre les agents et le besoin de capacités de perception de l'environnement par les agents. Nous présentons, dans cette partie, nos choix pour réaliser ces différentes communications et perceptions. D'après ce que nous venons de présenter, nous avons besoin de trois types de communication entre nos agents :

- Le **transfert d'information** d'un agent à un autre : par exemple, l'agent espace transfère au chef d'orchestre l'évaluation qu'il a faite de sa généralisation par un processus ou encore le registre transfère au chef d'orchestre la liste des processus adaptés à la généralisation d'un espace.
- L'**acquisition d'information** par un agent sur son voisinage.
- La **demande d'action** par un agent à un autre agent ayant les capacités pour réaliser l'action : par exemple, le chef d'orchestre demande à un processus de généraliser un espace ou encore l'espace demande au registre de trouver un processus adapté.

Les systèmes multi-agents utilisent principalement deux types de modèles : le modèle fondé sur une métaphore animale dans lequel les agents déposent l'information dans l'environnement comme les fourmis déposent des phéromones, et le modèle fondé sur une métaphore humaine dans lequel les agents communiquent explicitement en s'envoyant des messages. Le premier modèle est principalement utile lorsque le destinataire de la communication n'est pas préalablement connu : l'émetteur ne sait pas quel agent va percevoir les informations déposées dans l'environnement. Dans notre cas, les types de communication et les intervenants sont assez réduits et déjà connus, il s'agit d'une communication ciblée pour laquelle la métaphore humaine nous paraît plus appropriée. La communication par messages dans les SMA est principalement fondée sur les actes de langages (Searle, 1969), théorie sur laquelle sont basés deux standards de langages de communication, KQML (ARPA Knowledge Sharing Initiative, 1993) et FIPA ACL (FIPA, 2000). Ces standards déterminent la forme des messages, le contenu ou la méthode de transport des messages. Par exemple, le modèle de généralisation CartACom (Duchêne, 2004) permet la communication entre agents par échange de messages avec un langage plus simple que les normes KQML et FIPA ACL inspiré par (Ferber, 1995). Les messages sont ainsi de la forme « Informer(éliminé) » ou « DemanderFaire(Bouger) ». Toutefois, comme les communications entre nos agents sont assez simples et que nous avons manqué de temps pour travailler ce point, nous n'avons pas mis en œuvre ce type de communication entre agents par messages. Nos agents communiquent de manière plus directe : les agents pouvant communiquer ensemble ont accès aux méthodes ad-hoc des autres agents. Ainsi, le chef d'orchestre a accès au résultat de la méthode d'auto-évaluation des agents espaces, pour remplacer la communication de type transfert d'information. Pour remplacer la demande d'action, comme la demande à un processus de généraliser un espace, il a accès à la méthode déclenchant l'action, le résultat de l'action étant obtenu à travers le résultat de la méthode.

Concernant la perception de l'environnement, dans l'environnement logiciel, nous avons également opté pour une solution simpliste pour aller plus vite. Un objet environnement logiciel regroupe l'ensemble des entités présentes dans cet environnement, leur emplacement logiciel. Cet objet environnement est accessible à l'ensemble des agents évoluant dans cet environnement, c'est-à-dire les agents processus, correcteurs et coordinateurs. Dans l'environnement géographique, les agents

espaces sont capables de percevoir les agents situés dans leur voisinage (d'après notre définition du voisinage) et les portions de son voisinage sans agent. Pour cela, l'agent espace est doté de méthodes d'analyse spatiale assez simples comme la recherche des objets dans son buffer (dilatation de la géométrie).

D.3. REPARTITION DES CAPACITES ENTRE LES AGENTS

Nous avons présenté dans la partie précédente les agents de CollaGen, leur organisation et leurs interactions. Nous précisons dans cette partie, pour chacun des types d'agents, son cycle de vie et la répartition des actions, issues des composants présentés dans le chapitre précédent, entre ces agents.

D.3.1. L'Agent espace géographique

Le premier agent que nous décrivons est l'espace géographique, situé dans l'environnement géographique de notre système. Il dispose de capacités issues de plusieurs des composants de CollaGen que nous avons présentés dans le chapitre C. Il dispose tout d'abord d'une partie des outils d'analyse spatiale inclus dans CollaGen pour se caractériser (e.g. les objets de l'espace ont-ils une structure hiérarchique ?) mais aussi pour s'enrichir. Par exemple, l'espace réseau routier dispose d'algorithmes d'analyse spatiale comme ceux de (Touya, 2008) pour identifier ses ronds-points et ses pattes d'oies. Ces capacités étaient regroupées dans le composant d'enrichissement.

L'espace géographique est également doté des capacités d'auto-évaluation attribuées au composant d'évaluation et d'observation. En analysant l'évolution locale et globale des moniteurs de contrainte contenus dans son emprise géométrique et thématique, l'espace peut évaluer sa satisfaction de manière générale ainsi que la qualité interne de la généralisation qu'il vient de subir. De plus, l'agent espace dispose des capacités de déterminer un échantillon représentatif des moniteurs qu'il contient pour les communiquer à l'agent observateur.

L'espace géographique dispose enfin d'une partie des capacités du composant de gestion des effets de bord. Il est en effet capable d'analyser son voisinage pour identifier les différents types d'effets de bord qui se traduisent par des moniteurs de cohérence insatisfaits. L'espace est capable d'analyser ses moniteurs de cohérence pour identifier des effets de bord avec un autre espace ou simplement avec des objets extérieurs à son emprise.



Figure D-9. Cycle de vie d'un agent espace géographique. Cette figure, comme les suivantes dans cette partie, omet les interactions avec les autres agents, notamment celles qui déclenchent et terminent les périodes d'attentes (symbolisée par le sablier), pour clarifier la lecture.

Le cycle de vie d'un agent espace, une fois créé et activé par le chef d'orchestre, se déroule en trois temps (Figure D-9). Dans un premier temps, l'agent se caractérise et s'enrichit en objets meso et relations géographiques, puis, il s'évalue à son état initial pour savoir s'il a besoin d'être généralisé. L'agent attend alors que le chef d'orchestre lui propose de se généraliser. Quand le chef d'orchestre le désigne, l'agent espace fait une requête au registre et attend que le processus choisi effectue la généralisation. Quand la généralisation est effectuée, l'espace se réévalue. Si cette évaluation n'est pas jugée comme satisfaisante, d'autres généralisations peuvent être effectuées (annulation ou non et recours au processus suivant dans la liste). Si cette évaluation est jugée comme satisfaisante, l'espace recherche des effets de bord dans son voisinage (et demande une correction au cas où) puis se remet en attente du chef d'orchestre.

D.3.2. L'Agent processus

L'agent processus de généralisation a comme capacité principale de pouvoir s'exécuter sur les objets géographiques contenus dans un agent espace. Avant de pouvoir s'exécuter, il doit être capable de fournir une description formelle de ses capacités de généralisation à partir des connaissances qu'il a de lui-même. Il peut alors s'inscrire au registre en communiquant cette description formelle. Il dispose également d'une partie des capacités du composant de traduction de CollaGen car il sait comment s'auto-paramétrer à partir de la consultation des contraintes formelles et de l'ontologie. La fonction de traduction associée à un processus devient du point de vue SMA une action que peut réaliser l'agent pour fixer ses paramètres avant de s'exécuter.

L'agent processus récupère enfin une partie des capacités du composant d'observation, notamment pour gérer l'interruption de sa généralisation (s'il s'agit d'un processus 'interrompable') et la reprise sans incohérence après la généralisation d'un espace émergent entre temps. Il dispose donc des mécanismes de synchronisation avec l'observateur, provenant du composant d'observation.

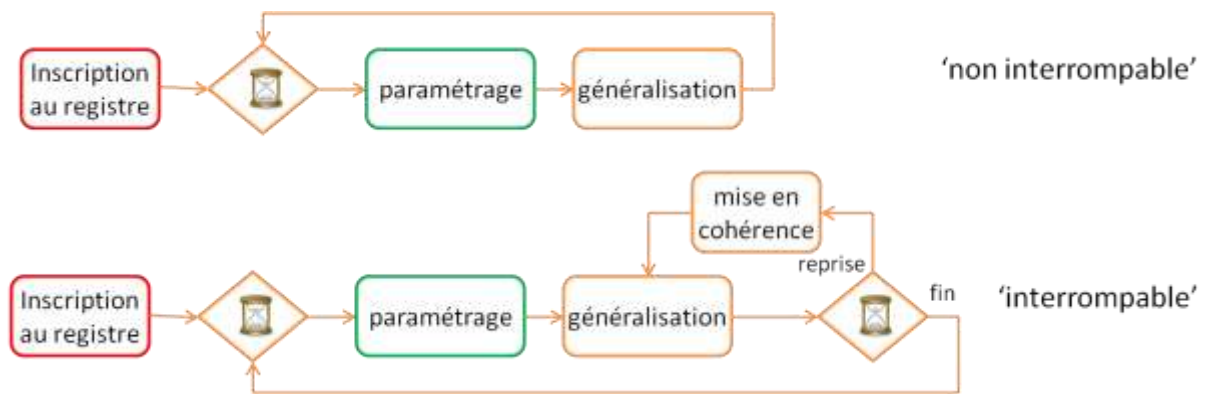


Figure D-10. Cycle de vie d'un agent processus de généralisation suivant qu'il est 'interrompable' ou non.

Nous proposons deux cycles de vie légèrement différents pour les processus que l'on peut interrompre au milieu de la généralisation (dits interrompables) et les autres (Figure D-10). Dans les deux cas, le cycle de vie se passe en deux temps. Dans un premier temps, le processus s'inscrit au registre en communiquant ses capacités formelles, puis, attend que le chef d'orchestre lui demande d'agir. Dans un deuxième temps, quand le chef d'orchestre lui a désigné un espace à généraliser, il détermine un paramétrage adapté, puis, effectue la généralisation. Dans le cas des processus interrompables, l'agent processus fait une pause tous les pas de temps (paramètre du système choisi

à la construction des agents processus), pause au cours de laquelle l'observateur cherche des espaces émergents. Les modifications éventuelles faites sur les espaces émergents sont mises en cohérence avec les connaissances du processus (pour désigner au processus les objets éliminés et/ou modifiés) avant la reprise du processus là où il s'était arrêté. Une fois la généralisation terminée, l'agent processus se remet en attente d'une demande du chef d'orchestre.

D.3.3. Le Chef d'orchestre

Le chef d'orchestre, comme son nom l'indique, a un rôle central dans notre système multi-agent car c'est lui qui prend les décisions d'orchestration, indiquant alors aux autres agents quand agir. Il est aussi capable de créer les agents situés du système et dispose donc des capacités de création d'espaces du composant d'enrichissement: il utilise des algorithmes d'analyse spatiale pour créer les espaces géographiques qui correspondent aux spécifications de l'utilisateur (et aux pré-conditions des processus). Le chef d'orchestre dispose également des capacités du composant d'enrichissement pour créer les moniteurs de contrainte à partir des contraintes formelles. Afin de réaliser ces deux opérations, il dispose aussi des capacités du composant de traduction qui permettent l'appariement entre concepts de l'ontologie ou contraintes formelles, et algorithmes présents sur la plateforme logicielle.

La capacité principale du chef d'orchestre, qui lui donne son nom, est celle qu'il assure comme composant d'orchestration : le choix, à chaque cycle de généralisation, de l'espace de généralisation à activer. Comme nous l'avons vu dans le chapitre C, le chef d'orchestre dispose d'un moteur d'inférence pour choisir l'espace à activer à partir de la position dans la généralisation collaborative et des règles d'orchestration dont il a connaissance. Le chef d'orchestre dispose également des autres capacités du composant d'orchestration, c'est-à-dire celles relatives à la gestion des états des objets géographiques transformés par les différentes généralisations : il sait comment manipuler la base de données comportant les données initiales de la généralisation pour stocker les états intermédiaires et mettre à jour l'état courant. Le chef d'orchestre est aussi chargé de réaliser le choix qui détermine si le nouvel état est valide ou si CollaGen doit revenir en arrière, ce qui est une des fonctions du composant d'évaluation.

Enfin, nous avons vu dans la partie C.16 que plusieurs solutions étaient envisageables pour corriger des effets de bord : trouver une position équilibrée, relâcher une contrainte ou choisir une des positions (avant ou après la généralisation causant les effets de bord). Ce choix nécessite une perspective plus globale sur les données que celle du (ou des) espaces concerné(s) et la responsabilité de faire ce choix est donc déléguée au chef d'orchestre.



Figure D-11. Cycle de vie d'un agent chef d'orchestre.

Du fait de son statut d'entité décisionnelle centralisée, le cycle de vie du chef d'orchestre est composé de nombreuses interventions ponctuelles (Figure D-11). Le cycle démarre par la création des espaces géographiques, puis, le chef d'orchestre réalise des cycles de généralisation : il choisit le

prochain espace à généraliser, il met en rapport le processus et l'espace d'après le choix du registre, valide et crée les nouveaux états après la généralisation, et enfin statue sur la méthode de correction des effets de bord éventuels.

D.3.4. Le Registre

L'agent registre est assez simple à définir car il reprend exactement les capacités du composant registre de CollaGen. Le registre est ainsi chargé de comprendre des requêtes simples (un espace demande à être généralisé) et des requêtes orientées par le chef d'orchestre (un espace demande à être généralisé par un certain type de processus, ce qui correspond à la possibilité de définir des règles d'orchestration dont la conclusion est un processus plutôt qu'un espace). Il analyse l'environnement géographique et les descriptions formelles pour établir sa réponse. Il dispose enfin de capacités d'apprentissage par renforcement sur les réussites et les échecs des processus qu'il a conseillés pour orienter ses choix.



Figure D-12. Cycle de vie d'un agent registre.

Le cycle de vie de l'agent registre est également très simple (Figure D-12). Dans un premier temps, l'agent enregistre les inscriptions des processus de généralisation disponibles. Puis il n'agit qu'en réponse à une requête d'un espace géographique : il enchaîne les trois étapes de la constitution de sa réponse, la communique au chef d'orchestre et attend la requête suivante.

D.3.5. L'Agent observateur

L'agent observateur est un agent assez indépendant des autres car son objectif est d'interrompre la généralisation en cours (car nous avons fait le choix de généralisations séquentielles) s'il observe l'émergence d'une ou plusieurs zones de conflits qui perdurent. Il dispose donc principalement des capacités décrites dans la partie C.14 sur l'observation en ligne de la généralisation. Il s'agit de capacités de perception de l'environnement géographique pour observer un échantillon de moniteurs ; de capacités d'analyse pour savoir distinguer un conflit persistant parmi les moniteurs observés ou une situation requérant l'annulation de la généralisation en cours ; de capacités d'analyse spatiale pour délimiter une zone émergente de conflits ; il s'agit enfin de capacités de synchronisation avec un agent processus (qui se met en pause et reprend).

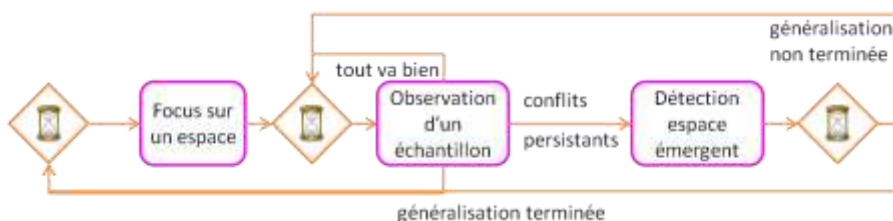


Figure D-13. Cycle de vie d'un agent observateur.

Le cycle de vie de l'agent observateur de notre système démarre en position d'attente d'une généralisation d'un espace par un processus (Figure D-13). Dès qu'un espace commence à être généralisé, l'observateur se concentre sur cet espace. Quand le processus se met en pause pour autoriser l'observation, l'agent analyse les moniteurs de l'échantillon et trois possibilités se présentent à lui : si tout va bien (pas de conflit persistant), soit l'observateur fait reprendre la généralisation au processus (en lui indiquant qu'il a terminé son observation), soit il fait continuer le cycle normal de généralisation si le processus est terminé ; si des conflits persistants sont identifiés, l'agent détecte l'espace émergent et le communique au chef d'orchestre qui va le construire et l'activer. Une fois l'espace émergent généralisé, soit la généralisation reprend (si elle n'était pas terminée), soit le système continue les cycles collaboratifs (si la généralisation était terminée).

D.3.6. L'Agent correcteur d'effets de bord

L'agent correcteur d'effets de bord fonctionne relativement de la même manière qu'un agent processus qui ne peut pas s'interrompre (non interrompable). La différence est que sa description formelle diffère du processus et qu'il ne travaille pas sur un espace géographique mais sur une portion du voisinage d'un espace, partagée ou non avec un autre espace. Dans le cycle de vie de l'agent correcteur (Figure D-14), il faut noter une participation à la façon de statuer sur la correction du chef d'orchestre (en lui indiquant quelles transformations il peut effectuer) avant le paramétrage de l'agent.

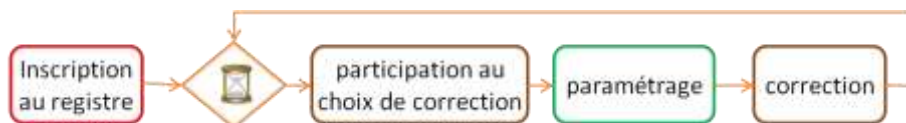


Figure D-14. Cycle de vie d'un agent correcteur d'effets de bord.

D.3.7. Bilan de la répartition des capacités entre agents

La Figure D-15 récapitule les répartitions de capacités entre les agents, décrites dans les parties précédentes. Le choix de cette dichotomie entre composants et agents découle de notre analyse du problème de la généralisation collaborative. Nous avons commencé par analyser les fonctions à remplir pour réaliser une généralisation collaborative ce qui nous a amené à la conception des composants en regroupant des proches thématiquement et méthodologiquement. La mise en œuvre de CollaGen par un SMA nous a amené à adopter un point de vue différent sur ces composants, orienté par les questions d'autonomie et de proactivité : qui d'un point de vue conceptuel et opérationnel réalise une fonction donnée dans CollaGen ? Nous avons toutefois tenu à conserver cette présentation en composants pour faciliter la compréhension de ce que nous considérons comme le cœur de notre modèle, c'est-à-dire les mécanismes constitutifs de ces composants.

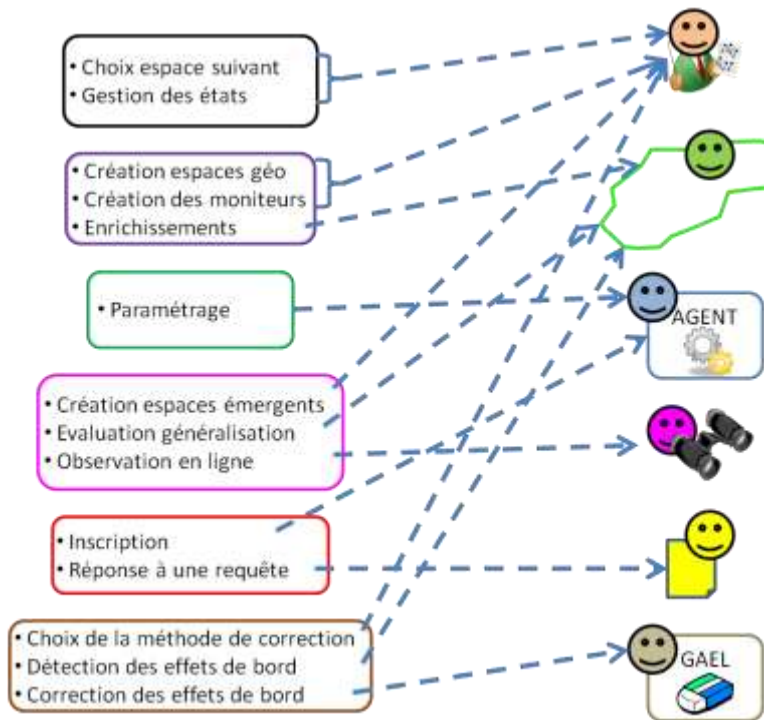


Figure D-15. Schéma récapitulatif de la répartition des fonctions (et capacités associées) des composants de CollaGen vers les agents de notre SMA.

D.3.8. Déroulement d'une généralisation CollaGen du point de vue SMA

Nous avons présenté dans les parties précédentes, les cycles individuels des différents agents de notre système sans montrer l'enchaînement des interactions entre les agents. La Figure D-16 montre un diagramme de séquences UML décrivant les interactions entre nos agents durant une généralisation (pour laquelle un seul espace est généralisé pour simplifier).

Cette figure montre comment les agents se coordonnent dans leur tâche en communiquant des informations (flèches pleines), des résultats d'action (flèches pointillées) ou des demandes d'action (bout de flèche avec un triangle blanc). Si la plupart des agents sont actifs durant des plages bien précises et attendent pendant le reste du cycle, ce schéma permet d'illustrer le rôle central du chef d'orchestre qui est le point de départ ou d'arrivée d'une grande partie des interactions. Dans le cas de plusieurs espaces, le diagramme reprend par l'action *choixEspace()* et les séquences qui suivent.

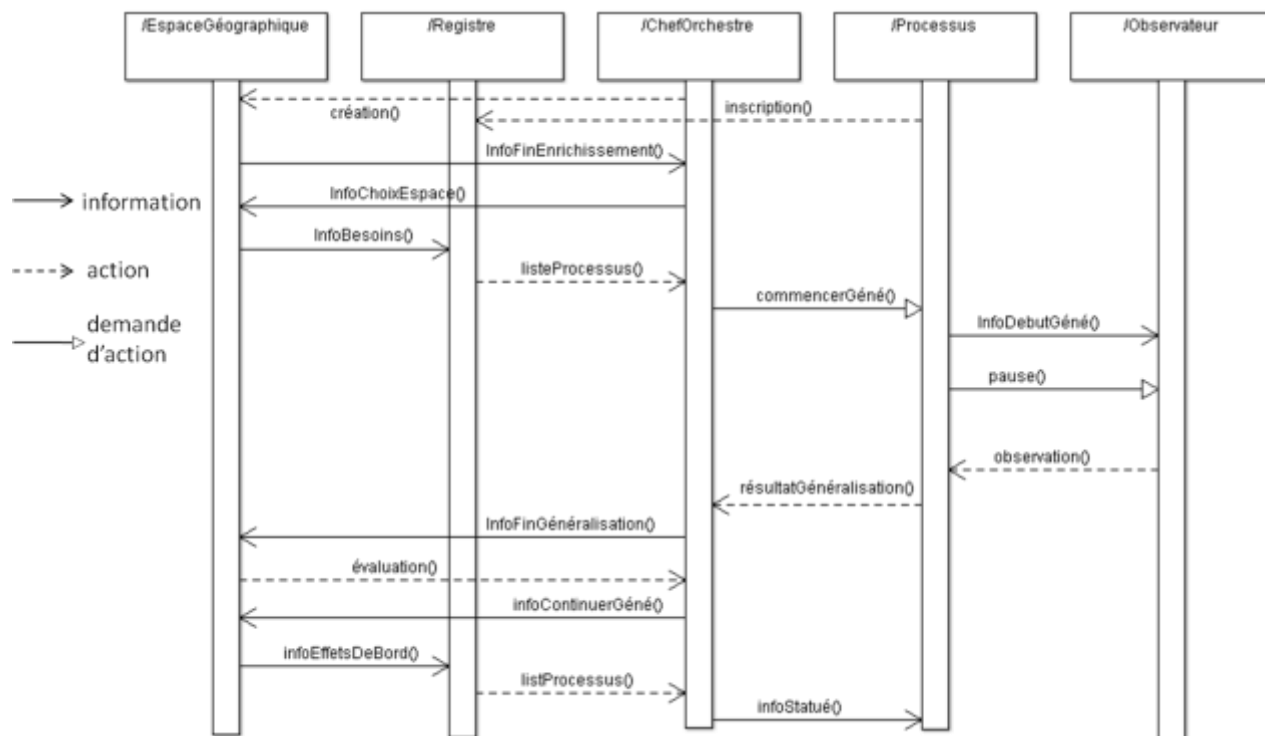


Figure D-16. Diagramme UML de séquence résumant le déroulement d'une généralisation collaborative par les séquences d'actions de ses agents (un seul espace est généralisé pour simplifier).

D.4. DISCUSSION

Nous avons montré, dans les parties précédentes, comment nous proposons de mettre en œuvre CollaGen dans une perspective SMA. Dans cette partie, nous discutons ce choix en nous demandant tout d'abord si notre système est vraiment un SMA. Puis nous discutons de l'intérêt de cette perspective SMA par rapport à une mise en œuvre plus littérale des composants de CollaGen présentés au chapitre C. Enfin, nous discutons d'autres considérations SMA comme la convergence générale de notre système ou sa complexité.

D.4.1. Analyse critique des agents de CollaGen

Nous avons défini un agent comme une entité capable de remplir ses objectifs de manière autonome, en coopération avec d'autres agents (B.1.2). Pour pouvoir justifier le choix d'une perspective SMA pour CollaGen, nous devons donc montrer que les entités que nous avons définies comme des agents sont bien autonomes et agissent en coopération avec les autres agents. De plus, (Franklin et Graesser, 1996) établissent que pour différencier un agent d'une entité logicielle quelconque, il faut considérer les propriétés suivantes : l'autonomie sur ses propres actions, la réactivité par rapport aux évènements, la pro-activité par rapport à ses objectifs et la continuité temporelle.

Concernant l'aspect « coopération entre agents » de la définition, les différentes interactions décrites dans le cycle de vie de la Figure D-16 montrent qu'ils communiquent bien entre eux pour se coordonner et prendre les bonnes décisions pour atteindre l'objectif commun du système, la bonne généralisation de la carte. Cette figure montre également la continuité temporelle de nos agents tout le long du processus collaboratif de généralisation, qu'ils soient en phase active ou en phase d'attente de nouvelle information. Pour les autres aspects, comme nous proposons l'utilisation d'agents assez différents dans leurs objectifs et leurs capacités, nous discutons de l'autonomie, de la réactivité et de la pro-activité de chacun d'eux séparément.

Les agents espaces géographiques sont ce que (Duchêne et al, 2011) appellent des agents géographiques, c'est-à-dire des entités possédant une empreinte géométrique et géographique dans une base de données géographique que l'on doit généraliser. L'objectif d'un agent géographique est de bien se généraliser tout en contrôlant la bonne généralisation de ses composants s'il y en a. Concernant la réactivité, (Duchêne, 2004, p.211) argumente que les agents géographiques de CartACom (et d'AGENT) sont réactifs dans la mesure où ils perçoivent les changements de l'environnement à travers leurs contraintes et y réagissent en se généralisant. Les agents espaces diffèrent des agents géographiques de CartACom car ils ne sont pas liés directement à des contraintes et ne se généralisent pas eux-mêmes (seulement leurs composants). Nous les considérons tout de même comme réactifs car ils perçoivent les moniteurs de contrainte contenus dans leur empreinte et y réagissent en cherchant un processus améliorant leur satisfaction globale. A propos de l'autonomie, (Duchêne, 2004) avance que les agents géographiques de CartACom choisissent eux-mêmes les actions à appliquer et en évaluent les effets de manière autonome. Les agents espaces ne choisissent pas eux-mêmes les actions à s'appliquer, ils laissent une part de leur autonomie au chef d'orchestre et au registre. Bien que moins autonomes que les agents de CartACom, nous considérons les agents espaces comme relativement autonomes car ils s'évaluent eux-mêmes et décident de la qualification de cette généralisation (bonne qualité interne par exemple). Ils choisissent par ailleurs de manière autonome les enrichissements qu'ils s'appliquent et s'ils font une requête au registre. De manière plus anecdotique, ils déterminent aussi de manière autonome l'échantillon de moniteurs qui doit être observé par l'observateur lors d'une généralisation. Enfin, concernant la proactivité, (Duchêne, 2004) considère que les agents de CartACom sont proactifs car ils essaient de satisfaire au mieux leurs contraintes en fonction de leur but (être bien généralisé individuellement). Nous pouvons reprendre la même argumentation pour les agents espaces, qui sont eux aussi guidés par l'objectif de la satisfaction globale des moniteurs qu'ils contiennent pour déclencher des actions et interactions avec les autres agents.

Considérons maintenant le cas des agents processus, que nous avons classés dans la catégorie des agents logiciels. Il serait assez facile d'imaginer CollaGen fonctionner comme un SMA sans que les processus soient considérés comme des agents mais plutôt comme des artefacts connus du registre qui peuvent être manipulés par les agents espaces pour se généraliser (de la même manière que les agents géographiques d'AGENT ou CartACom déclenchent des actions pour se généraliser). Nous avons proposé une solution dans laquelle les processus sont considérés comme des agents (contrairement aux actions d'AGENT et CartACom) car nous considérons que notre problème diffère de la généralisation simple d'entités (individuelles ou meso) géographiques. L'objectif d'un agent processus est, comme tous les fournisseurs de services, d'effectuer le plus possible de ces services de généralisation qu'il propose. Pour cela, il dispose d'un objectif secondaire qui est de bien généraliser les données pour lesquelles il a été choisi car le système d'apprentissage du registre favorise les

processus efficaces. Nous considérons l'agent processus comme relativement proactif car pour atteindre son objectif d'être choisi, il s'inscrit au registre en décrivant ses capacités. Un agent processus est également autonome, car, une fois choisi, il détermine lui-même son paramétrage pour s'adapter au mieux aux spécifications de la carte et aux caractéristiques de l'espace à généraliser. Un agent processus est enfin relativement réactif dans la mesure où il perçoit les interactions avec l'observateur et sait y réagir en mettant en cohérence les modifications apportées par la généralisation d'un espace émergent au milieu d'une généralisation.

Le registre est un agent qui suit les principes de la norme FIPA et des agents dits d'appariement, donc, on peut le considérer comme un agent autonome de notre système en s'appuyant sur cette littérature. L'observateur peut aussi être considéré comme un véritable agent, car il remplit les trois propriétés. Il est réactif, car il perçoit les moniteurs de contrainte d'un espace et peut y détecter un ensemble de conflits persistants émergent. Il est autonome, car il décide par lui-même de qualifier les groupes de conflits persistants d'espaces émergents. Il est enfin proactif, car il est guidé par son objectif de détection des endroits où la généralisation échoue localement.

Le dernier point à considérer est le cas du chef d'orchestre et plus généralement de l'approche centralisée de l'orchestration : le chef d'orchestre centralise une grande partie des décisions prises dans notre SMA. Le chef d'orchestre est clairement autonome, car il contrôle ses actions (le choix du prochain espace à généraliser par exemple) tout comme les actions de certains autres agents. Il est proactif, car toutes ses actions sont guidées par l'objectif global d'une bonne généralisation de la carte. Le chef d'orchestre est enfin réactif, car il est capable de recevoir des demandes des autres agents quand ceux-ci ont besoin d'une décision centralisée qu'ils ne peuvent prendre eux-mêmes, et il est capable d'y répondre de manière adaptée.

Nous considérons donc que notre perspective centrée-agent de CollaGen forme bien un véritable Système Multi-Agent dans lequel l'ensemble des agents, qu'ils soient situés ou logiciels, sont plus ou moins autonomes, réactifs, proactifs et coordonnés mais assez en tout cas pour être considérés comme de vrais agents.

D.4.2. En quoi est-il intéressant de mettre en œuvre CollaGen par un SMA

Dans la partie précédente, nous avons défendu l'idée que nous pouvions considérer CollaGen comme un véritable système multi-agents, mais nous devons maintenant nous interroger sur la pertinence de l'utilisation d'un SMA pour la résolution de notre problème de généralisation collaborative. Nous discutons donc dans cette partie de l'utilité d'une telle perspective, de son intérêt pour la mise en œuvre de CollaGen, par rapport à d'autres perspectives ou méthodes de mise en œuvre.

Alternative aux SMA pour mettre en œuvre CollaGen

Quels autres choix pouvions-nous faire pour modéliser CollaGen d'une manière efficace et qui permette une mise en œuvre relativement aisée ? L'alternative la plus réaliste est de mettre en œuvre les composants de manière littérale sous la forme d'un 'workflow' comme ceux proposés par

(Petzold et al, 2006). Nous proposons en Figure D-17 un workflow qui reprend la dynamique des composants de CollaGen dans le formalisme YAWL (Van der Aalst et ter Hofstede, 2005) inspiré des réseaux de Petri (Petri, 1962). Toutes les étapes ne sont pas complètement détaillées pour garder le schéma lisible.

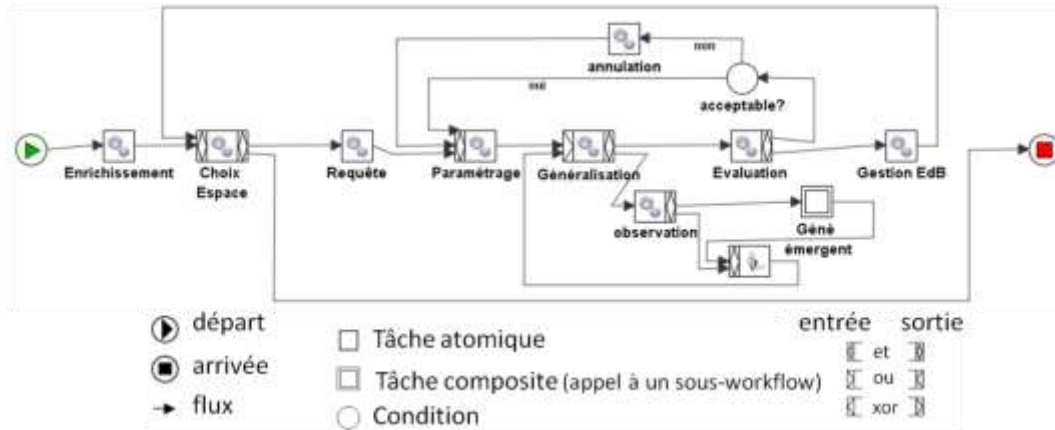


Figure D-17. Exemple de workflow pouvant servir d'alternative à la perspective SMA pour mettre en œuvre CollaGen. Le formalisme utilisé est celui de YAWL (Van der Aalst et ter Hofstede, 2005).

Comparaison de la perspective SMA à la perspective par Workflow

CollaGen est un modèle dont l'objectif est de réaliser une généralisation collaborative de manière pas à pas locale. Il est donc important de pouvoir adapter la décision de généralisation aux particularités de chaque situation : par ses capacités d'autonomie dans la prise de décision locale et de réactivité par rapport à la perception du voisinage, la perspective SMA de CollaGen convient bien mais convient-elle mieux que la perspective simple par workflow ? Tout d'abord, les capacités d'autonomie des agents permettent de distribuer des mécanismes de décision légèrement différents suivant les situations : un agent espace urbain et un agent espace réseau routier peuvent ne pas s'évaluer de la même manière dans notre perspective SMA alors que la tâche d'évaluation du workflow se doit d'avoir un mécanisme générique pour traiter tous les types d'espaces. De même, deux processus peuvent ne pas se mettre en cohérence de la même manière et l'autonomie permise par la perspective SMA nous permet cette flexibilité (il faudrait une tâche par type de processus et des flux plus complexes pour permettre cette flexibilité dans un workflow). En outre, la réactivité des agents dans notre perspective SMA permet une adaptation plus aisée aux cas particuliers pouvant intervenir dans la généralisation d'une carte topographique. Par exemple, dans la gestion du voisinage et des effets de bords, les principes de réactivité permettent à un agent espace de réagir à un cas particulier d'effet de bord perçu dans son voisinage, de manière plus spécifique que dans notre proposition par workflow. En effet, il faudrait prévoir par avance tous les cas de figure d'effets de bord pour permettre une adaptation identique aux situations particulières. De même, la possibilité d'envoyer des messages (même si elle n'est pas gérée dans notre mise en œuvre par manque de temps) permet une expressivité plus forte dans la réactivité qu'un workflow assez générique : face à une situation imprévue, l'agent peut envoyer un message pour qu'on l'aide à trouver une solution (ou juste pour dire qu'il n'y est pas arrivé).

D'autre part, la perspective SMA de CollaGen propose un intérêt non négligeable dans la facilité de modélisation, d'implémentation et d'évolution d'un prototype de CollaGen. En effet, les SMA sont un domaine très bien balisé car de nombreux modèles existent pour décrire les comportements, les modes de communication des agents. Il est par exemple très aisé de modéliser un agent registre en s'inspirant des normes et modèles existants dans la littérature. Les SMA ont donc l'intérêt de faciliter la modélisation des composants de CollaGen dérivant du domaine des SMA. En outre, la généralisation cartographique est un domaine de recherche appliquée et il est donc important de concevoir des prototypes pour mettre en pratique ses idées. L'implémentation doit donc être la plus facile possible. L'implémentation des séquences concurrentes de la Figure D-16 (même s'il s'agit plus de passages de relais que de travail en parallèle) est assez facile à implémenter par un SMA : les actions effectuées par un agent sont implémentées dans son cycle de vie et quand il doit passer la main, l'agent met simplement en pause son exécution pour permettre l'exécution d'un autre agent du système. Les SMA sont enfin une solution qui facilite les évolutions car il est plus facile de modifier ou ajouter un comportement d'un agent que d'apporter une modification à un workflow pour lesquels un petit changement peut nécessiter de repenser le système dans son ensemble. Par exemple, la modification du mécanisme d'émergence d'espaces peut se résumer à l'ajout d'un comportement sur l'agent observateur, alors que dans le cas du workflow, cela peut nécessiter la création de nouvelles tâches du workflow et une modification de la connexion entre les tâches. Or, nous avons vu que CollaGen tend à être un système en constante évolution. Ajouter de nouveaux processus, de nouveaux algorithmes dans la boîte à outils doit être facile et ne pas remettre en question tout le système.

En résumé, notre choix d'une perspective SMA pour CollaGen par rapport à une perspective par workflow s'appuie sur le besoin d'autonomie et de réactivité des composants du fait des spécificités de la généralisation collaborative. De plus, cette perspective SMA permet de faciliter la modélisation du système, son implémentation et ses futures évolutions.

D.4.3. Autres considérations SMA

Après avoir discuté de l'intérêt d'une perspective SMA pour CollaGen par rapport à d'autres solutions, dans cette partie, nous discutons deux considérations classiques dans la conception d'un système multi-agent : la convergence générale du système et sa complexité.

CollaGen dans sa perspective SMA peut se résumer en une série de cycles dans lesquels un agent espace, désigné par le chef d'orchestre, cherche à se généraliser au mieux en faisant appel à un processus par l'intermédiaire du registre. A l'intérieur de chaque cycle, plusieurs processus peuvent être appliqués itérativement, donc, il faut faire attention à la convergence à l'intérieur d'un cycle (comment un espace arrête d'essayer de se généraliser) et à la convergence des cycles de généralisation collaborative (comment le chef d'orchestre arrête de désigner des espaces pour la généralisation). Il faut donc s'assurer dans un premier temps de la convergence du cycle de vie d'un espace géographique. Deux points peuvent poser problème dans la convergence de ce cycle de vie : l'émergence de sous-espaces et le choix d'un processus tant que l'état global des moniteurs n'est pas suffisant. Concernant l'émergence de sous-espaces, nous avons limité le cycle de vie des espaces émergents pour éviter un phénomène récursif (l'espace émergent n'est généralisé qu'une fois par un processus). De plus, l'observateur n'observe plus les zones dans lesquelles un espace a

déjà émergé empêchant l'observation à l'infini de nouveaux espaces émergents (l'espace observé n'est pas infini). Ainsi, l'observation d'espaces émergents n'entraîne pas de boucle potentiellement infinie mais ne fait qu'allonger le déroulement de la généralisation. Concernant le nombre de généralisations d'un même espace, nous n'avons pas placé de garde-fou théorique. Nous avons simplement limité à un le nombre des applications possibles d'un même processus sur un espace (il faut noter que deux méthodes de paramétrage d'un processus signifient dans CollaGen que deux processus sont disponibles, ce qui autorise à essayer une autre fois un processus avec un paramétrage différent). Ainsi, un espace ne peut pas être généralisé plus de fois qu'il n'existe de processus disponibles adaptés à lui (selon la liste fournie par le registre). Ce nombre de processus disponibles adaptés n'étant pas infini (et généralement assez petit), le cycle de généralisation collaborative d'un espace finit par se terminer, même si l'état n'est pas complètement satisfaisant.

Si nous nous intéressons maintenant à la convergence à l'extérieur des cycles de généralisation d'un espace, deux nouveaux problèmes se posent : celui de la validation d'un état et celui du choix du prochain espace par le chef d'orchestre (il choisit un espace à généraliser à chaque itération). Dans le premier cas, il peut y avoir des problèmes de convergence si le chef d'orchestre est trop strict et qu'il revient trop souvent en arrière : il risque de ne pas trouver le meilleur état global des moniteurs possible avec les processus disponibles. Dans notre prototype, nous avons opté pour une méthode qui ne revient en arrière que si l'état global est significativement pire que l'état initial par la différenciation des trois types d'évaluation. Pour aller plus loin, il serait nécessaire de comparer plusieurs méthodes et d'évaluer leur impact sur le temps de convergence du système et la qualité du résultat final obtenu. Pour le deuxième cas, le système peut être confronté à un problème de blocage, il n'arrive pas à proposer le bon espace pour faire avancer la généralisation : ce problème peut notamment intervenir à la fin d'une des étapes de généralisation comme la « sélection » ou la « généralisation cartographique ». Nous proposons de nous appuyer sur les règles d'orchestration (C.8) pour nous assurer que le chef d'orchestre ne va pas rester bloqué à une étape. De la même manière que nous proposons des outils de vérification de la cohérence de la base de règles d'orchestration, il serait pertinent d'ajouter des outils permettant de qualifier la convergence globale du chef d'orchestre, *a priori*, d'après les règles présentes dans la base, ce que nous n'avons pas fait par manque de temps.

Après la convergence du système, le deuxième point à discuter concernant notre perspective SMA pour CollaGen est la complexité en termes de difficulté du paramétrage. CollaGen est un système assez complexe car il vise à l'automatisation de tâches de généralisation assez variées ce qui implique l'utilisation de ressources très diverses et de nombreux mécanismes possédant leurs propres paramètres. En effet, en plus des différentes ressources qui guident une généralisation par CollaGen comme les spécifications de la carte sous la forme de contraintes et règles formelles, les processus disponibles et leur description formelle, le contenu de la boîte à outils, l'ontologie ou les règles d'orchestration, d'autres paramètres peuvent faire varier fortement le résultat de la généralisation. Par exemple, le comportement de l'agent observateur est guidé par plusieurs paramètres comme le nombre et la proximité des conflits persistants pour constituer un espace émergent. De même, l'agent espace peut être paramétré en lui affectant différentes méthodes pour évaluer l'état global de ses moniteurs. CollaGen a donc le défaut d'être assez lourd à paramétrer mais il a l'avantage d'éviter le paramétrage précis de chacun des processus (qui est souvent assez complexe) par l'intermédiaire des fonctions de traduction.

Pour conclure ce chapitre, faisons un bilan des points clés abordés. Nous avons d'abord démontré que mettre en œuvre CollaGen par un Système Multi-Agents permettait d'avancer vers l'implémentation d'un prototype. Plusieurs types d'agents interagissent dans cette perspective pour tendre vers l'objectif partagé de l'obtention d'une carte complètement généralisée, en se partageant les capacités des composants de CollaGen :

- Les *agents géographiques*, proches des agents géographiques de (Ruas et Duchêne, 2007) à la différence qu'ils ne se généralisent pas eux-mêmes.
- Ils font plutôt appel aux *agents processus* qui peuvent se paramétrer eux-mêmes pour traiter au mieux les objets géographiques contenus dans un espace.
- Les agents processus sont mis en relation avec les agents espaces par l'intermédiaire de *l'agent registre*.
- *L'agent observateur* se concentre sur les généralisations en cours pour identifier, à travers l'analyse des moniteurs de contrainte, l'apparition de nouveaux agents espaces émergents. Il communique avec l'agent processus pour coordonner les phases d'observation et de généralisation afin d'éviter des incohérences.
- Enfin, *l'agent chef d'orchestre* communique avec les autres agents pour orchestrer leurs actions.

Nous avons démontré que ces agents peuvent bien être considérés comme des agents, et que cette perspective SMA apporte plus qu'une perspective, plus littérale, par Workflow.

CHAPITRE E. IMPLEMENTATION ET RESULTATS

Nous avons décrit dans les chapitres précédents le cadre générique de la généralisation collaborative, le modèle CollaGen et la modélisation orientée agents de CollaGen, du plus générique au plus proche de l'implémentation. Dans ce dernier chapitre de notre thèse, nous allons présenter l'implémentation d'un prototype de CollaGen et les résultats que nous avons obtenus à l'aide de ce prototype. Dans un premier temps, nous décrivons les développements connexes au prototype de CollaGen qu'ils soient de l'ordre de l'algorithme pour remplir la boîte à outils ou de l'ordre du processus de généralisation à rendre disponible. Puis, nous présentons l'implémentation du prototype dont les outils développés pour faciliter les interactions des différents rôles (modélisateur, développeur, fournisseur et utilisateur) avec le prototype. Nous présentons ensuite le cadre des tests effectués sur le prototype en décrivant les données utilisées et les ressources saisies (contraintes, descriptions, ontologie...). Nous présentons ensuite une série de résultats sur le fonctionnement des composants individuellement et du prototype dans son ensemble. Nous présentons, dans la partie suivante, une évaluation de ces résultats, en ayant notamment recours à des comparaisons avec d'autres méthodes de généralisation. Dans une dernière partie, nous discutons les résultats et leur évaluation.

E.1. DEVELOPPEMENTS CONNEXES A COLLAGEN

Dans cette première partie du chapitre, nous présentons une part importante de l'implémentation nécessaire à l'utilisation du prototype de CollaGen en décrivant les outils, algorithmes et processus que nous avons développés pour alimenter la boîte à outils du prototype. Dans un premier temps, nous présentons notre plateforme de travail. Puis les processus de généralisation utilisés, en insistant sur le processus de généralisation par moindres carrés que nous avons développé à partir du modèle de (Harrie et Sarjakoski, 2002). Nous présentons, dans les parties suivantes, différents algorithmes de création d'espaces géographiques que nous avons développés durant la thèse. Nous terminons cette partie par la présentation d'un processus de correction d'effets de bord par moindres carrés.

E.1.1. Description de notre plateforme de travail

Le prototype de CollaGen que nous avons implémenté a été intégré à la plateforme de travail de l'équipe de généralisation du COGIT afin de profiter des travaux précédents et de faciliter l'intégration des travaux de recherche dans les services de production de l'IGN. Cette plateforme logicielle est construite autour du logiciel de généralisation Clarity™ de la société 1Spatial™. Ce

logiciel est la nouvelle version du logiciel issu du projet européen AGENT (Barrault et al, 2001) et fournit donc une version du modèle de généralisation AGENT. Clarity fournit également une version du modèle des Beams (Bader et al, 2005). Il est donc très facile de fournir à notre prototype de CollaGen des processus basés sur ces deux modèles déjà intégrés à la plateforme. De plus, ce logiciel est construit sur un système de gestion de base de données orienté objet ce qui est très utile pour modéliser et stocker les enrichissements nécessaires à la généralisation collaborative, comme l'identifiait déjà (Buttenfield, 1995) pour les enrichissements, dans le cas de généralisation en général.

Par dessus cette souche logicielle, la plateforme de travail a été enrichie (par des prédécesseurs au laboratoire, puis, par nous) des nombreux travaux de recherche précédents de l'équipe de généralisation, qu'ils soient anciens et implémentés en Lull, langage propriétaire de 1Spatial ou plus récents et implémentés en Java. Nous pouvons citer notamment des travaux sur l'analyse de la ville (Boffet, 2001), le modèle de généralisation CartACom (Duchêne, 2004), l'analyse et la sélection des routes (Touya, 2010). De plus, de nombreux algorithmes et mesures d'analyse spatiale permettent de remplir la boîte à outils de CollaGen, notamment en ce qui concerne la mesure de la satisfaction des moniteurs de contrainte.

Depuis plusieurs mois, l'équipe de généralisation a également entamé la conception d'une nouvelle plateforme entièrement en Java qui doit intégrer de manière plus efficace l'ensemble des travaux passés de l'équipe et notamment les trois modèles multi-agents AGENT, CartACom et GAEL. Cette plateforme, CartAGen, permet déjà de réaliser des processus de généralisation basés sur le modèle AGENT (Renard et al, 2010). CartAGen s'appuie sur le SIG Open Source GeOxygene (Bucher et al, 2011) développé par le COGIT à partir des normes de l'Open Geospatial Consortium. Nous avons ainsi développé une interface pour relier ces deux plateformes (Clarity et CartAGen). Ainsi, notre plateforme de travail peut faire appel à des processus ou des outils de CartAGen. Ceci enrichit fortement les possibilités de fournir des processus de généralisation sur la plateforme de travail, ce qui est une des hypothèses de notre thèse (nous ne considérons pas les processus de généralisation appelés par service Web par exemple).

E.1.2. Un Modèle de généralisation par moindres carrés

Afin d'enrichir notre plateforme de travail avec un processus de généralisation assez différent de ceux déjà disponibles, nous avons décidé de développer un modèle de généralisation par moindres carrés et d'en dériver plusieurs processus de généralisations. En effet, la généralisation par moindres carrés est très bien adaptée à la généralisation graphique (Harrie et Sarjakoski, 2002) et est en cela très complémentaire des modèles dont nous disposons déjà par les travaux antérieurs du laboratoire. Nous nous sommes appuyé sur les modèles par moindres carrés existants (Harrie et Sarjakoski, 2002 ; Sester, 2005) que nous avons décrits en A.2.3. Les moindres carrés sont une méthode mathématique pour résoudre un système d'équations sur-contraint (qui possède plus d'équations que d'inconnues) et qui n'a donc pas de solution exacte. La solution approchée obtenue par moindres carrés minimise la somme des carrés des résidus (le résidu d'une équation étant l'écart par rapport à la solution exacte de l'équation). Dans cette partie, nous décrivons notre modèle et son implémentation, les processus que nous avons dérivés de ce modèle et enfin quelques résultats obtenus avec ces processus.

Notre modèle de généralisation par moindres carrés se fonde principalement sur celui présenté dans (Harrie, 1999 ; Harrie et Sarjakoski, 2002) : la généralisation des objets géographiques est effectuée en déplaçant les points qui composent leur géométrie. Le système d'équations qui va être résolu par moindres carrés a pour inconnues les écarts, entre les situations initiale et finale, des coordonnées x et y de chacun des points : le système comporte $2n$ inconnues pour n points dans les données. La Figure E-1 illustre sur un bâtiment simple comment mettre en équation la généralisation pour une résolution par moindres carrés.

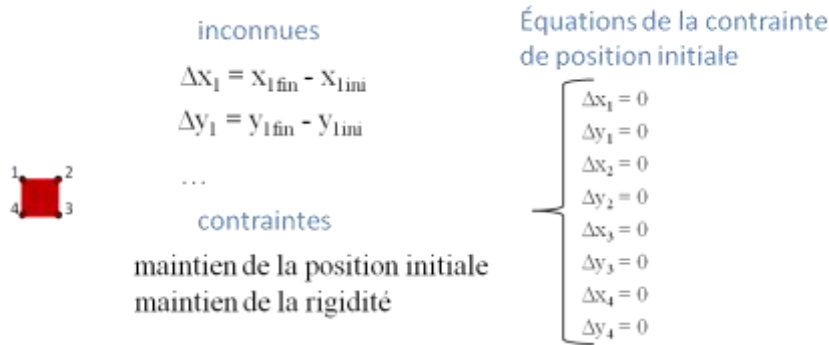


Figure E-1. Comment mettre le problème de la généralisation sous la forme d'équations linéaires en prenant ici en compte le maintien des objets dans leur position initiale.

Pour résoudre le système complet tenant compte de l'ensemble des contraintes sur l'ensemble des objets, nous représentons le système sous forme matricielle. Généraliser par moindres carrés revient alors à résoudre l'équation matricielle suivante :

Où A est la matrice représentant les équations dérivées des contraintes, P est la matrice des poids, O est le vecteur des observations (i.e. une observation est le terme de droite de l'équation, dans $\Delta x = 0$, l'observation vaut 0) et X est le vecteur des déplacements optimaux de chaque point pour satisfaire les équations. Le modèle repose sur la définition de contraintes internes et externes définies sur les objets à généraliser qui sont transformées en équations sur les coordonnées des points. Une contrainte interne est une contrainte qui concerne uniquement des points d'un même objet (plutôt donc des contraintes pour maintenir la forme initiale de l'objet). Une contrainte externe est une contrainte qui concerne des points appartenant à plusieurs objets (par exemple, une contrainte de proximité minimale entre deux bâtiments en est une). Un poids est associé à chacune des contraintes, poids qui est alors reporté dans la matrice des poids sur les lignes correspondant aux équations issues de la contrainte. Trois types d'objets sont définis, les objets fixes (qui ne doivent pas bouger durant la généralisation), les objets rigides (comme les bâtiments, qui ne peuvent être que déplacés) et les objets malléables qui peuvent être déformés (comme les routes ou les forêts). Définir un objet géographique comme un objet de tel type revient à lui affecter un certain type de contraintes internes : par exemple, si un bâtiment est un objet rigide, le modèle lui attribue une contrainte de position initiale (les points doivent garder leur position initiale) et une contrainte de rigidité (le vecteur entre deux points consécutifs doit rester le même).

Ce modèle a été implémenté sur notre plateforme de travail pour pouvoir être exécuté sur des données affichées à l'écran et pour être spécialisé facilement en processus de généralisation. Nous avons donc implémenté un certain nombre de contraintes internes et externes qui peuvent être

utilisées avec ce modèle : deux contraintes externes traitant de proximité d'une manière différente (par triangulation ou en utilisant des buffers) et sept contraintes internes pour maintenir la forme des objets rigides et malléables. Un scheduler a été implémenté et enchaîne les étapes de création des contraintes sur les objets à généraliser, de traduction en équations, d'assemblage des équations pour former une unique matrice, de résolution de l'équation matricielle (effectuée à partir de la bibliothèque Java de calcul matriciel Jama) et enfin, d'application des résultats sur les objets. Des spécifications formelles ont également été développées pour faciliter et stocker le paramétrage d'une généralisation par moindres carrés, spécifications que l'on peut saisir à partir d'une IHM ad-hoc (Figure E-2). Enfin, notre implémentation propose deux options de généralisation pour les objets linéaires situés en bord de zone généralisée : soit le système en fixe les extrémités (pour éviter des incohérences avec les objets non généralisés), soit il propage les modifications aux objets topologiquement voisins en utilisant la diffusion par les snakes (Burghardt et Meier, 1997) présente dans le logiciel Clarity™, socle de notre plateforme.

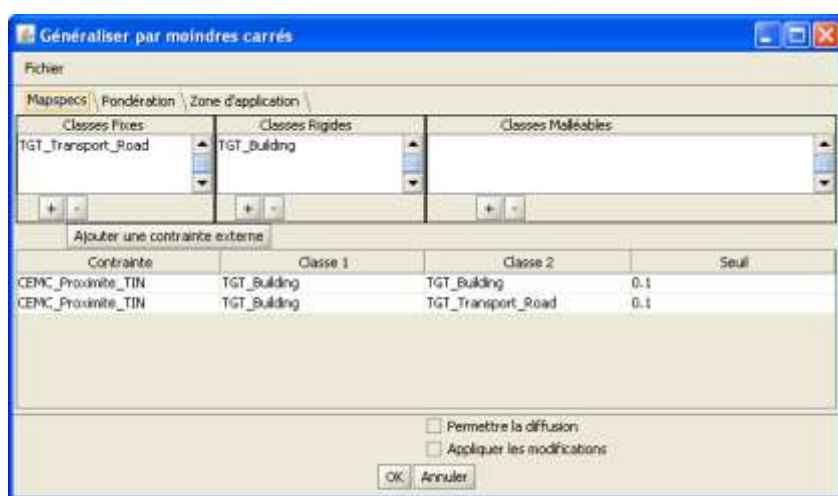


Figure E-2. Outil de saisie des spécifications formelles pour une généralisation par moindres carrés.

A partir de ce modèle et de son implémentation, trois processus ont été conçus : un processus dédié aux bâtiments en zone urbaine, un processus dédié aux zones périurbaines et rurales et un processus dédié aux réseaux. Dans chacun des cas, mettre en œuvre un processus revient à choisir les concepts (de l'ontologie) à généraliser, à les qualifier (les bâtiments sont rigides, les routes et les cours d'eau malléables par exemple) et à choisir les pondérations des contraintes ; le paramétrage intervient alors au niveau du choix des contraintes externes et de leur seuil. Pour illustrer ceci, le processus dédié à l'urbain ne considère que les bâtiments et autres équipements urbains (e.g. terrains de sport) comme objets rigides et les routes et cours d'eau comme objets fixes. La contrainte externe de proximité entre bâtiments a un poids maximum, celle entre bâtiment et route dispose d'un poids très fort, la contrainte interne de rigidité est pondérée fortement et enfin, celle de position initiale a un poids assez faible.

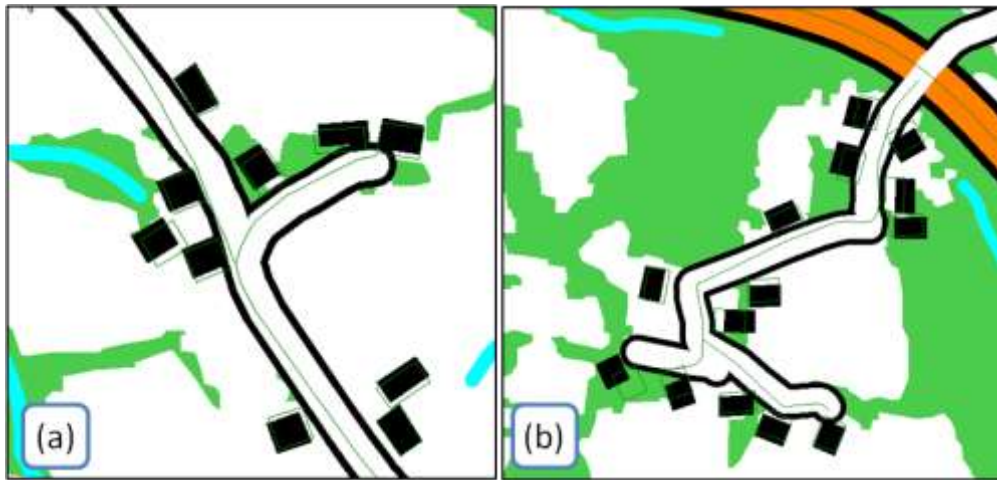


Figure E-3. Résultats obtenus avec le processus de généralisation par moindres carrés dédié aux zones rurales (les positions initiales sont représentées avec les traits fins).

La Figure E-3 montre deux exemples de résultats obtenus avec le processus de généralisation par moindres carrés dédié aux zones rurales pour une carte au 1 : 50000 avec les données de la BD Topo de l'IGN. Ce processus, contrairement à celui dédié aux zones urbaines, considère les routes, les chemins et les cours d'eau comme malléables (on voit bien sur la figure que les routes ont été déformées). La Figure E-4 montre un résultat obtenu sur le même type de données avec le processus dédié aux réseaux (pour lequel les poids des contraintes d'objets malléables sont adaptés pour maintenir au mieux la courbure des routes et des cours d'eau).

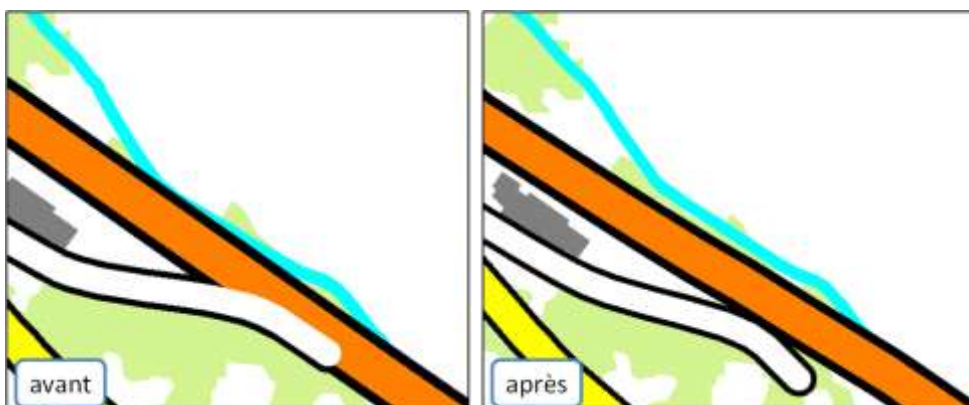


Figure E-4. Résultats obtenus avec le processus de généralisation par moindres carrés dédié aux réseaux.

La difficulté majeure pour une utilisation réussie d'un processus de généralisation par moindres carrés réside dans la définition de la matrice de pondération des équations (Harrie, 2003) et notre modèle, qui est une adaptation de celui de Lars Harrie, n'échappe pas à la règle. Dans notre modèle, la définition des poids est faite au niveau des contraintes internes et externes et traduite directement sur les équations. La Figure E-5 montre des résultats obtenus avec les mêmes contraintes et les mêmes données mais avec des poids différents sur les contraintes : une mauvaise pondération peut entraîner de mauvais résultats. Toutefois, dans une utilisation du modèle dans CollaGen, le choix des poids est fait par le fournisseur lors du passage de modèle à processus et l'utilisateur n'a pas à se soucier des poids.

Afin d'exploiter encore mieux ce modèle pour de nouveaux processus ou des processus plus efficaces, nous souhaitons définir plus de contraintes internes et externes pour gérer plus de

contraintes de généralisation qu'actuellement : il serait notamment très intéressant d'avoir des contraintes externes pour la préservation des relations topologiques comme 'le long de' ou 'traverse'. Nous pourrions également imaginer des contraintes internes qui requièrent de simplifier un objet et pas seulement de conserver ses caractéristiques. Il faudrait alors permettre de paramétrer plus facilement les contraintes internes associées à un type d'objet (rigide ou malléable) pour que la fonction de traduction associée au processus permette de déterminer celles qui sont utiles en fonction des spécifications formelles choisies par l'utilisateur.

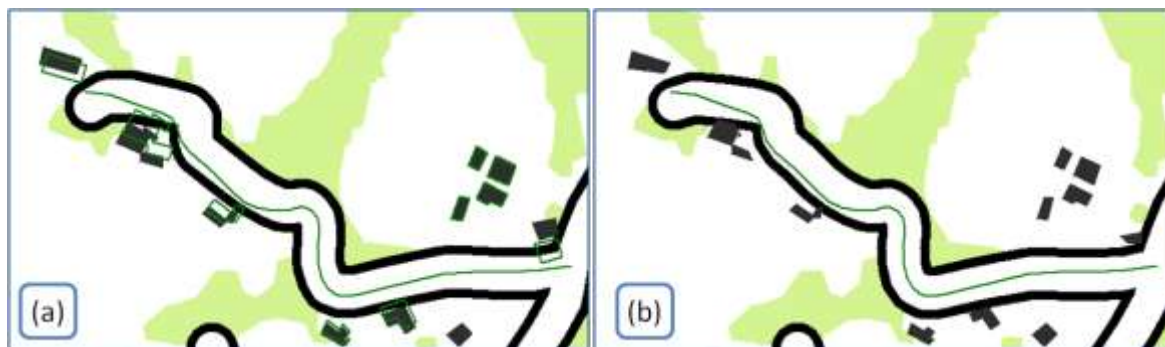


Figure E-5. Une même situation généralisée deux fois avec des poids différents sur les contraintes. (a) poids assez proches des optimaux empiriques. (b) poids très fort sur la contrainte de position initiale et faible sur la rigidité : les bâtiments sont déformés plutôt qu'éloignés de la route.

E.1.3. Amélioration de processus disponibles

A partir des modèles de généralisation disponibles sur notre plateforme de travail, nous avons réalisé des développements pour disposer de processus plus efficaces. Nous avons tout d'abord réalisé un processus fondé sur le modèle AGENT de CartAGen adapté à la généralisation des zones urbaines pour le 1 : 50000 et au-delà. Nous avons ainsi intégré l'algorithme de déplacement issu de (Ruas, 1998) aux actions suggérées par la contrainte de proximité entre bâtiments des agents meso (Figure E-6). Nous avons également implémenté un algorithme d'élimination des rues et des impasses issu de (Touya, 2010) qui peut être proposé par une contrainte de densité de rues sur l'agent meso ville (Figure E-6).



Figure E-6. (a) Algorithme de déplacement implémenté dans CartAGen appliqué sur un îlot urbain. (b) Algorithme d'élimination des rues et impasses appliqué sur le réseau routier d'une petite ville. Les situations initiales sont rappelées en traits bleus fins dans les deux cas.

Nous avons enfin développé une méthode pour déterminer les îlots de centre ville pouvant être grisés en zone bâtie car trop denses en bâtiments pour que ceux-ci soient représentés même généralisés. Celle-ci repose sur la méthode de décision multicritères ELECTRE TRI (Figueira et al, 2005) utile quand les critères que l'on veut utiliser sont nombreux et s'évaluent sur des échelles très différentes (Taillandier, 2008). La décision prise est de placer l'action (ici l'îlot de bâtiments) dans une des cinq catégories (de 'très mauvais' à 'très bon'), seuls les îlots 'très bons' étant gardés comme îlots de centre ville à griser. On peut donc qualifier cette méthode de classification supervisée. La méthode ELECTRE TRI permet notamment d'utiliser ensemble des critères dont on sait qu'ils n'ont pas le même impact sur la classification finale. Par exemple, nous avons utilisé un critère de présence d'une église dans l'îlot : nous avons paramétré ce critère pour que la présence effective d'une église influence fortement la classification finale (c'est très sûrement un îlot de centre ville), alors que la non présence n'aura que très peu d'impact sur la classification finale. De même, nous avons pu exprimer grâce à cette méthode que le critère de proximité aux bords de la ville diminuait la probabilité de se trouver en centre ville (la plupart des centres villes ne se trouvent pas aux bords de la ville) mais n'était jamais rédhibitoire (il existe des configurations de ville où le centre ville historique est au bord du fait de limites naturelles (fleuve, mer, lac) ou administratives (frontière)). Les critères que nous avons utilisés pour la détermination des îlots à griser sont les suivants :

- distance au centre de gravité de la ville (si l'îlot est près du centre il a plus de chance d'être dans le centre ville mais l'éloignement n'est pas rédhibitoire)
- présence d'une église
- taille de l'îlot par rapport aux îlots de la ville (les îlots de centre ville sont plutôt plus petits et un îlot très grand est vraiment rédhibitoire)
- densité en bâtiments (calculée en fonction de la surface des bâtiments et la largeur des symboles de routes)
- taille moyenne des bâtiments dans l'îlot par rapport à la moyenne de la ville (un îlot dont les bâtiments sont petits, donc *a priori* pavillonnaire, implique une non-classification comme îlot à griser, alors que des grands bâtiments n'indiquent pas forcément le centre ville)
- hauteur des bâtiments dans l'îlot par rapport à la moyenne de la ville (des bâtiments très bas ou très hauts ne sont pas des indicateurs de centre ville, du moins en France)
- proximité aux limites de la ville

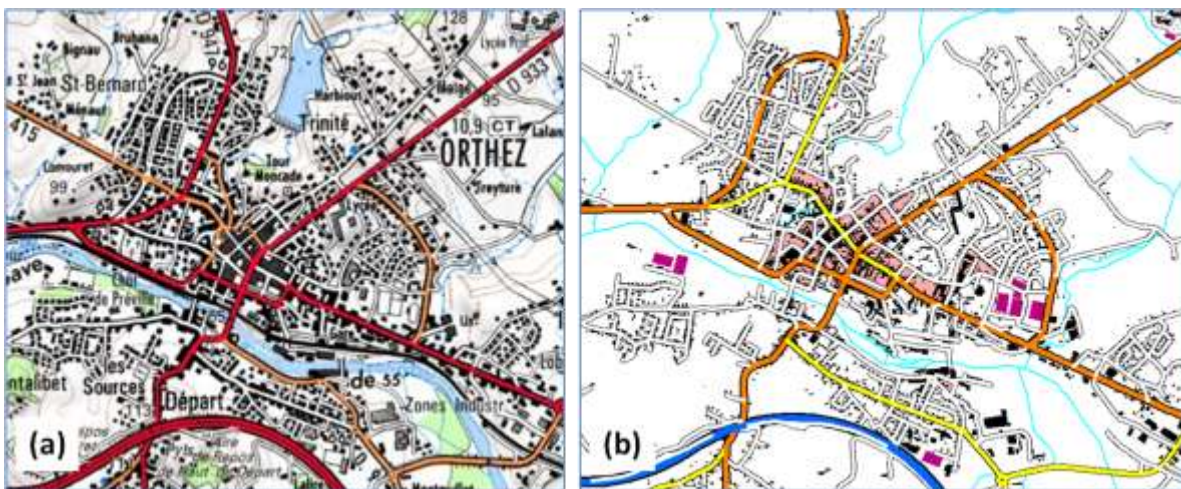


Figure E-7. Résultats obtenus par notre méthode de détermination multicritères des îlots à griser au 1 : 50000, comparés à une carte généralisée manuellement.

Nous avons déterminé les seuils utilisés dans ces critères de manière empirique puis testé l’algorithme sur des villes de différentes tailles et formes. La Figure E-7 montre les résultats obtenus sur la ville d’Orthez (64), en les comparant avec la carte IGN actuelle au 1 : 50000, généralisée manuellement.

Nous avons ensuite réalisé quelques adaptations pour porter un processus CartACom sur notre plateforme à partir du modèle développé (Duchêne, 2004) sur une version antérieure de Clarity™. Nous avons principalement modifié des paramètres pour adapter CartACom à des échelles proches du 1 : 50000 et également réalisé une interface pour appliquer le processus à différents types de données en entrée (la fenêtre courante, la sélection d’objets ou un espace géographique de CollaGen par exemple), indépendamment du schéma de données pour plus d’interopérabilités : comme pour les moindres carrés, nous avons conçu un modèle de spécification formelle que l’on peut éditer à partir des concepts de l’ontologie si on le souhaite (on peut toujours l’éditer avec les noms des classes) : un utilisateur peut choisir à quel type d’objet correspondent les types d’agents de CartACom, mais il peut également définir les contraintes qu’il souhaite appliquer à chaque type d’agent. Quelques erreurs ont également été corrigées. La Figure E-8 montre un exemple de résultat obtenu avec le processus CartACom ainsi porté sur notre plateforme de travail.

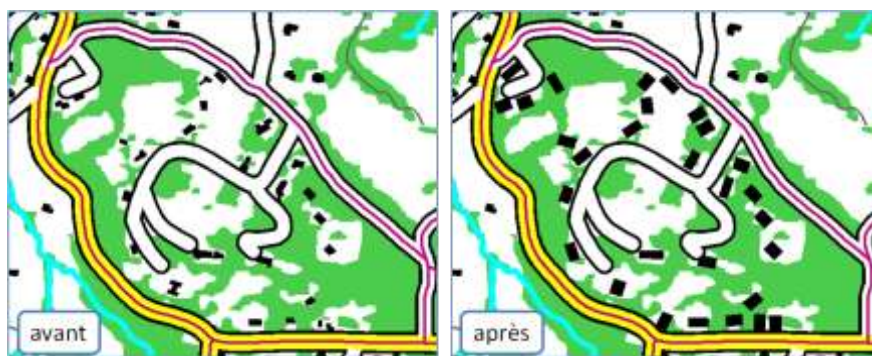


Figure E-8. Exemple de résultat obtenu avec notre processus CartACom au 1 : 50000 sur une zone rurale.

E.1.4. Des Processus spécialisés

A partir des algorithmes et développements disponibles sur notre plateforme de travail, nous avons conçu deux processus de généralisation assez spécialisés qui permettent de traiter des problèmes différents de ceux traités par les processus issus des modèles AGENT, CartACom, Beams ou Moindres Carrés. L’utilisation de ces processus permet également de tester certains aspects de CollaGen. Le premier de ces processus spécialisés est adapté à la simplification des carrefours routiers complexes composés de ronds-points et pattes d’oies. Le processus reprend des algorithmes que nous avons présenté dans (Touya, 2010) et simplifie ces carrefours en simples croisements entre routes en fonction de leur taille : les deux paramètres du processus sont donc l’aire minimum pour maintenir une patte d’oie sur la carte et le diamètre minimum pour maintenir un rond-point sur la carte. La fonction de traduction du processus cherche si des règles opérationnelles sont définies sur ces caractères précis et concernent une opération de type « squelettisation » (concept correspondant dans la taxonomie des opérateurs de généralisation). La Figure E-9 montre des résultats obtenus par l’application de ce processus sur l’espace thématique réseau routier avec les règles opérationnelles choisies pour nos tests (voir E.3.2).

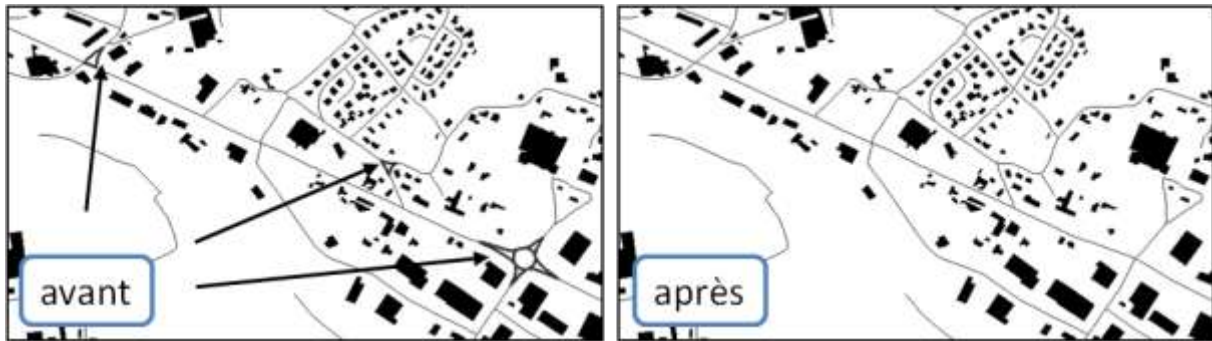


Figure E-9. Résultats obtenus sur le réseau routier par l'application de notre processus de simplification des carrefours complexes.

Le deuxième processus que nous avons conçu est dédié à la généralisation de la végétation. Inspiré des méthodes présentées dans (Mackaness et al, 2008 ; Touya et al, 2010), il permet de simplifier fortement la couche de végétation sur le niveau de détail des contours ou la taille des plus petits éléments et trous. Le processus est également capable d'éliminer les haies de la couche de végétation pour les représenter dans une couche spécifique après squelettisation (c'est de la généralisation de modèle d'après les grandes étapes que nous avons définies). La fonction de traduction s'appuie sur la recherche de différentes contraintes sur les caractères définis sur la végétation dans l'ontologie (aire, aire des trous, granularité du contour, largeur minimum). La Figure E-10 montre le résultat d'une application de ce processus avec comme paramètres ceux dérivés des contraintes formelles utilisées dans nos tests.

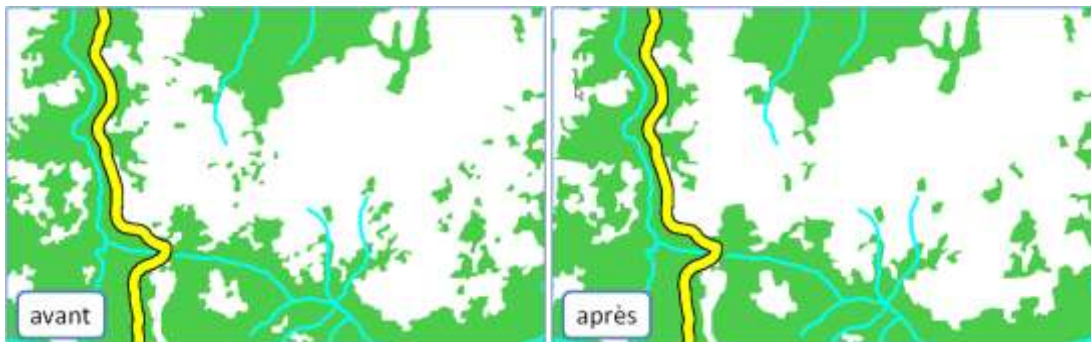


Figure E-10. Exemple d'application du processus de généralisation de la végétation.

Ces deux processus permettent de mettre en œuvre des processus de généralisation sur des espaces thématiques au contraire de processus issus des modèles AGENT, CartACom ou moindres carrés par exemple.

E.1.5. Algorithmes de création d'espaces géographiques

Nous avons vu en C.2 que CollaGen faisait appel à une boîte à outils composée d'algorithmes d'analyse spatiale pour créer les espaces géographiques et enrichir les données de structures meso et de processus capables de corriger des effets de bord. Plus (et mieux) cette boîte à outil est remplie, plus les possibilités de CollaGen sont étendues. Dans cette partie, nous présentons les principaux outils que nous avons développés pour enrichir la boîte à outils de notre prototype de CollaGen.

Création des espaces de montagne

Le premier outil que nous avons développé concerne la création des espaces géographiques de montagne. S'il existe de nombreux travaux sur la caractérisation du relief et ses formes particulières comme les crêtes ou les cols (Wood, 1996 ; Monier, 1997 ; Bonin et Poupeau, 2005, par exemple), la détection d'une entité géographique de haut niveau comme la montagne a été peu traitée. Le seul travail dont nous avons connaissance est celui de (Chaudhry et Mackaness, 2008c). La méthode proposée détermine les sommets d'une zone et cherche à partir des courbes de niveau l'étendue de leur zone de proéminence (zone dans laquelle ils sont le sommet le plus élevé). Si l'utilisateur ne garde que les sommets élevés et les plus proéminents, il se retrouve avec une zone de montagne hiérarchique (Figure E-11). Cette méthode permet de distinguer précisément toutes les collines et leur chaîne.

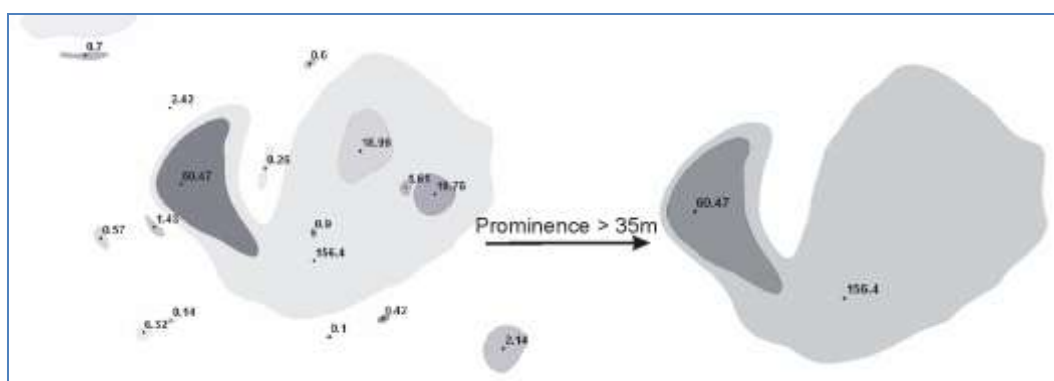


Figure E-11. Caractérisation des zones de montagne et hiérarchisation par la proéminence (Chaudhry et Mackaness, 2008c).

Notre vision d'un espace de montagne est moins précise et concerne de grandes zones dans lesquelles les pentes et l'altitude (ou plutôt la proéminence) sont élevées. La Figure E-12 montre un exemple sur la partie basque des Pyrénées de zone de montagne que l'on souhaite établir avec en comparaison un résultat très précis obtenu avec notre implémentation de la méthode de (Chaudhry et Mackaness, 2008c). Si l'on souhaite avoir un résultat plus grossier et sur des zones étendues, cette méthode devient vraiment trop coûteuse en temps de calcul, donc, nous décidons de développer une méthode plus grossière et plus rapide.



Figure E-12. À gauche, des espaces de montagne dessinés grossièrement à la main sur le côté basque des Pyrénées. À droite, les zones de montagnes précises obtenues par la méthode de (Chaudhry et Mackaness, 2008c) avec un zoom sur une petite partie.

La méthode que nous proposons est inspirée de (Walter, 2008) et repose sur l'utilisation d'une grille carrée que l'on plaque sur nos données (Figure E-13). (Walter, 2008) propose une méthode générique (appliquée à la détection des zones urbaines) pour réaliser une classification de chaque cellule de la grille selon un certain nombre de critères. Pour chacun des critères, l'algorithme attribue à la cellule une valeur de 1, 2 ou 3 suivant la valeur des données dans cette cellule. Il fait ensuite pour chaque cellule la somme pondérée (un critère peut être plus important qu'un autre et la somme des poids fait 1) des classes pour chaque critère ce qui nous donne une valeur de classe globale réelle pour la cellule comprise entre 1 et 3. Cette moyenne pondérée peut se justifier en reprenant les caractéristiques nécessaires à l'utilisation de méthodes multicritères plus complexes comme ELECTRE TRI : ces méthodes ne sont pertinentes que lorsque le nombre de critères est assez élevé (au moins cinq et que ceux-ci s'interprètent de manière très différente) (Taillandier, 2008). A l'inverse, dans le cas de critères assez comparables (c'est une contrainte pour choisir nos critères) et peu nombreux, une somme pondérée est tout aussi pertinente. Par rapport à la méthode générique de (Walter, 2008), nous avons introduit principalement des critères spécifiques aux zones de montagne, une méthode de simplification du contour et une amélioration de la méthode de regroupement des cellules en clusters représentant une zone homogène (une cellule voisine d'une zone de montagne est classée presque en zone de montagne, alors que ses autres voisins sont classés comme plaine, est ajoutée au cluster de la zone de montagne).

Les deux critères que nous avons utilisés sont la dénivelée et la densité en courbes de niveau. Par exemple, la Figure E-13 représente le critère de dénivelée: l'algorithme cherche dans un cercle de rayon donné (c'est un paramètre de l'algorithme) les courbes de niveau et il compare la plus haute à la plus basse. Dans ce cas, avec une valeur de dénivelée de 240 m, l'algorithme va attribuer une classe 3 (la plus haute) pour ce critère à cette cellule. Une fois l'ensemble des cellules classées, l'algorithme garde celles dont la classe est supérieure à un seuil (autre paramètre de l'algorithme) et agrège les cellules restantes qui sont contiguës. Les coins à angle droit sont lissés par une opération morphologique de fermeture, c'est-à-dire l'enchaînement d'une opération de dilatation puis d'érosion de même valeur (Serra, 1983). Une amélioration serait d'ajouter en suivant une ouverture (opération inverse de la fermeture) pour rendre le contour encore plus lisse. Enfin, l'algorithme élimine les zones trop petites.

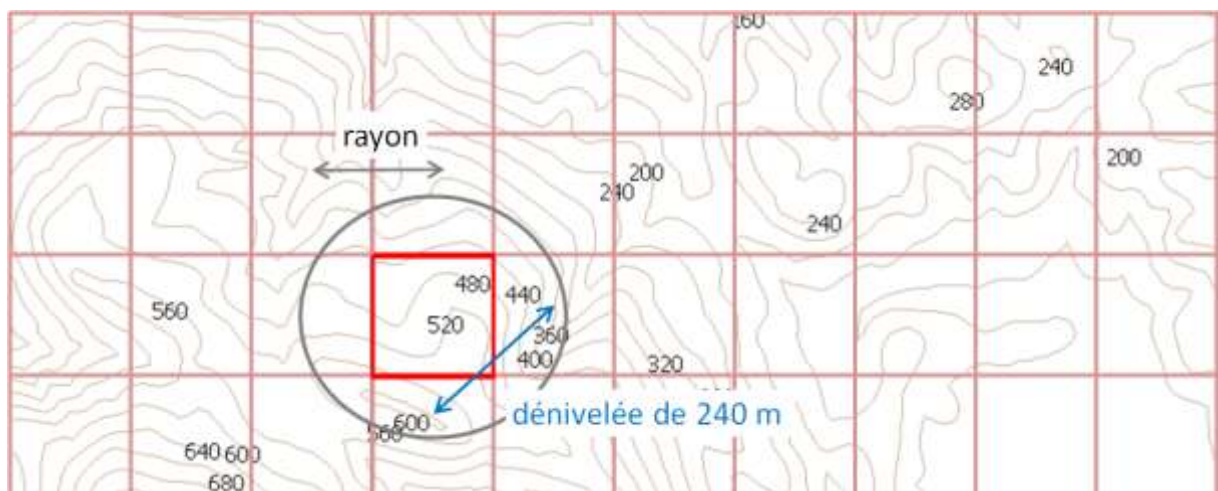


Figure E-13. Détection des montagnes par grille et critère de dénivelée : chaque cellule de la grille est caractérisée en fonction de la plus grande dénivelée entre les courbes situées dans un cercle de rayon donné (paramètre de l'algorithme) et centré sur la cellule.

Cette méthode avec deux critères de dénivelée et densité en courbes de niveau donne des résultats très satisfaisants avec un temps de calcul relativement faible. Seulement les paramètres sont nombreux : taille des cellules, rayon de recherche, deux seuils par critère pour déterminer la classe de la cellule (si la valeur est inférieure au seuil bas, la classe est 1, si elle est supérieure au seuil, la classe est 3), le seuil de classe globale à partir duquel l'algorithme considère que la cellule est une montagne, le poids des critères et enfin le seuil d'élimination des petites zones. Nous avons donc dû réaliser de nombreuses expérimentations pour déterminer l'ensemble de ces paramètres de manière empirique. La Figure E-14 montre de bons résultats sur la même zone des Pyrénées et quelques variations quand nous bougeons les paramètres. Suivant l'objectif, ces zones de montagne peuvent donc englober des parties plus ou moins vallonnées. Nous pouvons noter que cette méthode a été utilisée (avec à peu près les mêmes paramètres que nous) pour détecter les zones dans lesquelles les routes sont susceptibles d'être très déformées par la généralisation, ceci afin d'estimer l'erreur sur la longueur d'une route engendrée par le processus de généralisation (Girres, 2011).

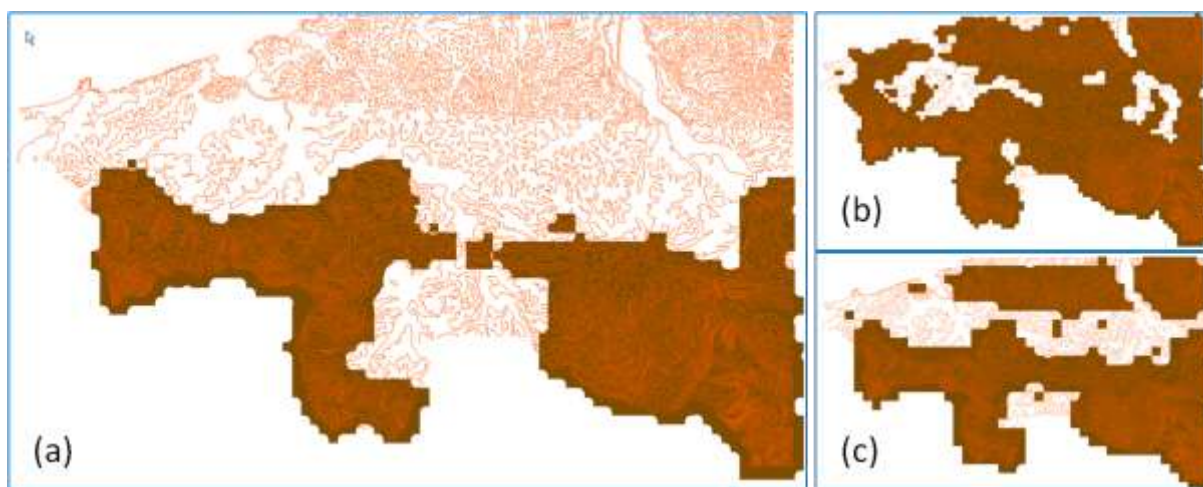


Figure E-14. Résultats de notre méthode de création de zones de montagne sur le côté basque des Pyrénées. (a) résultat choisi comme le meilleur. (b) résultat en étant plus tolérant dans les critères. (c) résultat avec des cellules deux fois plus grandes.

Création des espaces périurbains et ruraux

En géographie, la notion d'espace rural n'est plus directement liée à l'activité agricole mais désigne généralement tout ce qui n'est pas urbain (Braud et al, 1997). Avec l'évolution récente de l'urbanisation, les géographes ont vu apparaître un espace intermédiaire, moins dense que l'espace urbain et assez hétérogène dans sa dispersion que nous pouvons qualifier de périurbain quand il entoure un espace urbain ou de rurbain sinon (Braud et al, 1997). Les particularités de ces deux espaces en font des espaces clés qu'il faut identifier par un algorithme d'analyse spatiale. En effet, les zones périurbaines sont susceptibles de contenir à la fois des petites zones très denses (centre de bourgs) et des zones pavillonnaires composées d'alignements de bâtiments qu'il va falloir maintenir au mieux dans la généralisation. Concernant les zones rurales, les bâtiments représentés sur la carte sont plus rares et servent donc de points de repère à la lecture de la carte. Le maintien des relations géographiques avec les objets alentours comme l'orientation relative ou le positionnement par rapport à la route deviennent très importants à conserver et nous faisons l'hypothèse que les traiter nous donnera de meilleurs résultats.

Comme l'espace urbain, l'espace périurbain ou rurbain (nous traitons les deux de manière identique) est caractérisé par la présence de bâtiments représentant principalement du logement résidentiel individuel et des exploitations agricoles mais nous retrouvons aussi des bâtiments industriels ou commerciaux (les centres commerciaux sont souvent situés en périphérie). Comme pour la détection des zones urbaines que nous effectuons en utilisant l'algorithme issu de (Boffet, 2001), nous allons utiliser les bâtiments pour identifier les zones périurbaines. Ainsi, dans (Boffet, 2001), la géométrie des bâtiments est dilatée et ces empreintes dilatées sont assemblées quand elles s'intersectent (Figure E-15). Le rayon de dilatation utilisé pour les zones périurbaines est bien plus grand que pour les zones urbaines (75 m contre 25 m) car les bâtiments sont plus éloignés les uns des autres.

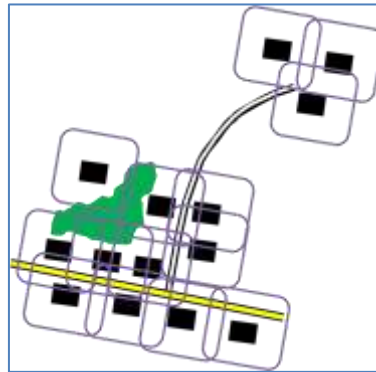


Figure E-15. Première étape de l'identification des espaces périurbains en assemblant les empreintes dilatées des bâtiments.

Comme dans le cas des zones urbaines, un certain nombre d'opérations sont appliquées sur les surfaces obtenues : nous éliminons les surfaces trop petites ; nous simplifions le contour par des opérations morphologiques (Serra, 1983) comme la fermeture et l'ouverture et par un filtrage (Douglas et Peucker, 1973) ; nous supprimons enfin les trous contenus dans ces surfaces car nous considérons qu'une zone périurbaine peut contenir un bois par exemple, sans que celui-ci en soit exclu ; enfin, nous enlevons, comme un emporte-pièce, les zones urbaines contenues dans nos espaces périurbains. La Figure E-16 montre des espaces périurbains construits autour de l'espace urbain de la ville d'Orthez et qui correspondent bien à nos attentes.

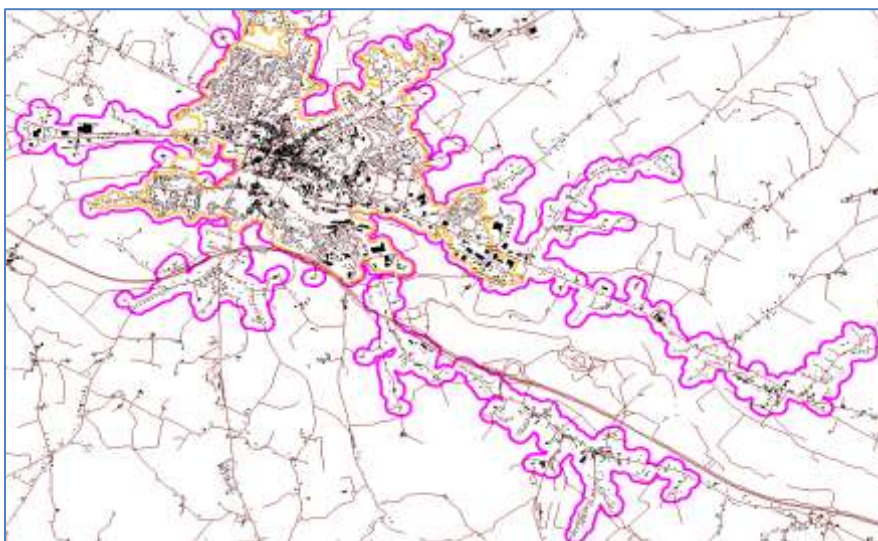


Figure E-16. Espace périurbain (en magenta) autour de l'espace urbain d'Orthez (64) (en orange).

En reprenant la définition de l'espace rural, nous considérons comme espace rural tout ce qui n'est ni urbain ni périurbain. Comme on le voit sur la Figure E-16, ces espaces peuvent être très étendus donc nous décidons de les découper à partir du réseau routier principal.



Figure E-17. Une méthode alternative pour les espaces ruraux : seules les zones autour des bâtiments sont concernées.

Nous proposons également, dans la boîte à outils, une méthode alternative pour déterminer les espaces ruraux en ne gardant, dans les espaces précédents, que les zones bâties que l'on détermine de la même manière que les espaces périurbains sans éliminer les petites zones (Figure E-17). Cette méthode a le défaut d'être centrée sur les bâtiments et s'éloigne donc de la définition géographique mais peut être bien adaptée si les processus dédiés à la généralisation des espaces ruraux se concentrent aussi sur les bâtiments et moins sur d'autres thèmes comme l'orographie, l'hydrographie ou les zones agricoles. Toutefois, en prenant une bande de terrain assez grande autour des bâtiments, ces zones rurales incluent tous les objets géographiques avec lesquels les bâtiments pourraient être en relation. Cette méthode alternative est plus centrée sur l'opérationnel de la généralisation, on pourrait considérer qu'elle permet plutôt de détecter des 'zones rurales bâties', sous-concept de l'espace rural dans notre ontologie et plutôt espace facilitateur que clé.

Création des espaces côtiers

La côte peut être définie comme la zone en relief dominant l'estran et située au-dessus du niveau des hautes mers alors que le littoral a une définition plus difficile à traduire numériquement, car il concerne toute la région sous l'influence de la mer (Braud et al, 1997). La zone côtière qui nous intéresse est la région dans laquelle la généralisation cartographique est influencée par la mer. Elle contient clairement la côte telle que définie plus haut mais aussi une partie du littoral car un point de repère comme un phare ou un sémaphore peut être situé à plusieurs centaines de mètres de la côte. Nous proposons donc empiriquement de prendre la zone située à moins de 2 km de la côte (définition inspirée par la loi littorale).

Nous définissons la côte comme la laisse des hautes mers car nous disposons des deux laisses dans nos données tests, mais un trait de côte unique permet un fonctionnement identique de l'algorithme. La première étape est de simplifier fortement ce trait de côte pour éviter d'être parasité par ses nombreux petits détails. Pour cela, nous réalisons un filtrage des points de la ligne par l'algorithme de (Visvalingham et Wyatt, 1993) puis nous réalisons un fort lissage de la ligne (Tobler,

1966) ce qui nous donne un trait de côte très simplifié (Figure E-18). Dans un deuxième temps, nous appliquons une dilatation de 2 km à ce trait de côte simplifié et pour palier à des défauts de continuité dans le trait de côte initial (les opérateurs chargés de la saisie des données ne pouvant pas faire contenir toute la côte dans une seule ligne continue), les petites zones proches sont agrégées jusqu'à une certaine taille (pour éviter d'avoir à manipuler des espaces géographiques trop grands).

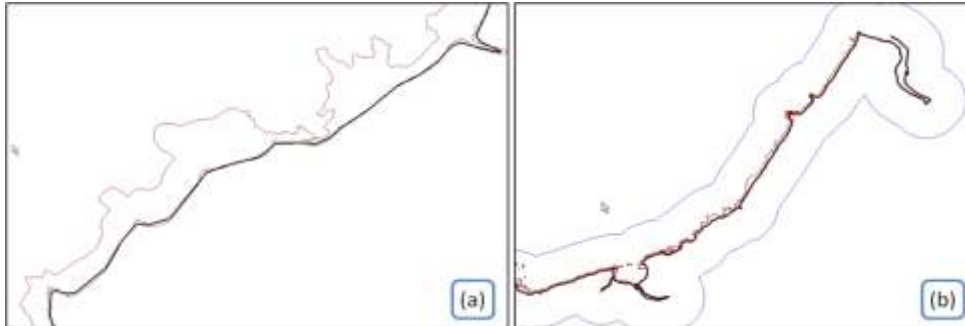


Figure E-18. Les laisses et le trait de côte simplifié (a) et l'espace côtier obtenu sur la côte basque (b).

E.1.6. Correction d'effets de bord par moindres carrés

La correction des effets de bord est la partie de l'implémentation que nous avons le moins avancée : nous n'avons implémenté qu'un processus de correction d'effets de bord, fondé sur notre modèle de généralisation par moindres carrés. Un tel processus sera choisi par le chef d'orchestre quand il voudra trouver une position moyenne entre celle avant la généralisation ayant entraîné les effets de bord et celle après. Par exemple, dans le cas d'effets de bord liés à des contraintes de proximité (la proximité entre deux bâtiments dans l'espace en a poussé un à l'extérieur de l'espace, le rapprochant trop d'un autre bâtiment non pris en compte), il peut être intéressant de chercher une position moyenne : dans le cas de la proximité, il vaut mieux deux contraintes moyennement satisfaites qu'une très satisfaite et l'autre très insatisfaite, surtout si le contexte analysé par le chef d'orchestre fait apparaître qu'on est à l'étape de généralisation graphique durant laquelle il vaut mieux ne pas éliminer ce bâtiment. Ce processus fonctionne en introduisant un nouveau type de contrainte externe (voir E.1.2) qui cherche à rétablir la position des objets, avant la généralisation ayant entraîné les effets de bord. Ces contraintes externes sont contrebalancées par des contraintes internes de maintien de la position et de la forme. Nous proposons trois paramétrages du processus : favoriser la position finale, favoriser la position initiale ou trouver une position équilibrée entre les deux. La Figure E-19 montre plusieurs exemples de correction d'effets de bord simulée entre une position avant et après généralisation par moindres carrés.

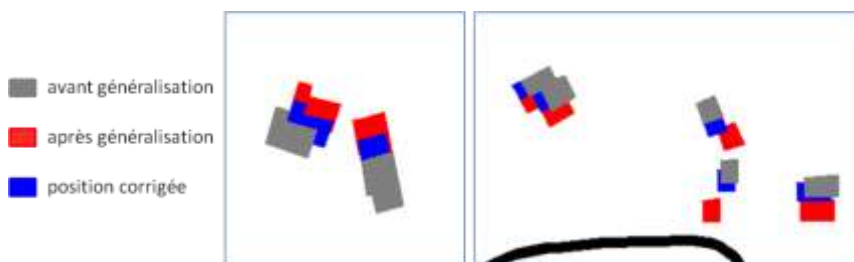


Figure E-19. Exemples de correction d'effets de bord (ici simulé) en se rapprochant de la position des objets avant généralisation.

E.2. DES OUTILS POUR FACILITER L'USAGE DE COLLAGEN

CollaGen nécessite un certain nombre de connaissances de guidage servant de ressource au système comme les contraintes formelles ou les descriptions formelles qui doivent être créées par le fournisseur de processus ou l'utilisateur. Dans le prototype de CollaGen, nous avons développé des outils interactifs pour faciliter ce travail de saisie du fournisseur et de l'utilisateur. Par ailleurs, nous avons développé des outils interactifs pour permettre la réalisation manuelle de certaines tâches du composant de traduction, notamment l'annotation sémantique des éléments du système par des concepts de l'ontologie (C.12.1).

E.2.1. Faciliter les saisie des ressources

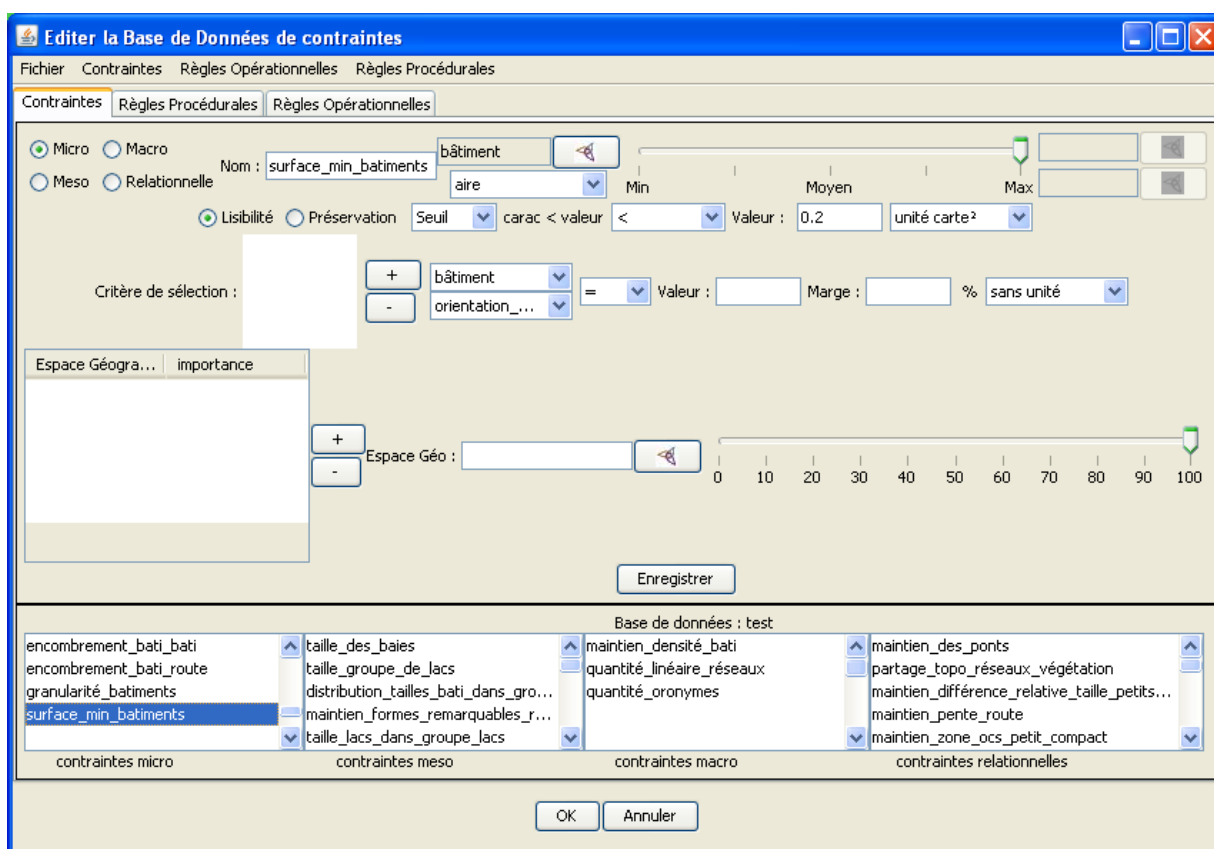


Figure E-20. Interface de saisie interactive des contraintes formelles dans CollaGen.

Les spécifications de la carte sont formalisées dans CollaGen sous la forme de contraintes et règles opérationnelles. Pour spécifier une carte complète, le nombre de ces contraintes et règles peut s'avérer très important. Par exemple, pour la spécification de zones tests dans le cadre du projet EuroSDR, les producteurs de données institutionnelles sollicités ont produit une version simplifiée des spécifications des cartes qu'ils produisent, comprenant en moyenne 75 contraintes (Stöter et al, 2010). Il paraît donc indispensable de faciliter la tâche de formalisation de ces spécifications pour qu'elle soit le moins fastidieuse possible pour l'utilisateur. Nous proposons donc sur notre plateforme de travail une fenêtre interactive de saisie des contraintes et règles formelles. Ces spécifications formelles font particulièrement appel à des concepts et propriétés de l'ontologie, source du vocabulaire partagé permettant l'interopérabilité. Cet outil interactif est donc directement

lié à l'ontologie de généralisation utilisée. La Figure E 19 montre une capture d'écran de cette fenêtre de saisie qui permet d'ajouter une nouvelle contrainte micro, macro, meso ou relationnelle en choisissant le concept et le caractère concernés parmi ceux de l'ontologie. Nous pouvons ensuite définir son expression et éventuellement un critère de sélection (seuls les objets résultats d'une requête sont concernés par la contrainte) et une restriction (la contrainte n'est définie que dans certains espaces géographiques). Dans un autre onglet de la fenêtre, les règles opérationnelles peuvent être définies de la même manière, en choisissant par exemple l'opération concernée parmi celles de la taxonomie des opérateurs intégrée à l'ontologie. Cet outil permet également de modifier ou supprimer une contrainte déjà définie. Les bases de contraintes et règles ainsi définies sont stockées dans la base de données orientée objet de notre plateforme mais peuvent également être exportées en XML pour être rechargées dans d'autres cas d'utilisation avec des données éventuellement différentes.

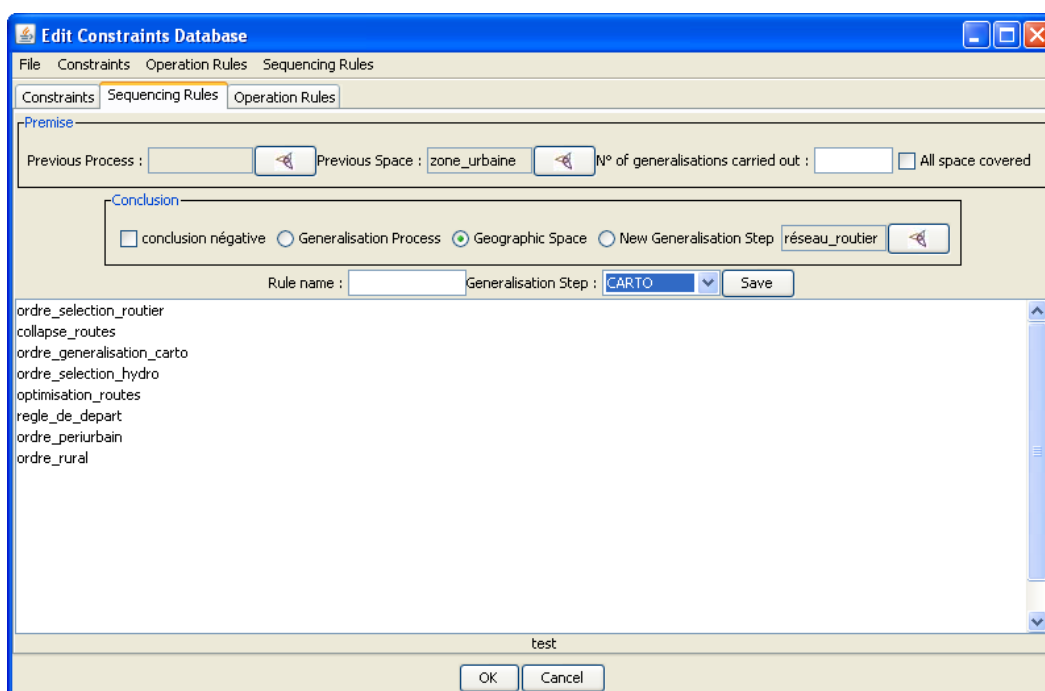


Figure E-21. Interface de saisie interactive des règles d'orchestration ou procédurales dans CollaGen. L'ordre affiché ici est uniquement celui de saisie mais pas l'ordre d'application des règles.

Le même outil permet saisir de nouvelles règles d'orchestration : l'utilisateur peut définir la prémisse sous forme d'une place dans le processus collaboratif à l'aide de concepts d'espaces géographiques ou de processus de l'ontologie ; l'utilisateur peut définir sa conclusion sous forme de concept d'espace, de processus ou sous la forme d'une fin de grande étape de généralisation (e.g. le système passe de la phase de sélection à la phase de généralisation cartographique). La Figure E-21 montre une capture d'écran de cet outil de saisie interactive.

Si elles sont moins nombreuses que les contraintes, les descriptions formelles de capacités des processus sont assez volumineuses et complexes (voir Figure C-34) et nous avons également développé un outil interactif pour faciliter leur saisie par le fournisseur de processus. Comme pour les outils précédents, le choix des pré-conditions ou des enrichissements nécessaires est fait en piochant dans les concepts de l'ontologie (le fournisseur indique quels concepts doivent être instanciés en les sélectionnant dans l'ontologie). Le choix des post-conditions est lui effectué en

piochant dans les contraintes et règles définies à partir de l'outil précédent. La Figure E-22 montre plusieurs des onglets permettant la saisie complète de la description formelle d'un processus de généralisation. L'onglet en Figure E-22a permet de saisir les pré-conditions de la description en choisissant un concept d'espace géographique dans l'ontologie, l'indice de confiance dans la pré-condition, la gamme d'échelle pour laquelle cette pré-condition est valable ou encore les caractéristiques plus précises de l'espace (thème dominant, structure hiérarchique, etc.). L'onglet en Figure E-22b permet de choisir les concepts avec lesquels les données doivent être enrichies tout en précisant si cet enrichissement peut-être local (réalisé uniquement à l'intérieur des espaces pour lesquels le processus est choisi) ou s'il est global (il n'a de sens que s'il est réalisé sur l'ensemble des données).

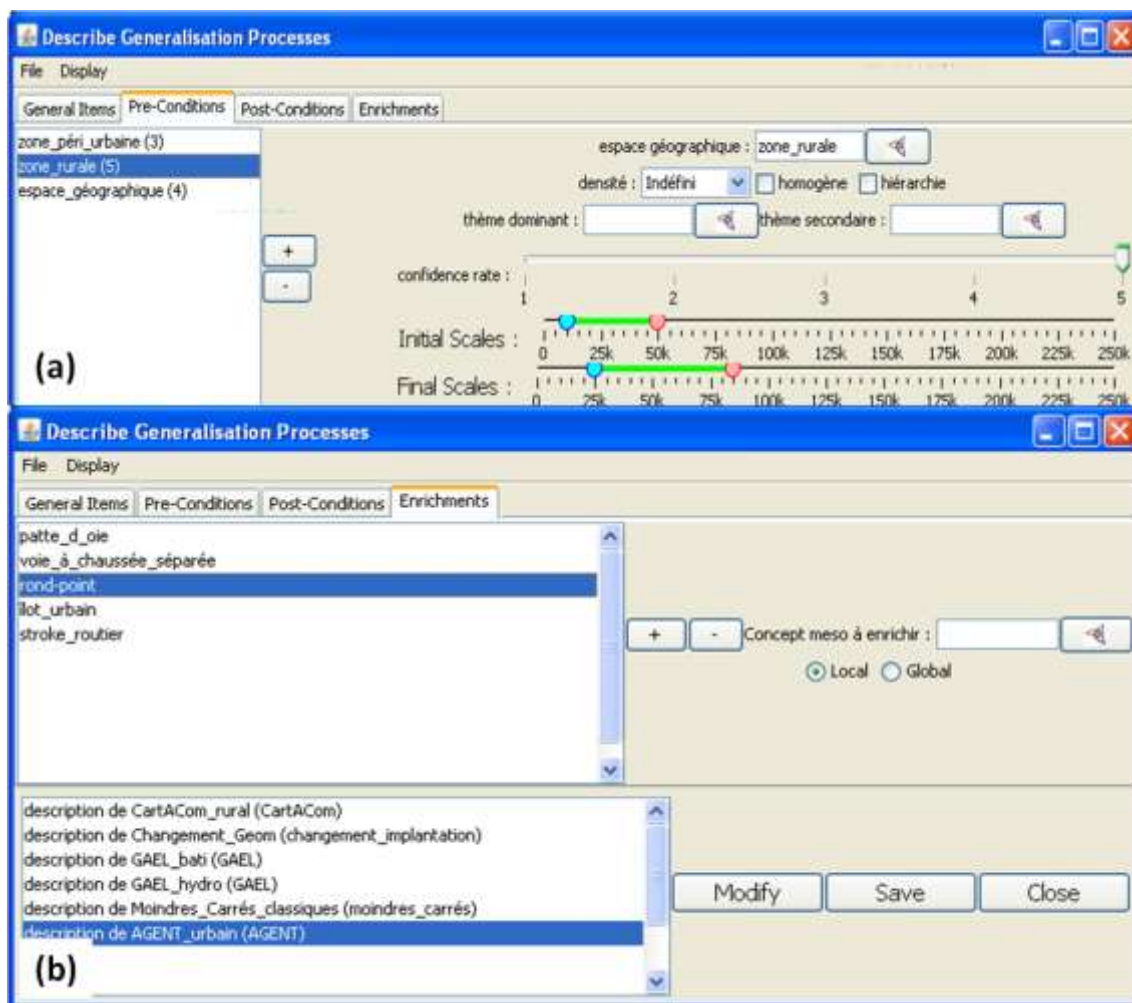


Figure E-22. Plusieurs onglets de l'interface de saisie interactive des descriptions formelles de processus.

E.2.2. Outils de traduction interactive

Comme mentionné dans la description du composant de traduction de CollaGen, nous avons opté pour des solutions interactives plutôt qu'automatiques pour une partie des fonctionnalités du composant, dont l'annotation sémantique du schéma de données et l'instanciation des contraintes formelles par des moniteurs. Dans les deux cas, nous avons donc développé des outils pour saisir ces informations de traduction et les stocker de manière formelle (pour les utiliser automatiquement

dans CollaGen). Tout d'abord, l'outil interactif d'annotation sémantique permet d'associer des éléments de schéma de données (classe d'objet, attribut, lien entre classes) avec des concepts et propriétés de l'ontologie de généralisation (Figure E-23). Cet outil récupère les concepts correspondant à des entités géographiques et permet pour chacun de lui associer une classe d'objets de la base de données. Nous avons fait ce choix car notre plateforme dispose du SGBD orienté-objet de Clarity ce qui permet de gérer par une superclasse les cas où plusieurs classes correspondent à un unique concept (plusieurs classes de routes par exemple). Sur une autre plateforme, il serait plus judicieux de réaliser l'appariement dans l'autre sens (chaque classe est annotée par un concept de l'ontologie). Nous ne gérons pas ici le cas complexe d'appariements n-m entre classes et concepts, que nous n'avons pas rencontré dans nos données mais qui peut tout à fait exister suivant l'ontologie et le schéma de données (cependant, plus l'ontologie est détaillée, plus on évite ce type de situations). De la même manière, les propriétés de l'ontologie peuvent être associées à des attributs du schéma de données et les relations entre concepts à des liens entre classes. De plus, pour les relations géographiques (concepts particuliers de l'ontologie), elles peuvent être associées à un nom de classe ainsi qu'à un nom de classe Java connue de la plateforme qui permet de calculer la satisfaction et la valeur courante de la relation. Cela permet de réaliser sur le même outil l'annotation sémantique et l'appariement entre concept ontologique et algorithme de la boîte à outils capable d'instancier ce concept. L'ensemble des informations de l'annotation sémantique est finalement représenté dans un objet *annotation sémantique* manipulable par nos agents logiciels et coordinateurs, mais aussi stocké dans un fichier XML pour être réutilisé dans d'autres cas d'application.

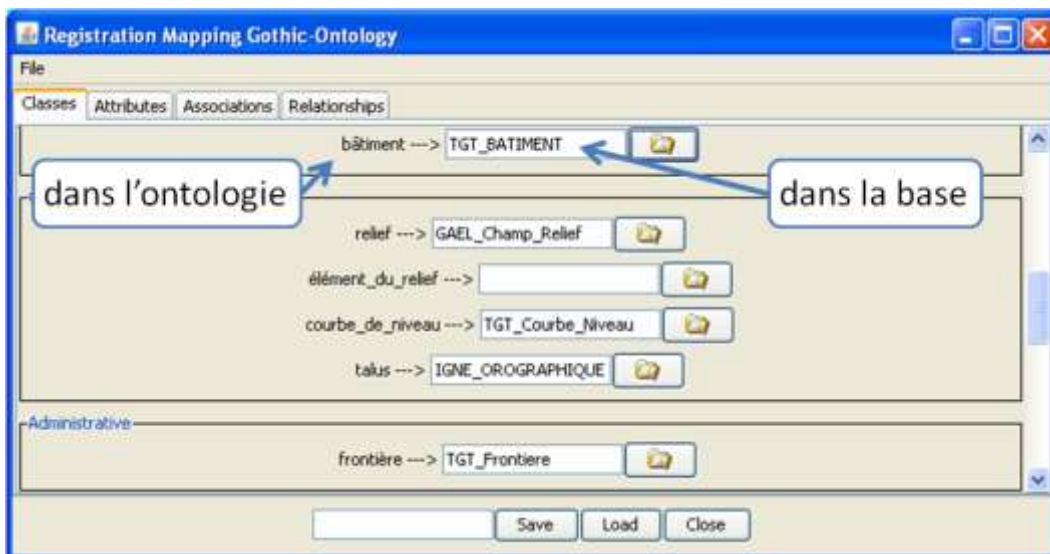


Figure E-23. Outil interactif d'annotation sémantique du schéma de données à généraliser.

Le deuxième outil interactif assurant une des fonctionnalités du composant de traduction permet d'associer les contraintes formelles (et les règles opérationnelles) avec les moniteurs qui vont s'assurer, dans les données, du respect de ces contraintes. Ainsi, il est possible d'associer interactivement une contrainte formelle avec une classe du schéma de données (dans laquelle seront créées les instances de moniteurs) et une classe Java (qui contient les méthodes pour évaluer à tout moment la satisfaction de la contrainte et son évolution). En effet, notre plateforme manipule les objets géographiques, agents et moniteurs en Java mais les stocke dans le SGBD de Clarity, non directement lié à Java, ce qui explique la nécessité de préciser à la fois la classe Java et la classe du

SGBD pour instancier nos contraintes formelles en moniteurs. Encore une fois, cette information est stockée formellement dans un fichier XML utilisable par l'agent du système en charge de la création des moniteurs, soit le chef d'orchestre.

E.3. DESCRIPTION DU PROTOCOLE DE TEST

Afin de tester le prototype de CollaGen implémenté et ses différents composants sur notre plateforme de travail sur un cas réel de généralisation cartographique, nous avons mis en place un protocole de test avec différents jeux de données et un jeu de ressource identique pour lequel nous avons joué le rôle de fournisseur et d'utilisateur. Ce protocole de test se concentre sur la généralisation de cartes au 1 : 50000 de type IGN à partir de la base de données BD TOPO de l'IGN dont l'échelle de référence est le 1 : 15000. Dans cette partie, nous commençons par présenter les trois jeux de données utilisées et les raisons de leur choix. Puis, nous décrivons l'ensemble des ressources utilisées dans ce test afin de permettre la compréhension et l'interprétation des résultats que nous présentons dans la partie E.4.

E.3.1. Description des données utilisées

Nous avons choisi trois zones pour réaliser nos tests du prototype de CollaGen, la première entre Hendaye et Saint-Jean-de-Luz au Pays Basque, la deuxième autour d'Orthez dans les Pyrénées Atlantiques (Figure E-24) et la troisième autour de l'Alpe d'Huez. Les données pour ces trois zones ont été extraites de la base BD TOPO® de l'IGN de résolution métrique avec une échelle de référence de 1 : 15000. La première, qui mesure 42 km sur 25 km, a été choisie pour la très forte diversité de paysages contenue sur une faible étendue. Il y a en effet dans cette zone des parties montagneuses, côtières, urbaines, périurbaines et rurales. La deuxième zone autour d'Orthez mesure 40 km sur 20 km et présente des paysages essentiellement ruraux avec deux petites villes. Cette zone a été choisie car elle a été utilisée pour mettre en place la chaîne de production de cartes au 1 : 25000 à partir de la BD TOPO® en utilisant des outils de généralisation automatique (essentiellement issus d'AGENT et les Beams). Elle constitue donc un bon support de comparaison pour l'évaluation de CollaGen. La troisième zone autour de l'Alpe d'Huez mesure 11 km sur 11 km et correspond à l'une des quatre zones utilisées lors du projet EuroSDR d'état de l'art des logiciels du commerce en généralisation cartographique (Stöter et al, 2010). Cette zone offre donc, elle aussi, des possibilités intéressantes de comparaisons avec les meilleurs résultats obtenus lors de ce projet. Par ailleurs, le schéma de données ayant été modifié et traduit en anglais pour les besoins du projet, cette zone permet de tester les capacités d'interopérabilité de notre prototype face à l'interopérabilité des données.



Figure E-24. Deux de nos zones tests délimitées sur la carte régionale de l'IGN (source : Géoportail®).

E.3.2. Description des ressources formelles utilisées

Nous commençons donc notre présentation du protocole de test utilisé par la description des ressources de CollaGen utilisées. Nous décrivons ainsi les processus et espaces géographiques utilisés, notre ontologie de généralisation, les spécifications formalisées sous forme de contraintes et règles, les descriptions formelles de processus et enfin, les règles d'orchestration utilisées.

Espaces et processus utilisés

Dans ce test, nous avons utilisé les espaces géographiques pour lesquels nous disposions d'algorithmes de notre boîte à outils, c'est-à-dire les espaces urbains, périurbains, ruraux, de montagne et le réseau routier comme espaces clés ; les routes de montagne, les graphes de flexibilité (Lemarié et Lecordix, 2003) et la végétation comme espaces facilitateurs et les espaces côtiers comme espaces spécifiés (construits si des contraintes existent avec une restriction sur cet espace).

Les processus de généralisation disponibles sont issus des modèles implémentés sur notre plateforme de travail et ils sont au nombre de neuf dans ces tests :

- Deux processus sont fondés sur le modèle AGENT : « AGENT urbain » dédié à la généralisation des zones plutôt denses en bâtiments et « AGENT routier » dédié à la généralisation des routes de montagne.
- Un processus est fondé sur CartACom : il correspond à celui présenté en application de (Duchêne, 2004).

- Trois processus sont fondés sur les moindres carrés présentés précédemment dans ce chapitre.
- Un processus dédié à la simplification des carrefours.
- Un processus dédié à la généralisation de la végétation.
- Un processus fondé sur les Beams.

Les descriptions formelles de tous ces processus sont décrites plus loin dans cette partie pour connaître le champ d'action de ces processus et comprendre la manière dont ils sont utilisés par CollaGen.

Construction manuelle de l'ontologie de généralisation

Plusieurs méthodes sont présentées dans la littérature pour la construction d'ontologies géographiques. La plus simple, et celle que nous avons choisie, est de la constituer à la main à l'aide de textes, de dictionnaires ou glossaires du domaine. Trois approches principales existent pour construire manuellement une ontologie :

- l'approche bottom-up (Van Der Vet et Mars, 1998) dans laquelle on détermine les concepts spécifiques puis on les généralise.
- l'approche top-down (Sowa, 1995) dans laquelle on détermine d'abord les concepts génériques puis on les spécialise jusqu'au niveau souhaité.
- l'approche middle-out (Ushold et Grüninger, 1996) dans laquelle on détermine les concepts les plus importants puis on les généralise et les spécialise.

Les travaux en intégration de données géographiques du COGIT ont conduit à la conception d'une taxonomie de concepts topographiques exprimée dans un langage d'ontologie, OWL2 (Abadie et Mustière, 2010). Nous avons donc choisi d'utiliser une approche middle-out en prenant les concepts de la taxonomie topographique comme concepts de base. Puis, nous avons généralisé et spécialisé ces concepts tout en en ajoutant d'autres, plus spécifiques à la généralisation collaborative (voir C.4). Puis, nous avons intégré manuellement à l'ontologie l'ensemble des propriétés que nous connaissions comme susceptibles d'être soumises à une contrainte de généralisation. Nous avons également intégré les relations et les axiomes sur ces relations nécessaires au fonctionnement de nos tests : par exemple, la relation de composition des objets meso a été ajoutée et restreinte pour les groupes de bâtiments qui contiennent au moins un bâtiment et éventuellement des routes (Figure E-25). Notre ontologie n'est pas exhaustive et nécessite encore un enrichissement qui est fait petit à petit. Tout ce travail de construction de l'ontologie est effectué à l'aide du logiciel Protégé qui permet de créer et gérer des ontologies en OWL2.



Figure E-25. Extrait de notre ontologie, issu du logiciel Protégé, centré sur la relation 'est_composé_de' entre le concept entité_meso et le concept général 'entité_géographique' pour le meso 'groupe_de_bâtiments': les deux axiomes précisent qu'un groupe_de_bâtiments est composé d'au moins un bâtiment et éventuellement de routes.

Contraintes et règles opérationnelles formelles

Pour construire le jeu de contraintes formelles et règles opérationnelles qui représentent les spécifications de notre protocole de test, nous avons utilisé les outils interactifs présentés en E.2.1. Afin de réaliser des tests réalistes et d'éprouver la généricité de notre modèle formel, nous avons synthétisé les spécifications classiques et représentatives d'une carte topographique généralisée en 70 contraintes et 8 règles opérationnelles. Les contraintes se répartissent de la manière suivante : 30 contraintes micro, 20 contraintes relationnelles, 15 contraintes meso et 5 contraintes macro. Ces contraintes comprennent notamment les contraintes suivantes (correspondant à celles de la p. 39):

- Taille minimum sur les bâtiments (>0.2 mm² carte), les lacs, les forêts.
- Maintien de la position initiale des bâtiments, routes, cours d'eau, chemins.
- Taille minimum des trous dans les bâtiments, l'eau surfacique (cours d'eau et lacs), les forêts.
- Granularité des bâtiments, forêts, lacs, routes, cours d'eau, voies ferrées, chemins.
- Eviter l'empâtement des symboles de routes, cours d'eau, chemins.
- Largeur locale minimum des bâtiments et des forêts.
- Forme générale des bâtiments, forêts, routes, cours d'eau.
- Maintien des alignements de bâtiments et de leur orientation.
- Harmonisation des alignements de bâtiments.
- La densité, la distribution des tailles et la répartition spatiale dans les groupes de bâtiments.
- Maintien des formes remarquables du relief, des grands ronds-points et pattes d'oies.
- Proximité entre éléments de réseaux, entre bâtiments, entre bâtiment et élément de réseau et entre élément de réseau et talus.
- Maintien de l'orientation relative entre bâtiment et route.
- Harmonisation des orientations relatives proches entre bâtiment et route.
- Maintien de l'écoulement d'une rivière dans son talweg.
- Maintien de la différence relative de taille entre deux bâtiments.
- Maintien de l'altitude d'un bâtiment.
- Maintien de la position relative entre un bâtiment et une impasse menant au bâtiment.
- Maintien des différences de densité des réseaux et bâtiments.
- Réduction de la quantité de linéaires de réseau.

Les huit règles formalisées sont les suivantes : la squelettisation des petits ronds-points, la squelettisation des petites pattes d'oies, l'interdiction de l'agrégation des bâtiments en ville, l'interdiction de l'élimination des bâtiments importants, le grisage des îlots en centre ville denses, l'élimination des petits strokes hydrographiques, l'élimination des tous petits bâtiments (aire < 0.012 mm² carte) et la squelettisation des parties fines des cours d'eau surfaciques.

Bien que ce jeu ne soit pas complètement exhaustif, il nous paraît suffisant pour guider et évaluer correctement les processus de généralisation disponibles sur notre plateforme de test : il nous a paru inutile de saisir trop de contraintes dont nous savons pertinemment qu'aucun des processus à notre disposition ne serait capable de les gérer.

Nous nous sommes d'autant plus restreint que nous n'avons pas eu le temps d'implémenter un moniteur de contrainte (ou de règle) pour chacune de ces spécifications formelles. En effet, l'implémentation d'un moniteur nécessite la conception de plusieurs algorithmes d'analyse spatiale

pour mesurer la valeur courante du caractère contraint et la satisfaction de la contrainte à un moment donné. Seule une vingtaine de ces contraintes (et règles) a donc été instanciée en moniteur pour pouvoir être évaluée. Les autres ne sont utiles dans nos tests qu'à l'auto-paramétrage des processus par leur fonction de traduction.

Descriptions formelles des processus

Pour chacun des neuf processus disponibles sur la plateforme, nous avons rempli la description formelle (voir C.7.3) en utilisant l'outil interactif de saisie développé. La description des pré-conditions (les espaces sur lesquels le processus peut s'appliquer) est faite en utilisant les concepts d'espace géographique issus de l'ontologie et les post-conditions (les types de spécifications gérées *a priori* par le processus) sont exprimées à partir des 78 éléments de spécification formelle que nous avons saisis pour ces tests. Les indices de confiance (valeur de 1 à 5) associés à chaque pré ou post-condition sont marqués entre parenthèse dans le tableau récapitulatif suivant :

| Nom du processus | Echelle optimale | Enrichissements | Pré-conditions | Post-conditions |
|------------------|------------------|---|---|--|
| AGENT urbain | 15k-50k | 'îlot_urbain', 'strokes_routier' 'rond-point', 'patte_d_oie' | Espace urbain (5) Espace périurbain (3) Espace rural (2) | Maintien densité îlots (5) Bâti (taille, granularité...) (5) Répartition spatiale dans îlots (4) Interdiction agrégation (5) Proximité entre bâtiments (4) Proximité bâti/route (4) Grisage des îlots denses (5) |
| AGENT routier | 15k-50k | - | Routes de montagne (5) Réseau routier (4) | Empâtement des routes (5) Maintien forme des routes (5) Maintien position ini routes (5) |
| CartACom | 15k-25k | 'Petits compacts' | Espace rural (5) Espace périurbain (3) Espace peu dense de bâtiments (3) | Bâti (taille, granularité...) (5) Maintien orientation relative (5) Posit. relatif bâti/impasse (5) Proximité entre bâtiments (5) Proximité bâti/route (5) Caricature des orient. relatives proches bâti/route (5) |
| MC urbain | 25k-50k | 'îlot_urbain' | Espace urbain (3) | Proximité entre bâtiments (5) Proximité bâti/route (5) Forme bâtiments (3) Position initiale bâti (3) Position initiale routes (5) |
| MC rural | 25k-50k | - | Espace rural (4) Espace périurbain (4) | Proximité entre bâtiments (5) Proximité bâti/route (5) Position initiale bâti (3) Position initiale routes (3) Maintien forme bâti (3) Maintien forme des routes (3) |
| MC réseaux | 25k-50k | - | Remontées méca. (5) Graphes de flexibilité (4) | Proximité réseau/réseau (5) Maintien forme des réseaux (3) |
| Beams | 10k-25k | - | Graphes de flexibilité (5) | Proximité réseau/réseau (5) Maintien forme des réseaux (4) Empâtement des routes (5) |
| Carrefours | 15k-50k | 'rond-point', 'patte_d_oie' | Réseau routier (5) | Squelettisation rond-point (5) Squelettisation patte d'oie (5) |

| | | | | |
|------------|---------|---|--------------------------|---|
| Végétation | 10k-50k | - | Couche de végétation (5) | Granularité végétation (4) Taille min végétation (5) Taille trous végétation (5) Maintien forme végétation (4) |
|------------|---------|---|--------------------------|---|

Nous avons inséré ci-dessous un extrait de la description formelle du processus issu de CartACom dans son format de stockage XML.

```
- <description-processus>
  - <ressources-utilisées>
    <ontologie>file:///E:/users/gtouya/Documents/These/ontologie/OntoGene
ralisation.owl</ontologie>
    <base-de-contraintes>test</base-de-contraintes>
    <base-de-règles>test</base-de-règles>
  </ressources-utilisées>
  <nom>CartACom_rural</nom>
  <processus>CartACom</processus>
  <classejava>fr.ign.gothic.cogit.guillaume.orchestration.collagenv1.pr
ocessus.ProcessusCartAComRural</classe-java>
  <nom-traducteur>traduireToCartAComRural</nom-traducteur>
  - <pre-condition>
    <espace-geographique>zone_rurale</espace-geographique>
    - <caractéristiques>
      <densité>0</densité>
      <homogénéité>>false</homogénéité>
      <hiérarchie>>false</hiérarchie>
    </caractéristiques>
    - <gamme-échelles>
      <échelle-ini-basse>10000</échelle-ini-basse>
      <échelle-ini-haute>50000</échelle-ini-haute>
      <échelle-gen-basse>25000</échelle-gen-basse>
      <échelle-gen-haute>85000</échelle-gen-haute>
    </gamme-échelles>
    <indice-confiance>5</indice-confiance>
  </pre-condition>
  ...
  - <post-condition>
    <élément-spécification>
      <type>contrainte</type>
      <nom>orientation_relative_proche_bati_route</nom>
    </élément-spécification>
    <indice-confiance>5</indice-confiance>
  </post-condition>
  - <post-condition>
    <élément-spécification>
      <type>contrainte</type>
      <nom>maintien_orient_relative_bati_route</nom>
    </élément-spécification>
    <indice-confiance>5</indice-confiance>
  </post-condition>
  <enrichissements />
</description-processus>
```

Règles d'orchestration utilisées

Nous avons enfin saisi des règles d'orchestration pour guider le choix de l'espace à généraliser par le chef d'orchestre. Comme précédemment, ces règles ont été saisies grâce à l'outil interactif que nous avons développé. Nos premières expérimentations nous ont poussé à saisir des règles pour obtenir la séquence suivante dans l'étape dite de généralisation cartographique (qui succède aux étapes de généralisation de modèle, de changement de géométrie et de sélection) :

1. Généralisation de modèle
2. Changement de géométrie
 - a. Réseau routier
3. Sélection
 - a. Réseau hydrographique
 - b. Réseau routier
4. Généralisation cartographique
 - a. espace urbain
 - b. graphes de flexibilité
 - c. espace périurbain
 - d. espace rural
 - e. espace de montagne
 - f. végétation
5. Généralisation graphique
 - a. Réseau routier
 - b. espace urbain
 - c. espace périurbain

Toutefois, nous avons réalisé des tests en faisant varier cet ordre pour en évaluer la pertinence.

E.4. RESULTATS DE TESTS DU PROTOTYPE

Maintenant que nous avons décrit notre implémentation du prototype de CollaGen et les conditions dans lesquelles il est testé, nous présentons dans cette partie des résultats obtenus à l'aide de ce prototype sur ces données et avec ces spécifications. Nous proposons, dans un premier temps, des résultats du fonctionnement des différents composants pris individuellement, puis, nous présentons des résultats de généralisations globales sur différentes zones tirées des données test. Nous présentons, dans un dernier temps, des expérimentations sur l'ordre d'orchestration des espaces à généraliser.

E.4.1. Résultats individuels des composants de CollaGen

Dans cette partie, nous présentons individuellement les résultats obtenus lors de l'utilisation automatique de certains des composants de CollaGen dans le cadre du protocole de test. Nous commençons par des résultats sur le fonctionnement du registre, puis, des résultats d'observation en ligne et d'évaluation globale d'une généralisation et enfin, des résultats de délibérations du chef d'orchestre pour choisir l'espace à généraliser à un moment donné.

Le Registre

L'agent registre que nous avons implémenté reprend exactement les principes présentés en C.13 pour répondre à une requête de généralisation : il choisit les processus compatibles en fonction des pré-conditions et les ordonne en fonction de l'adéquation entre les post-conditions et les moniteurs de contrainte effectivement instanciés à l'intérieur de l'espace. Comme nous n'avons pas assez de processus disponibles dans nos tests pour qu'il y ait réellement une concurrence sur la plupart des espaces, nous avons concentré nos tests sur les espaces périurbains pour lesquels le choix du processus optimal dépend de la configuration de l'espace et des moniteurs qui découlent de cette configuration. La Figure E-26 montre les réponses du registre à deux espaces. Le premier, peu dense (la grande majorité des moniteurs correspond à des contraintes micro et relationnelles), donne un très net avantage à CartACom sur AGENT et les moindres carrés ce qui correspond bien au choix qu'un spécialiste de la généralisation aurait effectué. Le deuxième est plus dense et même si CartACom est encore choisi, l'écart est plus resserré car AGENT peut gérer les quelques moniteurs meso au contraire de CartACom.

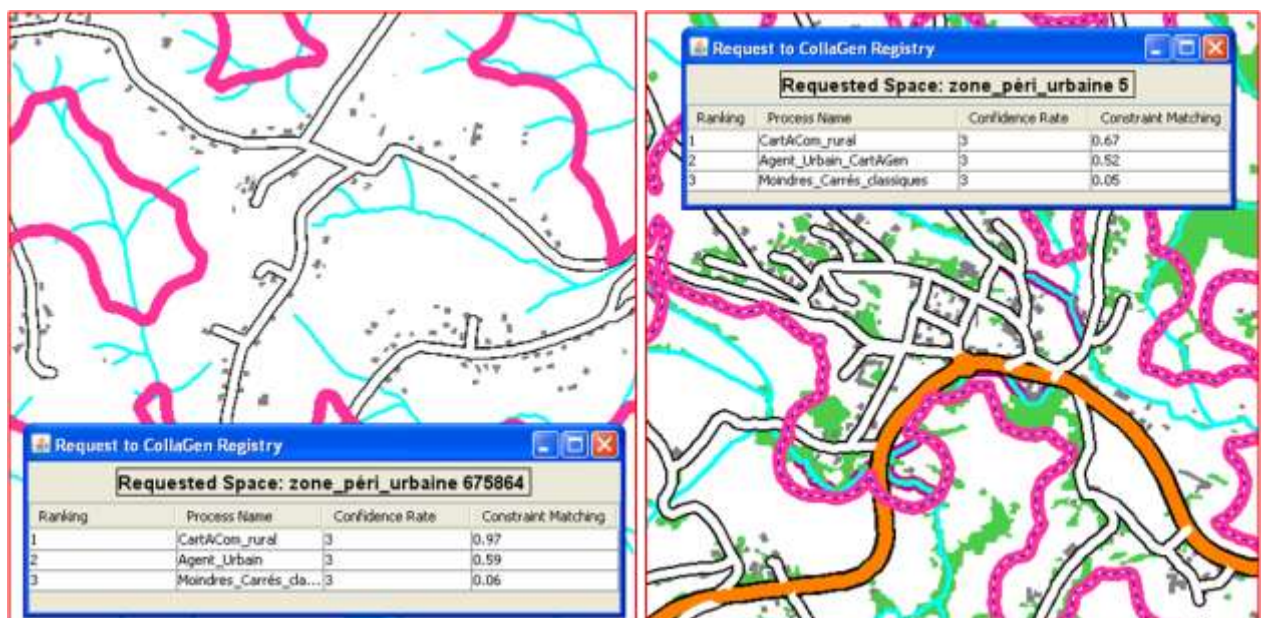


Figure E-26. Réponses du registre à la requête de deux espaces périurbains. Le deuxième étant plus dense, il y a dans l'espace des moniteurs de maintien de densité et l'écart de pertinence entre CartACom et AGENT est plus réduit même si CartACom reste le premier choix. Le 'confidence rate' est l'indice de confiance dans la pré-condition et l'indice de 'Constraint matching' correspond à l'adéquation entre post-conditions et moniteurs dans l'espace.

Nous avons testé si la réponse du registre était très dépendante de la méthode de calcul de l'adéquation des post-conditions aux moniteurs en comparant pour un même espace périurbain une réponse mettant tous les moniteurs à égalité (ils sont pondérés uniquement par leur importance) et une réponse utilisant la fonction d'adéquation présentée en C.13 (p. 139). Dans les deux cas, l'ordre des processus conseillés est le même (Figure E-27) mais l'écart est différent : en effet, cet espace dispose à peu près de 55 % de moniteurs micro, 40 % de moniteurs relationnels et 5 % de moniteurs meso, donc, notre fonction d'adéquation favorise plus nettement CartACom qui sait gérer les contraintes relationnelles correspondant à ces moniteurs (les moniteurs meso ne sont pas assez nombreux pour compenser et faire pencher la balance côté AGENT).

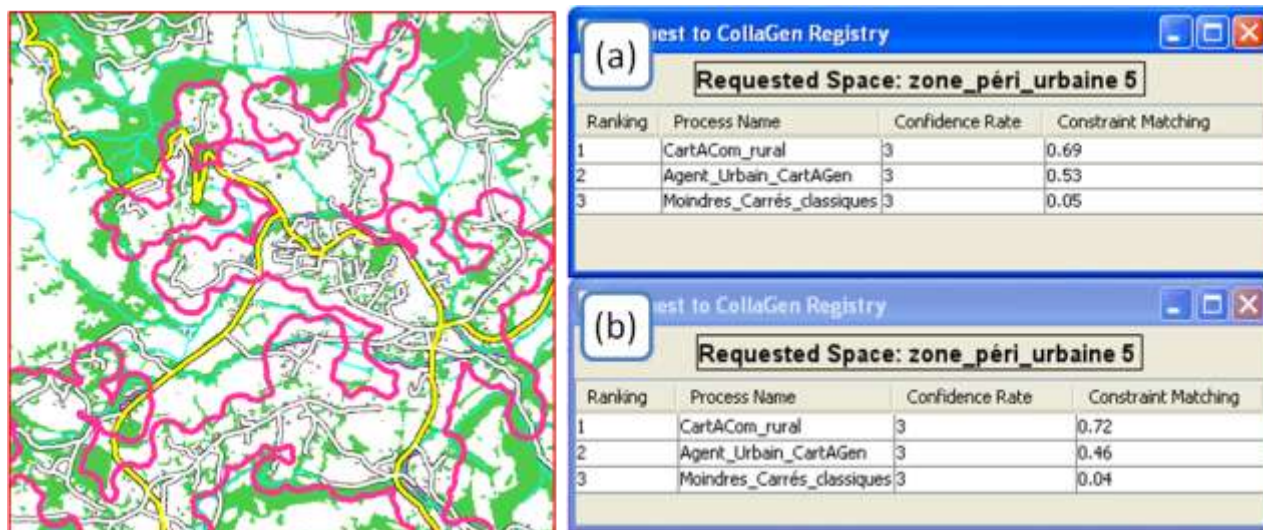


Figure E-27. Variation des réponses du registre lorsque l'on fait varier l'importance relative des contraintes entre elles. (a) toutes les contraintes se valent et sont pondérées seulement par leur importance. (b) les contraintes sont pondérées par le nombre d'objets sur lesquels porte le moniteur (un moniteur de contrainte relationnelle a deux fois le poids d'un moniteur de contrainte micro).

Nous avons également fait des tests en faisant varier le nombre de contraintes prises en compte par le registre et, sans surprise, cela peut faire varier fortement la réponse du registre : le registre ne choisit donc pas le processus le plus adapté à un espace dans l'absolu mais le processus le plus adapté à un espace compte tenu des spécifications. Par exemple, sur le même espace que la Figure E-27, si les moniteurs de contraintes relationnelles ne sont plus considérés, AGENT urbain devient le processus le plus adapté selon le registre.

L'Agent Observateur

Nous avons implémenté un agent observateur (C.14.3), qui peut réaliser des observations en parallèle des processus interrompables ou à la fin des processus non interrompables pour détecter des espaces émergents composés de moniteurs en conflits persistants. Afin d'évaluer les capacités d'observation, nous avons réalisé des tests en observant des généralisations par CartACom d'espaces périurbains assez denses (espaces pour lesquels CartACom atteint ses limites et pour lesquels le registre conseille plutôt AGENT mais nous forçons comme cela des situations d'émergence). Il existe des endroits, dans ces espaces, que le processus n'arrive pas à généraliser. La Figure E-28b montre une zone dans laquelle l'observateur a identifié des groupes de conflits persistants avec les paramètres d'observation déterminés empiriquement comme les plus pertinents. Ces conflits

persistants sont principalement des conflits de proximité entre un bâtiment et une route. La Figure E-28c montre la même zone mais avec une observation paramétrée de manière moins pertinente : tous les groupes de conflits ne sont pas identifiés et le groupe le plus grand est ici scindé en trois. Un bon choix des paramètres est donc déterminant pour observer tous les groupes de conflits et les regrouper au mieux. La Figure E-28d montre la même zone, mais avec une observation pour laquelle la méthode de détermination d'un conflit persistant à partir d'un moniteur a été changée : nous ne considérons, comme en conflit, que les moniteurs pour lesquels l'insatisfaction est importante (très insatisfait ou forte baisse de la satisfaction). Les résultats sont comparables à ceux obtenus en considérant plus de conflits, donc, il nous paraît plus pertinent d'utiliser cette dernière méthode pour gagner un peu de temps de calcul.

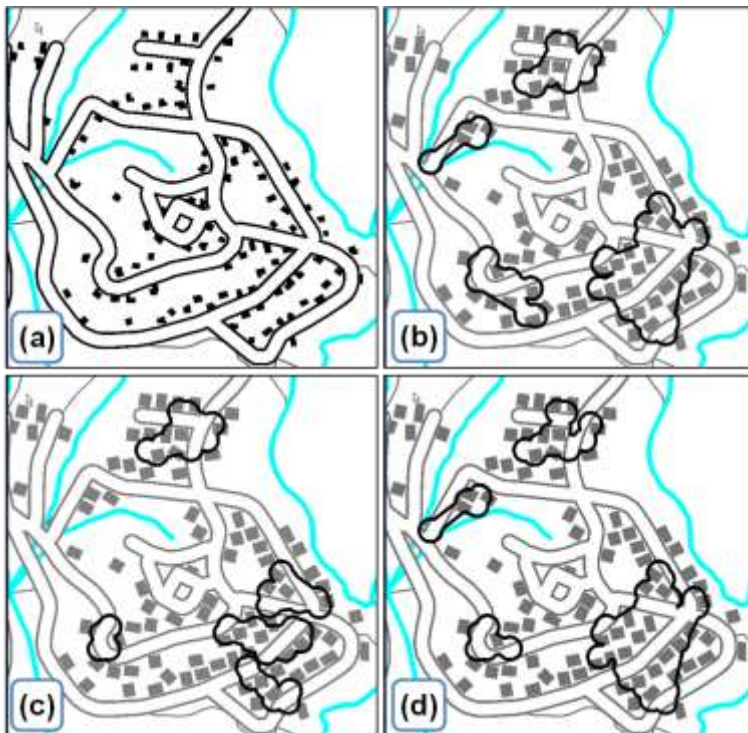


Figure E-28. Groupes de conflits résiduels issus après une généralisation par CartACom : (a) données initiales (b) groupes identifiés après généralisation concernant des conflits de proximité entre bâtiments et routes (c) groupes avec des paramètres d'émergence moins pertinents (d) groupes avec des alertes seulement pour les conflits importants.

Dans la Figure E-29a, nous travaillons sur la même zone et construisons cette fois les espaces émergents de manière complète (on étend spatialement les groupes de conflits résiduels aux objets liés aux moniteurs, puis, aux objets voisins comme décrit en C.14.4 p. 153). En Figure E-29b et c, nous zoomons sur un des espaces émergents qui est généralisé par le processus de moindres carrés ruraux choisi ici manuellement (la requête par un espace émergent n'a pas été implémentée sur l'agent registre).

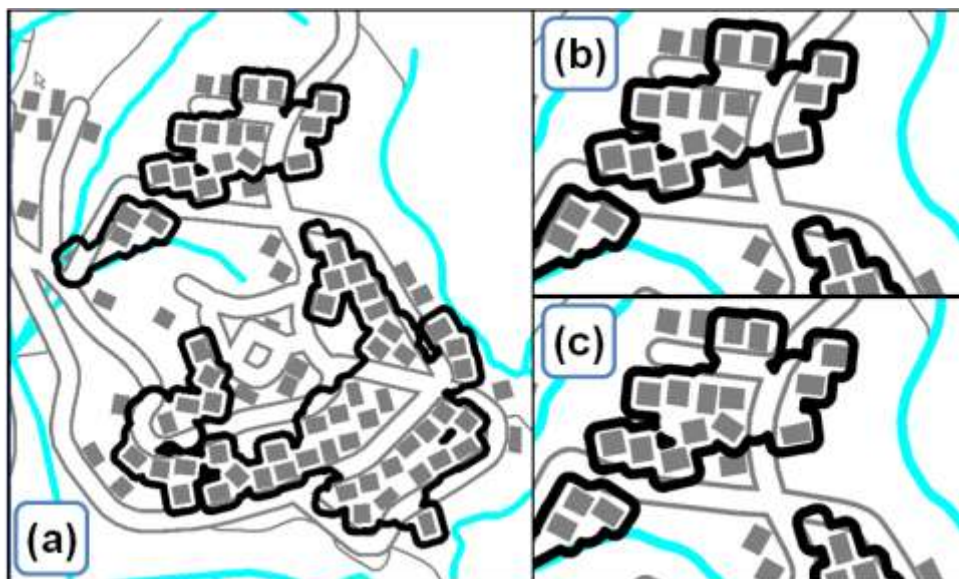


Figure E-29. (a) Espaces émergents construits pendant une généralisation par CartACom. (b) Zoom sur deux des groupes émergents. (c) Ces groupes corrigés par un autre processus (moindres carrés ruraux).

Nous avons également réalisé des tests permettant d'évaluer la méthode d'échantillonnage des contraintes à observer. En moyenne, sur la généralisation de plusieurs villes de tailles différentes par AGENT, le temps d'observation est 4 fois plus court que lorsque l'observation est faite sur l'ensemble des contraintes, pour une quantité de groupes créés équivalente et des formes similaires. Toutefois, la Figure E-30 montre que l'échantillonnage peut entraîner l'absence d'observation de petits groupes de conflits émergents. En effet, l'image (a) montre une construction durant l'application de CartACom d'un espace émergent à partir de deux moniteurs de proximité entre bâtiment et route non satisfaits. Cette observation est réalisée en utilisant un échantillon aléatoire de deux tiers des contraintes. L'image (b) est obtenue avec le même protocole mais pour un échantillon aléatoire de la moitié des contraintes : une seule des deux contraintes insatisfaites est dans l'échantillon, ce qui ne permet pas d'observer le groupe émergent. Ce problème est cependant réglé par mécanisme de zoom d'observation qui permet d'observer l'ensemble des moniteurs autour d'un moniteur insatisfait à l'observation suivante.



Figure E-30. Influence de l'échantillonnage dans l'observation des espaces émergents : (a) les deux conflits résiduels sont dans l'échantillon et un espace émergent est observé. (b) un seul conflit est dans l'échantillon (celui marqué par une croix entourée) et aucun espace émergent n'est observé.

En outre, notre méthode d'observation des groupes de conflits persistants ne repose pas sur une qualification des conflits comme dans CartACom: toutes sortes de contraintes peuvent être présentes dans l'échantillon observé et tous les types de conflits sont observés de la même manière, alors que dans CartACom, l'observation est ciblée sur un type de conflit particulier à résoudre par un traitement de groupe particulier (Duchêne et Touya, 2010). Nous estimons toutefois nécessaire de faire une classification des zones émergentes sur les types de moniteurs, leur nombre et leur disposition, pour éviter de les traiter de manière uniforme : parfois, il ne faut pas s'arrêter mais recommencer avec d'autres paramètres (voir mécanisme d'arrêt en p. 150) ; d'autres fois il faut que le groupe soit très gros avant d'intervenir etc. Une telle différenciation permettrait également de choisir la correction adaptée (que nous ne faisons pas de manière automatique pour l'instant). La Figure E-31b illustre bien ce propos en montrant comment des groupes de conflits de natures différentes sont traités actuellement de manière uniforme : les conflits persistants sur la végétation font émerger un groupe, ce qui est peu pertinent car la version de CartACom utilisée ne traite pas ces problèmes (il faudrait plutôt les laisser de côté pour appliquer en suivant un processus dédié à ces problèmes) ; sur la gauche, des conflits de proximité entre bâtiments sont regroupés avec des conflits dus à une trop grande densité en routes (or ces conflits mériteraient d'être différenciés pour être traités indépendamment). Mais il faut bien distinguer ce qui est de l'ordre de l'échantillonnage 'thématique' des moniteurs par un agent espace *a priori* (par exemple, l'espace urbain ne considère pas les moniteurs de maintien d'orientation relative entre bâtiments et routes, car ça n'est pas pertinent), et ce qui est de l'ordre de la différenciation *a posteriori* comme le cas des moniteurs de contrainte sur la végétation : ces moniteurs pourraient être dans un espace émergent avec des moniteurs de proximité, mais un espace émergent composé uniquement de ces moniteurs n'est pas pertinent car le processus utilisé ne sait en traiter aucun.

De plus, l'étendue de la zone de conflit émergente apparaît relativement dépendante de la méthode d'analyse spatiale intégrée à la méthode d'observation. La Figure E-31a montre en bas à gauche deux groupes de conflits proches (de proximité avec les routes et les bâtiments) mais séparés par une route qui ont été regroupés. Ils auraient été séparés si nous avions utilisé un seuil de dilatation un peu moins important (par exemple 8 m au lieu de 10 m ici). En rejoignant le point précédent, il serait peut-être pertinent de lier la méthode d'analyse spatiale utilisée pour délimiter un groupe émergent aux types de contraintes concernées par les conflits persistants.



Figure E-31. Deux résultats d'observation de zones de conflits émergentes lors de l'application de CartACom par CollaGen. (a) deux groupes de conflits ont été regroupés en bas à gauche. (b) les conflits à droite dus à des contraintes sur la végétation sont très différents des conflits de proximité à gauche.

Nous avons également comparé les résultats de notre observateur avec ceux du superviseur de CartACom qui a les mêmes responsabilités, de manière interne à CartACom, d'observer des espaces émergents en cas de conflits persistants (Duchêne et Touya, 2010). La Figure E-32 montre une même zone sur laquelle les groupes observés par l'observateur et le superviseur sont comparés. Contrairement à notre observateur, le superviseur de CartACom a accès à des informations sur le déroulement de la généralisation, dont les conversations entre agents, mais les résultats sont, dans cet exemple, relativement similaires.

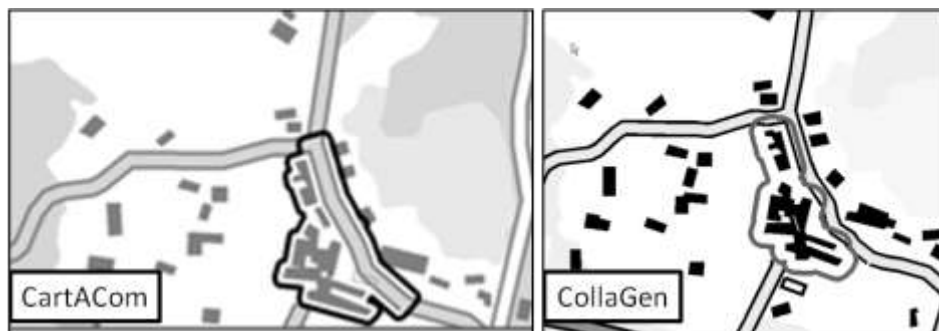


Figure E-32. Comparaison des résultats de l'observation d'espaces émergents dans CartACom et dans CollaGen lors de l'application de CartACom : les résultats sont équivalents malgré des méthodes différentes.

La Figure E-33 montre une comparaison équivalente sur une autre zone. Les espaces émergents observés sont ici relativement différents car le superviseur de CartACom ne forme des groupes émergents qu'avec des objets ayant participé à une conversation avec les objets en conflits et les routes servent à bloquer l'extension d'un groupe. De son côté, l'observateur de CollaGen ne se fie qu'à la proximité spatiale et forme dans l'exemple un espace plus étendu que le superviseur, mais cette délimitation reste pertinente. Dans le cas des espaces émergents, le risque est plutôt de former des espaces trop petits pour encapsuler le contexte local permettant la généralisation donc le fait que notre méthode puisse de temps en temps inclure quelques objets en trop dans l'espace n'est pas vraiment un problème.

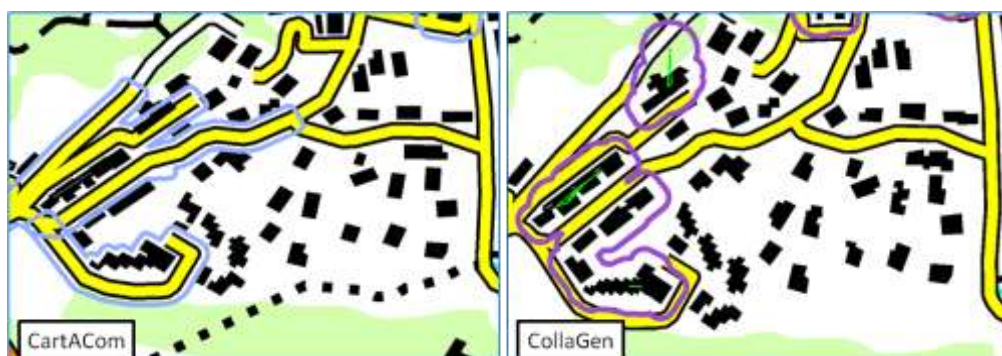


Figure E-33. Autre comparaison des deux méthodes sur une même zone : les espaces émergents observés par CartACom sont séparés car ils reposent sur les conversations et sont bloqués par les routes alors que les espaces émergents de CollaGen sont uniquement formés par proximité spatiale.

La Figure E-34 montre une différence d'observation sur une autre zone pour laquelle CartACom n'observe pas de zone de conflits émergente car les deux conflits de proximité persistants, bien que proche spatialement, sont entre un bâtiment et une route (agent bloquant) et pas entre les deux bâtiments. La méthode dans CollaGen apparaît donc moins précise mais plus systématique (moins perturbée par des configurations particulières).

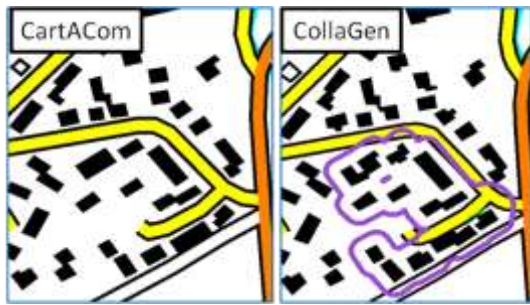


Figure E-34. Autre comparaison des deux méthodes sur la même zone : *CartACom n’observe pas de zone émergente car la route qui partage le conflit de proximité entre les deux bâtiments est un agent bloquant alors que CollaGen se base uniquement sur la proximité spatiale des deux conflits.*

Enfin, nous avons comparé le déroulement de deux généralisations par CartACom, sur la zone de la Figure E-32. Dans la première, CollaGen observe les conflits toutes les 50 itérations, arrête quand les conflits apparaissent, les corrige par un traitement approprié et reprend la généralisation. Dans la deuxième généralisation, nous laissons se dérouler le processus CartACom jusqu’au bout sur la même zone pour observer et corriger les conflits à la fin. Nous remarquons que corriger les conflits au milieu du traitement génère plus d’itérations que de laisser CartACom se dérouler jusqu’au bout (176 activations d’agents contre 158) mais que le temps de calcul est inférieur de quelques 50 millisecondes (pour un temps total de 38 s), pour un résultat identique (Figure E-35). Cela s’explique pour un nombre important de messages supplémentaires à traiter par les agents quand l’observateur corrige des problèmes au milieu du processus (pour rétablir la cohérence). Mais le temps de calcul montre que ces activations supplémentaires sont courtes et que les activations restantes visant à terminer la généralisation sont plus courtes donc *a priori* plus simples. Si cela nécessite d’approfondir ces tests, le principe d’interrompre la généralisation pour simplifier son déroulement après correction externe s’avère vérifié même s’il n’est pas très significatif pour des petits groupes de conflits comme celui de la Figure E-32.

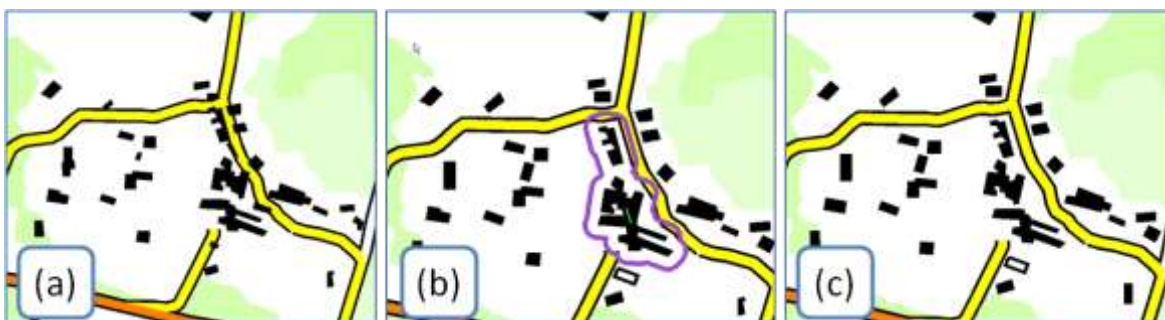


Figure E-35. Les trois étapes de l’observation en ligne d’une généralisation CartACom : (a) avant, (b) à l’émergence d’un groupe de conflits et (c) après sa résolution (ici par un processus enchainant les algorithmes d’élimination et de déplacement d’AGENT choisi manuellement).

Résultats d’évaluation globale

Nous présentons maintenant quelques résultats sur l’évaluation globale effectuée par chaque agent espace après une généralisation par un processus. L’évaluation globale de la généralisation d’un espace géographique est fondée sur l’analyse de la distribution des satisfactions de moniteurs inclus dans l’espace. La Figure E-36 montre deux histogrammes représentant les distributions avant et après généralisation par CartACom d’un espace rural : le nombre de moniteurs satisfaits a augmenté

d'environ 100 (10% des moniteurs pondérés par leur importance) alors que le nombre de moniteurs de satisfaction « passable », « peu satisfait » et « très peu satisfait » a diminué d'autant.

Nous avons vu dans la partie C.14 (p. 155) que l'évaluation globale utilise ces distributions de satisfactions individuelles pour estimer la qualité interne du processus de généralisation, sa qualité globale et si une amélioration significative a été remarquée.

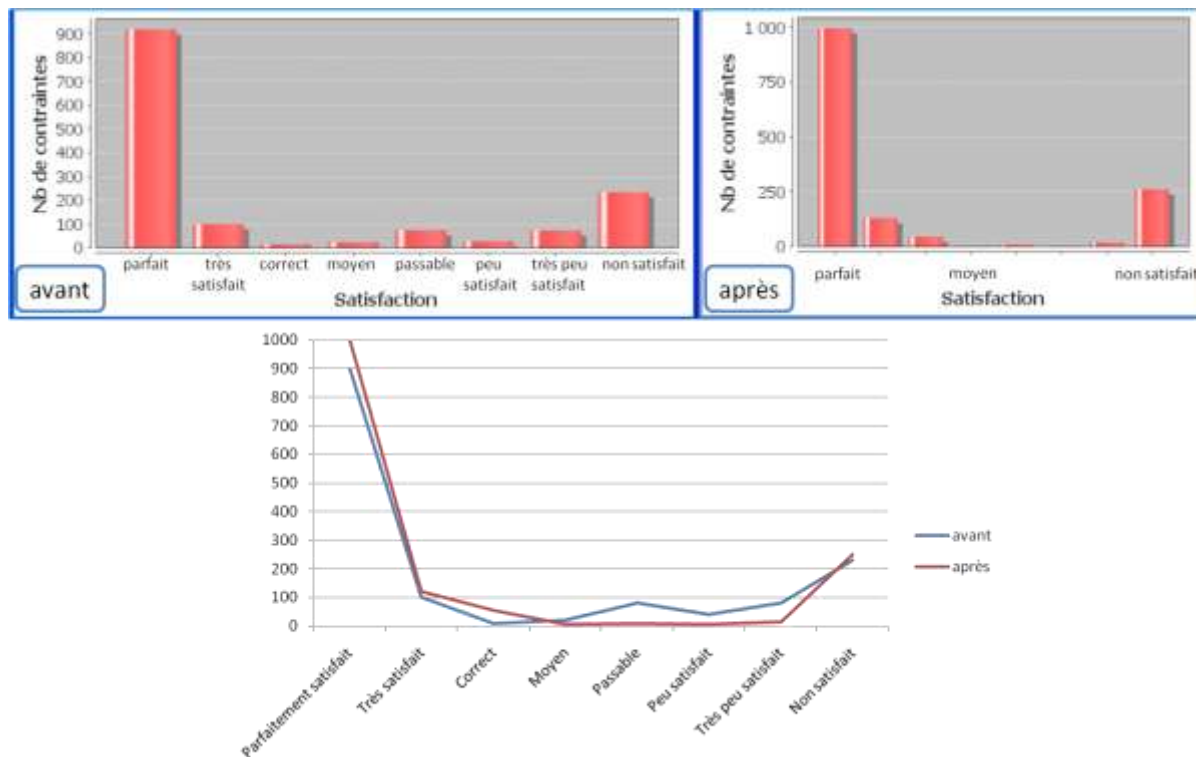


Figure E-36. Evolution de la distribution des satisfactions des contraintes d'un espace rural après généralisation de l'espace par CartACom.

La Figure E 34 montre deux exemples de généralisation d'espace géographique évaluée globalement. Dans l'exemple (a), un espace urbain est généralisé par AGENT et son évaluation montre une très bonne qualité interne (les contraintes décrites dans les post-conditions du processus sont bien satisfaites), une amélioration significative car le nombre de moniteurs non satisfaits a diminué quand le nombre de moniteurs très satisfaits a augmenté (de 30% dans les deux cas) et enfin une bonne qualité globale avec une bonne moyenne pour la satisfaction des contraintes. Cet exemple montre par ailleurs que la méthode d'évaluation de la qualité globale par la moyenne et l'écart-type n'est pas vraiment satisfaisante car elle traduit assez mal que les moniteurs sont, soit très satisfaits, soit très insatisfaits. Cela appuie bien notre intuition qu'une autre méthode est nécessaire pour évaluer de manière plus précise la qualité globale. Dans l'exemple (b), un espace rural généralisé par CartACom montre une qualité interne parfaite (98% des moniteurs gérés *a priori* sont très satisfaits), une amélioration significative et une qualité globale qui n'est pas jugée suffisante du fait d'un écart-type important (car les contraintes gérées par CartACom sont ici moins nombreuses que les contraintes non prises en compte (comme celles sur la végétation ou la proximité entre réseaux routier et hydrographique)). Dans ce cas, le chef d'orchestre décide de valider l'état (car la qualité interne est bonne et l'état a été amélioré), sans qu'il soit considéré comme suffisant et il décide donc de chercher si un autre processus peut être appliqué sur l'espace pour satisfaire les contraintes

restantes (ici, celles sur la végétation). Par ailleurs, la figure montre sur cet exemple le tableau qui récapitule notre évaluation qualitative pour mieux décrire les moniteurs satisfaits et insatisfaits.

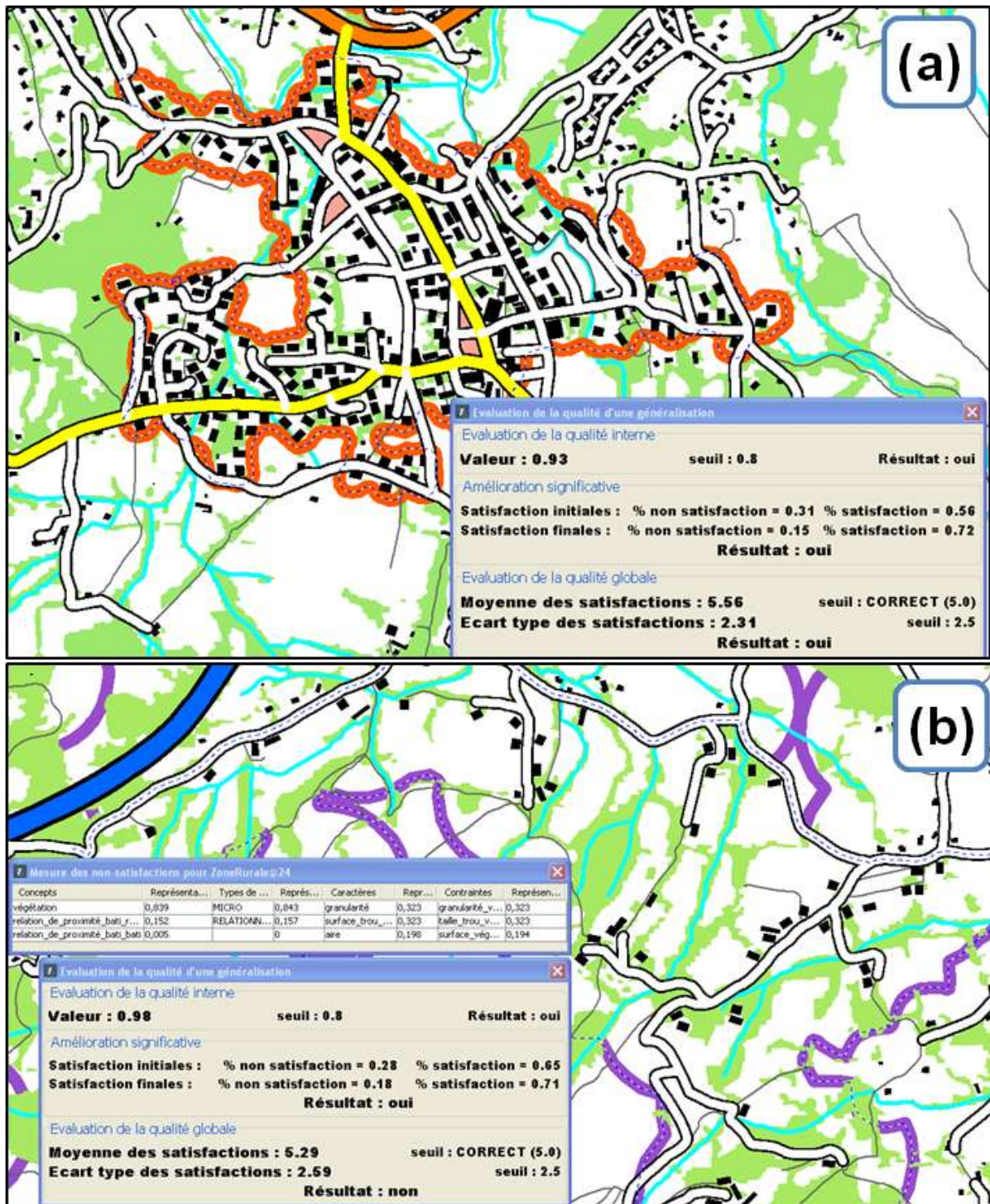


Figure E-37. Description en détail des trois aspects de l'évaluation globale pour un espace urbain généralisé par AGENT (a) et un espace rural généralisé par CartACom (b). Dans l'exemple (b), l'évaluation qualitative est également affichée pour montrer que les moniteurs encore insatisfaits concernent la végétation à 84%.

Enfin, la Figure E-38 montre un exemple de généralisation d'un espace urbain par les moindres carrés urbains (avant AGENT) que le chef d'orchestre ne va pas valider, préférant essayer autre chose à la place. En effet, il y a trop de bâtiments pour que les moindres carrés trouvent une solution moyenne acceptable et même la qualité interne est alors faible.

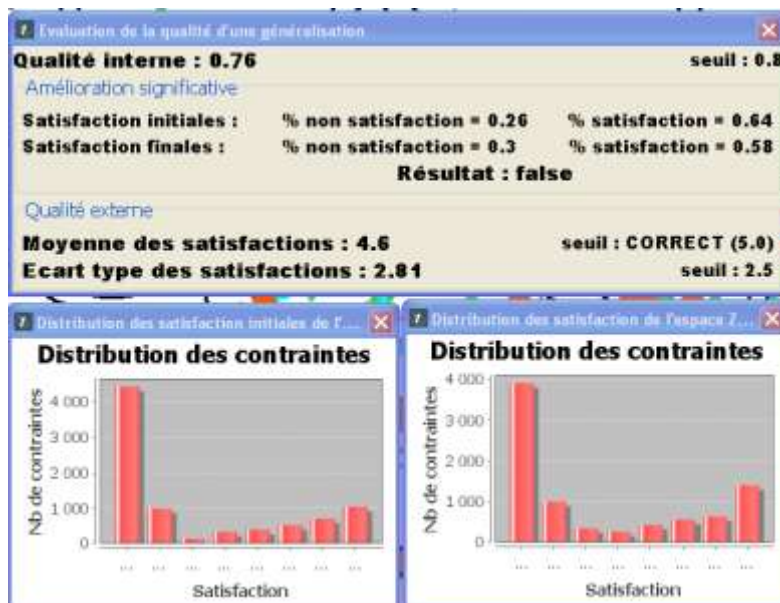


Figure E-38. Evaluation globale de la généralisation d'un espace urbain par le processus de moindres carrés urbains seulement : la progression n'est pas validée car même la qualité interne est faible (il y a trop de bâtiments pour que les moindres carrés trouvent une solution acceptable).

Chef d'orchestre

Le mécanisme de choix du prochain espace à généraliser par le chef d'orchestre a également été implémenté dans notre prototype. Le chef d'orchestre prend bien des décisions en cohérence avec les règles d'orchestration dont il a connaissance, comme l'illustre la Figure E-39. Cette figure représente l'affichage automatique de vérification que renvoie le chef d'orchestre : son choix se situe au début de la phase de généralisation cartographique. En suivant les règles utilisées dans ces tests (présentées en E.3.2), le chef d'orchestre désigne les 'zones urbaines' comme prochain type d'espace à généraliser et ordonne les 23 instances d'espaces urbains en fonction de la quantité de conflits, évaluée ici à partir du taux de moniteurs de contraintes insatisfaits (pondérées par leur importance) et le nombre absolu de moniteurs insatisfaits. Dans ce cas, cela correspond à traiter principalement les grandes zones urbaines d'abord, ce qui est le choix que nous aurions fait interactivement.

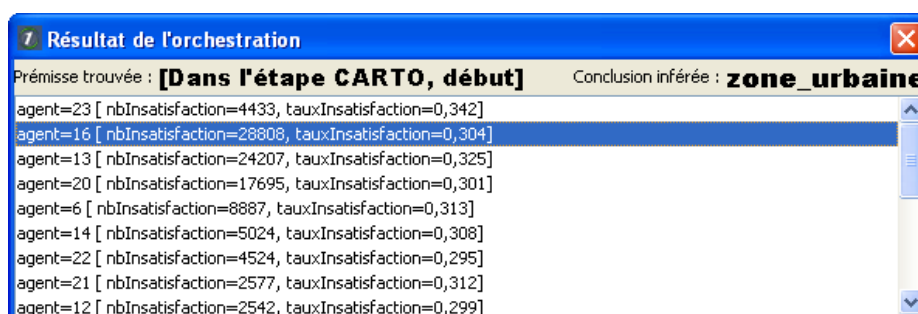


Figure E-39. Choix du chef d'orchestre au début de la phase de généralisation cartographique : il choisit les espaces de type zone_urbaine d'après la règle trouvée et ordonne les instances en fonction de la quantité de conflits (représentée par le nombre de moniteurs insatisfaits et leur représentativité dans l'ensemble des moniteurs)

E.4.2. Etude de sensibilité de CollaGen au choix de la limite des zones urbaines

Nous avons suggéré, en C.9.1, que le choix de la méthode de délimitation d'un espace géographique pouvait se fonder sur une analyse de la sensibilité. Nous avons mis en œuvre cette méthodologie pour étudier la sensibilité de la généralisation au choix des limites des espaces urbains. En effet, les espaces urbains se caractérisent par des définitions assez différentes et il existe des méthodes donnant des résultats assez différents. Il faut noter qu'une grande partie de ce travail a été effectuée par Camille Delahaye lors d'un stage de master que nous avons encadré (Delahaye, 2010). La Figure B-21 présentée plus tôt dans ce mémoire donnait un exemple des limites générées dans le cadre de notre étude de sensibilité avec les quatre méthodes choisies (Boffet, 2001 ; Chaudhry et Mackaness, 2008b ; Walter, 2008 ; Delahaye, 2010). Pour chacune, les paramètres ont été identifiés et nous les avons fait varier en prenant des valeurs logiques tirées des travaux de ces chercheurs ou de nos propres tests. Nous utilisons également des valeurs aléatoires de paramètres pour obtenir une variabilité statistiquement significative des paramètres qui contrôlent la variabilité du résultat (i.e. la limite de la ville). Par exemple, pour la limite utilisant l'algorithme de (Chaudhry et Mackaness, 2008b), la méthode est basée sur une dilatation des bâtiments proportionnelle à un indice d'urbanisation ('citiness'), indice calculé en fonction de sa congestion par les autres bâtiments et de son aire. Les deux paramètres que nous avons fait varier sont le facteur de dilatation fixe que l'on multiplie à l'indice, ainsi que le nombre de bâtiments à prendre en compte pour calculer la congestion. Quarante-limites (vingt pour chaque algorithme) ont ainsi été générées auxquelles s'ajoutent quatre limites dessinées manuellement par des personnes différentes.

Une vingtaine de mesures sont appliquées à l'ensemble des limites générées : des mesures sur la forme de la limite (aire, compacité, taille des trous...), des mesures sur le contenu de l'espace urbain ainsi délimité (distance maximum entre un bâtiment et son plus proche voisin, taille des espaces vides (sans bâtiment ni route) et des mesures sur les données situées juste à l'extérieur de la limite (densité en bâtiments dans une bande de 250 mètres autour de la limite, taille et nature de ces bâtiments). Les processus de généralisation dédiés aux zones urbaines sont *a priori* adaptés au traitement de zones congestionnées denses en bâtiments, donc les mesures sur la taille des espaces vides, la distance maximum entre un bâtiment et son plus proche voisin ou la densité à l'extérieur peuvent être des indicateurs d'une limite bien adaptée à cette généralisation.

Ensuite, la sensibilité est étudiée en généralisant pour chaque limite l'intérieur avec le processus AGENT urbain et l'extérieur (une bande de 250 m autour de la limite) avec le processus CartACom de notre prototype. La Figure E-40 montre quelques exemples de différences avec quatre limites assez différentes : le choix de la limite de l'espace urbain a une influence certaine sur le résultat de la généralisation collaborative. En effet, nous pouvons noter que les zones de densité moyenne en bâtiments sont généralisées assez différemment suivant qu'elles sont incluses ou non dans l'espace urbain (par exemple, le processus dédié à l'espace urbain réalise plus d'éliminations). Nous pouvons également remarquer qu'avec le processus choisi pour généraliser l'espace urbain, les bâtiments ne peuvent pas sortir des limites, ce qui contraint des bâtiments à rester trop près des routes. Une bonne limite de l'espace urbain pour ce processus doit donc laisser assez d'espace aux bâtiments pour s'éloigner des routes. Nous pouvons noter ensuite que le processus dédié à l'espace urbain traite moins bien les endroits où les bâtiments sont alignés le long de la route (à gauche de la zone 2

sur la figure) car le processus ne possède pas à l'heure actuelle d'outils de détection et de traitement de ces structures. Une bonne limite de l'espace urbain doit donc éviter d'inclure ces zones dans l'espace urbain. Il est enfin important de noter que certaines limites entraînent plus d'effets de bord que les autres : sur la zone 3 par exemple, les seuls effets de bord sur la proximité des bâtiments apparaissent sur la quatrième image. Une bonne délimitation des zones urbaines doit donc aussi minimiser les effets de bord possibles en limitant les bâtiments juste à l'extérieur de l'espace urbain. Une dernière remarque non visible sur la figure concerne le temps de calcul, notamment pour la zone 2 : quand des zones trop denses sont laissées en espace rural, le processus chargé de généraliser l'espace rural met beaucoup plus de temps que le processus dédié à l'urbain. Donc le choix de la limite influence à la fois la qualité de la généralisation et le temps pour la réaliser, et une bonne limite de l'espace urbain devrait inclure ces zones pour ne pas alourdir le temps de calcul. Il convient de noter que ces remarques ne sont valables que pour AGENT Urbain et CartACom. Le résultat de l'étude de sensibilité serait peut-être différent avec des processus aux capacités différentes pour traiter les espaces urbains et ruraux.

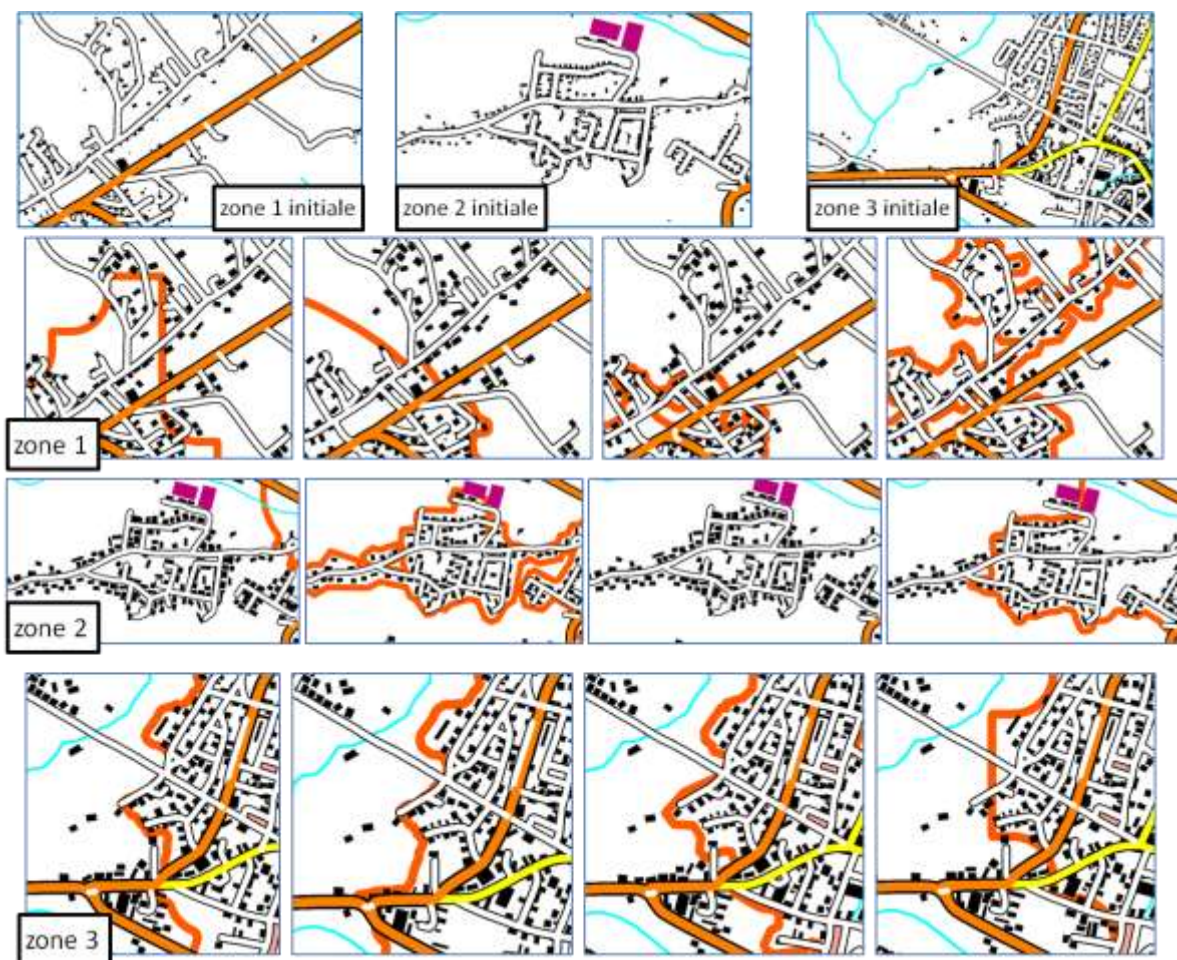


Figure E-40. Influence de la méthode de délimitation de la ville sur sa généralisation collaborative avec trois exemples autour de la ville d'Orthez et quatre limites générées par des méthodes différentes. Les limites apparaissent dans chaque cas avec un trait épais orange. Les résultats varient significativement suivant la limite de ville utilisée.

Par ailleurs, nous avons étudié l'impact de ces différentes limites sur le calcul des îlots à griser comme zones bâties par le processus AGENT, calcul qui prend en compte l'ensemble de l'espace (voir E.1.3). Certains des résultats obtenus sont présentés en Figure E-41. Nous pouvons noter que le cœur

des zones bâties reste le même mais il y a des différences notables à la marge. La généralisation des zones urbaines est donc bien sensible sur plusieurs points au choix de la limite de ces zones.

Cette étude de sensibilité prouve bien que le résultat de la généralisation collaborative dépend de la manière de délimiter les espaces géographiques, comme nous le supposions. L'intensité de cette dépendance peut cependant varier suivant les spécifications. En effet, dans notre cas, nous avons utilisé des spécifications simplifiées et identiques pour évaluer les différences, ce qui limite les différences à la manière dont les processus traitent des spécifications identiques : par exemple la proximité est traitée globalement par groupe dans AGENT et deux à deux dans CartACom. Mais si les spécifications sont en plus assez différentes en zone urbaine et en zone rurale (par exemple, agrégation autorisée seulement en rural et maintien de la répartition des bâtiments seulement en urbain), les différences entre deux délimitations de l'urbain peuvent devenir très importantes. La dépendance est également moins importante dans le cas des zones urbaines si, comme nous le proposons dans notre prototype, nous utilisons un espace périurbain entre l'espace urbain et l'espace rural, car, suivant sa densité en bâtiment et la hiérarchisation de sa structure, l'espace périurbain sera généralisé par des processus plus dédiés à l'urbain ou au rural (plus il y a de moniteurs de contraintes relatifs à l'espace rural comme la densité des groupes de bâtiments, plus le registre va préférer un processus dédié à l'urbain pour traiter l'espace périurbain). L'espace périurbain permet une transition plus douce de l'urbain au rural et atténue l'impact de la délimitation de l'urbain.

L'objectif de cette étude de sensibilité étant également de permettre le choix d'une méthode pour déterminer le contour d'un espace, nous proposons deux solutions *a priori* pour réaliser ce choix. Ces deux solutions n'ont pas été expérimentées par manque de temps et devront donc l'être ultérieurement pour résoudre ce problème. Une des solutions est de réaliser sur quelques cas, avant toute généralisation, et pour chacun des types d'espaces géographiques dont CollaGen possède plusieurs possibilités de délimitation dans la boîte à outils, une telle étude de sensibilité : en étudiant l'évaluation globale de la généralisation autour de la limite pour chacun des cas, nous pouvons déterminer la méthode de délimitation qui donne les meilleurs résultats et l'utiliser directement durant la généralisation de l'ensemble des données. La deuxième solution est de faire un choix une fois la généralisation collaborative démarrée : pour cela, le système dispose des mesures effectuées lors de notre étude sur chaque méthode de délimitation et essaie de mettre en adéquation ces mesures avec les descriptions des processus disponibles *a priori* dédiés à la généralisation de cette espace. Par exemple, un processus plutôt efficace pour la généralisation d'îlots denses de centre ville sera bien compatible avec une méthode de délimitation de la ville qui considère les habitations pavillonnaires plutôt comme extérieures à la ville.

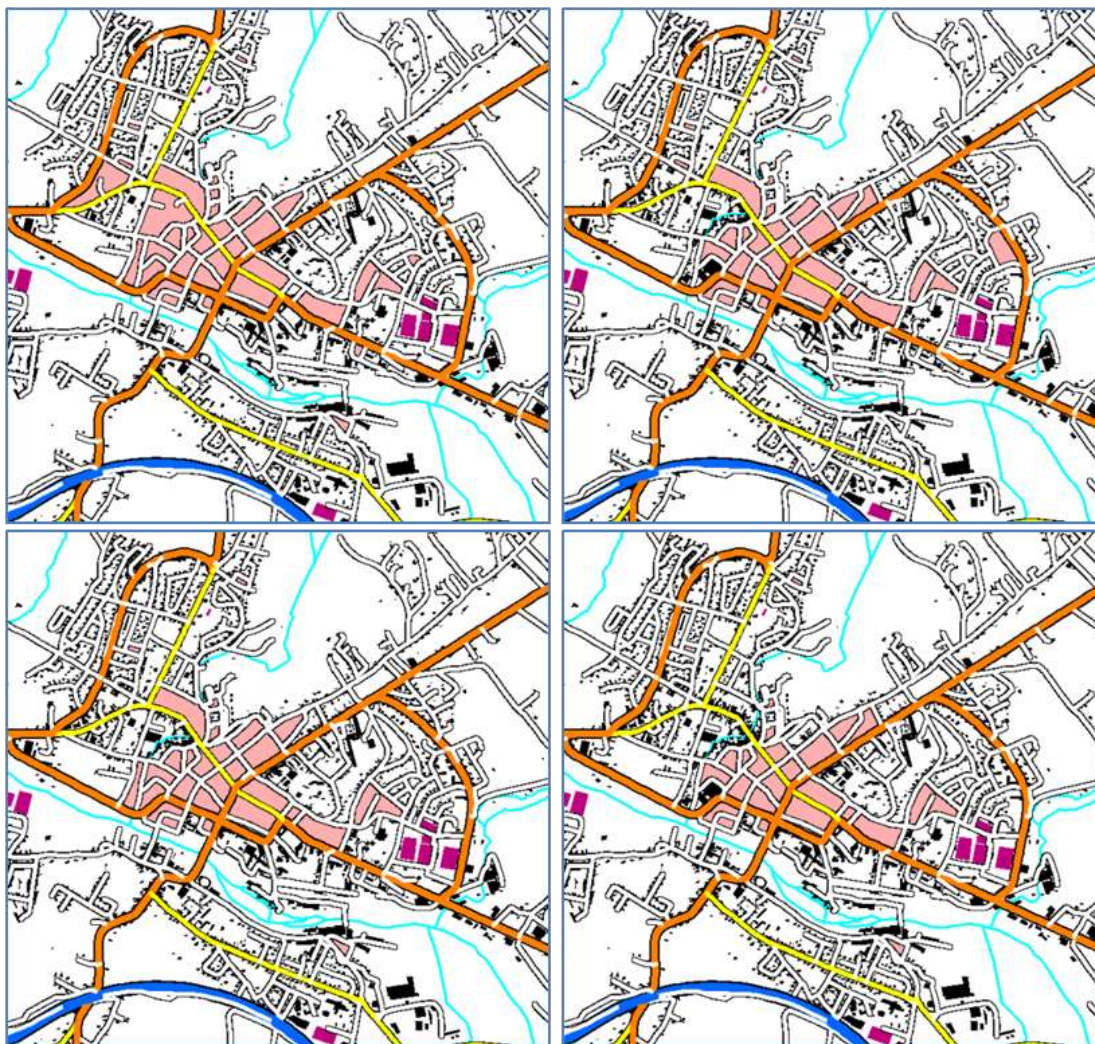


Figure E-41. Îlots transformés en zones bâties par le processus AGENT que nous avons intégré à notre prototype, selon différents choix de limites pour l'espace urbain généralisé par ce processus.

Pour conclure cette partie, nous avons montré par l'exemple des espaces urbains que le choix de la méthode pour délimiter un espace géographique avait une influence significative sur la généralisation collaborative de cet espace et de ses voisins. Nous avons montré qu'une étude de sensibilité était une bonne méthode pour identifier les caractéristiques idéales d'une limite d'espace (dans le but de limiter les erreurs, les effets de bord et le temps de calcul). Une perspective de ce travail de thèse est de fournir une méthodologie détaillée et générique pour déterminer, dans tous les cas, la meilleure limite pour un espace géographique.

E.4.3. Résultats de généralisations complètes par le prototype

Nous présentons, dans cette partie, des généralisations complètes qui exploitent les processus disponibles pour essayer de satisfaire les contraintes testées dans les espaces géographiques utilisés dans nos tests. Si les différents composants (hors gestion des effets de bord) sont complètement implémentés, le prototype ne peut pas encore enchaîner automatiquement les différentes étapes (il manque l'implémentation complète du cycle de vie de certains agents (chef d'orchestre, espaces

géographiques). Pour obtenir ces résultats, nous avons donc déclenché les différentes opérations en cliquant sur notre interface sur les méthodes commandant ces opérations. Par exemple, nous cliquons sur le lancement des enrichissements, puis, nous cliquons sur le choix du prochain espace à généraliser ; nous sélectionnons manuellement le premier espace de la liste des instances et cliquons sur le lancement d'une requête de cet espace au registre, etc. La Figure E-42 montre un résultat obtenu de cette manière autour du village d'Urrugne dans la zone d'Hendaye, la première de nos zones test. Dans cet exemple, seulement trois processus ont été rendus disponibles (AGENT Urbain, CartACom et les Beams) car ils suffisent pour bien généraliser la carte complète comme le montre la figure.



Figure E-42. Résultat d'une généralisation par CollaGen au 1 : 50000 autour d'Urrugne. La zone urbaine apparaît avec un contour épais orange pour permettre la distinction avec les autres espaces.

La Figure E-43 montre une généralisation complète sur une zone plus étendue autour de Cambo-les-Bains, toujours au Pays Basque. Cette fois, l'ensemble des processus de notre plateforme a été rendu disponible et trois types d'espaces géographiques territoriaux ont été créés : un espace urbain, délimité en orange, deux petits espaces périurbains au nord et à l'ouest de l'espace urbain et des espaces ruraux ailleurs. Le réseau routier a également été utilisé comme espace thématique ainsi que la couche de végétation. Enfin, des graphes de flexibilité (extraits du réseau contenant des conflits de superposition entre routes et cours d'eau, et aussi, suffisamment d'arcs pour permettre de diffuser les déformations (Lemarié et al, 2003)) ont été créés comme espaces facilitateurs pour le processus des Beams. Il est intéressant de noter que l'espace périurbain au nord a été (bien) généralisé par AGENT Urbain (choisi pour la prédominance dans l'espace de moniteurs de maintien de la densité des groupes de bâtiments) alors que l'espace périurbain à l'ouest a été (bien) généralisé par CartACom. Il est également intéressant de noter que deux instances (sur une vingtaine) d'espace rural ont vu l'échec de leur généralisation par CartACom (présence d'un bug dans ce cas). Ces généralisations ont été mal évaluées et les moindres carrés ont été utilisés comme deuxième choix, donnant un résultat convenable à défaut d'être parfait. Nous pouvons noter également que les résultats des zones généralisées par AGENT Urbain (que ce soit la zone urbaine ou la zone périurbaine en haut à gauche) ne sont pas encore complètement satisfaisants. La non prise en compte des alignements de bâtiments est vraiment problématique par endroits et des grands bâtiments sont éliminés alors qu'ils auraient dû être conservés : comme aucun processus n'est capable de faire mieux sur notre plateforme, CollaGen ne peut produire de meilleurs résultats.

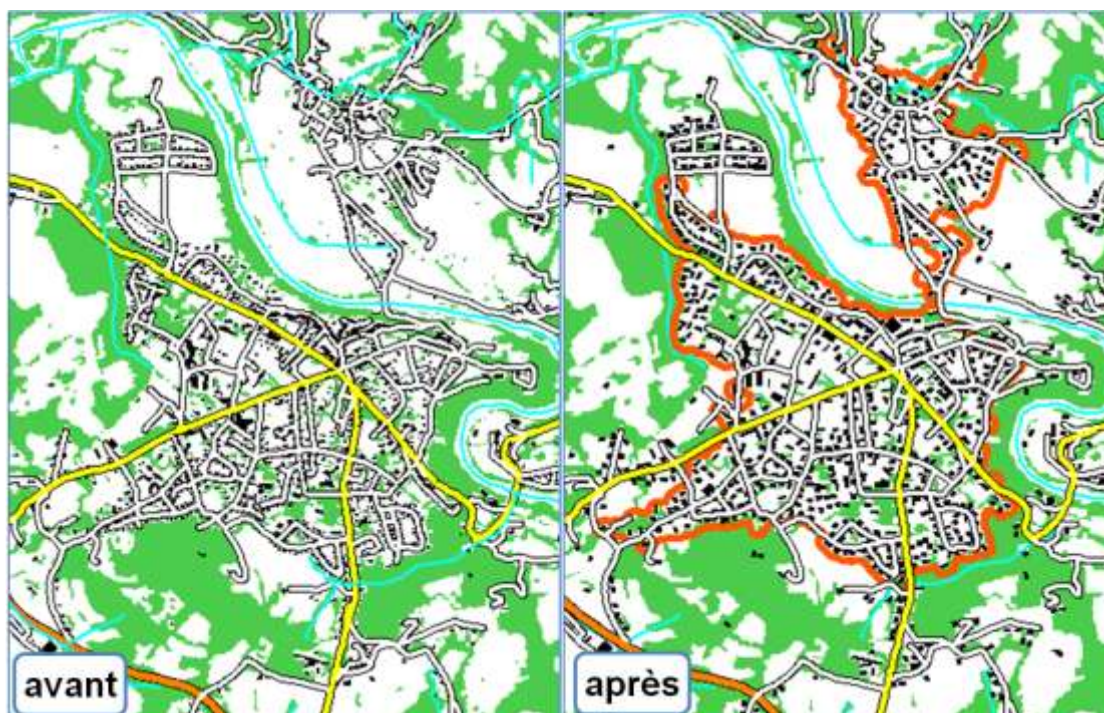


Figure E-43. Résultat d'une généralisation par CollaGen au 1 : 50000 autour de Cambo-les-Bains. La zone urbaine apparaît avec un contour épais orange pour permettre la distinction avec les autres espaces.

La Figure E-44 montre la généralisation complète d'une zone autour de l'Alpe d'Huez (Isère), issue de notre troisième zone test. Comme dans l'exemple précédent, des espaces urbain, périurbain, ruraux ont été construits. Cette zone a également vu la construction d'un espace de routes de montagne et une couche de remontées mécaniques (traits noirs sur la carte). Pour ces derniers, le registre a logiquement choisi les Beams plutôt que le processus par moindres carrés pour les réseaux du fait d'un meilleur indice de confiance (indices que nous avons choisis suite à l'expérience des tests EuroSDR (Stöter et al, 2010)) dans les post-conditions des descriptions formelles.

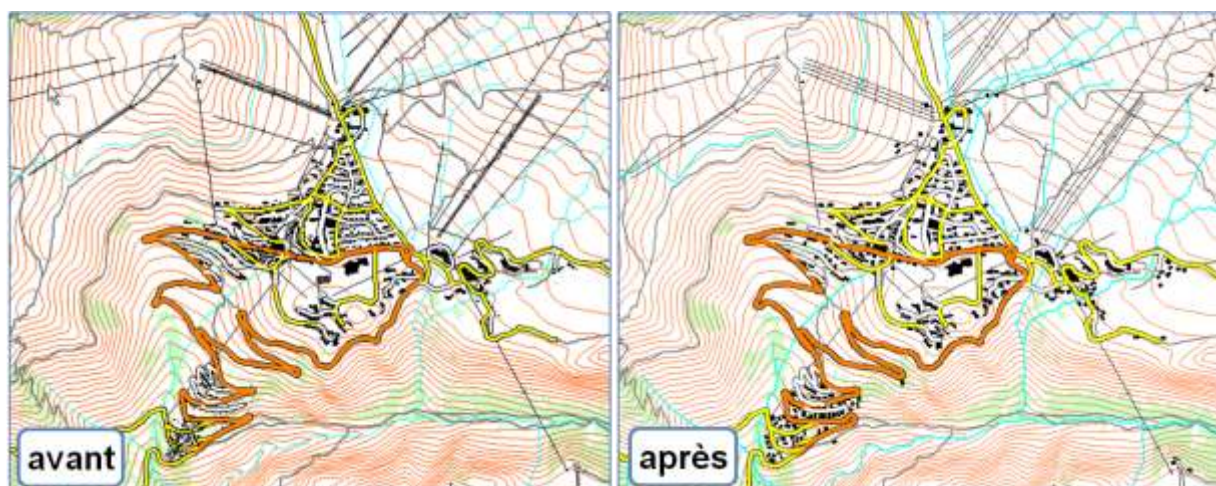


Figure E-44. Résultat d'une généralisation par CollaGen au 1 : 50000 autour de l'Alpe d'Huez.

La Figure E-45 montre un autre résultat autour de la ville d'Orthez (dans la deuxième zone test) sur une étendue beaucoup plus large que les autres cas. L'ensemble des processus disponibles de notre cas d'étude a été exploité. Ce test a démontré encore plus la pertinence de l'approche collaborative de CollaGen, car, sur une zone si étendue, il est arrivé plusieurs fois que le processus utilisé fasse des

erreurs (les processus ne sont pas complètement robustes) et la possibilité d'utiliser un autre processus (même s'il est moins adapté que le premier choisi), permet d'obtenir des résultats acceptables même dans ces espaces.

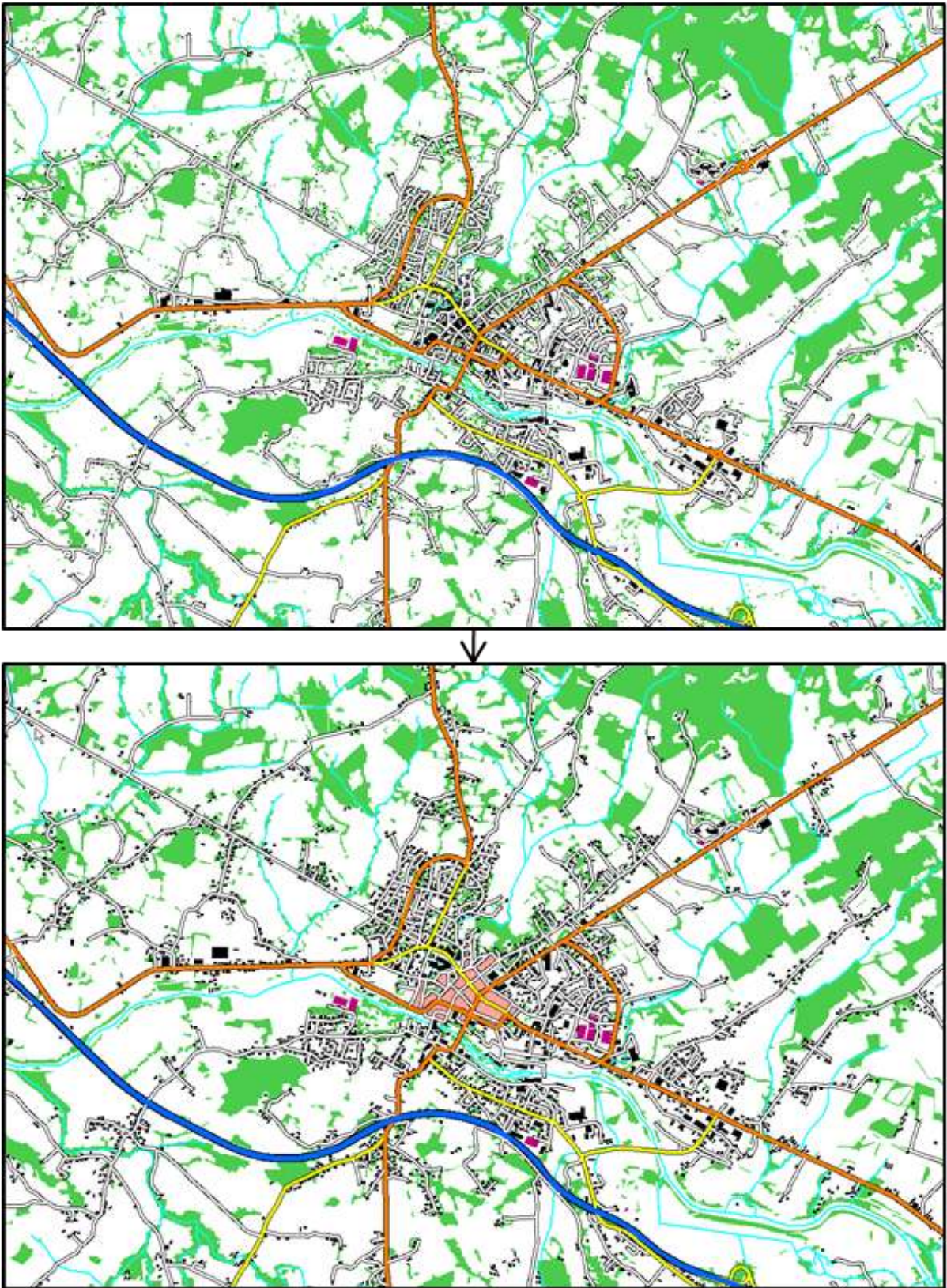


Figure E-45. Résultat d'une généralisation par CollaGen au 1 : 50000 autour de la ville d'Orthez.

Si nous zoomons sur une partie du résultat de la Figure E-45, nous pouvons trouver un exemple intéressant d'espace émergent durant la généralisation d'un espace périurbain, un peu trop dense pour que CartACom trouve une solution à cet endroit (Figure E-46). Un espace émergent a été créé suite à l'observation d'un groupe important de conflits de proximité, concernant des bâtiments entre eux et avec une route. Cette partie n'étant pas automatisée, nous avons choisi manuellement de corriger cette zone avec le processus AGENT urbain qui a donné une bien meilleure solution.



Figure E-46. Zoom sur un espace émergent durant la généralisation collaborative de la zone autour d'Orthez. (a) espace émergent durant la généralisation de l'espace périurbain par CartACom. (b) résultat après généralisation de l'espace émergent par AGENT.

E.4.4. Expérimentations sur l'ordre induit par l'orchestration des espaces

Afin de s'assurer que notre intuition sur l'influence de l'ordre des généralisations induit par l'orchestration était valide, nous avons réalisé quelques expérimentations en faisant varier cet ordre ainsi que les processus utilisés. Nous nous sommes concentrés sur l'influence de l'ordre de déclenchement des types d'espaces fixé par l'orchestration à l'intérieur de l'étape de généralisation cartographique. En effet, il nous paraît assez évident que les grandes étapes que nous avons énoncées ne devraient pas être inversées pour des raisons à la fois logiques (à quoi bon déplacer une impasse pour des questions de lisibilité si elle doit être éliminée plus tard par la sélection ?) et d'efficacité (il vaut mieux commencer par squelettiser les ronds-points pour éviter que les routes qui le composent s'écartent et empiètent sur les bâtiments autour lors de l'étape de généralisation cartographique). La Figure E-47 montre une expérimentation sur l'ordre de généralisation entre un espace rural et un graphe de flexibilité (créé autour du conflit entre la route jaune et l'impasse et englobant tous les tronçons de route de l'image). Les résultats (3) et (4) montrent qu'il est plus efficace de généraliser le graphe de flexibilité (avec les Beams ici) avant l'espace rural (généralisé avec CartACom ici) car cela n'engendre pas d'effets de bord. Pour formaliser ce résultat, nous pouvons considérer que les espaces qui contiennent de nombreux objets dans leur adhérence (voir définition p. 62), comme c'est le cas de ce graphe de flexibilité (les bâtiments au bord de la route sont dans son adhérence), devraient être prioritaires dans l'ordre d'orchestration. Les règles

d'orchestration devraient traduire cette priorité. Le résultat (2), bien que légèrement moins bon que le (4), montre que cet ordre peut être relativisé par le choix des processus : l'espace rural ayant été généralisé par AGENT Urbain, puis, les moindres carrés, il n'y a pas besoin en théorie de généraliser le graphe de flexibilité (traité avant si on se fie aux règles d'orchestration choisies). Ce résultat montre donc que la méthode d'orchestration que nous proposons devrait éventuellement être adaptée pour prendre en compte les processus disponibles pouvant s'appliquer sur l'espace concerné par la règle : nous pourrions faire des règles plus complexes ou modifier le mode de raisonnement du chef d'orchestre pour qu'il ne prenne pas la même décision si les espaces ruraux peuvent être généralisés plutôt par moindres carrés ou plutôt par CartACom.

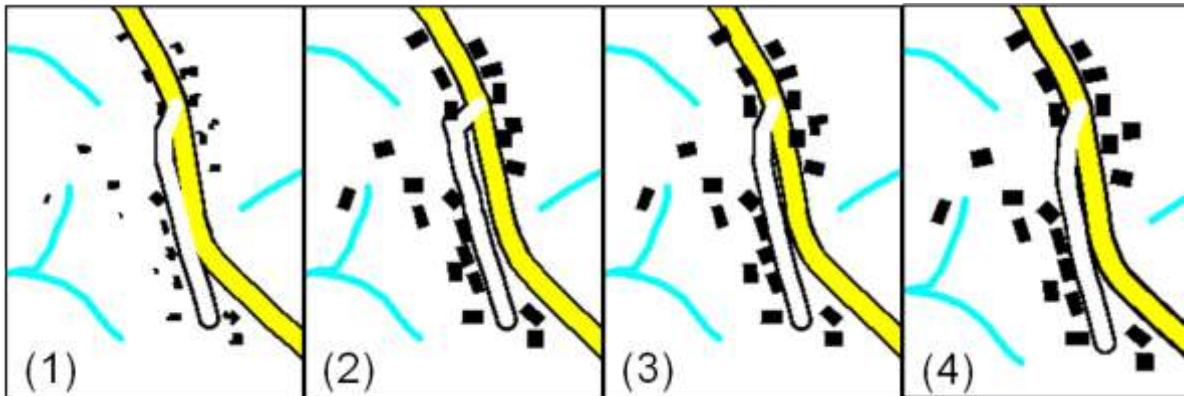


Figure E-47. Illustration de l'influence de l'ordre d'orchestration: (1) avant généralisation. (2) généralisation par AGENT Urbain puis les moindres carrés. (3) généralisation par CartACom puis les Beams. (4) généralisation par les Beams puis CartACom.

E.5. EVALUATION DES RESULTATS

Dans cette partie, nous allons évaluer les résultats obtenus avec le prototype de CollaGen et présentés dans la partie précédente. La première évaluation traite de la formalisation des ressources et notamment des spécifications. La deuxième évaluation compare les résultats obtenus grâce à CollaGen à ceux que l'on peut obtenir en utilisant les logiciels du commerce. Les résultats sont ensuite évalués par comparaison avec le résultat d'une généralisation issue de l'utilisation d'un unique processus. Enfin, les résultats sont évalués par comparaison avec une approche ad-hoc dans laquelle la séquence d'application des processus a été établie une fois pour toutes.

E.5.1. Evaluation de la formalisation des spécifications

Nous avons proposé, dans CollaGen, des modèles formels pour saisir des contraintes de généralisation et des règles opérationnelles. Nous évaluons dans cette partie la généralité de ces modèles formels lorsqu'ils sont confrontés à la saisie de données réelles. Ainsi, nous avons essayé de saisir l'ensemble des contraintes exprimées par au moins un des producteurs de données institutionnelles participant aux tests du projet EuroSDR, sous la forme de contraintes formelles et de

règles opérationnelles. Si nous avons réussi à toutes les exprimer dans notre formalisme, certaines formalisations n'étaient pas directes. Si nous prenons l'exemple de la contrainte de type objet-champ décrite dans (Gaffuri et al, 2008) 'une rivière doit couler dans son talweg'. La difficulté de la saisie réside dans l'obligation d'utiliser une contrainte de type relationnelle sur la relation 'd'écoulement d'un cours d'eau dans son talweg', décrite dans notre ontologie comme une saisie relation objet-champ. Il faut ensuite traduire la contrainte en choisissant la propriété d'écoulement moyen de la relation (Gaffuri, 2008) avec une expression de type seuil (la valeur de la propriété d'écoulement moyen de la relation doit être inférieure à un seuil). Prenons un autre exemple avec la contrainte 'un bâtiment presque parallèle à une route doit devenir parallèle': il faut ici contraindre la relation d'orientation relative entre un bâtiment et une route et sa propriété de 'différence d'orientation'. Il faut alors utiliser une expression de type seuil, un opérateur '=' et une valeur de '0' qui signifie que la valeur de ce caractère doit être nulle.

Afin de valider complètement ce modèle, nous souhaiterions mettre en place des tests auprès d'utilisateurs de processus de généralisation, pour voir s'ils sont capables de saisir des contraintes facilement et sans erreur. Toutefois, une perspective de cette thèse étant de disposer d'une large bibliothèque de contraintes classiques sur lesquelles un utilisateur pourrait s'appuyer, cela retire un peu d'importance à la facilité d'utilisation de ce formalisme.

Il y a également un point à noter dans l'évaluation de la généralisation produite par CollaGen en termes de satisfaction de contraintes. En effet, nous avons réalisé une réelle dichotomie *Contrainte formelle/Moniteur de contrainte* dans la modélisation du comportement d'une contrainte. Bien qu'un moniteur de contrainte soit censé encapsuler localement la signification d'une contrainte, il le fait à l'aide d'algorithmes intégrés à CollaGen par le développeur. L'utilisateur, qui saisit ses spécifications sous forme de contraintes formelles, ne fait qu'apparier ces contraintes formelles avec le moniteur qu'il souhaite utiliser pour modéliser la contrainte et évaluer sa satisfaction. Les moniteurs utilisent les valeurs de seuil saisies dans les contraintes formelles pour établir leur satisfaction. Toutefois, deux moniteurs peuvent utiliser la même valeur mais donner des valeurs de satisfaction légèrement différentes (notamment quand la contrainte n'est pas complètement satisfaite, la distance à la valeur souhaitée peut être interprétée de manières différentes). La satisfaction des contraintes, utilisée dans les composants de CollaGen est en fait toujours une extension de celle formulée assez simplement dans la contrainte formelle par l'utilisateur. Les résultats présentés dans la partie précédente doivent donc être vus à travers le filtre de cette dichotomie. Pour pallier à ce risque de hiatus, deux possibilités existent : nous pouvons trouver un modèle pour rendre les contraintes formelles plus explicites sur la manière de déterminer leur satisfaction, avec des fonctions d'évaluation par exemple, comme nous le proposons en C.5.3 ; nous pouvons également avoir recours à une description plus détaillée des contraintes et de la manière dont elles sont satisfaites suivant les valeurs, dans l'ontologie : il suffirait alors d'annoter sémantiquement le moniteur avec le comportement correspondant dans l'ontologie pour que le lien contrainte formelle/moniteur soit plus direct.

E.5.2. Comparaison avec des logiciels du commerce

Afin d'évaluer l'intérêt de l'approche de généralisation collaborative et du modèle CollaGen, nous avons comparé les résultats obtenus avec le prototype avec ceux obtenus avec des logiciels du

commerce proposant des processus de généralisation cartographique. Ces logiciels ont été testés lors du projet EuroSDR sur l'état de l'art des logiciels du commerce en généralisation cartographique (Stöter et al, 2010) et nous avons utilisé les meilleurs résultats obtenus avec ces logiciels sur notre zone test de l'Alpe d'Huez.

Tout d'abord, le logiciel Clarity™ de la société 1Spatial (qui sert de support à notre plateforme de travail) permet d'exécuter des processus basés sur le modèle AGENT et les Beams. Sur la Figure E-48, trois zones sont mises en valeur pour analyser les différences entre les deux cartes. La zone 1 met en avant le problème de généraliser une zone périurbaine comme n'importe quelle zone urbaine : le résultat est de bien moins bonne qualité que la version CollaGen qui a identifié une zone périurbaine et l'a généralisée par CartACom. La zone 2 souligne le même problème pour les zones rurales : en identifiant une zone rurale et en choisissant CartACom pour la généraliser, CollaGen produit de bien meilleurs résultats. La zone 3 est centrée sur des remontées mécaniques trop rapprochées que les Beams n'ont pas réussi à écarter : CollaGen a permis dans ce cas d'essayer dans un deuxième temps un processus par moindres carrés qui a donné entière satisfaction. La Figure E-49 montre les mêmes résultats affichés à la taille réelle de la carte et comparés aux données initiales.

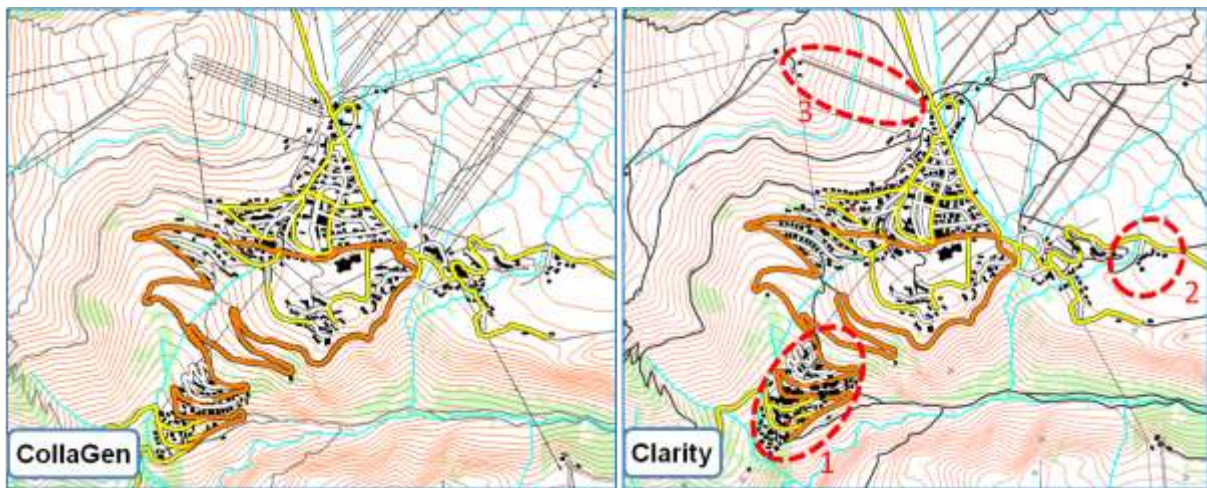


Figure E-48. Comparaison CollaGen/Clarity™ avec les meilleurs résultats obtenus sur Clarity durant le projet EuroSDR de (Stöter et al, 2010). Les trois zones d'intérêt en pointillé montrent des espaces sur lesquels CollaGen amène des résultats nettement meilleurs.

Le logiciel suivant est plutôt une suite logicielle comprenant Change®, Push® et Typify®, tous développés par l'Université de Hanovre (Sester, 2005) que l'on résume sous le terme CPT. Dans cet exemple, nous avons mis en valeur cinq zones intéressantes pour la comparaison (Figure E-50). La première zone est centrée sur les routes de montagne. CPT traite les routes de montagne indifféremment des autres problèmes de superposition entre symboles de réseaux, c'est-à-dire par moindres carrés, ce qui entraîne un trop fort écartement des têtes de virages. La solution par CollaGen permet de différencier la superposition entre objets différents de l'empâtement, ce qui produit un résultat de meilleure qualité. La zone 2 est centrée sur une zone rurale pour laquelle un traitement de typification plutôt adapté aux zones périurbaines a été utilisé ce qui efface les différences entre bâtiments, bien conservées par CartACom dans le résultat obtenu par CollaGen (Typify commence par transformer tous les bâtiments en carrés avant de les typifier). La zone 3 montre que CPT ne dispose pas de traitement spécifique pour les forêts ce qui illustre une faiblesse générale de ces solutions commerciales (ou de recherche pour CPT) : elles ne peuvent pas traiter la généralisation d'une carte complète. Les zones 4 et 5 illustrent cette fois une des limites de

CollaGen : la dépendance à la qualité intrinsèque des processus de généralisation disponibles. La zone 4 illustre une des faiblesses connues du processus AGENT Urbain utilisé par CollaGen (et non du modèle AGENT) : la gestion des alignements de bâtiments beaucoup mieux traité ici par Typify, car il manque des algorithmes pour que le processus AGENT Urbain les prennent en compte. Enfin, la zone 5 montre une portion de réseau routier sur laquelle les Beams ont manifestement échoué dans CollaGen (car l'implémentation utilisée n'est pas très fiable) alors que Push a produit un bon résultat.

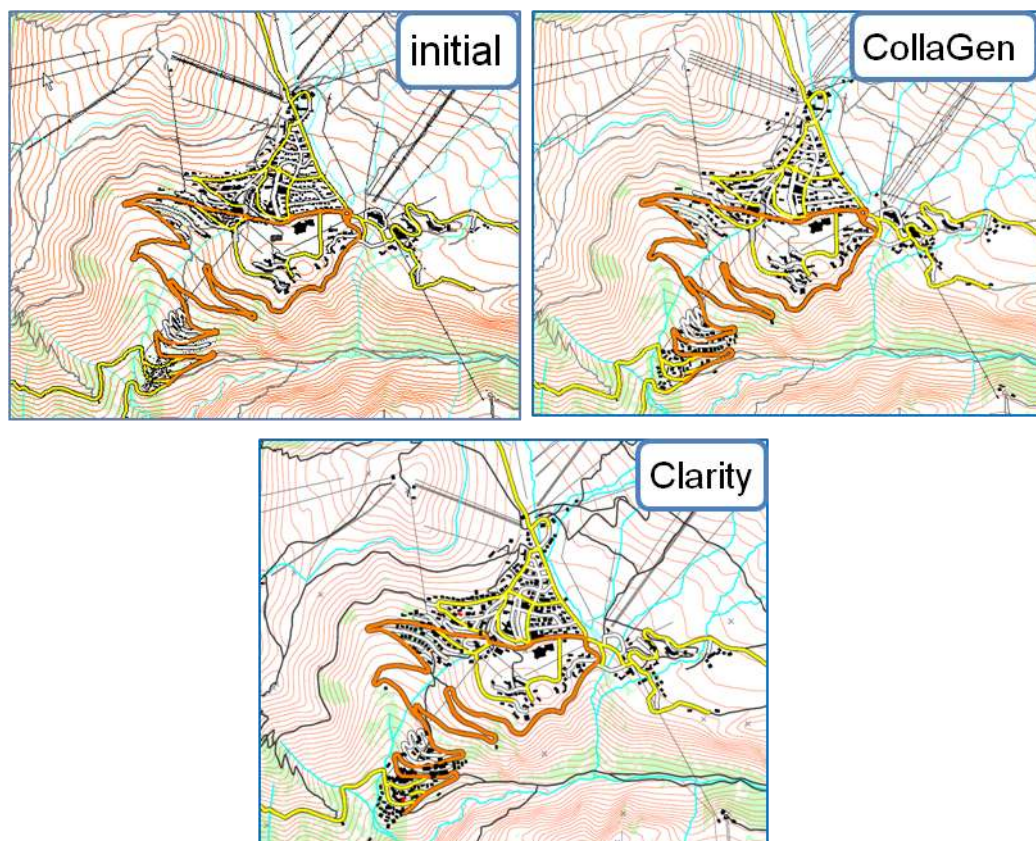


Figure E-49. Comparaison CollaGen/Clarity™ à la taille réelle de visualisation de la carte au 1 : 50000.

La Figure E-51 montre les résultats de la Figure E-50 à la taille réelle de la carte au 1 : 50000.

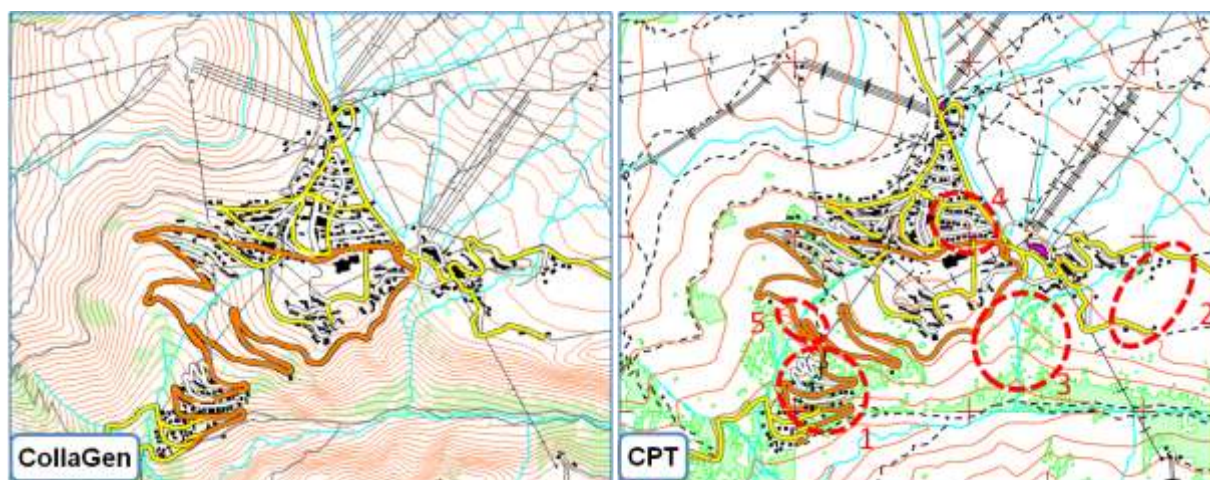


Figure E-50. Comparaison CollaGen/Change-Push-Typify (CPT) avec les meilleurs résultats obtenus sur CPT durant le projet EuroSDR de (Stöter et al, 2010).

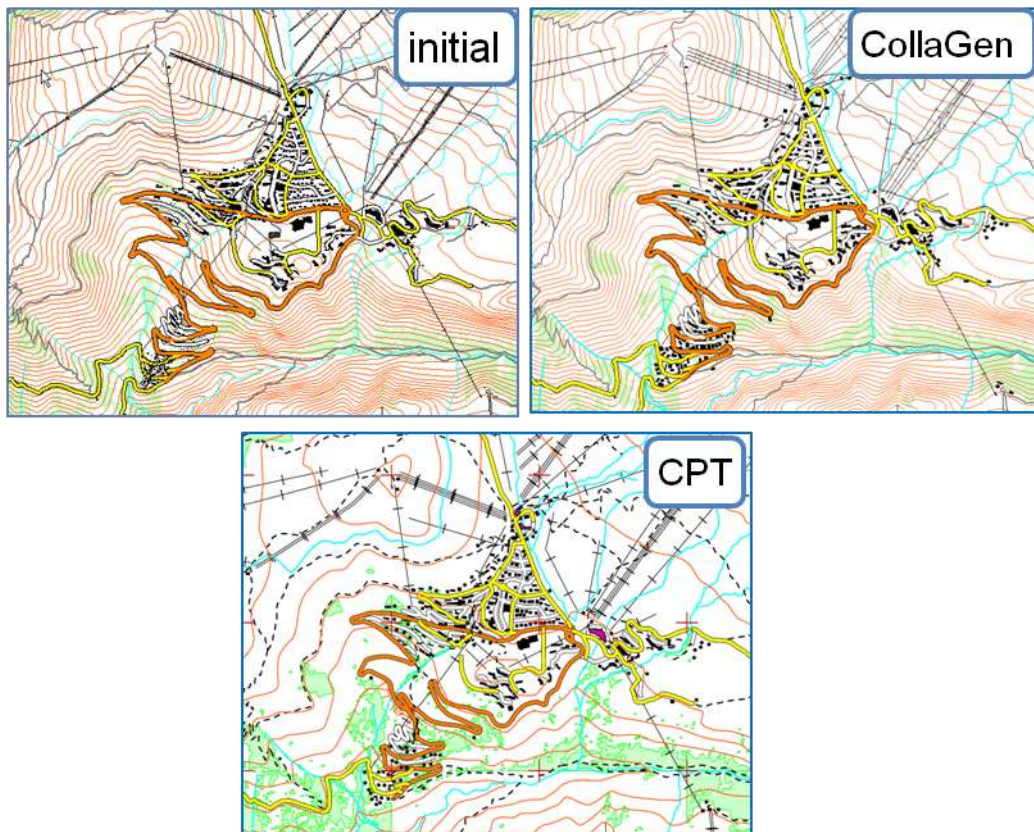


Figure E-51. Comparaison CollaGen/CPT à la taille réelle de visualisation de la carte au 1 : 50000.

La Figure E-52 montre une comparaison avec un dernier logiciel testé lors du projet EuroSDR : Aexpand de la société Axes Systems®. La première zone mise en avant dans cette comparaison fait apparaître des ronds-points qui ont été dilatés pour être lisibles au lieu d'être squelettisés avant la phase de généralisation cartographique. Cet exemple illustre l'intérêt, de distinguer plusieurs étapes de généralisation dans l'orchestration des processus : CollaGen simplifie les petits ronds-points lors de l'étape de 'Changement de géométrie', puis, lors de la phase de généralisation cartographique, les ronds-points restants peuvent être dilatés pour être plus lisibles.

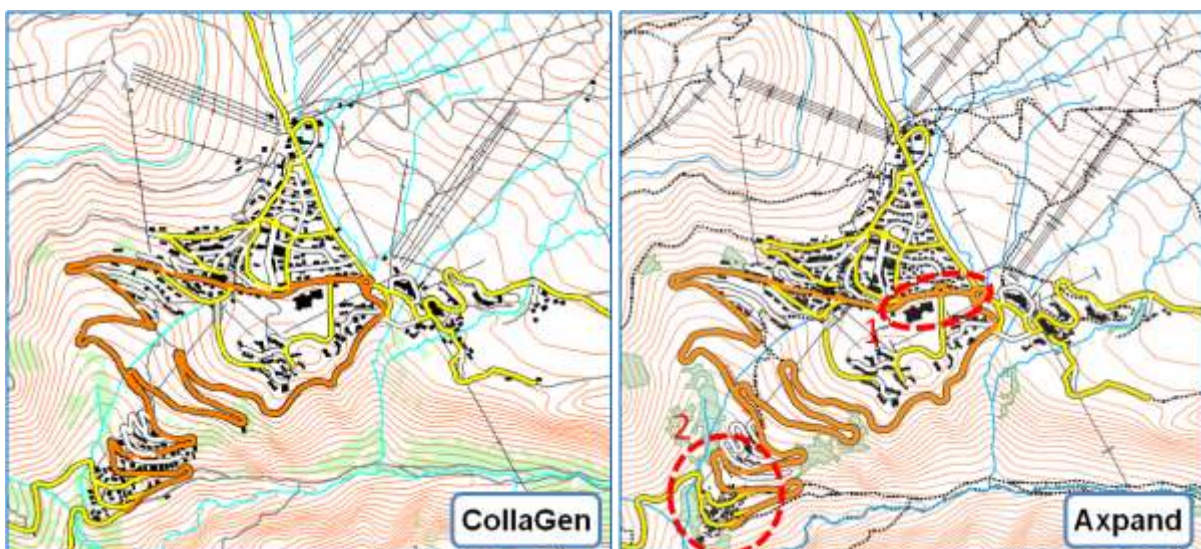


Figure E-52. Comparaison CollaGen/Aexpand avec les meilleurs résultats obtenus sur Aexpand durant le projet EuroSDR de (Stöter et al, 2010).

La deuxième zone montre à nouveau l'intérêt d'identifier des espaces périurbains pour leur appliquer des processus plus adaptés. La montre les résultats de la Figure E-52 à la taille réelle de la carte au 1 : 50000, comparés aux données initiales.

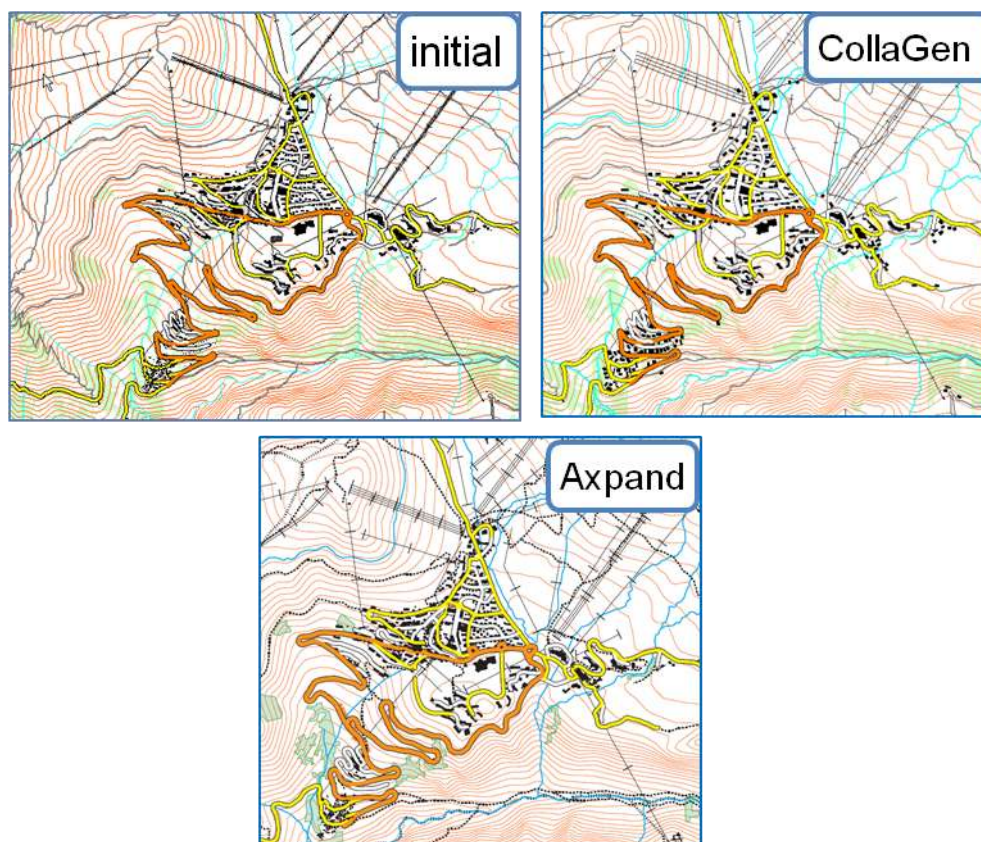


Figure E-53. Comparaison CollaGen/Axpand à la taille réelle de visualisation de la carte au 1 : 50000.

E.5.3. Comparaison avec un modèle unique

Nous avons émis comme hypothèse, au départ de ce travail de thèse, qu'il n'était pas intéressant d'essayer de réaliser une généralisation complète avec un seul processus de généralisation. Nous vérifions donc rapidement cette hypothèse en comparant dans une zone, plutôt rurale, ne présentant pas de conflits trop importants, une généralisation par CollaGen, une généralisation utilisant uniquement le modèle AGENT, paramétré différemment suivant le type d'objets meso rencontrés (les paramètres ne sont pas les mêmes pour les îlots urbains que pour les îlots ruraux) et une généralisation par CartACom. CartACom n'est pas arrivé à généraliser les zones les plus denses en bâtiment, car il atteint ses limites de conception et n'est pas très tolérant aux pannes. Il permet néanmoins une généralisation équivalente à celle de CollaGen (qui a aussi utilisé CartACom), sur les parties les plus rurales de la zone. La Figure E-54 montre les résultats obtenus avec CollaGen et ceux obtenus avec un processus AGENT spécialisé pour traiter différents types de meso (urbain, périurbain, rural, en utilisant la classification de (Trévisan et Gaffuri, 2004)).

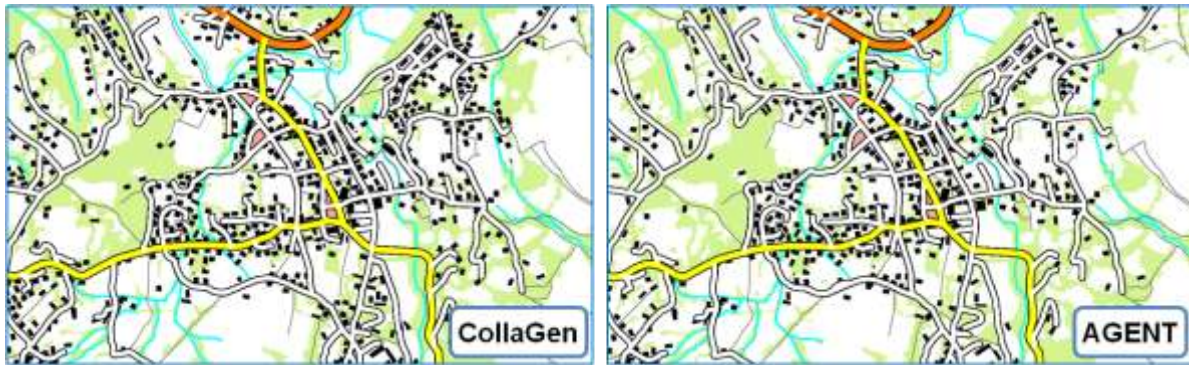


Figure E-54. Comparaison d'une généralisation par CollaGen avec une généralisation par AGENT seul sur la zone rurale autour d'Ascain au Pays Basque.

Même paramétré différemment, AGENT ne permet pas d'obtenir des résultats aussi fins que CollaGen sur les zones rurales comme celle en haut, à gauche de la Figure E-55. Même si les résultats sont assez proches (AGENT permet donc d'obtenir seul des résultats acceptables), l'harmonie de la densité en bâtiments entre les espaces ruraux et urbains semble mieux conservée par CollaGen car CartACom permet de mieux préserver la densité en zone rurale.



Figure E-55. Comparaison CollaGen/AGENT seul à taille réelle (pour le 1 : 50000), avec les données initiales.

Le recours à AGENT sur des îlots ruraux, même s'ils sont construits différemment qu'en zone urbaine, ne permet pas de gérer les relations géographiques entre objets relativement isolés : les algorithmes d'élimination ont tendance à détruire les alignements (car ce n'est pas ce qu'on leur demande) et le déplacement ne maintient pas les orientations relatives (idem), contrairement à CartACom utilisé par CollaGen (Figure E-56). De plus, CollaGen a la possibilité de faire appel à un processus par moindres carrés quand la situation l'exige comme dans le cas, assez fréquent dans cet exemple, où il faut déplacer une impasse pour laisser à AGENT ou CartACom la place pour déplacer les bâtiments.



Figure E-56. Zoom sur un extrait de la figure précédente qui montre qu'AGENT n'a pas géré les orientations relatives entre bâtiment et route au contraire de CollaGen qui a utilisé CartACom sur cette zone rurale.

E.5.4. Comparaison avec un processus ad-hoc

CollaGen est un modèle qui permet d'orchestrer la collaboration entre processus de généralisation de manière automatique. Mais, avec un système dans lequel l'espace adapté à chaque processus disponible a été étudié, il est possible de mettre en place une orchestration manuelle ad-hoc avec un choix de séquences de généralisation prédéfini adapté à un problème de généralisation donné. L'IGN est en train de mettre en place une nouvelle chaîne de production de séries de cartes au 1 : 25000 utilisant plusieurs processus de généralisation automatiques en utilisant cette approche (Lecordix et al, 2007). Dans ce système, plusieurs processus issus des modèles AGENT, GAEL, Beams, ainsi qu'un processus de généralisation de la végétation sont combinés pour produire la carte généralisée. Il y a même une gestion intégrée (à l'aide de GAEL) des effets de bord induits sur les bâtiments par l'application des Beams sur les routes. La Figure E-57 montre un extrait d'une carte générée automatiquement. Il convient de positionner la contribution de CollaGen par rapport à ce type de système. Du point de vue théorique tout d'abord, la contribution principale de CollaGen est d'être un modèle générique qui peut faciliter la mise en place de chaîne de production. Il permet par exemple d'ajouter un nouveau processus sans changer toute la chaîne de production. La production de cartes au 1 : 25000 est l'aboutissement du travail d'une équipe de trois à quatre ingénieurs pendant six ans, même si tout le temps n'a pas été consacré uniquement à la généralisation cartographique. Il n'est pas envisageable de réaliser une nouvelle fois ce travail pour une échelle légèrement différente ou des spécifications légèrement différentes et CollaGen permet ce type d'adaptation assez souple : dans CollaGen, le changement d'échelle est géré par le registre qui va proposer des réponses différentes et de nouvelles spécifications peuvent aussi entraîner des changements dans les processus choisis automatiquement par le registre.



Figure E-57. Extrait d'une carte au 1 : 25000 produite automatiquement à l'IGN.

Du point de vue des résultats cartographiques, les cartes produites sont de très bonne qualité donc il y a peu de chances que CollaGen apporte quoi que ce soit en ayant à disposition les mêmes processus. Néanmoins, pour la production des cartes au 1 : 50000, les problèmes de généralisation sont plus nombreux et complexes à résoudre. Ainsi, les premiers essais de production de carte au

1 : 50000 à partir de la chaîne de production du 1 : 25000 donnent des résultats moins convaincants (Figure E-58). L'étude de ces résultats fait apparaître que le même processus ne peut pas être utilisé pour tous les paysages (Marchais, 2010) ce qui légitime le recours à la généralisation collaborative. De plus, ces résultats font apparaître que la généralisation peut donner de bons résultats dans une zone avec certains paramètres et de mauvais dans une zone différente mais qui possède des caractéristiques proches. Ce point plaide pour l'utilisation d'un système comme CollaGen capable de s'adapter à des situations non prévues en essayant d'autres processus, même si des tests sur les mêmes données seront nécessaires pour s'en assurer. Nous pouvons donc dire que la contribution de CollaGen, par rapport à une telle approche ad-hoc de la généralisation complète, est la genericité qui permet de s'adapter plus facilement à des changements de spécifications sans devoir tout redévelopper et de réactivité qui permet d'adapter le ou les processus à utiliser dans un espace géographique donné. Cette capacité de réaction plus importante est aussi valable pour la gestion des effets de bord qui ne sont pas prévus à l'avance dans notre système, comme cela doit être le cas dans une approche ad-hoc.



Figure E-58. Résultat préliminaire de la mise en place d'une chaîne de production de cartes au 1 : 50000 par généralisation automatique à l'IGN.

E.6. DISCUSSION

Plusieurs points doivent être discutés pour compléter la présentation des résultats de CollaGen et leur évaluation. Tout d'abord, nous souhaitons mentionner les difficultés que nous avons rencontrées pour avoir à disposition des processus de généralisation donnant des résultats acceptables. (Renard et al, 2010) décrivent les difficultés pour capitaliser sur le code des recherches passés et nous y avons été confronté. Nous avons donc passé une partie de notre temps de recherche sur la capitalisation de processus pour pouvoir réellement tester notre modèle avec des

processus relativement efficaces et variés. Ces difficultés expliquent l'absence de résultats utilisant un processus issu du modèle GAEL pour traiter les zones de montagne.

Par ailleurs, comme nous l'avons rappelé plusieurs fois dans ce mémoire, la qualité des résultats de CollaGen est directement dépendante de la qualité des processus disponibles que le système utilise, processus qui sont souvent eux-mêmes dépendants de la qualité des algorithmes qu'ils manipulent. La qualité des résultats de CollaGen en zone urbaine dépend de la qualité d'AGENT Urbain qui dépend directement de la qualité de ses algorithmes (nous avons dû développer des algorithmes de déplacement et d'élimination des rues pour obtenir des résultats corrects au 1 : 50000). Les résultats de CollaGen sont aussi fortement dépendants du nombre des processus et pas seulement de leur qualité. En effet, nous avons vu que le premier processus choisi par le registre n'est pas toujours suffisant pour bien généraliser certains espaces dont la configuration est particulière : si CollaGen peut faire appel à un processus alternatif, il a beaucoup plus de chances d'aboutir à un résultat global satisfaisant. Pour résumer, la redondance de processus permet une meilleure robustesse aux pannes. Toutefois, des tests mettant en jeu plus de processus sont nécessaires pour valider ce postulat.

Sur un plan complètement différent, nous souhaitons aborder la difficulté à remplir les rôles de fournisseur et d'utilisateur dans CollaGen, lorsque l'on n'est pas le développeur (donc contrairement à nous qui remplissons tous les rôles). Les tâches assignées au fournisseur, décrites en C.2, ne sont pas très longues mais nécessitent de bien connaître le fonctionnement du processus rendu disponible, notamment pour la description formelle ou la création des méthodes relatives aux interactions avec l'observateur (mise en pause puis mise en cohérence des modifications). Il serait donc important de réaliser des tests auprès de personnes ayant développé un processus de généralisation, pour évaluer s'il est facile pour eux de fournir leur processus à CollaGen. Concernant le rôle d'utilisateur, les tâches sont plus nombreuses et elles ont une influence importante sur le résultat d'une généralisation : les utilisateurs doivent correctement saisir les spécifications formelles, annoter sémantiquement leur schéma de données, choisir les moniteurs chargés d'évaluer leurs spécifications ou encore choisir les règles d'orchestration à appliquer. Nous pensons qu'un spécialiste de la généralisation, comme ceux chargés de mettre en place une chaîne de production de cartes généralisées à l'IGN (Lecordix et al, 2007), serait capable de remplir ce rôle (bien que nos interfaces d'aide aux utilisateurs n'aient pas été testées par manque de temps). Par contre, dans le cadre d'une cartographie à la demande sur Internet par exemple, un utilisateur *lambda* ne serait pas capable de saisir tous ces renseignements et un mécanisme d'aide à la saisie des besoins utilisateurs, inspiré de (Hubert, 2008) par exemple, serait indispensable.

Le temps de calcul n'a pas vraiment été abordé dans cette thèse, que ce soit du point de vue modélisation ou implémentation. Ce travail ne s'inscrit pas du tout dans le cadre de la généralisation en temps réel (Sarjakoski et Sarjakoski, 2007), la généralisation d'une carte topographique étant un processus long. Pour donner un ordre d'idée, le temps de calcul clé pour une chaîne de production correspond à une nuit de calcul (l'opérateur n'est pas ralenti par la durée du processus automatique car il peut le lancer en partant le soir et récupérer les résultats le lendemain matin). Comme pour sa qualité, le temps de calcul dépend directement du temps de calcul des différents processus utilisés. En général, les processus de généralisation ne sont pas conçus pour une exécution en temps réel et ne considèrent pas le temps d'exécution comme un point clé. L'objectif d'une généralisation collaborative étant de généraliser une carte complète, la quantité de données est donc plus importante que pour les cas d'applications classiques de processus de la littérature. Dans l'absolu, le

temps de calcul est donc plus long avec CollaGen mais ceci est compensé par le découpage en espaces géographiques de tailles raisonnables : CollaGen réalise alors beaucoup de généralisation sur une petite partie des données. Par rapport à d'autres solutions basées sur l'utilisation d'un seul processus sur l'ensemble des données, plusieurs phases de CollaGen augmentent le temps d'exécution (mais pour obtenir de meilleurs résultats). Ainsi, la construction indispensable des relations géographiques et des moniteurs de contrainte alourdit le temps d'exécution. Et la recherche d'effets de bord après chaque généralisation réussie allonge d'autant plus le temps d'exécution. Par contre, les phases de prises de décision du chef d'orchestre, du registre, la traduction des paramètres, ou le temps d'observation pendant lequel la généralisation est stoppée, sont réellement négligeables par rapport au temps d'exécution d'un processus de généralisation et ces phases n'alourdissent pas le déroulement de la généralisation d'une carte complète. De même, la gestion centrée-processus de l'enrichissement permet un gain de temps (les enrichissements ne sont calculés que dans les espaces généralisés par le processus qui les nécessite). Bien que des tests supplémentaires soient nécessaires pour valider cette affirmation, nous considérons que la généralisation, par CollaGen, d'une carte au 1 : 50000 reste dans les délais d'une nuit de calculs.

Nous souhaitons, par ailleurs, aborder la question de la cohérence du modèle CollaGen : comment peut-on l'évaluer globalement alors qu'il est constitué de nombreux petits composants ? L'évaluation individuelle de ces composants fait-elle une évaluation du modèle CollaGen et de son prototype ? Nous avons choisi dans les parties précédentes d'évaluer CollaGen de deux manières pour gérer cette question de la cohérence du système : évaluation de chacune des briques par rapport à ce que le système attend d'elles et évaluation globale en termes de résultats cartographiques. Nous partons du principe que si les briques fonctionnent comme attendu, un bon résultat cartographique traduit un bon fonctionnement de CollaGen dans son ensemble. Nous avons vu cependant en C.14.5 que nous proposons une évaluation descriptive après chaque généralisation d'un espace par un processus. Si nous mettons ces évaluations bout à bout, nous obtenons le déroulement tracé de la généralisation collaborative, qui pourrait être exploitée pour aller plus loin dans l'évaluation de la cohérence globale de CollaGen. Cela permettrait peut-être d'identifier des éléments redondants, inutiles ou inefficaces du système.

Pour finir, nous souhaitons discuter de l'influence de notre démarche sur les choix de modélisation faits pour CollaGen : notre approche par analyse fonctionnelle de la généralisation collaborative nous a poussé vers la modélisation par composants puis par SMA, mais d'autres choix étaient-ils envisageables dès le départ ? Si nous réalisons une abstraction de la généralisation collaborative, il s'agit de déterminer la séquence d'application d'un processus sur un espace qui produit le résultat optimal. Nous pouvons décrire notre problème comme une variante de l'exploration informée d'arbres d'états (Taillardier, 2008), problème bien connu du domaine de l'intelligence artificielle. Ce problème peut être résolu à l'aide d'heuristiques (comme nous le faisons de manière indirecte avec les règles d'orchestration, le registre ou le mécanisme d'évaluation), mais aussi par des méthodes d'optimisation comme le recuit simulé (Kirkpatrick et al, 1983), les algorithmes génétiques (Holland, 1975) ou la recherche tabou (Glover, 1989). Cependant, en identifiant comme fonctions de la généralisation collaborative l'observation en ligne de groupes de conflits émergents ou la gestion des effets de bord, nous nous sommes éloignés de cette exploration abstraite d'arbre d'états. Résoudre le problème de la généralisation collaborative de façon complètement différente est donc imaginable mais nécessite de repenser le cadre théorique de la généralisation collaborative et sa dynamique associée avec un point de vue différent de celui que nous avons adopté dans cette thèse.

CONCLUSION ET PERSPECTIVES

RAPPEL DE L'OBJECTIF

Cette thèse s'inscrit dans la continuité des recherches en généralisation cartographique au laboratoire COGIT, symbolisées par les trois modèles complémentaires de généralisation AGENT (Ruas, 1999 ; Barrault et al, 2001), CartACom (Duchêne, 2004) et GAEL (Gaffuri, 2008). Ces trois modèles comme d'autres processus développés ailleurs sont efficaces sur une partie des problèmes posés par la généralisation complète d'une carte mais aucun ne peut réellement tous les traiter.

Cette thèse avait pour objectif de proposer un nouveau modèle permettant de faire collaborer des processus de généralisation complémentaires pour réaliser une généralisation complète optimale. Le modèle CollaGen devait permettre d'enchaîner des applications de processus sur des parties pertinentes de l'espace pour finalement couvrir au mieux tous les problèmes posés par la généralisation complète d'une carte.

CONTRIBUTIONS DE NOTRE THESE

La contribution majeure de cette thèse réside dans la conceptualisation de la généralisation collaborative et la conception du modèle CollaGen associé qui permet l'utilisation optimale de plusieurs processus pour généraliser une carte. Ce modèle a été mis en œuvre pour la généralisation de cartes topographiques au 1 : 50000. CollaGen intègre plusieurs contributions pour le domaine de la généralisation cartographique que nous allons maintenant détailler.

Un modèle formel de spécifications d'une carte à généraliser

Afin d'exprimer l'ensemble des spécifications d'une carte d'une manière standardisée et formelle qui permette une interprétation par des processus automatiques, nous avons proposé un modèle de contraintes et de règles opérationnelles. Ce modèle formel est un premier pas vers un standard de spécification de carte généralisé : une telle spécification signale les caractéristiques que doivent vérifier les différents objets d'une carte pour être affichés de manière lisible.

Un modèle pour l'interopérabilité des processus de généralisation

Nous avons proposé dans CollaGen une solution complète pour permettre d'enchaîner automatiquement et de manière interopérable des processus de généralisation. Cette solution s'appuie comme pivot sur une ontologie de la généralisation, sur les spécifications formelles et sur leur pendant instancié, les moniteurs de contraintes. Des mécanismes de traduction sont alors mis en place pour convertir les éléments hétérogènes comme les paramètres de chaque processus en éléments standardisés par les formats pivots.

Un modèle pour mettre en adéquation un espace géographique avec un processus

Nous avons proposé un modèle pour mettre en adéquation un espace géographique à généraliser avec les capacités, décrites formellement, d'un processus pour pouvoir choisir à tout moment le processus de généralisation le plus adapté à un ensemble d'objets. Ce modèle prend en compte l'échelle mais surtout les moniteurs de contraintes effectivement instanciés à l'intérieur de l'espace géographique, qui sont comparés avec la description des contraintes formelles *a priori* gérées par un processus. Ce modèle de mise en adéquation est intégré à un registre permettant de choisir automatiquement le processus le plus adapté à un espace donné d'après des spécifications.

Un modèle pour choisir l'ordre de déclenchement des généralisations

Nous avons proposé un modèle pour choisir l'ordre de déclenchement des généralisations pour des couples 'processus sur un espace'. Ce modèle peut déterminer à tout moment de la généralisation collaborative le meilleur prochain espace (ou processus) à traiter ; l'autre élément du couple est déterminé par le mécanisme de mise en adéquation (de la conclusion précédente). Ce modèle de choix s'appuie sur une base de règles d'orchestration à partir de laquelle le choix est inféré automatiquement.

Un modèle d'évaluation combinant observation en ligne et validation a posteriori

Nous avons proposé un modèle d'évaluation qui permet d'évaluer le résultat d'une généralisation de manière globale après l'exécution du processus mais qui permet aussi d'évaluer le déroulement d'une généralisation en l'observant en ligne. Cette observation en ligne permet des interventions sur des groupes de conflits persistants qui corrigent localement les situations que le processus n'arrive pas à régler.

Un modèle pour gérer les effets de bord d'une généralisation

Nous avons aussi proposé un modèle pour gérer les effets de bord provoqués par l'application d'un processus sur seulement un extrait des données (l'espace géographique) : les conflits pouvant

apparaître à l'extérieur d'un espace géographique ont été classés et un modèle de contraintes de cohérence permet de les identifier après une généralisation. Un mécanisme pour une correction automatique a été proposé.

Un Système Multi-Agents combinant agents situés et agents logiciels

Le modèle CollaGen que nous avons proposé repose sur une modélisation multi-agents qui couple une approche de résolution de problèmes pour des agents géographiques situés, avec une approche de 'matchmaking' entre besoins et fournisseurs de services. Ce modèle contribue donc au domaine des SMA en proposant une application unissant des types d'agents habituellement traités différemment.

La généralisation des cartes topographiques

Nous avons enfin contribué à l'automatisation de la généralisation de cartes topographiques par la mise en œuvre de notre prototype de CollaGen. Ce prototype produit des résultats inédits en optimisant le recours aux différents processus existants : ils ne sont utilisés que sur les espaces sur lesquels ils donnent de bons résultats. Notre contribution à la généralisation des cartes topographiques porte aussi sur l'acquisition de nouvelles connaissances grâce aux tests que nous avons réalisés. Nous avons notamment plus de connaissances sur l'espace d'application de certains processus comme AGENT, CartACom ou les moindres carrés et sur l'ordre dans lequel ces processus peuvent être utilisés au mieux.

ENSEIGNEMENTS

Au début de cette thèse, nous avons émis l'hypothèse que le principe de la collaboration de processus de généralisation était le meilleur moyen pour aller plus loin dans l'automatisation de la généralisation d'une carte complète. Nous considérons que cette hypothèse, sur laquelle nous nous sommes appuyé pour définir le cadre formel de généralisation collaborative, a été validée par les résultats obtenus grâce au prototype implémenté de ce modèle. De la même manière que la composition de services web a permis de résoudre des problèmes dans d'autres domaines de l'informatique appliquée, la collaboration de processus de généralisation permet de s'approcher de la généralisation automatique d'une carte topographique complète (certains processus et outils, briques du système collaboratif, manquent encore ou doivent être améliorés) et de faire un pas vers la cartographie automatique et de qualité à la carte, ce qui devient un des objectifs à moyen terme des producteurs de données géographiques qu'ils soient institutionnels, comme l'IGN, ou non.

Si nous avons vu dans cette thèse que les contraintes étaient un formalisme maintenant communément admis pour formaliser les spécifications d'une généralisation, notre travail confirme tout l'intérêt qu'elles peuvent représenter pour guider un processus de généralisation. Alors que

(Ruas, 1999) avait déjà montré que les contraintes instanciées étaient un bon outil pour guider et évaluer un processus de généralisation. Notre modèle a apporté la dichotomie contrainte formelle / moniteur qui permet de ne plus lier une contrainte instanciée à un simple processus mais de transformer les contraintes en outil interopérable de guidage et d'évaluation de la généralisation.

Enfin, ce travail de thèse nous a permis de vérifier l'utilité des ontologies dans le cadre de l'orchestration de la généralisation cartographique. L'hypothèse émise dans (Regnauld, 2007) se vérifie avec les capacités de CollaGen à rendre interopérables des processus conçus sans standard, de manière différente et dans des langues différentes. Nous considérons que les premiers résultats de cette thèse doivent inciter la communauté en généralisation cartographique à utiliser encore plus (nous avons vu dans l'état de l'art qu'elle les utilise déjà) les travaux de recherche de la communauté de recherche en ontologie pour aller plus loin vers la généralité et l'interopérabilité.

PERSPECTIVES

Amélioration du prototype

Nous avons mentionné dans le chapitre E que l'implémentation du prototype CollaGen n'était pas terminée donc la première des perspectives concerne l'amélioration de ce prototype en terminant ce travail d'implémentation. Il convient d'abord d'implémenter complètement le composant de gestion des effets de bord dont seule, la partie détection est implémentée sur les agents espaces géographiques. Plusieurs points n'ont pas été intégrés : le mécanisme particulier de requête de correction d'effets de bord du registre, le mécanisme permettant au chef d'orchestre de statuer sur le choix de correction et l'évaluation de la satisfaction des moniteurs de contrainte de cohérence.

De plus, si les différentes capacités d'action et de raisonnement des agents de CollaGen peuvent être déclenchées automatiquement dans le prototype, l'enchaînement de ces actions, qui correspond à la Figure D-16 n'a pas été implémenté : CollaGen ne peut pas encore réaliser une généralisation complète en cliquant sur un bouton. Il nous paraît donc indispensable de pallier rapidement à ce manque, en complétant l'implémentation du cycle de vie de l'ensemble des agents du système, pour donner plus d'utilisabilité à notre prototype et par conséquent au modèle CollaGen.

Nous souhaitons également intégrer le prototype de CollaGen à la nouvelle plateforme logicielle de recherche de l'équipe généralisation du COGIT, CartAGen (Renard et al, 2010). Pour l'instant, nous pouvons faire appel à la nouvelle plateforme depuis la notre, mais l'inverse n'est pas possible car le stockage des enrichissements, des moniteurs et des états intermédiaires est encore dépendant de la souche Clarity™ de notre plateforme. L'intégration de CollaGen dans CartAGen passe donc par l'ajout d'un mécanisme de persistance de certains objets de CartAGen, ce qui ne devrait pas représenter un travail trop important. Cette intégration permettrait une meilleure utilisation des outils communs de l'équipe mais surtout une prise en main de CollaGen par les autres membres de l'équipe ce qui accélérerait son amélioration.

D'autres pistes existent pour améliorer le prototype de CollaGen : nous avons vu plusieurs fois que sa qualité dépendait fortement de celle de ses processus donc nous souhaitons encore améliorer les processus déjà disponibles sur le prototype et surtout en intégrer de nouveaux, notamment des processus issus du modèle GAEL.

Amélioration du modèle

Nous souhaitons également apporter des améliorations au modèle CollaGen indépendamment de son prototype. Nous pensons que ces améliorations passent par une phase intensive de tests qui permettront de valider ou non les nombreux mécanismes que nous avons modélisés dans CollaGen. Ces tests passent par des généralisations complètes pour mettre en lumière d'éventuels points faibles mais aussi par des protocoles plus précis pour valider des mécanismes précis. Par exemple, il nous paraît indispensable de tester plus précisément le fonctionnement du registre, notamment sur deux points : l'influence de l'échelle dans le choix du processus le plus adapté et la pertinence de notre méthode de quantification des conflits (par les moniteurs) pour mettre en adéquation les conflits effectifs avec les post-conditions des descriptions formelles.

Par ailleurs, un des objectifs de notre thèse était de prendre en compte la question de la masse de données à généraliser dans le problème du partitionnement des données en espaces géographiques. Le découpage proposé dans CollaGen n'est actuellement guidé que par l'optimisation de l'efficacité des processus (à l'exception près des espaces ruraux, redécoupés par les réseaux pour réduire leur taille), ce découpage gérant le problème de la masse de données de manière indirecte : quand on découpe une zone en espaces urbain et rural, on diminue la masse de données de chaque généralisation. Mais la généralisation de grands espaces périurbains par CartACom a fait réapparaître ce problème de masse de données dans nos tests : quand l'espace contient trop d'objets, le processus de généralisation peut dysfonctionner (CartACom ne peut pas généraliser en même temps 500 bâtiments et 300 routes par exemple). Nous souhaitons donc intégrer de manière plus claire la problématique de la masse de données dans le mécanisme de découpage en espace géographique de CollaGen. Les travaux présentés dans (Chaudhry et Mackaness, 2008) sont un bon point de départ pour prendre en compte la masse de données dans le découpage.

Généralisation de cartes topographiques

Nos travaux de thèse ouvrent également plusieurs pistes de recherche intéressantes pour améliorer la généralisation cartographique et progresser vers l'objectif de la carte à la carte (carte non produite en série mais correspondant aux besoins plus spécifiques d'un utilisateur). Le sujet qui nous paraît prioritaire concerne l'évaluation de la généralisation. De nombreux précédents travaux ont également identifié l'évaluation globale de la généralisation comme un verrou théorique (Bard, 2004 par exemple), et nos quelques propositions dans ce domaine confirment qu'il s'agit d'un point que nous ne savons pas traiter correctement. Si l'évaluation d'une contrainte de type micro ne pose plus vraiment de problème, l'évaluation de contraintes meso et surtout l'évaluation globale sur un territoire (dans un espace dans notre cas) reste complexe : est-il mieux d'avoir beaucoup de contraintes très satisfaites et quelques unes très insatisfaites ou plutôt moins de contraintes satisfaites mais aucune contrainte très insatisfaite ? Nous nous proposons d'explorer ce que peut

nous apporter une analogie avec le bien-être général d'une population, étudié en économie et déjà adapté dans le domaine des SMA (Nongillard et al, 2010).

De plus, si CollaGen est une solution pour réduire la saisie des paramètres d'un processus de généralisation à leur expression formelle sous forme de contraintes, la prise en main d'un système de généralisation par un utilisateur non spécialisé dans le domaine s'avère toujours très problématique. Si nous souhaitons qu'un utilisateur lambda puisse faire la demande d'une carte à la carte, nous devons travailler sur des méthodes pour faciliter l'expression de ces besoins et pour traduire ces besoins en ressources formelles. Les travaux d'aide à la saisie de besoins par dialogue homme-machine (Hubert, 2008 ; Christophe, 2009 ; Taillandier et Gaffuri, 2010) nous paraissent des bons points de départ pour tendre vers cet objectif.

Nous avons également identifié, durant nos tests de généralisation, que nos processus dédiés à la généralisation des espaces urbains n'étaient pas encore complètement satisfaisants. Nous avons notamment remarqué que la non prise en compte des différents types d'alignements de bâtiments par les processus AGENT pouvait entraîner des généralisations de qualité moyenne dans les zones d'habitation pavillonnaire. Il nous paraît donc intéressant d'étudier l'intégration des travaux sur les alignements comme (Hangouët, 1998 ; Regnauld, 2001 ; Christophe et Ruas, 2002) dans un processus AGENT. Ce travail vient d'ailleurs de démarrer au laboratoire.

Nous considérons aussi qu'il est souhaitable d'aller vers une intégration plus intime de l'ensemble des modèles du laboratoire basés sur des agents géographiques pour mettre en œuvre les pistes explorées dans (Duchêne et Gaffuri, 2008). Ces modèles manipulent différents niveaux d'agents micro, meso et maintenant espaces géographiques qui peuvent contenir ou traverser (e.g. le réseau routier) d'autres niveaux. Dans ce cadre, le laboratoire a entamé une réflexion sur l'intégration d'aspects du modèle de simulation SMA PADAWAN (Picault et al, 2010) dans nos modèles agents. Le modèle PADAWAN formalise les notions d'encapsulation d'un environnement par un agent et de situation d'un agent dans un ou plusieurs environnements (de niveaux différents), avec des comportements différents dans chaque environnement. Nous pensons de plus que ce travail de recherche pourrait être une solution intéressante pour mieux formaliser et gérer les effets de bord dans CollaGen : il peut y avoir des effets de bord quand un objet sort de l'environnement encapsulé par l'agent espace ou quand il adopte des comportements différents car il est encapsulé par plusieurs espaces.

Un dernier sujet nécessite d'être étudié de manière plus approfondie, il s'agit de l'ontologie de généralisation. Nous proposons notamment d'étudier comment intégrer dans l'ontologie des contraintes « classiques » qui serviraient de structure à la saisie des contraintes formelles de CollaGen. Nous proposons également de travailler sur une véritable ontologie de haut niveau des relations spatiales. Si nous sommes capables de modéliser dans l'ontologie qu'une 'route peut longer une forêt' ou qu'un 'chemin traverse une forêt' (Mathet, 2000) en plus des relations topologiques classiques comme l'adjacence, nous serons capables de modéliser automatiquement l'impact de la généralisation d'une des entités en relation sur l'autre (i.e. automatisation de la détection et correction des effets de bord dans le cas de voisinages thématiques dans CollaGen). En effet, la description formelle de telles relations topologiques peut permettre des vérifications automatiques de la même manière que le modèle de 9-intersections de (Egenhofer, 1989) a permis de mieux automatiser la recherche des relations topologiques simples. Par exemple, si on modélise dans une

ontologie la relation spatiale « la rivière traverse la forêt dans sa longueur » (Figure 0-1) en décrivant que cela correspond à traverser le polygone de la forêt de part en part en suivant à peu près le squelette du polygone, on peut détecter automatiquement ces relations dans les données et le maintenir après que la forêt ou la rivière aient été généralisés.

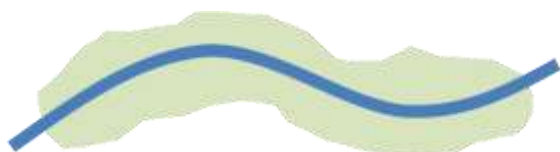


Figure 0-1. Relation spatiale : « la rivière traverse la forêt dans sa longueur ».

Diversification des applications de CollaGen

Nous avons évoqué brièvement, à plusieurs endroits de cette thèse, que CollaGen était un modèle conçu avec une portée générique mais testé sur le cas classique de la généralisation de carte topographique au 1 : 50000, et que pour vérifier sa généricité, il devait être utilisé dans d'autres contextes. Nous souhaitons donc trouver d'autres contextes d'application, que ce soit en termes d'échelles ou de type de carte (carte touristique, carte d'occupation du sol ou géologique), et tester CollaGen dans ces nouveaux contextes avec de nouvelles contraintes, de nouveaux processus et de nouveaux types d'effets de bord à gérer. Le laboratoire COGIT vient notamment de démarrer un travail de thèse sur la prise en compte de données externes dans la généralisation d'une carte : si on ajoute une couche de données sur les risques d'avalanche ou sur le positionnement des radars routiers, quels changements cela implique-t-il dans les processus de généralisation ? Il sera donc intéressant de déterminer quelle place peut avoir le modèle CollaGen dans la résolution de ce problème.

Enfin, nous avons émis comme hypothèse au début de cette thèse que les processus disponibles dans CollaGen seraient uniquement des processus effectivement implémentés sur la plateforme hébergeant CollaGen et non des processus disponibles sous la forme de services Web, pour simplifier notre problème. Nous pensons que la diversification des applications de CollaGen passe par l'ouverture à de tels processus. Il faudra alors étudier les changements que l'utilisation de services Web impliquent et quelles sont les limitations techniques, notamment en termes de transfert de données à travers le réseau, mais surtout quels sont les changements induits dans la modélisation de la généralisation collaborative. Un pas supplémentaire sera ensuite de rendre CollaGen manipulable à travers un réseau, en permettant peut-être à nos agents de se déplacer dans un réseau.

BIBLIOGRAPHIE

- Abadie, N. (2009).** Schema Matching Based on Attribute Values and Background Ontology. In *12th International Conference on Geographic Information Science (AGILE'09)*, Hanovre (Germany).
- Abadie, N. and S. Mustière (2010).** Constitution et exploitation d'une taxonomie géographique à partir des spécifications de bases de données. *Revue Internationale de Géomatique* 20 (2), 145-174.
- Abadie, N., A.-M. Olteanu, and S. Mustière (2007).** Comparaison de la nature d'objets géographiques. In *Actes de la conférence Ingénierie des connaissances, journée OGHS « Ontologies et Gestion de l' Hétérogénéité Sémantique »*.
- Abd-el Kader, Y. and B. Bucher (2006).** Cataloguing GI Functions provided by Non Web Services Software Resources Within IGN. In *9th AGILE Conference*.
- AGENT (1998).** Constraint Analysis. Deliverable A2. Technical Report, <http://agent.ign.fr/deliverable/DA2.html>
- Altay, I. G. (2010).** Améliorer l'Ordonnanceur Principal du Modèle de Généralisation CartACom. Master's thesis, Master IAD, Université Pierre et Marie Curie, Paris.
- Anders, K.-H., M. Sester, and J. Bobrich (2007).** Incremental Update in a MRDB. In *23rd International Cartographic Conference*. ICA.
- ARPA Knowledge Sharing Initiative (1993),** Specification of the KQML agent-communication language. ARPA Knowledge Sharing Initiative, External Interfaces Working Group.
- Baader, F., D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (2003).** *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.
- Bader, M. and M. Barrault (2000).** Improving snakes for linear feature displacement in cartographic generalization. In *Proceedings of the 5th International Conference on GeoComputation*, Greenwich, UK.
- Bader, M. and M. Barrault (2001).** Cartographic Displacement in Generalization: Introducing Elastic Beams. In *4th workshop on progress in automated map generalisation*. ICA.
- Bader, M., Barrault, M., and Weibel, R. (2005).** Building displacement over a ductile truss. *International Journal of Geographical Information Science* 19 (8), 915-936.
- Bădică, A. and C. Bădică (2008).** Conceptualizing Interactions with Matchmakers and Front-Agents Using Formal Verification Methods. pp. 390-394.
- Baeijs, C. (1998).** *Fonctionnalité émergente dans une société d'agents autonomes; étude des aspects organisationnels dans les systèmes multi-agents réactifs*. Ph. D. thesis, institut national polytechnique de Grenoble.
- Baeijs, C., Y. Demazeau, and L. Alvares (1996).** Sigma : Application of multi-agent systems to cartographic generalization. In W. Van de Velde and J. W. Perram (Eds.), *Agents Break Away. Proceedings of the European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Volume 1038 of *LNAI*, pp. 163-176. Springer.
- Baella, B. and M. Pla (2003).** An Example of Database Generalization Workflow: The Topographic Database of Catalonia at 1:25 000. In *5th Workshop on Progress in Automated Map Generalization*, Paris, France. ICA.

- Baella, B., J. Palomar-Vázquez, J. E. Pardo-Pascual, and M. Pla (2007).** Spot heights generalization: deriving the relief of the Topographic Database of Catalonia at 1:25,000 from the master database. In *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation*. Moscow, Russia.
- Balley, S. (2007).** Pourquoi et comment décrire la structure d'un jeu de données géographiques. *Le Monde des cartes* 194, 9-16.
- Bard, S. (2004).** Quality Assessment of Cartographic Generalisation. *Transactions in GIS* 8 (1), 63-81.
- Barde, J. (2005).** *Mutualisation de données et de connaissances pour la Gestion Intégrée des Zones Côtières. Application au projet SYSCOLAG*. Ph. D. thesis, Université Montpellier II.
- Barrault, M., N. Regnaud, C. Duchêne, K. Haire, C. Baeijs, Y. Demazeau, P. Hardy, W. A. Mackaness, A. Ruas, and R. Weibel (2001).** Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalisation. In *20th International Cartographic Conference*, Volume 3, pp. 2110-2116. ICA. Beijing, China.
- Barillot, X. (2002).** Mesures et structures d'analyse. In A. Ruas (Ed.), *Généralisation et représentation multiple*, Information Géographique et Aménagement du Territoire, pp. 187-201. Paris: Hermès, Lavoisier.
- Beard, K. (1991).** Constraints on rule formation. In B. Buttenfield and R. McMaster (Eds.), *Map Generalization*, pp. 121-135. Longman Pages.
- Beguín, M. and D. Pumain (1994).** *La Représentation des données géographiques*. Paris: Armand Colin.
- Bellicha, A., B. Neveu, et al (1994).** Autour Du Problème De Satisfaction De Contraintes. In *Actes des 5èmes journées nationales du PRC GDR Intelligence Artificielle*, pp. 159-178.
- Beni, S. and G. Hackwood (1992).** The Maximum Entropy Principle and Sensing in Swarm Intelligence. In *Toward a Practice of Autonomous Systems*, pp. 153-160. Cambridge: MIT Press.
- Bertin, J. (1967).** *Sémiologie graphique*. Paris: Mouton, Gauthier-Villars.
- Bishr, Y. (1998).** Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science* 12 (4), 299-314.
- Bjerhammar, A. (1973).** *Theory of errors and generalized matrix inverses*. Amsterdam: Elsevier.
- Bobrich, J. (2001).** Cartographic Map Generalization in Urban Districts. In *Proceedings of 9th annual conference on GIS Research in United Kingdom*, pp. 513-515.
- Boffet, A. (2000).** Creating urban information for cartographic generalisation. In *International Symposium on Spatial Data Handling (SDH)*.
- Boffet, A. (2001).** *Méthode de création d'informations multi-niveaux pour la généralisation de l'urbain*. Ph. D. thesis, Université Marne La Vallée.
- Bonabeau, E., J.-L. Dessalles, and A. Grumbach (1995).** Characterizing Emergent Phenomena (1): A Critical Review. *Revue Internationale de Systémique* 9 (3), 327-346.
- Bonabeau, E., J.-L. Dessalles, and A. Grumbach (1995b).** Characterizing Emergent Phenomena (2): A Conceptual Framework. *Revue Internationale de Systémique* 9 (3), 347-371.
- Bonin, O. (1998).** Attribute uncertainty Propagation in Vector Geographic Information Systems: Sensitivity Analysis. In *Proceedings of 10th International Conference on Scientific and Statistical Database Management, Ssdm '98*, pp. 254-259.
- Bonin, O. and B. Poupeau (2005).** Identification of simple landscape features on triangulated contour lines. In *Proceedings of 13th Conference on GIScience and Research in UK (GISRUK'05)*, Glasgow, UK.
- Brassel, K. E. and Weibel, R. (1988).** A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Science* 2 (3), 229-244.

- Braud, P., P. Bourgeat, and C. Bras (1997).** *Dictionnaire de géographie* (2ème ed.). Paris: Hatier.
- Brazile, F. (2000).** *Semantic Infrastructure and Methods to Support Quality Evaluation in Cartographic Generalization*. Ph. D. thesis, Department of Geography, University of Zurich.
- Brewer, C. and B. Buttenfield (2009).** Mastering map scale: balancing workloads using display and geometry change in multi-scale mapping. *GeoInformatica* 14 (2), 221-239.
- Briot, J.-P. and Y. Demazeau (2001).** Introduction. In J.-P. Briot and Y. Demazeau (Eds.), *Principes et architecture des systèmes multi-agents*, Chapter 1, pp. 17-25. Hermès.
- Brodie, M. L. (1992).** The Promise of Distributed Computing and the Challenges of Legacy Information Systems. In P. Gray and R. Lucas (Eds.), *Advanced Database Systems: Proceedings of the 10th British National Conference on Databases*, pp. 1-28. Berlin: Springer-Verlag.
- Brosset, D. and T. Devogele (2004).** Algorithmes génétiques et généralisation de MNT marins. In *Cassini 2004, 7ième conférence du GDR SIGMA*. Grenoble, France.
- Brüggemann, H. (1985).** Die topographisch-kartographische Datenbank in der Standardisierung. *Nachrichten aus dem Karten- und Vermessungswesen, Series I* 95, 43+.
- Brunet, R., R. Ferras, and H. Théry (1992).** *Les mots de la géographie, dictionnaire et critique*. Paris: La Documentation française.
- Bucher, B. (2006).** Décrire formellement les processus de manipulation de données géographiques. *Ingénierie des Systèmes d'Information* 11 (1), 61-82.
- Bucher, B. and S. Balley (2007).** A generic preprocessing service for more usable geographical data processing services. In *10th AGILE International Conference on Geographic Information Science*. AGILE.
- Bucher, B. and L. Jolivet (2008).** Acquiring service oriented descriptions of GI processing software from experts. In *11th AGILE International Conference on Geographic Information Science*. AGILE.
- Bucher, B., S. Balley, D. Richard, G. Cébéliu, and J.-F. Hangouët (2005).** Shareable descriptions of geographical data production processes. In *8th AGILE Conference*.
- Bucher, B., M. Brasebin, E. Buard, E. Grosso, and M. Sébastien (2011).** GeOxygene: Built on top of the expertness of the french NMA to host and share advanced GI science research results. In E. Bocher and O. Ertz (Eds.), *Proceedings of the First International OpenSource Geospatial Research Symposium (OGRS'09)*, Lecture Note in Computer Sciences. Berlin: Springer. In press.
- Bundy, G. L., C. B. Jones, and E. Furse (1995).** Holistic generalization of large-scale cartographic data. In J.-C. Müller, J.-P. Lagrange, and R. Weibel (Eds.), *GIS And Generalisation: Methodology And Practice*, Volume 1 of *GISDATA*, pp. 106-119. London: Taylor & Francis.
- Burghardt, D. (2010).** ICA Generalisation workshops (1995-2010), from past generalisation workshops to future works on the field. In *13th Workshop of the ICA commission on Generalisation and Multiple Representation*, Zurich.
- Burghardt, D. and Cecconi, A. (2007).** Mesh simplification for building typification. *International Journal Geographic Information Science* 21 (3), 283-298.
- Burghardt, D. and S. Meier (1997).** Cartographic Displacement using the Snakes Concept. In W. Foerstner and L. Pluemer (Eds.), *Semantic Modelling for the Acquisition of Topographic Information from Images and Maps*. Basel, Swiss: Birkhauser.
- Burghardt, D. and M. Neun (2006).** Automated Sequencing of Generalisation Services Based on Collaborative Filtering. In M. Raubal, H. J. Miller, A. U. Frank, and M. F. Goodchild (Eds.), *Geographic Information Science - 4th International Conference GIScience*, pp. 41-46. Münster, Germany: IFGI prints.

- Burghardt, D., S. Schmid, and J. Stoter (2007).** Investigations on cartographic constraint formalisation. In *10th ICA Workshop on Generalisation and Multiple Representation*.
- Buttenfield, B. (1991).** A rule for describing line feature geometry. In B. Buttenfield and R. McMaster (Eds.), *Map Generalization : Making Rules for Knowledge Representation*, pp. 150-171. Harlow: Longman Group.
- Buttenfield, B. (1995).** Object-Oriented map generalization. In J.-C. Müller, J.-P. Lagrange, and R. Weibel (Eds.), *GIS and generalization: methodology and practice*, pp. 91-105. Taylor & Francis.
- Buttenfield, B. and D. P. Mark (1991).** Expert systems in cartographic design. In D. R. F. Taylor (Ed.), *Geographic Information Systems: The Microcomputer and Modern Cartography*, pp. 129-150. Oxford: Pergamon Press.
- Buttenfield, B., L. V. Stanislawski, and C. A. Brewer (2010).** Multiscale Representations of Water: Tailoring Generalization Sequences to Specific Physiographic Regimes. In *Proceedings of GIScience 2010*.
- Campbell, M. (2002).** Deep Blue. *Artificial Intelligence* 134 (1-2), 57-83.
- Cauvin, C. (1984).** *Espaces cognitifs et transformations cartographiques. Les conditions de la comparaison des espaces cognitifs : de la carte aux configurations ; exemples de l'espace urbain Strasbourgeois*. Ph. D. thesis, Université Louis Pasteur, Strasbourg.
- Cecconi, A. (2003).** *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping*. Ph. D. thesis, University of Zurich.
- Chaib-Draa, B., I. Jarras, and B. Moulin (2001).** Systèmes multi-agents : principes généraux et applications. In J.-P. Briot and Y. Demazeau (Eds.), *Principes et architecture des systèmes multi-agents*, Chapter 1, pp. 27-70. Paris: Hermès, Lavoisier.
- Chandrasekaran, B., J. R. Josephson, and V. R. Benjamins (1998).** The Ontology of Tasks and Methods. In *Proceedings of KAW'98, 11th Workshop on Knowledge Acquisition*, Canada.
- Charif, Y. and N. Sabouret (2009).** Un protocole de coordination d'agents introspectifs pour la chorégraphie dynamique de services. *Revue d'Intelligence Artificielle* 23 (1), 47-80.
- Chaudhry, O. Z. (2007).** *Modelling Geographic Phenomena at Multiple Levels of Detail: A Model Generalisation Approach Based on Aggregation*. Ph. D. thesis, University of Edinburgh.
- Chaudhry, O. Z. and W. A. Mackaness (2008).** Partitioning Techniques to Make Manageable the Generalisation of National Spatial Datasets. In *ICA Workshop on Generalisation and Multiple Representation*, Montpellier, France.
- Chaudhry, O. Z. and W. A. Mackaness (2008b).** Automatic identification of urban settlement boundaries for multiple representation databases. *Computers, Environment and Urban Systems* 32 (2), 95-109.
- Chaudhry, O. Z. and W. A. Mackaness (2008c).** Creating Mountains out of Mole Hills: Automatic Identification of Hills and Ranges Using Morphometric Analysis. *Transactions in GIS* 12 (5), 567-589.
- Christophe, S. (2009).** Making legends by means of painters' palettes. In *Cartography and Art, Lecture Notes in Geoinformation and Cartography*, Chapter 8, pp. 1-11. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Christophe, S. and A. Ruas (2002).** Detecting building structures for generalisation purposes. In *Proceedings of 10th International Symposium on Spatial Data Handling (SDH'02)*, Ottawa (Canada).
- Cullot, N., C. Parent, S. Spaccapietra, and C. Vangenot (2003).** Des Ontologies pour données géographiques. *Revue Internationale de Géomatique* 13 (3), 285-306.
- Davis, E. (1990).** *Representations of commonsense knowledge*. San Francisco, USA: Morgan Kaufmann Publishers Inc.
- Decker, K., K. P. Sycara, and M. Williamson (1997).** Middle-Agents for the Internet. In *15th International Joint Conference on Artificial Intelligence (IJCAI'97)*.

- Delahaye, C. (2010).** Dire que ma maison est en ville ou à la campagne change-t-il sa généralisation cartographique ? Master's thesis, Université Paris 8, Vincennes-Saint-Denis.
- Demazeau, Y. (1997).** Steps Towards Multi-agent Oriented Programming. In *1st International Workshop on Multi-agent Systems (IWMAS)*, Boston, USA.
- Denègre, J. and Salgé, F. (1996).** *Les systèmes d'information géographique. Que sais-je ?* Presses universitaires de France.
- Dorigo, M., V. Maniezzo, and A. Colorni (1996).** The Ant System: Optimization by a colony of cooperating agents. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B* 26 (1), 29-41.
- Dougenik, J. (1980).** WHIRLPOOL: A geometric processor for polygon coverage data. In *Auto-Carto 4*, pp. 304-311. ACSM-ASPRS.
- Douglas, D. H. and Peucker, T. K. (1973).** Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10 (2), 112-122.
- Drogoul, A. (1993).** *De la simulation Multi-Agents à la Résolution Collective de Problèmes*. Ph. D. thesis, Université Paris 6.
- Drogoul, A. and J. Ferber (1994).** Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies. In *Artificial Social Systems*, pp. 2-23.
- Duchêne, C. (2003).** Automated Map Generalisation Using Communicating Agents. In *Proceedings of the 21st International Cartographic Conference*, pp. 160-169. ICA. Durban, South Africa.
- Duchêne, C. (2004).** *Généralisation Cartographique par Agents Communicants : le modèle CartACom. Application aux données topographiques en zone rurale*. Ph. D. thesis, Université Paris 6.
- Duchêne, C. (2004b).** The CartACom model: a generalisation model for taking relational constraints into account. In *7th ICA Workshop on Generalisation and Multiple Representation*, Leicester, UK. ICA.
- Duchêne, C. and J. Gaffuri (2008).** Combining Three Multi-agent Based Generalisation Models: AGENT, CartACom and GAEL. In A. Ruas and C. Gold (Eds.), *Headway in Spatial Data Handling*, pp. 277-296.
- Duchêne, C. and G. Touya (2010).** Emergence de zones conflits dans deux modèles de généralisation cartographique multi-agents. In M. Occello and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux*, Toulouse, pp. 33-42. Cépaduès.
- Duchêne, C., S. Bard, X. Barillot, A. Ruas, J. Trevisan, and F. Holzapfel (2003).** Quantitative and qualitative description of building orientation. In *fifth workshop on progress in automated map generalisation*, Paris, France.
- Duchêne, C., G. Touya, J. Gaffuri, P. Taillandier, A. Ruas, and J. Renard (2011).** Multi-Agents Systems for Cartographic Generalisation: feedback from past and on-going research works. In B. Buttenfield, C. Brewer, and K. Clarke (Eds.), *Generalization and Data Integration*. A paraître.
- Dukaczewski, D., E. Bielecka, and J. Bac-Bronowicz (2007).** Topographic Databases of 10 000 in EU Countries. Comparison of the Thematic Scope. In *Proceedings of 23rd International Cartographic Conference*. Moscow, Russia.
- Dussaix, A.-M. and J.-M. Grosbras (1996).** *Les sondages : Principes et méthodes*. Que sais-je ? Presses Universitaires de France.
- Dutton, G. and A. Edwardes (2006).** Ontological Modeling of Geographical Relationships for Map Generalization. In *9th ICA Workshop on Generalisation and Multiple Representation*, Portland, USA.

- Egenhofer, M. (1989).** A formal definition of binary topological relationships. In W. Litwin and H.-J. Schek (Eds.), *Foundations of Data Organization and Algorithms*, Volume 367 of *Lecture Notes in Computer Science*, pp. 457-472. Springer Berlin / Heidelberg.
- Egenhofer, M. J. and R. D. Franzosa (1991).** Point-set topological spatial relations. *International Journal of Geographical Information Systems* 5 (2), 161-174.
- El Fallah Seghrouchni, A. (2001).** Les modèles de coordination d'agents cognitifs. In J.-P. Briot and Y. Demazeau (Eds.), *Principes et architecture des systèmes multi-agents*, Chapter 4, pp. 139-176. Paris: Hermès.
- Ferber, J. (1995).** *Les systèmes multi-agents, vers une intelligence collective*. Information Intelligence Artificielle. InterEditions.
- Ferber, J. and O. Gutknecht (1998).** A Meta-Model for the analysis and design of organizations in Multi-Agent systems. In *Proceedings of the 3rd International Conference on Multi Agent Systems, ICMAS '98*, Washington, DC, USA. IEEE Computer Society.
- Figueira, J., V. Mousseau, and B. Roy (2005).** ELECTRE methods. In J. Figueira, S. Greco, and M. Ehrgott (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, pp. 133-162. Springer-Verlag.
- Filippovska, Y., V. Walter, and D. Fritsch (2008).** Quality Evaluation of Generalization Algorithms. In *ISPRS Commission II, WG II/7*. ISPRS.
- FIPA (2000).** FIPA ACL Message Structure Specification. FIPA report XC 00061D. <http://www.fipa.org/specs/>
- FIPA (2004).** FIPA Agent Management Specification. <http://www.fipa.org/specs/fipa00023/>.
- Foerster, T., J. Morales, and J. Stoter (2008).** A Classification of Generalisation Operators Formalised in OCL. In E. Pebezma and T. Bartoschek (Eds.), *GI-Days*, pp. 141-156.
- Foerster, T., Stoter, J., and Köbben, B. (2007).** Towards a Formal Classification of Generalization Operators. In *Proceedings of 23rd International Cartographic Conference*. ICA.
- Forrest, S. (1991).** *Emergent Computation* (First ed.). Cambridge: The MIT Press.
- Franklin, S. and A. Graesser (1997).** Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, London, UK, pp. 21-35. Springer-Verlag.
- Friedberg, R. M. (1958).** A learning machine: Part I. *IBM Journal* 2, 2-13.
- Gaffuri, J. (2007).** Field deformation in an agent-based generalisation model: the GAEL model. In F. Probst, C. Kessler (Eds.), *GI-days 2007 - young researches forum*, IFGI prints 30, pp. 1-24.
- Gaffuri, J. (2008).** *Généralisation automatique pour la prise en compte des thèmes champ : le modèle GAEL*. Ph. D. thesis, Université Marne-la-Vallée.
- Gaffuri, J. (2009).** Three reuse example of a generic deformation model in map generalisation. In *24th International Cartographic Conference*. Santiago, Chile.
- Gaffuri, J. and J. Trévisan (2004).** Role of urban patterns for building generalisation: An application of AGENT. In *ICA Workshop on Generalisation and Multiple representation, commission on map generalisation*. ICA.
- Gaffuri, J., C. Duchêne, and A. Ruas (2008).** Object-field relationships modelling in an agent-based generalisation model. In *Proceedings of 12th workshop on generalisation and multiple representation*, Montpellier, France.
- Galanda, M. (2003).** Modelling constraints for polygon generalisation. In *Proceedings of 5th workshop on progress in automated map generalisation*. ICA. Paris, France.

- Galanda, M. and R. Weibel (2002).** An agent-based framework for polygonal subdivision generalization. In D. Richardson and P. van Oosterom (Eds.), *Advances in spatial data handling, 10th international symposium on spatial data handling*, pp. 121-136. Springer.
- Galton, A. and M. Duckham (2006).** What Is the Region Occupied by a Set of Points? In *GIScience*, Volume 4197 of *Lecture Notes in Computer Science*, pp. 81-98. Springer.
- Gao, W., A. Song, and J. Gong (2004).** Constraint-Based Generalization of Soil Maps. In *Proceedings of ISPRS Commission IV, WG IV/3*. ISPRS.
- Gesbert, N. (2005).** *Etude de la formalisation des spécifications des bases de données géographiques en vue de leur intégration*. Ph. D. thesis, Université Marne La Vallée.
- Girres, J.-F. and G. Touya (2010).** Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS* 14 (4), 435-459.
- Goodchild, M. F. (2007).** Citizens as voluntary sensors: spatial data infrastructure in the world of Web 2.0. *International Journal of Spatial Data Infrastructures Research* 2, 24-32.
- Gruber, T. (1995).** Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* 43 (5-6), 907-928.
- Grünreich, D. (1985).** Ein Vorschlag zum Aufbau einer grossmassstäbigen topographischkartographischen Datenbank unter besonderer Berücksichtigung der Grundrissdateides ALK-Systems. *Nachrichten aus dem Karten- und Vermessungswesen, Series I* 95, 55+.
- Guarino, N. (1998).** Formal Ontology and Information Systems. In N. Guarino (Ed.), *Formal Ontology in Information Systems. Proceedings of FOIS'98*, Amsterdam, pp. 3-15. IOS Press.
- Haas, H. and A. Brown (2004).** Web Services Glossary. Technical report, W3C.
- Handschuh, S. and S. Staab (Eds.) (2003).** Annotation for the Semantic Web (Frontiers in Artificial Intelligence and Applications) (1 ed.). IOS Press.
- Hangouët, J.-F. (1998).** *Approche et méthodes pour l'automatisation de la généralisation cartographique; application en bord de ville*. Ph. D. thesis, Université Marne-la-Vallée.
- Hannoun, M., O. Boissier, J. Sichman, and C. Sayettat (2000).** MOISE: An organizational model for multi-agent systems. In M. Monard and J. Sichman (Eds.), *Advances in Artificial Intelligence*, Volume 1952 of *Lecture Notes in Computer Science*, Chapter 17, pp. 156-165. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Hake, G., Grünreich, D., and Meng, L. (1994).** Kartographie. Visualisierung raum-zeitlicher Informationen.
- Hardy, P., M. Hayles, and P. Revell (2003).** Clarity - a new Environment for Generalisation using AGENTS, JAVA, XML and Topology. In *Proceedings of ICA Generalisation Workshop*, Paris.
- Hardy, P., J.-L. Monnot, and D. Lee (2006).** An Optimization Approach To Constraint-Based Generalization In a Commodity GIS Framework. In *Proceedings of 9th ICA Workshop on Generalisation and Multiple Representation*. Portland, USA.
- Harrie, L. E. (1999).** The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science* 26 (1), 55-69.
- Harrie, L. (2003).** Weight-Setting and Quality Assessment in Simultaneous Graphic Generalization. *The Cartographic Journal* 40 (3), 221-233.
- Harrie, L. and T. Sarjakoski (2002).** Simultaneous Graphic Generalization of Vector Data Sets. *Geoinformatica* 6 (3), 233-261.
- Harrie, L. and R. Weibel (2007).** Modelling the Overall Process of Generalisation. In A. Ruas, W. A. Mackaness, and T. Sarjakoski (Eds.), *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pp. 67-87. Amsterdam: Elsevier.

- Hauert, J.-H. (2007a).** Optimization Methods for Area Aggregation in Land Cover Maps. In *23rd International Cartographic Conference*, Moscow, Russia. ICA.
- Hauert, J.-H. (2007b).** Efficient area aggregation by combination of different techniques. In *10th ICA Workshop on Generalisation and Multiple Representation*. ICA. Moscow, Russia.
- Heinzle, F. and K.-H. Anders (2007).** Characterising Space via Pattern Recognition Techniques: Identifying Patterns in Road Networks. In W. A. Mackaness, A. Ruas, and T. Sarjakoski (Eds.), *The Generalisation of Geographic Information : Models and Applications*, Chapter 12. Elsevier.
- Hoët, S. and N. Sabouret (2010).** Simplification de la mémoire des agents en utilisant des FMDP. In M. Occello and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux, JFSMA'10*, Mahdia, Tunisie, pp. 13-22. Cépaduès.
- Hojholt, P. (1998).** Solving Local and Global Space Conflicts in Map Generalisation: Using a Finite Element Method Adapted from Structural Mechanics. In T. K. Poiker and N. Chrisman (Eds.), *Proceedings of 8th International Symposium on Spatial Data Handling (SDH'98)*, pp. 679-689. Vancouver.
- Hojholt, P. (2000).** Solving Space Conflicts in Map Generalization: Using a Finite Element Method. *Cartography and Geographic Information Science*, 65-74.
- Hopfstock, A. (2007).** A Comparative Study of European Topographic Maps at Scale 1:250.000. In *Proceedings of 23rd International Cartographic Conference*. Moscow, Russia.
- Horrocks, I. (2008).** Ontologies and the semantic web. *Commun. ACM* 51 (12), 58-67.
- Horton, R. E. (1945).** Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin* 56 (3), 275-370.
- Hubert, F. (2008).** CartAble, système d'aide au paramétrage de traitements géographiques complexes. *Revue Internationale de Géomatique* 18 (1), 41-65.
- Hubner, J. F., J. S. Sichman, and O. Boissier (2007).** Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *Int. J. Agent-Oriented Softw. Eng.* 1, 370-395.
- Iosifescu, I. and L. Hurni (2007).** Towards Cartographic Ontologies or "How Computers Learn Cartography". In *Proceedings of 23rd International Cartographic Conference*, Moscow, Russia. ICA.
- Jabeur, N., B. Moulin, and E. Gbei (2003).** Une approche par compétition d'agents pour la résolution de l'encombrement spatial lors de la génération automatique de cartes. In *Journées Francophones sur les Systèmes Multi-Agents JFSMA 2003*, Hammamet, Tunisie, pp. 161-173.
- Jabeur, N., B. Boulekrouche, and B. Moulin (2006).** Using Multiagent Systems to Improve Real-Time Map Generation. In A. Sattar, B. H. Kang (Eds.), *Advances in Artificial Intelligence*, pp. 37-48.
- Jackson, P. (1998).** *Introduction to Expert Systems* (3rd ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Jamal, S. (2005).** *Environnement de procédé extensible pour l'orchestration. Application aux services web*. Ph. D. thesis, l'Université Joseph Fourier Grenoble I.
- Jolivet, L. (2009).** Characterizing maps to improve on-demand cartography - the example of European topographic maps. In *17th Conference on GIScience and Research in UK (GISRUK'09)*.
- Jones, C. (1997).** *Geographical Information Systems and Computer Cartography* (1st ed.). Prentice Hall.
- Joubran Abu Daoud, J. and Y. Doytsher (2004).** A combined automated generalization model of spatial active objects. In *Proceedings of ISPRS Commission IV, WG IV/3*, Istanbul, Turkey. ISPRS.
- Joubran Abu Daoud, J. and Y. Doytsher (2005).** A Combined Automated Generalization Model Based on the Relative Forces Between Spatial Objects. In *Proceedings of 22nd International Cartographic Conference*. ICA. La Coruña, Spain.

- Joubran Abu Daoud, J. and Y. Doytsher (2008).** An Automated Cartographic Generalization Process: A Pseudo-Physical Model. In *Proceedings of ISPRS Commission II, WG II/3*. ISPRS. Beijing, China.
- Kirkpatrick, S., C. D. Gelatt, Jr, and M. P. Vecchi (1983).** Optimization by Simulated Annealing. *Science* 220, 671-680.
- Klien, E. and M. Lutz (2005).** The Role of Spatial Relations in Automating the Semantic Annotation of Geodata. In A. Cohn and D. Mark (Eds.), *Spatial Information Theory*, Volume 3693 of *Lecture Notes in Computer Science*, Chapter 9, pp. 133-148. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Kubera, Y., P. Mathieu, and S. Picault (2011).** IODA: an interaction-oriented approach for multi-agent based simulations. *Autonomous Agents and Multi-Agent Systems*, 1-41.
- Kulik, L., M. Duckham, and M. J. Egenhofer (2005).** Ontology-driven map generalization. *Journal of Visual Languages and Computing* 16 (2), 245-267.
- Lacomme, L., Y. Demazeau, and V. Camps (2009).** Classification des mécanismes organisationnels dans les réseaux d'agents. In *17ème Journées Francophones des Systèmes Multi-Agents*, pp. 79-88. Cepaduc.
- Lamy, S., A. Ruas, Y. Demazeau, M. Jackson, and W. A. Mackaness (1999).** The Application of Agents in Automated Map Generalisation. In *19th International Cartographic Conference*.
- Lecordix, F., J.-M. Le Gallic, and L. Gondol (2007).** Development of a new generalisation flow line for topographic maps. In *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation*. Moscow, Russia.
- Lecordix, F., Plazanet, C., and Lagrange, J.-P. (1997).** A Platform for Research in Generalization: Application to Caricature. *Geoinformatica* 1 (2), 161-182.
- Lee, D. (2003).** Recent generalisation development and road ahead. In *Proceedings of 5th workshop on progress in automated map generalisation*. ICA. Paris, France.
- Legrand, C., C. Duchêne, and F. Lecordix (2005).** Propagation of the displacements and deformations during a generalisation process. In *International Cartographic Conference*. ICA.
- Lemarié, C. (2003).** Generalisation process for top100: research in generalisation brought to fruition. In *5th ICA Workshop on progress in automated map generalisation*. ICA. Paris, France.
- Le Men, H. (1996).** Généralisation cartographique pour l'occupation du sol : application au passage v1 à v2 pour la BD carto. *Revue Internationale de Géomatique* 6 (2-3), 227-248.
- Lemmens, R. (2006).** *Semantic interoperability of distributed geo-services*. Ph. D. thesis, International Institute for Geo-Information Science and Earth Observation (ITC), Enschede, The Netherlands.
- Lemmens, R. (2008).** Lost and found, the importance of modelling map content semantically. In M. Peterson (Ed.), *International Perspectives on Maps and the Internet*, Lecture Notes in Geoinformation and Cartography, Chapter 24, pp. 377-396. Springer.
- Lodwick, W. A., W. Monson, and L. Svoboda (1990).** Attribute error and sensitivity analysis of map operations in geographical information systems: Suitability analysis. *International Journal of Geographical Information Science* 4 (4), 413-428.
- Lüscher, P., D. Burghardt, and R. Weibel (2007).** Ontology-driven Enrichment of Spatial Databases. In *Proceedings of 10th ICA Workshop on Generalisation and Multiple Representation*, Moscow, Russia.
- Lüscher, P., R. Weibel, and D. Burghardt (2008a).** Alternative options of using processing knowledge to populate ontologies for the recognition of urban concepts. In *11th ICA Workshop on Generalisation and Multiple Representation*. ICA. Montpellier, France.

- Lüscher, P., R. Weibel, and W. A. Mackaness (2008b).** Where is the Terraced House? On the Use of Ontologies for Recognition of Urban Concepts in Cartographic Databases. In Ruas, A., Gold, C. (Eds.), *Headway in Spatial Data Handling*, pp. 449-466.
- Lutz, M. (2007).** Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. *GeoInformatica* 11 (1), 1-36.
- Lutz, M., R. Lucchi, A. Friis-Christensen, and N. Ostländer (2007).** A Rule-Based Description Framework for the Composition of Geographic Information Services. In F. Fonseca, M. Rodríguez, and S. Levashkin (Eds.), *GeoSpatial Semantics*, Volume 4853 of *Lecture Notes in Computer Science*, Chapter 8, pp. 114-127. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Lutz, M., C. Riedemann, and F. Probst (2003).** A Classification Framework for Approaches to Achieving Semantic Interoperability between GI Web Services. In W. Kuhn, M. F. Worboys, and S. Timpf (Eds.), *Spatial Information Theory*, Volume 2825 of *Lecture Notes in Computer Science*, Chapter 13, pp. 186-203. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Mackaness, W. A. (2007).** Understanding Geographic Space. In Mackaness, W., Ruas, A., Sarjakoski, T. (Eds.), *The Generalisation of Geographic Information : Models and Applications*, Chapter 1. Elsevier.
- Mackaness, W. A. and Edwards, G. (2002).** The Importance of Modelling Pattern and Structure in Automated Map Generalisation. In *Proceedings of the Joint ISPRS/ICA Workshop on Multi-Scale Representations of Spatial Data*, pp. 7-8.
- Mackaness, W. A. and P. F. Fisher (1987).** Automatic Recognition and Resolution of Spatial Conflicts in Cartographic Symbolisation. In *Proceedings of Auto-Carto 8*, pp. 709-718.
- Mackaness, W. A., P. F. Fisher, and G. G. Wilkinson (1986).** Towards a cartographic expert system. In *Proceedings of Auto-Carto 7*, London, UK, pp. 578-587.
- Mackaness, W. A., S. Perikleous, and O. Z. Chaudhry (2008).** Representing Forested Regions at Small Scales: Automatic Derivation from Very Large Scale Data. *Cartographic Journal, The* 45 (1), 6-17.
- Mackaness, W. and A. Ruas (2007).** Evaluation in the Map Generalisation Process. In W. A. Mackaness, A. Ruas, and L. T. Sarjakoski (Eds.), *Generalisation of Geographic Information*, pp. 89-111. London: Elsevier.
- Mao, B., H. Fan, L. E. Harrie, Y. Ban, and L. Meng (2010).** City Model Generalization Similarity Measurement using Nested Structure of Earth Mover's Distance. In *Proceedings of 13th Workshop of the ICA commission on Generalisation and Multiple Representation*, Zurich.
- Marchais, E. (2010).** Adaptation du nouveau processus de production des cartes topographiques au 1:25 000 pour une cartographie à façon au 1:50 000. Master's thesis, Université de Provence, Aix-Marseille.
- Mathet, Y. (2000).** *Etude de l'expression en langue de l'Espace et du Déplacement: analyse linguistique, modélisation cognitive et leur expérimentation informatique*. Ph. D. thesis, Université de Caen.
- McMaster, R. B. and Shea, K. S. (1988).** Cartographic Generalization in Digital Environment: A Framework for implementation in a GIS. In *GIS/LIS'88*, pp. 240-249.
- McMaster, R. and Shea, K. S. (1992).** *Generalization in Digital Cartography*. Association of American Cartographers.
- Meurisse, T. (2001).** Le Contrôle de l'Ordonnement dans les Simulations Multi-Agents, Etude et Propositions. In *Actes de l'Atelier Méthodologie et Environnement pour les Systèmes Multi-Agents*, Grenoble, France. Plateforme AFIA.
- Meyer, U. (1986).** Software developments for computer-assisted generalization. In *Proceedings of Auto-Carto London*, pp. 247+.

- Monier, P. (1997).** *Caractérisation du terrain en vue de son traitement numérique. Application à la généralisation de l'orographie*. Ph. D. thesis, Université Louis Pasteur, Strasbourg.
- Monnot, J.-L., D. Lee, and P. Hardy (2007).** Topological constraints, actions, and reflexes for generalization by optimization. In *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation*. Moscow, Russia.
- Morvan, G., A. Veremme, D. Dupont, and D. Jolly (2009).** Démonstration : stratégies d'observation de simulations orientées agent. In *17ème Journées Francophones des Systèmes Multi-Agents*, pp. 233-236. Cépaduès.
- Moulin, H. (2004).** *Fair Division and Collective Welfare*. The MIT Press.
- Müller, J.-C. (1990).** Rule based generalization: Potentials and impediments. In *Proceedings of 4th International Symposium on Spatial Data Handling*, Zurich, Switzerland, pp. 317-334.
- Müller, J.-P. (2004).** Emergence of Collective Behavior and Problem Solving. In A. Omicini, P. Petta, and J. Pitt (Eds.), *ESAW 2003*, LNAI 3071, pp. 1-20. Springer-Verlag.
- Müller, J.-C. and Wang, Z. (1992).** Area-patch generalisation: a competitive approach. *The Cartographic Journal* 29 (2), 137-144.
- Müller, J.-C., R. Weibel, J.-P. Lagrange, and S. Salgé (1995).** Generalization: state of the art and issues. In J.-C. Müller, J.-P. Lagrange, and R. Weibel (Eds.), *GIS and Generalization*, pp. 3-17. London: Taylor & Francis.
- Mustière, S. (1998).** GALBE: Adaptive Generalisation - The need for an Adaptive Process for Automated Generalisation, an Example on Roads. In *1st GIS'PlaNNet conference*, Lisbon, Portugal.
- Mustière, S. (2001).** *Apprentissage supervisé pour la généralisation cartographique*. Ph. D. thesis, Université Pierre et Marie Curie.
- Mustière, S. and B. Moulin (2002).** What Is Spatial Context in Cartographic Generalisation? In *Joint International Symposium and Exhibition on Geospatial Theory, Processing and Applications*, Volume 34, pp. 274-278. ISPRS & SIS.
- Nakos, B., J. Gaffuri, and S. Mustière (2008).** A Transition from Simplification to Generalisation of Natural Occuring Lines. In *Proceedings of 11th ICA Workshop on Generalisation and Multiple Representation*, Montpellier, France.
- Neun, M., D. Burghardt, and R. Weibel (2008).** Web service approaches for providing enriched data structures to generalisation operators. *International Journal of Geographical Information Science* 22 (2), 133-165.
- Neun, M., D. Burghardt, and R. Weibel (2009).** Automated processing for map generalization using web services. *Geoinformatica* 13 (4), 425-452.
- Nickerson, B. G. (1988).** Automated Cartographic Generalization for Linear Features. *Cartographica* 25 (3), 15-66.
- Nickerson, B. G. and H. Freeman (1986).** Development of a Rule-Based System for Automatic Map Generalization. In *Proceedings of the Second International Symposium on Spatial Data Handling*, pp. 537-556.
- Niles, I. and A. Pease (2001).** Towards a standard upper ontology. In N. Guarino, B. Smith, and C. Welty (Eds.), *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, New York, NY, USA, pp. 2-9. ACM.
- Nongaillard, A., P. Mathieu, and P. Everaere (2010).** Bien-être de Nash : enjeux et difficultés pour une société d'agents. In M. Ocelllo and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux*, pp. 129-138. Cépaduès.
- Nyerges, T. (1991).** Representing geographical meaning. In B. Buttenfield and R. McMaster (Eds.), *Map Generalization : Making Rules for Knowledge Representation*, pp. 59-85. Harlow: Longman Group.
- OMG (2003).** UML 2.0 OCL Specification. Technical report.

- Partridge, C. (2002).** The Role of Ontology in Integrating Semantically Heterogeneous Databases. Technical report, LADSEB-CNR, Padova, Italie.
- Peltz, C. (2003).** Web Services Orchestration and Choreography. *Computer* 36 (10), 46-52.
- Perret, J., A. Boffet-Mas, and A. Ruas (2009).** Understanding Urban Dynamics : the use of vector topographic databases and the creation of spatio-temporal databases. In *Proceedings of 24th International Cartography Conference*, Santiago, Chile.
- Petri, C. A. (1962).** *Kommunikation mit Automaten*. Ph. D. thesis, Fakultät für Mathematik und Physik, Technische Hochschule Darmstadt, Darmstadt, Germany.
- Petzold, I., D. Burghardt, and M. Bobzien (2006).** Workflow Management and Generalisation Services. In *Workshop of the ICA Commission on Map Generalisation and Multiple Representation*. ICA. Portland, USA.
- Peyton Jones, S. (2003).** *Haskell 98 Language and Libraries: The Revised Report*. Cambridge University Press.
- Picault, S., P. Mathieu, and Y. Kubera (2010).** PADAWAN, un modèle multi-échelles pour la simulation orientée interactions. In M. Ocelllo and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux, JFSMA'10*, Mahdia, Tunisie, pp. 193-202. Cépaduès.
- Pinet, F., M. Duboisset, and M. Schneider (2009).** Modélisation de contraintes d'intégrité spatiales avec OCL. *Revue Internationale de Géomatique* 19 (1).
- Piunti, M., A. Ricci, O. Boissier, and J. F. Hübner (2009).** Manipulation implicite d'une organisation multiagent via l'environnement. In *17ème Journées Francophones des Systèmes Multi-Agents*, pp. 69-78.
- Plazanet, C. (1996).** *Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes)*. Ph. D. thesis, Université Marne-la-Vallée.
- Popov, B., A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov (2003).** KIM – Semantic Annotation Platform. In D. Fensel, K. Sycara, and J. Mylopoulos (Eds.), *The SemanticWeb - ISWC 2003*, Volume 2870 of *Lecture Notes in Computer Science*, Chapter 53, pp. 834-849. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Pumain, D. and T. Saint-Julien (1997).** *L'analyse spatiale. Localisations dans l'espace*. Paris: Armand Colin.
- Punt, E. and D. Watkins (2010).** User-Directed Generalization of Roads and Buildings for Multi-Scale Cartography. In *Proceedings of 13th ICA Workshop on Generalisation and Multiple Representation*, Zurich.
- Ramezani, S. and U. Endriss (2010).** Nash Social Welfare in Multiagent Resource Allocation. In *Agent-Mediated Electronic Commerce: Designing Trading Strategies and Mechanisms for Electronic Markets*, Volume 59 of *Lecture Notes in Business Information Processing*, pp. 117-131. Springer-Verlag.
- Regnauld, N. (1998).** *Généralisation du bâti: Structure spatiale de type graphe et représentation cartographique*. Ph. D. thesis, Université de Provence.
- Regnauld, N. (2001).** Contextual building typification in automated map generalization. *Algorithmica* 30 (2), 312-333.
- Regnauld, N. (2007).** Evolving from automating existing map production systems to producing maps on demand automatically. In *10th ICA Workshop on Generalisation and Multiple Representation*. ICA, Moscow, Russia.
- Regnauld, N. and W. Mackaness (2006).** Creating a hydrographic network from its cartographic representation: a case study using Ordnance Survey MasterMap data. *International Journal of Geographical Information Science* 20 (6), 611-631.
- Regnauld, N. and R. C. McMaster (2007).** A synoptic View of Generalisation Operators. In W. A. Mackaness, A. Ruas, and L. T. Sarjakoski (Eds.), *Generalisation of Geographic Information*, pp. 37-66. Elsevier.

- Renard, J., J. Gaffuri, and C. Duchêne (2010).** Capitalisation problem in research - example of a new platform for generalisation : CartAGen. In *11th ICA Workshop on Generalisation and Multiple Representation*, Zurich. ICA.
- Revell, P. (2007a).** Generic Tools For Generalising Ordnance Survey Base Scale Landcover Data. In *Proceedings of 10th ICA Workshop on Generalisation and Multiple Representation*. ICA. Moscow, Russia.
- Revell, P. (2007b).** Automated Generalisation and Representation of Ordnance Survey Polygonal Landcover Data at 1:10 000 Scale. In *Proceedings of ICC2007, the International Cartography Association Conference*. Moscow, Russia.
- Revell, P., N. Regnaud, and S. Thom (2007).** Generalising and Symbolising Ordnance Survey Base Scale Data to Create a Prototype 1:50 000 Scale Vector Map. *The Cartographic Journal* 44 (3), 251-257.
- Ricci, A., M. Viroli, and M. Piunti (2010).** Formalising the Environment in MAS Programming: A Formal Model for Artifact-Based Environments. In L. Braubach, J.-P. Briot, and J. Thangarajah (Eds.), *Programming Multi-Agent Systems*, Volume 5919 of *Lecture Notes in Computer Science*, Chapter 9, pp. 133-150. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Rich, E. and K. Knight (1990).** *Artificial Intelligence* (2nd ed.). McGraw-Hill Higher Education.
- Richards, N. and M. J. Ware (2010).** Ant Colony Optimization Applied to Map Generalization. In *13th Workshop of the ICA commission on Generalisation and Multiple Representation*. ICA.
- Roth, R. E., M. Stryker, and C. A. Brewer (2008).** A typology of multi-scale mapping operators. In *Proceedings of GIScience 2008*, Park City, Utah, USA.
- Ruas, A. (1998).** A method for building displacement in automated map generalisation. *International Journal of Geographical Information Science* 12 (8), 789-803.
- Ruas, A. (1999).** *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. Ph. D. thesis, Université de Marne-la-Vallée.
- Ruas, A. (2000).** The Roles of Meso Objects for Generalisation. In *Spatial Data Handling*, pp. 3b. 50-63.
- Ruas, A. (2001).** Automating the generalisation of geographical data: the age of maturity? In *20th International Cartographic Conference*, pp. 1943-1953. ICA.
- Ruas, A. (2001b).** Automatic generalisation project: Learning process from interactive generalisation. Technical report, OEEPE Official Publication n°39.
- Ruas, A. (2004).** *Le Changement de niveau de détail dans la représentation de l'information géographique*, mémoire d'Habilitation à Diriger les Recherches, Université de Marne la Vallée.
- Ruas, A. and C. Duchêne (2007).** A Prototype Generalisation System Based on the Multi-Agent System Paradigm. In W. A. Mackaness, A. Ruas, and L. T. Sarjakoski (Eds.), *Generalisation of Geographic Information*, pp. 269-284. Elsevier.
- Ruas, A. and J.-P. Lagrange (1995).** Data and knowledge modelling for generalization. In J. C. Müller, J. P. Lagrange, and R. Weibel (Eds.), *GIS and generalization: methodology and practice*, pp. 73-90. London: Taylor & Francis.
- Ruas, A. and Libourel, T. (2002).** Introduction. In Ruas, A. (Ed.), *Généralisation et représentation multiple*, Chapter 1, pp. 17-22. Paris: Hermès Lavoisier.
- Ruas, A. and W. A. Mackaness (1997).** Strategies for Urban Map Generalization. In *Proceedings of 18th International Cartographic Conference*, pp. 1387-1394.
- Ruas, A. and Plazanet, C. (1996).** Strategies for automated generalization. In *7th international symposium on spatial data handling*, Delft, Pays-Bas, pp. 319-336.
- Russel, S. and P. Norvig (2006).** *Intelligence Artificielle* (Second ed.). Pearson Education France.

- Sabouret, N. (2003).** Active Semantic Web Services: A programming model for agents in the semantic web. In *Proceedings of European Multi-Agent Systems Conference (EUMAS)*.
- Sabouret, N. (2010).** *Interactions sur le fonctionnement dans les SMA ouverts et hétérogènes*. Editions universitaires européennes.
- Saltelli, A., K. Chan, and E. M. Scott (Eds.) (2000).** *Sensitivity Analysis: Gauging the Worth of Scientific Models*. New York: Wiley.
- Sanders, L., D. Pumain, H. Mathian, F. Guérin-Pace, and S. Bura (1997).** SIMPOP: a multiagent system for the study of urbanism. *Environment and Planning B: Planning and Design* 24 (2), 287-305.
- Sarjakoski, T. and L. T. Sarjakoski (2007).** A Real-Time generalisation and map adaptation approach for Location-Based services. In W. A. Mackaness, A. Ruas, and L. T. Sarjakoski (Eds.), *Generalisation of Geographic Information*, Chapter 7, pp. 137-159. Oxford, UK: Elsevier.
- Schade, S., A. Sahlmann, M. Lutz, F. Probst, and W. Kuhn (2004).** Comparing Approaches for Semantic Service Description and Matchmaking. In R. Meersman and Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, Volume 3291 of *Lecture Notes in Computer Science*, pp. 1062-1079. Springer Berlin / Heidelberg.
- Schneider, L. (2003).** How to Build a Foundational Ontology. In A. Günther, R. Kruse, and B. Neumann (Eds.), *KI 2003: Advances in Artificial Intelligence*, Volume 2821 of *Lecture Notes in Computer Science*, Chapter 10, pp. 120-134. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Schylberg, L. (1992).** Cartographic amalgamation of area objects. In *Proceedings of ISPRS'92*, pp. 135-138. ISPRS.
- Searle, J. R. (1969).** *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.
- Sellami, Z., V. Camps, M.-P. Gleizes, and S. Rougemaille (2010).** Système Multi-Agent pour la construction et l'évolution d'ontologies. In M. Ocelllo and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux*, Toulouse, pp. 151-160. Cépaduès.
- Serra, J. (1983).** *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc.
- Servat, D., E. Perrier, J.-P. Treuil, and A. Drogoul (1998).** When Agents Emerge from Agents: Introducing Multi-scale Viewpoints in Multi-agent Simulations. In *Multi-Agent Systems and Agent-Based Simulation*, pp. 183-198.
- Sester, M. (2000).** Generalization Based on Least Squares Adjustment. *International Archives of Photogrammetry and Remote Sensing XXXIII (B4)*.
- Sester, M. (2005).** Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science* 19 (8), 871-897.
- Shafer, G. (1976).** *Mathematical Theory of Evidence*. Princeton University Press.
- Shea, K. S. and McMaster, R. B. (1989).** Cartographic Generalization in a Digital Environment: When and How To Generalize. In *Proceedings of Auto-Carto 9*, pp. 56-67.
- Sinha, G. and D. Mark (2010).** Toward A Foundational Ontology of the Landscape. In *Proceedings of GIScience 2010*, Zurich.
- Skopeliti, A. and L. Tsoulos (1999).** On the Parametric Description of the Shape of the Cartographic Line. *Cartographica: The International Journal for Geographic Information and Geovisualization* 36 (3), 53-65.
- Skopeliti, A. and L. Tsoulos (2001).** A knowledge-based approach for the cartographic generalization of linear features. In *Proceedings of the 20th International Cartographic Conference*, Beijing, China. ICA.
- Smith, B. (1998).** The Basic Tools of Formal Ontology. In N. Guarino (Ed.), *Formal Ontology in Information Systems. Proceedings of FOIS'98*, Frontiers in Artificial Intelligence and Applications, pp. 19-28. Amsterdam: IOS Press.

- Smith, B. and P. Grenon (2004).** The Cornucopia of Formal-Ontological Relations. *Dialectica* 58 (3), 279-296.
- Smith, B. and D. M. Mark (2001).** Geographical categories: an ontological investigation. *International Journal of Geographical Information Science* 15 (7), 591-612.
- Smith, B. and A. C. Varzi (2000).** Fiat and Bona Fide Boundaries. *Philosophy and Phenomenological Research* 60 (2), 401-420.
- Sowa, J. F. (1995).** Top-level ontological categories. *Int. J. Hum.-Comput. Stud.* 43, 669-685.
- Steiniger, S. (2007).** *Enabling Pattern-Aware Automated Map Generalization*. Ph. D. thesis, University of Zurich.
- Steiniger, S., D. Burghardt, and R. Weibel (2006).** Recognition of island structures for map generalization. In *GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, New York, NY, USA, pp. 67-74. ACM.
- Steiniger, S., T. Lange, D. Burghardt, and R. Weibel (2008).** An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques. *Transactions in GIS* 12 (1), 31-59.
- Steiniger, S. and R. Weibel (2007).** Relations among Map Objects in Cartographic Generalization. *Cartography and Geographic Information Science* 34 (3), 175-197.
- Stoter, J., D. Burghardt, C. Duchêne, B. Baella, N. Bakker, C. Blok, M. Pla, N. Regnaud, G. Touya, and S. Schmid (2009).** Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems* 33 (5), 311-324.
- Stoter, J. E., B. Baella, C. Blok, D. Burghardt, C. Duchêne, M. Pla, N. Regnaud, and G. Touya (2010).** State-of-the-art of automated generalisation in commercial software. Technical report, EuroSDR series on research projects, Amsterdam.
- Stoter, J. E., J. M. Morales, R. L. G. Lemmens, B. M. Meijers, P. J. M. Oosterom, C. W. Quak, H. T. Uitermark, and L. Brink (2008).** A Data Model for Multi-scale Topographical Data. In A. Ruas and C. Gold (Eds.), *Headway in Spatial Data Handling*, Lecture Notes in Geoinformation and Cartography, Chapter 14, pp. 233-254. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Stuckenschmidt, H. and F. van Harmelen (2005).** *Information Sharing on the Semantic Web*. Berlin: Springer-Verlag.
- Sutton, R. S. and A. G. Barto (1998).** *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. Cambridge: The MIT Press.
- Sycara, K., S. Widoff, M. Klusch, and J. Lu (2002).** Larks: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems* 5 (2), 173-203.
- Taillandier, P. (2008).** *Modèle de révision de connaissances dans un processus de généralisation de données géographiques à base d'agents*. Ph. D. thesis, Université Paris-Est.
- Taillandier, P. (2008b).** Diagnosis in systems based on an informed tree search strategy: application to cartographic generalisation. In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, CSTST '08, pp. 589-594. ACM.
- Taillandier, P. and J. Gaffuri (2010).** Designing generalisation evaluation function through human-machine dialogue. In *Proceedings of GIScience 2010*, Zurich.
- Taillandier, P., C. Duchene, and A. Drogoul (2009).** Using Belief Theory to Diagnose Control Knowledge Quality: Application to Cartographic Generalisation. In T. Cao, R.-D. Kutsche, and A. Demaille (Eds.), *Computing and Communication Technologies, RIVF '09*, pp. 1-8. IEEE.
- Thomson, R. C. (2006).** The 'stroke' concept in geographic network; generalization and analysis. In Riedl, A., Kainz, W., and Elmes, G. A. (Eds.), *Progress in Spatial Data Handling 12th International Symposium on Spatial Data Handling*, pp. 681-697.

- Thomson, R. C. and Brooks, R. (2007).** Generalisation of Geographic Networks. In Mackaness, M., Ruas, A., and Sarjakoski, L. (Eds.), *Generalisation of Geographic Information: Cartographic Modelling and Applications.*, Chapter 13. Elsevier.
- Thomson, R. C. and Richardson, D. (1999).** The "Good Continuation" principle of Perceptual Organization applied to the Generalization of Road Networks. In *19th International Cartographic Conference*. ICA.
- Tobler, W. R. (1966).** Numeric map generalization. *Cartographica* 26 (1), 9-25.
- Touya, G. (2005).** Etude de la dérivation de la BD CARTO® à partir de la BD TOPO®Pays. Bilan du lot 3 "Modélisation pour les tests de dérivation". Technical report, Document interne COGIT.
- Touya, G. (2007).** Un processus de sélection du réseau hydrographique, basé sur la détection des structures. *Le Monde des Cartes* 194, 46-56.
- Touya, G. (2008).** Enrichissement de données par analyse spatiale pour la généralisation des réseaux. In *Colloque International de Géomatique et d'Analyse Spatiale SAGEO'08*.
- Touya, G. (2010).** A Road Network Selection Process Based on Data Enrichment and Structure Detection. *Transactions in GIS* 14 (5), 595-614.
- Touya, G., C. Duchêne, and S. Mustière (2010).** Généralisation et intégration pour un fond vert commun entre l'IFN et l'IGN. *Revue Internationale de Géomatique* 20 (1), 65-86.
- UDDI (2004).** UDDI Version 3.0.2. Technical standard, OASIS.
- Uitermark, H., P. van Oosterom, N. Mars, and M. Molenaar (1999).** Ontology-Based Geographic Data Set Integration. In M. Böhlen, C. Jensen, and M. Scholl (Eds.), *Spatio-Temporal Database Management*, Volume 1678 of *Lecture Notes in Computer Science*, Chapter 4, pp. 60-78. Berlin, Heidelberg: Springer Berlin / Heidelberg.
- Uni-Zh (1999),** University of Zürich, Department of Geography. *Selection of basic measures*. Rapport DC1 du project européen AGENT, agent.ign.fr/deliverable/DC1.pdf
- Uschold, M. and M. Grüninger (1996).** Ontologies: principles, methods, and applications. *Knowledge Engineering Review* 11 (2), 93-155.
- Van der Aalst, W. and A. ter Hofstede (2005).** YAWL: yet another workflow language. *Information Systems* 30 (4), 245-275.
- van der Vet, P. E. and N. J. I. Mars (1998).** Bottom-up construction of ontologies. *IEEE Transactions on Knowledge and Data Engineering* 10 (4), 513-526.
- van Oosterom, P. (1995).** The GAP-tree, an approach to 'on the fly' map generalization of an area partitioning. In J.-C. Müller, J.-P. Lagrange, and R. Weibel (Eds.), *GIS and Generalization: Methodology and Practise*, pp. 120-132. London: Taylor & Francis.
- van Smaalen, J. (2003).** *Automatic Aggregation of Geographic Objects: A New Approach to the Conceptual Generalisation of Geographic Databases*. Ph. D. thesis, Wageningen University, the Netherlands.
- Vauglin, F. (1997).** *Modèles statistiques des imprécisions géométriques des objets géographiques linéaires*. Ph. D. thesis, Université Marne la Vallée.
- Visvalingham, M. and Wyatt, J. D. (1993).** Line Generalization by Repeated Elimination of Points. *The Cartographic Journal* 30 (1), 46-51.
- Walter, V. (2008).** Automatic Interpretation of Vector Databases with a Raster-Based Algorithm. In *ISPRS Commission II, WG II/4*.
- Ware, J. M. and C. B. Jones (1998).** Conflict Reduction in Map Generalization Using Iterative Improvement. *Geoinformatica* 2 (4), 383-407.

- Ware, M. J., C. B. Jones, and N. Thomas (2003).** Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science* 17 (8), 743-769.
- Ware, M. J. and Richards, N. (2010).** Ant Colony Optimization Applied to Network Schematization. In *Sixth international conference on Geographic Information Science, Extended abstracts*. Zurich, Switzerland.
- Weger, G. (1997).** *Sémiologie graphique*, cours de cartographie de l'Ecole Nationale des Sciences Géographiques, IGN, Marne la Vallée.
- Weibel, R. and G. Dutton (1998).** Constraint-based Automated Map Generalization. In *8th SDH 98*, Vancouver, Canada, pp. 214-224.
- Weibel, R. and G. Dutton (1999).** Generalising Spatial Data and Dealing with Multiple Representations. In P. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind (Eds.), *Geographical Information Systems, Vol. 1, Principles and Technical Issues* (Second ed.), pp. 125-150. Wiley.
- Weiss, G. (1999).** *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge, MA, USA: MIT Press.
- Werder, S. (2009).** Formalization of Spatial Constraints. In *12th AGILE International Conference on Geographic Information Science*.
- Wilson, I. D., M. J. Ware, and A. J. Ware (2003).** A genetic algorithm approach to cartographic map generalisation. *Comput. Ind.* 52 (3), 291-304.
- Wong, H. C. and K. P. Sycara (2000).** A Taxonomy of Middle-Agents for the Internet. In *4th International Conference on Multi-Agent Systems (ICMAS 2000)*, pp. 465-466. IEEE Computer Society.
- Wood, J. (1996).** *The geomorphological characterisation of digital elevation models*. Ph. D. thesis, , department of geography, university of Leicester.
- Wooldridge, M. (1999).** Intelligent Agents. In G. Weiss (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp. 27-77. Cambridge: The MIT Press.
- W3C working group (2004).** Web Services Architecture. Technical Standard.
- Yaolin, L., Molenaar, M., and Tinghua, A. (2001).** Frameworks for Generalization Constraints and Operations Based on Object-Oriented Data Structure in Database Generalization. In Du, H. L. (Ed.), *20th International Cartographic Conference*, Volume 3.
- Yaolin, L., M. Molenaar, and T. Ai (2003).** Categorical database generalization aided by data model. In *Proceedings of the 21st International Cartographic Conference : Cartographic Renaissance, ICC 2003*, Durban, South Africa, pp. 2308-2318. ICA.
- Zhang, X., T. Ai, and J. Stöter (2010).** Characterization And Detection Of Building Patterns In Cartographic Data: Two Algorithms. In E. Guilbert, B. Lees, and Y. Leung (Eds.), *Proceedings of Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science*, Volume XXXVIII. ISPRS.
- Zhang, X., J. Stoter, and T. Ai (2008).** Formalization and Automatic Interpretation of Map Requirements. In *11th ICA Workshop on Generalisation and Multiple Representation*. ICA. Montpellier, France.
- Zhang, X., J. Stöter, T. Ai, and M.-J. Kraak (2010b).** Formalization and Data Enrichment for Automated Evaluation of Building Pattern Preservation. In E. Guilbert, B. Lees, and Y. Leung (Eds.), *Proceedings of Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science*, Volume XXXVIII. ISPRS archives.
- Zhou, S., N. Regnaud, and C. Roensdorf (2008).** Towards a Data Model for Update Propagation in MR-DLM. In *11th ICA Workshop on Generalisation and Multiple Representation*. ICA. Montpellier, France.

GLOSSAIRE

Agent : entité logicielle ou physique à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents (Briot et Demazeau, 2001).

Algorithme de généralisation : nous appelons algorithme de généralisation un processus qui réalise une opération de généralisation parmi celles qui sont présentées dans la partie A.1.2 grâce à un programme informatique automatique.

Approche de généralisation : nous appelons enfin approche de généralisation un ensemble de modèles de généralisations dont le point de vue est le même.

Modèle de généralisation : nous appelons modèle de généralisation un modèle conceptuel (éventuellement implémenté) qui permet d'enchaîner et/ou combiner différentes opérations de généralisations pour des cas d'applications différents. La mise en œuvre d'un modèle de généralisation pour un cas d'application particulier est considérée ici comme un processus de généralisation.

Ontologie : un ensemble organisé de concepts, propriétés et relations partagés au sein d'un domaine restreint, ainsi que de règles sur ces concepts, propriétés et relations.

Opération de généralisation : un type de transformation que l'on applique à un objet géographique pour réaliser sa généralisation. L'agrégation et la squelettisation sont deux exemples d'opérations de généralisation.

Processus de généralisation : nous appelons processus de généralisation un processus qui enchaîne et/ou combine différentes opérations de généralisation de manière automatique.

Schéma adapté : le schéma adapté est la version enrichie (avec des classes, des attributs et des liens supplémentaires) du schéma de données initial qui permet de décrire tous les concepts géographiques à expliciter pour réaliser une généralisation (il faut par exemple expliciter les ronds-points par une classe et un lien vers les routes pour simplifier les carrefours routiers).

PUBLICATIONS

- Touya, G.**, E. Grosso, S. Mustière, and C. Duchêne (2006). Generalization or data matching for integration? insights from experiments on databases at different levels of details. In M. Raubal, H. Miller, A. U. Frank, and M. F. Goodchild (Eds.), *4th International Conference on Geographic Information Science (GIScience'2006) – Extended Abstracts*, IFGI Prints, pp. 373-376.
- Touya, G.** (2007). River network selection based on structure and pattern recognition. In *International Cartographic Conference*, Moscow, Russia. International Cartographic Association.
- Touya, G.** (2007). A road network selection process based on data enrichment and structure detection. In *workshop on generalisation and multiple representation*, Moscow, Russia. ICA.
- Touya, G.** (2008). First thoughts for the orchestration of generalisation methods on heterogeneous landscapes. In *workshop on generalisation and multiple representation*, Montpellier, France.
- Burghardt, D., S. Schmid, C. Duchêne, J. Stoter, B. Baella, N. Regnaud, and **G. Touya** (2008). Methodologies for the evaluation of generalised data derived with commercial available generalisation systems. In *Workshop on Generalisation and Multiple Representation*.
- Stoter, J., D. Burghardt, C. Duchêne, B. Baella, N. Bakker, C. Blok, M. Pla, N. Regnaud, **G. Touya**, and S. Schmid (2009). Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems* 33 (5), 311-324.
- Touya, G.** and J.-F. Girres (2009). Première évaluation de la qualité des données libres d'OpenStreetMap en France. In *Colloque International de Géomatique et d'Analyse Spatiale SAGEO'09*, Saint-Mandé, France.
- Touya, G.** (2010). Enrichissement automatique de données par analyse spatiale pour la généralisation de réseaux. *Revue Internationale de Géomatique* 20 (2), 175-200.
- Girres, J.-F. and **G. Touya** (2010). Quality assessment of the french OpenStreetMap dataset. *Transactions in GIS* 14 (4), 435-459.
- Touya, G.** (2010). A road network selection process based on data enrichment and structure detection. *Transactions in GIS* 14 (5), 595-614.
- Touya, G.**, C. Duchêne, and S. Mustière (2010). Généralisation et intégration pour un fond vert commun entre l'IFN et l'IGN. *Revue Internationale de Géomatique* 20 (1), 65-86.
- Touya, G.** (2010). Relevant space partitioning for collaborative generalisation. In *Proceedings of 12th ICA Workshop on Generalisation and Multiple Representation*, Zurich.

Touya, G., C. Duchêne, and A. Ruas (2010). Collaborative generalisation: Formalisation of generalisation knowledge to orchestrate different cartographic generalisation processes. In S. Fabrikant, T. Reichenbacher, M. van Kreveld, and C. Schlieder (Eds.), *Geographic Information Science*, Volume 6292 of *Lecture Notes in Computer Science*, Chapter 19, pp.264-278. Berlin, Heidelberg: Springer Berlin / Heidelberg.

Duchêne, C. and **G. Touya** (2010). Emergence de zones conflits dans deux modèles de généralisation cartographique multi-agents. In M. Ocelllo and L. Rejeb (Eds.), *Systèmes Multi-Agents, Défis Sociétaux*, Toulouse, pp. 33-42. Cépaduès.

Touya, G. and C. Duchêne (2011). CollaGen: Collaboration between Automatic Cartographic Generalisation Processes. In Ruas (ed), *Advances in Cartography and GISciences Volume 1*, pp. 541-558. Berlin, Heidelberg: Springer Berlin / Heidelberg.

Duchêne, C., **G. Touya**, J. Gaffuri, P. Taillandier, A. Ruas, and J. Renard (2011). Multi-Agents Systems for Cartographic Generalisation: feedback from past and on-going research works. In B. Buttenfield, C. Brewer, and K. Clarke (Eds.), *Generalization and Data Integration*. Soumis.

RESUME

Cette thèse traite de l'automatisation de la généralisation cartographique qui est le procédé de simplification d'une base de données géographique vectorielle pour sa représentation sur une carte lisible. La recherche dans le domaine a abouti aujourd'hui au développement de nombreux processus automatiques de généralisation cartographique, chacun étant spécialisé pour un problème particulier comme un type de paysage, un thème de donnée, un type de conflit ou un mélange des trois (proximité entre bâtiments en zone urbaine). L'objectif de cette thèse est de tirer parti de cette diversité pour mettre en place la généralisation complète d'une carte en faisant collaborer des processus de généralisation complémentaires.

Pour répondre à cet objectif, nous proposons le modèle CollaGen (Collaborative Generalisation) qui permet, par un système multi-agent, la collaboration des processus : les données sont découpées de manière pertinente par rapport aux processus à disposition en espaces géographiques (une zone urbaine ou le réseau routier par exemple) ; la généralisation d'un espace par un processus est ensuite orchestrée par CollaGen. CollaGen associe de manière itérative un espace à généraliser et un processus adapté, notamment par un mécanisme de registre type pages jaunes. L'interopérabilité entre les processus est assurée par une ontologie du domaine sur laquelle s'appuie un format de spécifications formelles d'une carte généralisée. Chaque généralisation est évaluée globalement en temps réel pour permettre un retour en arrière en cas de problème. Enfin, du fait du principe de découpage en espaces, CollaGen doit vérifier après chaque généralisation si des effets de bord sont apparus avec les objets géographiques situés juste à l'extérieur de l'espace, auquel cas il les corrige au mieux. Dans, cette thèse, le modèle CollaGen est mis en œuvre pour la généralisation de cartes topographiques (notamment au 1 : 50000) et les résultats sont comparés à d'autres approches.

ABSTRACT

This phd thesis deals with cartographic generalisation, the process that simplifies a geographic database to allow its representation on legible map. Past research lead to the development of many automatic generalisation processes, each one being specialised for a specific problem like a particular landscape, a given data theme, a particular graphic conflict or a mix of the three (like 'proximity between buildings in urban areas). The aim of the thesis is to benefit from this diversity to carry out a complete map generalisation by collaboration between complementary processes.

To meet this objective, the CollaGen model is proposed (Collaborative Generalisation) as it allows, based on multi-agent techniques, generalisation processes collaboration: data is relevantly partitioned into geographic spaces (e.g. an urban area or the road network); then, CollaGen orchestrate the generalisation of a space by an adapted process. CollaGen iterately maps a space to be generalised and an adapted process thanks to a yellow pages registry mecanism. The interoperability between processes is managed by a domain ontology on which formal map specifications are based. Each generalization is globally assessed online to allow backtracks if necessary. Finally, because of the space partitioning, CollaGen has to check after each generalisation if side effects appeared with spaces just outside the one that has been generalised. If some side effects occurred, they are corrected. In this thesis, CollaGen is implemented for topographic map generalisation (to 1: 50000) and results obtained are compared to other approaches and discussed.