

MAMANE NOURI SABO

**INTÉGRATION DES ALGORITHMES DE
GÉNÉRALISATION ET DES PATRONS
GÉOMÉTRIQUES POUR LA CRÉATION DES
OBJETS AUTO-GÉNÉRALISANTS (SGO) AFIN
D'AMÉLIORER LA GÉNÉRALISATION
CARTOGRAPHIQUE À LA VOLÉE**

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en Sciences Géomatiques
pour l'obtention du grade de Philosophiae Doctor (Ph. D.)

FACULTÉ DE FORESTERIE ET DE GÉOMATIQUE
UNIVERSITÉ LAVAL
QUÉBEC

2007

Résumé

Le développement technologique de ces dernières années a eu comme conséquence la démocratisation des données spatiales. Ainsi, des applications comme la cartographie en ligne et les SOLAP¹ qui permettent d'accéder à ces données ont fait leur apparition. Malheureusement, ces applications sont très limitées du point de vue cartographique car elles ne permettent pas une personnalisation flexible des cartes demandées par l'utilisateur. Pour permettre de générer des produits plus adaptés aux besoins des utilisateurs de ces technologies, les outils de visualisation doivent permettre entre autres de générer des données à des échelles variables choisies par l'utilisateur. Pour cela, une solution serait d'utiliser la généralisation cartographique automatique afin de générer les données à différentes échelles à partir d'une base de données unique à grande échelle. Mais, compte tenu de la nature interactive de ces applications, cette généralisation doit être réalisée à la volée. Depuis plus de trois décennies, la généralisation automatique est devenue un sujet de recherche important. Malheureusement, en dépit des avancées considérables réalisées ces dernières années, les méthodes de généralisation cartographique existantes ne garantissent pas un résultat exhaustif et une performance acceptable pour une généralisation à la volée efficace. Comme, il est actuellement impossible de créer à la volée des cartes à des échelles arbitraires à partir d'une seule carte à grande échelle, les résultats de la généralisation (i.e. les cartes à plus petites échelles générées grâce à la généralisation cartographique) sont stockés dans une base de données à représentation multiple (RM) en vue d'une éventuelle utilisation. Par contre, en plus du manque de flexibilité (car les échelles sont prédéfinies), la RM introduit aussi la redondance à cause du fait que plusieurs représentations de chaque objet sont stockées dans la même base de données. Tout ceci empêche parfois les utilisateurs d'avoir des données avec un niveau d'abstraction qui correspond exactement à leurs besoins.

Pour améliorer le processus de la généralisation à la volée, cette thèse propose une approche basée sur un nouveau concept appelé SGO (objet auto-généralisant: Self-

¹ Le SOLAP (Spatial OnLine Analytical Processing) est une plate-forme visuelle supportant l'exploration et l'analyse spatio-temporelle faciles et rapides des données selon une approche multidimensionnelle à plusieurs niveaux d'agrégation via un affichage cartographique, tabulaire ou en diagramme statistique (Bédard 1997).

Generalizing Object). Le SGO permet d'encapsuler des patrons géométriques (des formes géométriques génériques communes à plusieurs objets de la carte), des algorithmes de généralisation et des contraintes d'intégrité dans un même objet cartographique. Les SGO se basent sur un processus d'enrichissement de la base de données qui permet d'introduire les connaissances du cartographe dans les données cartographiques plutôt que de les générer à l'aide des algorithmes comme c'est typiquement le cas. Un SGO est créé pour chaque objet individuel (ex. un bâtiment) ou groupe d'objets (ex. des bâtiments alignés). Les SGO sont dotés de comportements spécifiques qui leur permettent de s'auto-généraliser, c.-à-d. de savoir comment généraliser l'objet qu'ils représentent lors d'un changement d'abstraction (ex. changement d'échelle).

Comme preuve de concept, deux prototypes basés sur des technologies Open Source ont été développés lors de cette thèse. Le premier permet la création des SGO et l'enrichissement de la base de données. Le deuxième prototype basé sur la technologie multi-agent, utilise les SGO créés pour générer des données à des échelles arbitraires grâce à un processus de généralisation à la volée. Pour tester les prototypes, des données réelles de la ville de Québec à l'échelle 1 : 1000 ont été utilisées.

Abstract

With the technological development of these past years, geospatial data became increasingly accessible to general public. New applications such as Webmapping or SOLAP which allow visualising the data also appeared. However, the dynamic and interactive nature of these new applications requires that all operations, including generalization processes, must be carried on-the-fly. Automatic generalization has been an important research topic for more than thirty years. In spite of recent advances, it clearly appears that actual generalization methods can not reach alone the degree of automation and the response time needed by these new applications.

To improve the process of on-the-fly map generalization, this thesis proposes an approach based on a new concept called SGO (Self-generalizing object). The SGO allows to encapsulate geometric patterns (generic geometric forms common to several map features), generalization algorithms and the spatial integrity constraints in the same object. This approach allows us to include additional human expertise in an efficient way at the level of individual cartographic features, which then leads to database enrichment that better supports automatic generalization. Thus, during a database enrichment process, a SGO is created and associated with a cartographic feature, or a group of features. Then, each created SGO is transformed into a software agent (SGO agent) in order to give them autonomy. SGO agents are equipped with behaviours which enable them to coordinate the generalization process.

As a proof of concept, two prototypes based on Open Source technologies were developed in this thesis. The first prototype allows the creation of the SGO. The second prototype based on multi-agents technology, uses the created SGO in order to generate data on arbitrary scales thanks to an on-the-fly map generalization process. Real data of Quebec City at scale 1: 1000 were used in order to test the developed prototypes.

Avant-Propos

Je voudrais tout d'abord remercier ma femme, mes enfants et tous mes proches pour le soutien moral qu'ils m'ont apporté lors de la réalisation de cette thèse.

Je remercie énormément mon directeur de thèse, Yvan Bédard pour son soutien, sa patience et la confiance qu'il m'a accordée tout au long de cette thèse. J'aimerais aussi le remercier pour le support financier, sans lequel cette thèse n'aurait probablement pas eu lieu.

Un gros merci à mon codirecteur, Bernard Moulin pour ses conseils judicieux.

Merci à Stéphane Roche pour la pré-lecture de la présente thèse.

Un grand merci à tous les professionnels de recherche et les étudiants de la chaire CRSNG de recherche industrielle en bases de données géospatiales décisionnelles pour les supports moraux, techniques et scientifiques. Mes remerciements vont particulièrement à Eveline Bernier pour ses commentaires pour les articles rédigés lors de cette thèse.

Merci à la chaire de recherche industrielle en bases de données géospatiales décisionnelles et à tous les organismes (le réseau GEOIDE, Recherche et Développement Défense Canada, Ressources Naturelles Canada, Ressources Naturelles, Faune et Parc Québec, Intergraph) qui ont contribué à la réalisation de cette thèse en finançant les projets GEMURE et MUSCAMAGS.

Et enfin, j'aimerais remercier tous les membres du Centre de Recherche en Géomatique (CRG) qui ont fait du CRG un lieu d'échange agréable.

À mes parents, ma femme et mes enfants

Table des matières

Introduction.....	13
1.1 Contexte de recherche.....	13
1.2 Problématique.....	16
1.2.1 Démocratisation des données spatiales vs personnalisation du niveau d'abstraction des données.....	16
1.2.2 Interactivité des nouvelles applications vs temps de réponse et automatisation complète.....	18
1.3 Hypothèse et objectif de recherche.....	19
1.3.1 Hypothèse.....	19
1.3.2 Objectifs de la thèse.....	20
1.4. Les différentes étapes de la méthodologie utilisée.....	20
1.5 Présentation de la thèse.....	24
1.6 Références.....	25
Chapitre 2 Revue de littérature.....	27
2.1 La généralisation cartographique.....	27
2.1.1. Définition et objectifs de la généralisation.....	27
2.1.2. L'automatisation de la généralisation.....	28
2.1.2.1. La généralisation automatique.....	28
2.1.2.2. La conceptualisation de la généralisation.....	30
2.1.2.3. Les opérateurs et les algorithmes de généralisation.....	32
2.1.3. Les différentes approches de généralisation cartographique.....	35
2.1.3.1. L'approche séquentielle.....	35
2.1.3.2. Les approches à base de connaissances.....	35
2.1.3.2.1. L'approche basée sur les systèmes experts à base de règles.....	36
2.1.3.2.2. L'approche à base de contraintes.....	36
2.1.3.2.3. L'approche multi-agent.....	38
2.1.3.3. Les approches à base de connaissances et l'acquisition des connaissances de généralisation.....	39
2.1.3.4. L'approche interactive ou intelligence amplifiée.....	40
2.2. La représentation multiple.....	42
2.2.1. Définition et concepts de la représentation multiple.....	42
2.2.2. Les structures de gestion des représentations multiples.....	43
2.2.2.1. La structure multicouche.....	43
2.2.2.2. La structure hiérarchique.....	43
2.2.2.3. Les arbres hiérarchiques ou représentation multiple implicite.....	44
2.2.2.4. L'estampillage.....	44
2.2.2.5. Les vuels.....	44
2.2.2.6. Les patrons géométriques.....	45
2.2.3. Les problèmes de la représentation multiple.....	46
2.3 La cartographie sur demande.....	46
2.3.1 La carte sur demande.....	46
2.3.2. Exemples d'applications de cartographie sur demande.....	47
2.3.2.1. La cartographie en ligne.....	47
2.3.2.2. Les SOLAP.....	50

2.4.	La généralisation à la volée	51
2.4.1.	Définition	51
2.4.2.	Les différentes approches	51
2.4.2.1.	L'approche orientée-traitements	52
2.4.2.2.	L'approche multi-représentation.....	52
2.4.2.3.	L'approche combinée	53
2.5.	Conclusion	54
2.6.	Références.....	55
Chapitre 3	Introduction du concept des patrons géométriques.....	63
3.1.	Introduction.....	64
3.2.	Les différentes notions de patron.....	69
3.2.1.	Les patrons dans la vie de tous les jours.....	69
3.2.2.	Les patrons informatiques.....	69
3.2.3.	Les patrons en géomatique	70
3.2.4.	Les patrons géométriques	71
3.3.	Enrichissement et application du concept de patrons géométriques à la cartographie et à la généralisation des bâtiments.....	72
3.3.1.	Construction de la base de données de patrons géométriques	72
3.3.1.1.	L'analyse des cartes existantes et détection des patrons de forme	72
3.3.1.2.	Choix des primitives et classification des patrons de forme.....	73
3.3.1.3.	La sélection du patron commun le plus détaillé	74
3.3.1.4.	La construction du patron géométrique initial.....	75
3.3.1.5.	La création des patrons géométriques dérivés et peuplement de la base de données des patrons géométriques.....	79
3.3.2.	Saisie des patrons géométriques	80
3.3.3.	La généralisation des bâtiments à l'aide des patrons géométriques	82
3.4.	Les résultats des tests.....	85
3.4.1.	Données et matériel utilisés	85
3.4.2.	Résultats de la création de la base de données de patrons géométriques	86
3.4.3.	Vérification du taux de remplacement des bâtiments par des patrons géométriques.....	87
3.4.4.	Tests de performance des patrons.....	90
3.4.4.1.	La généralisation des bâtiments individuels	90
3.4.4.2.	La généralisation des groupes de bâtiments	92
3.5.	Conclusion	95
3.6.	Remerciements.....	96
3.7.	Références.....	97
Chapitre 4	Les objets auto-généralisants (SGO)	100
4.1.	Introduction.....	101
4.2.	On-demand mapping and on-the-fly map generalization	104
4.2.1.	On-demand mapping.....	104
4.2.2.	On-the-fly map generalization.....	107
4.2.2.1.	What is on-the-fly map generalization?.....	107
4.2.2.2.	State of the art of on-the-fly map generalization	108
4.2.2.2.1.	Approach based upon generalization algorithms.....	108
4.2.2.2.2.	Multi-representations approach	109

4.2.2.2.3.	Approach combining generalization algorithms and multiple representations	110
4.3.	Self-Generalizing Object- SGO	110
4.3.1.	What is a SGO?.....	110
4.3.2.	SGO components	112
4.3.2.1.	Geometric patterns.....	112
4.3.2.2.	Process patterns.....	117
4.3.2.3.	Spatial integrity constraints	120
4.3.3.	Types of SGOs.....	122
4.3.4.	Generalization process and the SGO	123
4.3.4.1.	Relationships and interaction between SGOs.....	124
4.3.4.2.	Behaviour pattern.....	124
4.4.	Applying the SGO concept.....	126
4.4.1.	A prototype for the creation and enrichment of SGOs (The <i>SGO Creation Engine</i>)	126
4.4.1.1.	Creation of geometric and process pattern databases	127
4.4.1.2.	Importation of a map's features and the creation of simple SGOs.....	128
4.4.1.3.	Enrichment of created SGOs	128
4.4.2.	On-the-fly Map Generalization System.....	130
4.4.2.1.	The Administration Module.....	131
4.4.2.2.	The On-the-fly Map Generation Module.....	131
4.5.	Conclusion	132
4.6.	Acknowledgements.....	134
4.7.	References.....	134
Chapitre 5	La généralisation à la volée à l'aide des objets auto-généralisants (SGO).....	138
5.1.	Introduction.....	139
5.2.	Self-Generalizing Objects (SGO)	144
5.2.1.	Concept of SGO.....	144
5.2.2.	SGO components	146
5.2.2.1.	Geometric patterns.....	146
5.2.2.2.	Process pattern	148
5.2.2.3.	Spatial integrity constraints	149
5.2.2.4.	Behavior pattern.....	150
5.2.3.	Types of SGOs.....	150
5.2.4.	Creation of Self-generalizing objects' (SGOs').....	151
5.2.4.1.	SGOs' creation prototype and data used.....	151
5.2.4.2.	Buildings' SGOs creation	152
5.2.4.3.	Road network SGOs' creation	152
5.3.	Applying the SGO approach to on-the-fly map generalization	153
5.3.1.	System architecture.....	153
5.3.1.1.	Administration Module.....	154
5.3.1.2.	On-the-fly Map Generation Module	155
5.3.2.	Multi-agent architecture.....	157
5.3.2.1.	The Coordination Sub-Module's agent: System Agent.....	158
5.3.2.2.	The Map Generalization Sub-Module's agents	159
5.3.2.2.1.	Map Agent	159
5.3.2.2.2.	Thematic Agent.....	160

5.3.2.2.3.	SGO agents	161
5.3.2.2.4.	The role of a SGO agent	162
5.3.2.2.5.	SGO agent perception	163
5.3.2.2.6.	Interactions and communication between SGO agents	164
5.3.2.3.	The visualization Sub-Module's agent: Drawing Agent	167
5.4.	Implementation of generalization sub-module's agents	168
5.4.1.	Implementation of buildings' agents	169
5.4.1.1.	Implementation of buildings' Thematic Agent.....	169
5.4.1.2.	Implementation of buildings' SGO agents	169
5.4.1.2.1.	SGO agents of aligned buildings	170
5.4.1.2.2.	SGO agents of buildings in block.....	172
5.4.1.2.3.	SGO agents of scattered buildings.....	174
5.4.2.	Implementation of road network agents	177
5.4.2.1.	Implementation of road network Thematic Agent.....	177
5.4.2.2.	Implementation of road network SGO agents	177
5.4.3.	The use of the system and test results.....	179
5.5.	Conclusion and Discussion.....	182
5.6.	Acknowledgments	183
5.7.	References.....	184
Conclusion	188
6.1	Sommaire	188
6.2	Discussions	190
6.3	Conclusions.....	194
6.4	Limitations de l'approche et perspectives de recherche	195
6.5.	Références.....	197
Bibliographie	199
ANNEXES	212

Liste des tableaux

Tableau 1- Paramètres d'un patron géométrique de bâtiments	78
Tableau 2- Generalization results using the prototype (the scales are those that were used at the time of generalization, and the scales of the images can differ slightly from those of the generated maps)	181

Liste des figures

Figure 1- Les différentes étapes de la méthode de recherche	23
Figure 2- La généralisation comme séquence de modèles (après Grünreich 1985)	29
Figure 3- Trois vues servant à définir sémantiquement ou à représenter géométriquement et graphiquement une même réalité (Bédard <i>et al.</i> 2002)	45
Figure 4- Architecture d'UMapIT (Bernier <i>et al.</i> 2005).....	49
Figure 5- Un groupe de patrons et leur patron commun le plus détaillé.....	75
Figure 6- Patron géométrique d'un bâtiment.....	77
Figure 7- Diagramme de classe UML représentant le patron géométrique	79
Figure 8- Mesure du détail minimal d'un bâtiment.....	82
Figure 9- Les principales étapes de la généralisation des bâtiments à l'aide des patrons géométriques.....	84
Figure 10- Extrait des bâtiments de la zone test	86
Figure 11- Arbre hiérarchique des patrons géométriques.....	87
Figure 12- Taux des bâtiments remplacés par des patrons géométriques selon les échelles	88
Figure 13- Taux des bâtiments remplacés par type de patron géométrique	88
Figure 14- Fréquence d'utilisation des patrons géométriques selon le nombre d'instances de primitives secondaires.....	89
Figure 15- Proportion des différents patrons géométriques utilisés pour remplacer 2844 bâtiments.....	91
Figure 16- Temps de réponse moyen (sec.) pour la généralisation des bâtiments individuels	92
Figure 17- Temps de réponse moyen (sec.) pour la généralisation des groupes de bâtiments	92
Figure 18- Exemples de généralisation: a) données initiales; b) utilisation des patrons géométriques; c) agrégation et simplification avec Area Generalizer; d) agrégation et simplification avec Line Generalizer.....	94
Figure 19- Quebec City's road map and bicycle route map with heterogeneous levels of abstraction.....	106
Figure 20- The SGO philosophy.....	112
Figure 21- Detected patterns of shape on a map (left picture) and their corresponding geometric patterns (right picture).	113
Figure 22- Use of a geometric pattern in a cartographic generalization process: A - Initial map; B – Initial feature replaced by a geometric pattern; C - Pattern's orientation; D - Pattern's scaling; E – Result.....	113
Figure 23- UML class diagram for the geometric pattern's data structure.....	115
Figure 24- Geometric pattern's generalization operations.....	115
Figure 25- A geometric pattern and its different simplified forms obtained by iterative generalization (primitive's elimination) (Sabo <i>et al.</i> 2005a).	116
Figure 26- Generalization of a road segment using a geometric pattern (Weibel <i>et al.</i> 2002).	117
Figure 27- UML class model of the process pattern.....	120
Figure 28- Simplified UML class diagram of the data structure to manage SGO.....	123

Figure 29- A simplified view of the architecture including both developed systems: the SGO Creation Engine and the On-the-fly Map Generalization System.	126
Figure 30- User interface of the SGO creation prototype. A- Complex SGO creation panel; B- Constraint specification panel; C- Process pattern specification panel; D- Geometric pattern (on top of the hierarchy) and its simplified forms.	129
Figure 31 UML sequence diagram of the enrichment of database (the stars represent interactive processes).....	130
Figure 32- Interface of the administration (A) and map generation modules of the on-the-fly, multi-agent map-generalization prototype, and generalized data of different scales (C, D).	132
Figure 33- Simplified UML class model of a SGO associated with a cartographic feature	146
Figure 34- A geometric pattern and its generalization (Sabo et al. 2007).	147
Figure 35- Overview of the on-the-fly map generation system architecture (numbers in parentheses indicate the sequence of interactions in the system).	154
Figure 36- Multi-agent On-the-fly Map Generation System interface	156
Figure 37- UML sequence diagram of system modules	157
Figure 38- UML state diagram of the System Agent's Behavior.....	159
Figure 39- UML state diagram of the Map Agent's Behavior	160
Figure 40- UML state diagram of the Thematic Agent's Behavior	161
Figure 41- Transformation of a transversal (R1) interaction to oblique (R2) during a generalization process.....	167
Figure 42- UML state diagram of Drawing Agent's Behavior	168
Figure 43- Simplified UML class model of Buildings' SGO Agents	170
Figure 44- UML state diagram of complex SGO agent for aligned buildings	173
Figure 45- UML state diagram of a simple scattered building SGO agent	176
Figure 46- UML state diagram of complex scattered building SGO agent	176
Figure 47- UML state diagram of a road segment SGO agent (simple SGO agent)	178
Figure 48- UML state diagram of a road SGO agent (complex SGO agent)	179
Figure 49- Simplified UML sequence diagram of agents' interactions.....	180
Figure 50- Times necessary for generating various map scales (all scales are generated from the 1K scale. These arbitrary scales were computed according to the map extension selected by the user.)	182

Introduction

1.1 Contexte de recherche

Depuis l'antiquité lointaine, les hommes ont toujours utilisé des cartes comme un des moyens privilégiés pour représenter ou découvrir leur environnement. Pour certains, cette utilisation de la carte pour représenter l'espace remonte même avant l'invention de l'écriture. Même si le but principal des cartes, celui de représenter l'espace et les phénomènes qui nous entourent est resté le même, les techniques et les supports cartographiques ont quant à eux constamment évolué, suivant ainsi les progrès techniques du moment. L'évolution des techniques et des supports cartographiques engendre le plus souvent, d'une part l'amélioration de la qualité des produits et d'autre part l'apparition de nouveaux défis techniques vis-à-vis de la production cartographique.

Parmi tous les progrès techniques les plus récents qui ont marqué la cartographie, la révolution numérique de ces dernières années est l'une qui a le plus profondément bouleversé la manière de concevoir et d'utiliser les produits cartographiques. Jadis, les cartes étaient produites sur papier pour un usage bien déterminé et pour des utilisateurs très restreints qui étaient le plus souvent des experts d'un domaine spécifique. Mais, avec le développement technologique de ces dernières années, les produits cartographiques jadis réservés à une poignée de spécialistes sont devenus aujourd'hui accessibles au grand public. Ainsi, on observe l'apparition de nouvelles applications utilisant des données spatiales comme par exemple la cartographie en ligne ou les outils SOLAP (Spatial On-Line Analytical Processing).

L'Internet est de loin le média qui a le plus favorisé l'expansion de l'utilisation des données spatiales car il permet d'une part, de diffuser quasi instantanément les données à l'échelle de la planète et d'autre part, de faciliter l'accès à ces données. Selon Internet World Stats (2006), le nombre d'internautes dépasse aujourd'hui le milliard d'individus. Avec le développement d'Internet plusieurs sites permettant d'accéder aux données à référence

spatiale ont vu le jour (ex. Map24²; MapQuest³; Google Maps⁴). D'une manière générale, tous les internautes sont des utilisateurs potentiels des données spatiales. Aujourd'hui, l'Internet est devenu le média le plus important en ce qui concerne le nombre de produits cartographiques diffusés (Kraak *et al.* 2001) car selon Peterson (1999), environ 40 millions de cartes étaient déjà générées quotidiennement sur Internet il y a près d'une dizaine d'années, nombre qui a certainement augmenté de façon substantielle aujourd'hui.

Avec l'apparition de ces nouvelles applications, la cartographie voit apparaître d'autres défis liés d'un côté aux exigences technologiques (ex. les taux de transfert des données pour la cartographie en ligne ou la diversité des supports d'affichage) et de l'autre aux utilisateurs de ces technologies (ex. le manque d'expertise dans le domaine de la production cartographique de certains utilisateurs ou la diversité des besoins due au nombre croissant des utilisateurs). Si dans un passé récent, les utilisateurs avaient seulement la possibilité de visualiser les cartes sur un support-papier ou sur un PC standard, aujourd'hui, ils peuvent utiliser des médias comme les PDA ou les téléphones cellulaires pour accéder aux données spatiales et réaliser diverses requêtes. En ce qui concerne la visualisation, chaque type de média d'affichage (ex. PDA), chaque type d'application (ex. la cartographie en ligne) et chaque utilisateur possèdent ses propres exigences. Ceci a comme conséquence que chaque carte doit être personnalisée et adaptée afin de répondre d'une part aux contraintes qu'imposent ces technologies et ces médias et d'autre part aux exigences des utilisateurs.

Quel que soit le type d'utilisation ou de médias sur lequel la carte doit être affichée, il est évident que, lorsqu'on change l'échelle de visualisation le contenu de la carte peut nécessiter une adaptation afin d'assurer la lisibilité de la carte. Cette adaptation du contenu de la carte requiert souvent un processus de généralisation.

La généralisation cartographique est un procédé qui permet de simplifier le contenu d'une carte en fonction des besoins spécifiques. Actuellement, la généralisation cartographique est considérée comme un processus très complexe, long et coûteux qui demande l'intervention d'un cartographe expérimenté. Depuis l'avènement de la cartographie

² <http://www.map24.com>

³ <http://www.mapquest.ca>

⁴ <http://maps.google.com>

numérique, plusieurs travaux cherchant à automatiser ce processus ont été entrepris à travers le monde (cf. 2.1.2.). Hélas! même si des avancées considérables ont été réalisées en généralisation cartographique ces dernières années, les méthodes existantes ne garantissent pas un résultat exhaustif.

Comme il est actuellement impossible de créer à la volée des cartes à des échelles arbitraires à partir d'une seule carte à grande échelle, les résultats de la généralisation pour les plus petites échelles (i.e. les cartes à plus petites échelles générées grâce à la généralisation cartographique) sont stockés en vue d'une éventuelle utilisation. Face aux contraintes de temps de production et de volume ces cartes ne peuvent pas être créées pour un grand nombre d'échelles (ex. échelles urbaines les plus utilisées comme 1000, 1200, 1500, 2000, 2400, 2500, 3600, 4800, 5000, 7500, 10000, etc.). Conséquemment, les degrés d'abstraction (ex. les échelles) des différentes cartes produites peuvent différer considérablement (ex. 1000, 5000, 10000). D'autre part, en terme de personnalisation du niveau d'abstraction, parfois cette personnalisation doit être poussée jusqu'au niveau des occurrences d'objets contrairement à ce qui se fait actuellement, où le niveau d'abstraction est souvent uniforme pour toutes les occurrences d'une classe d'objets. Par contre, en cartographie traditionnelle, lors de la création des cartes thématiques le niveau d'abstraction des différentes occurrences d'une classe d'objets ne doit pas être toujours uniforme. Par exemple, sur une carte touristique, les bâtiments habitants les musées sont représentés avec un niveau d'abstraction moins élevé que celui des bâtiments résidentiels. Tout cela fait que les utilisateurs ne peuvent pas toujours avoir des données au niveau d'abstraction qui correspond toujours à leurs besoins. C'est pourquoi les données extraites d'une telle base de données nécessitent le plus souvent un raffinement à l'aide des opérations de généralisation cartographique.

C'est dans ce contexte que le projet **GEMURE** (Généralisation multiéchelles et représentation multiple pour la production et la diffusion efficaces de données cartographiques (projet **MEN#BED** du réseau d'excellence **GEOIDE**)) a été initié. L'objectif du projet GEMURE était de développer de nouveaux outils et méthodes permettant la combinaison des opérations de généralisation cartographique et de la représentation multiple afin de faciliter la généralisation dans un contexte de cartes sur

demande. L'approche utilisée dans le projet GEMURE était basée sur une architecture multi-tiers où est tout d'abord produit un entrepôt de données multiéchelles puis des comptoirs de données (datamarts) spécialisés par thème et plage d'échelles (Bernier et Bédard 2007). Les travaux de la présente thèse s'inscrivent aussi dans l'optique de ce projet.

1.2 Problématique

1.2.1 Démocratisation des données spatiales vs personnalisation du niveau d'abstraction des données

L'explosion du nombre d'utilisateurs des données spatiales favorisée par le développement du réseau Internet entraîne de nouvelles exigences vis-à-vis des outils de diffusion des données spatiales car les besoins en données spatiales augmentent, se diversifient et se complexifient constamment. En effet, chaque utilisateur possède ses propres besoins en termes de personnalisation de la carte (ex. l'échelle de visualisation globale ou spécifique à un objet). À cela il faut ajouter les contraintes liées aux technologies utilisées pour diffuser ou accéder à ces données (ex. le taux de transfert des données ou la taille réduite de certains médias d'accès). Même si les outils actuels permettant d'accéder aux données spatiales (ex. la cartographie en ligne) ont connu une nette amélioration ces dernières années, il n'en demeure pas moins qu'ils sont présentement loin de satisfaire tous les besoins des utilisateurs. Entre autres, ces outils ne permettent pas de créer des cartes à des échelles arbitraires selon les spécifications de l'utilisateur. Par exemple, beaucoup des sites de cartographie en ligne sont équipés seulement de simples fonctions de zoom sans aucune réelle capacité de généralisation et les cartes générées sont parfois de piètre qualité (Cecconi *et al.* 2002). Par exemple, des sites de cartographie en ligne comme MapQuest⁵ ou Google Maps⁶ utilisent une approche basée sur le stockage des cartes à différentes échelles. Et lorsque l'échelle de visualisation change, une carte est entièrement remplacée par une autre à différente échelle. Avec une telle approche, le niveau d'abstraction des données présentées à l'utilisateur peut être considérablement différent de celui souhaité.

⁵ <http://www.mapquest.ca>

⁶ <http://maps.google.com>

D'autre part, dans une telle approche, la mise à jour des données devient une tâche de longue haleine car tous les changements doivent être répercutés dans toutes les cartes.

Dans le passé, les besoins en données géographiques étaient limités, ce qui permettait de créer à l'avance des cartes génériques (ex. série des cartes topographiques, les cartes géologiques) à des échelles prédéfinies afin de combler les besoins du moment. Mais, aujourd'hui, avec un tel nombre d'utilisateurs et une telle diversité d'usages des produits cartographiques et des technologies utilisées pour y accéder, il est évident que tous les besoins ne peuvent pas être connus à l'avance, ce qui exclut la possibilité de créer les cartes à l'avance pouvant combler tous les besoins. En effet, chaque requête d'un utilisateur correspond à une carte particulière qui nécessite un degré d'abstraction spécifique et un habillage spécifique. Donc, vu les paramètres à prendre en compte lors de la création de ces cartes et le nombre phénoménal des cartes générées quotidiennement, les outils de visualisation utilisés par ces technologies doivent permettre aux utilisateurs de créer eux-mêmes leurs cartes sans aucune intervention externe des experts. Pour cela, ces outils doivent être entièrement automatiques et inclure des outils de généralisation automatique. Malheureusement, les outils actuels de visualisation ne sont dotés d'aucun mécanisme de généralisation et d'autre part, dans leur forme actuelle, les méthodes de généralisation automatique existantes sont incapables de supporter l'interactivité de tels outils. Ces méthodes de généralisation sont soit très complexes, soit elles exigent une intervention humaine accrue qui dépasse de loin les capacités d'un utilisateur non expert.

Jadis, la production cartographique était une affaire des cartographes qui génèrent des cartes pour différents utilisateurs, généralement pour un domaine spécifique (ex. les cartes géologiques). Cette production cartographique inclut toujours des étapes de contrôle qui permettent de s'assurer de la qualité des produits générés. Cette phase cruciale est le plus souvent réalisée par des cartographes expérimentés. Mais, aujourd'hui, avec des technologies comme la cartographie en ligne ou les SOLAP qui permettent aux utilisateurs non experts du domaine de la production cartographique de générer des cartes sans intervention des experts, le contrôle de la qualité des cartes générées échappe en partie aux producteurs des données. Un élément de solution est d'encapsuler certaines connaissances du cartographe dans la base de données via un processus d'enrichissement.

Vu tous ces problèmes (ex. l'impossibilité de créer les cartes à l'avance, le manque d'expertise des utilisateurs) dus à la démocratisation des données cartographiques, il est urgent d'avoir des méthodes et des outils de généralisation flexibles et fiables permettant aux utilisateurs (y compris des non-experts) de générer eux-mêmes leurs cartes conformément à leurs désirs.

1.2.2 Interactivité des nouvelles applications vs temps de réponse et automatisation complète

Une des grandes particularités des nouvelles applications utilisant les données spatiales (ex. la cartographie en ligne, les SOLAP) est leur caractère interactif. L'interactivité permet à un utilisateur d'interagir avec les données en interrogeant les différentes parties de la carte afin d'obtenir les informations nécessaires. L'avantage de cette manière d'interroger les données est de faciliter l'exploration des données tout en permettant à l'utilisateur de suivre le fil de ses idées sans qu'il ne soit inondé d'informations. Par exemple, pour des applications utilisant les données géographiques, l'interactivité permet de sélectionner une partie de la carte, de changer des paramètres comme les thèmes de la carte, l'échelle de visualisation ou l'habillage de la carte. La majorité des applications permettent de changer ces paramètres. Par contre, le changement de la plupart de ces paramètres nécessite le plus souvent un processus de généralisation afin d'adapter le contenu de la carte. Actuellement, ces applications n'offrent pas des outils de généralisation capables de supporter convenablement cette interactivité.

Bien que l'interactivité présente un réel avantage lors de l'exploration interactive des phénomènes géographiques du fait qu'elle permet de ne pas perdre le fil conducteur de ses idées, il n'en demeure pas moins qu'elle engendre plusieurs défis vis-à-vis des outils de visualisation des données géographiques. En effet, pour supporter un tel processus (l'interactivité), d'une part, le délai entre la requête de l'utilisateur et la présentation des résultats doit être assez court et d'autre part, le processus doit être entièrement automatique afin de permettre à l'utilisateur de se concentrer sur les résultats de ses requêtes et non sur le processus de production de ces résultats. Pour cela, toutes les transformations qui sont appliquées aux données doivent être réalisées quasi instantanément et sans aucune intervention humaine. Pour ce qui est du temps de réponse nécessaire à la transformation

des données (y compris la généralisation si cette dernière est requise), il doit être conforme à la bande cognitive de Newell (1990) qui est inférieure à dix secondes, temps d'attente au-delà duquel le fil conducteur lors d'une analyse quelconque peut être rompu. Actuellement, plusieurs opérations simples (ex. changement de symbologie) peuvent être réalisées à la volée lors de la visualisation des données. Par contre, il n'en est pas de même pour des opérations plus complexes comme la généralisation cartographique qui reste encore un processus complexe, long et souvent semi-automatique.

Question de recherche : comment supporter un processus de généralisation cartographique à la volée, soit une généralisation quasi-instantané et ne demandant aucune intervention humaine afin de faciliter la personnalisation du niveau d'abstraction des cartes générées par des applications interactives comme la cartographie en ligne ou les outils SOLAP?

1.3 Hypothèse et objectif de recherche

Vu, d'une part, la problématique liée à la démocratisation des données géographiques (la diversité des besoins et la difficulté de créer des cartes à l'avance, capables de combler les différents besoins) et à l'interactivité des nouvelles applications utilisant les données géographiques (transformation des données à la volée à l'aide des processus entièrement automatiques) et d'autre part les limites des approches actuelles de généralisation (ex. le caractère souvent semi-automatique de la généralisation cartographique), il est évident que de nouvelles approches permettant d'adapter le contenu de la carte s'imposent. Ces nouvelles approches doivent permettre entre autres la généralisation à la volée des données spatiales. Dans cette thèse, nous nous attaquons à ces problèmes de la généralisation à la volée.

1.3.1 Hypothèse

L'hypothèse principale de la thèse est que l'intégration d'algorithmes de généralisation, de patrons géométriques et de contraintes d'intégrité spatiales peut supporter un processus de généralisation à la volée. Cette dernière peut grandement faciliter la personnalisation du niveau d'abstraction des données générées par des applications comme la cartographie en ligne ou les outils SOLAP.

1.3.2 Objectifs de la thèse

Afin de démontrer l'hypothèse énumérée ci-dessus, l'objectif principal de la thèse est de développer un nouveau concept appelé SGO capable de supporter un processus de généralisation à la volée qui exige un temps de réponse quasi instantané et aucune intervention humaine, puis d'en tester le potentiel grâce à des données réelles. Pour atteindre le principal objectif de la thèse, les objectifs spécifiques suivants doivent être réalisés :

- Développer le concept théorique des SGO, incluant autant les aspects algorithmiques que ceux des patrons géométriques et des contraintes d'intégrité spatiales;
- Enrichir le concept des patrons géométriques existant afin de l'adapter aux exigences des SGO;
- Intégrer les algorithmes de généralisation sous forme de patrons de traitements afin de faciliter leur utilisation;
- Développer informatiquement des SGO, de même que l'outil qui permet de les enrichir;
- Créer un outil de généralisation à la volée utilisant les SGO. Cet outil permettra de tester et de valider le concept développé.

1.4. Les différentes étapes de la méthodologie utilisée

La méthode de recherche proposée dans cette thèse suit une approche expérimentale dans laquelle l'hypothèse de départ est que l'intégration d'algorithmes de généralisation, des patrons géométriques et des contraintes d'intégrité spatiales peut supporter un processus de généralisation à la volée. Pour vérifier cette hypothèse, plusieurs tests ont été réalisés et deux prototypes ont été développés. La présente thèse est composée de différents articles dont chacun des ces articles présente une partie de la méthodologie suivie (par exemple, l'article présenté au chapitre 3 expose la méthodologie suivie pour développer le concept

des patrons géométriques). Dans ce qui suit, nous présentons les différentes étapes de cette méthodologie.

Une étude bibliographique traitant des différents aspects reliés au sujet de la thèse (la généralisation cartographique, la représentation multiple, la cartographie sur demande, la généralisation à la volée) fut effectuée dans un premier temps. Cela a permis de mieux définir le projet de recherche, ainsi que le contexte de la recherche, la problématique, et l'hypothèse de recherche. La revue de littérature a continué tout au long du projet afin d'une part de raffiner certains concepts et d'autre part pour prendre en compte les développements récents.

La deuxième étape de la méthodologie nous a permis de développer le concept théorique des SGO. Ainsi, différents modèles UML des SGO et leurs composants ont été développés. Les SGO encapsulent entre autres des patrons géométriques qui sont des concepts récents. Ainsi, les présents travaux de recherches bénéficient des travaux du mémoire de maîtrise de Cardenas (2004) qui s'inscrivent aussi dans le cadre du projet GEMURE. Dans son mémoire, il a développé le concept théorique des patrons géométriques introduit par Bernier (2002), ainsi qu'une méthodologie pour l'utilisation des patrons géométriques multi-échelles comme support à la généralisation automatique dans le contexte des cartes routières. D'autre part, les patrons géométriques n'ont jamais été implémentés, donc le gain réel qu'ils peuvent procurer lors d'un processus de généralisation n'a jamais été vérifié. En plus, de par leur création et leur utilisation, les patrons géométriques du réseau routier diffèrent considérablement de ceux des autres thèmes de la carte (ex. des bâtiments). C'est pourquoi, un enrichissement du concept afin de l'adapter aux besoins des SGO a été nécessaire. Dans cette phase, la pertinence d'utiliser des patrons géométriques lors d'un processus de généralisation à la volée a été vérifiée. Cette vérification a été réalisée à l'aide de différents tests qui ont permis de vérifier d'une part le nombre des objets cartographiques de la zone test qui peuvent être remplacés par des patrons géométriques et d'autre part la performance des opérations d'ajustement des patrons (voir premier article présenté au chapitre 3). Le concept développé a fait l'objet de plusieurs communications lors des conférences scientifiques ou des workshops du projet GEMURE. Les discussions lors de ces communications nous ont permis de raffiner le concept. Pour valider le concept

des SGO, deux prototypes ont été développés. Ainsi, une fois que le concept théorique des SGO a été stabilisé, le modèle des SGO développé a été implémenté et un prototype (*SGO Creation Engine*) permettant la création des SGO et l'enrichissement de la base de données a été développé en Java (voir deuxième article présenté au chapitre 4). Le prototype est basé sur des bibliothèques Open Source : JTS (Java Topological suite) (JTS 2006) et Geotools (GeoTools 2006). Le prototype nous a permis de créer des SGO à partir des données réelles de notre zone test (une portion de la ville de Québec). Une description de ces données est présentée à la Section 3.4.1.

La troisième étape a porté sur l'utilisation des SGO dans un processus de généralisation à la volée (voir troisième article présenté au chapitre 5). Ainsi, un prototype (*On-the-fly Map Generalization System*) de généralisation à la volée basé sur la technologie multi-agent a été développé. Le prototype permet aux utilisateurs de généraliser automatiquement, eux-mêmes leur carte et ceci sans aucun délai notable et à n'importe quelle échelle d'affichage à partir d'une seule source cartographique à grande échelle (1K). La plage d'échelles qui peuvent être générées dépend du type d'enrichissement introduit dans les données. Pour tester le prototype, des SGO ont été créés à l'aide du prototype *SGO Creation Engine* (deuxième étape de la méthodologie). Les SGO créés à l'aide de ce prototype sont ensuite injectés dans le prototype *On-the-fly Map Generalization System* destiné à la généralisation à la volée où ils sont transformés en agents logiciels. Ce prototype (*On-the-fly Map Generalization System*) est basé sur la plateforme multi-agent JADE (JADE 2006). Ce prototype est décrit à la Section 5.3. Pour tester le prototype (*On-the-fly Map Generalization System*), des cartes à différentes échelles (de 1K à 50K) ont été générées à la volée. Pour chaque échelle générée, des vérifications de la performance et de la qualité visuelle des données ont été réalisées. Ainsi, pour ce qui est de la performance, le temps nécessaire pour la génération de chaque carte est mesuré (figure 50, Section 5.4.3.). Ce temps doit être conforme à la bande cognitive de Newell (1990) qui est en dessous de 10 secondes. Ce seuil correspond au temps maximum nécessaire pour réaliser des opérations cognitives tout en conservant le fil de ses idées. Lors de l'analyse visuelle des données générées, la qualité esthétique des données générées, de même que le respect des contraintes d'intégrité spécifiées dans le système (ex. la séparation minimale entre deux objets) sont vérifiés.

La quatrième et dernière étape consiste à la rédaction de la thèse. Lors de cette étape, les différents articles (quatre) rédigés tout au long de cette recherche ont été intégrés et complétés par une révision de la revue de littérature (chapitre 2) de tous les domaines qui touchent à la thèse. Un diagramme d'activités UML des différentes étapes de la méthodologie utilisée dans cette thèse est montré à la figure 1.

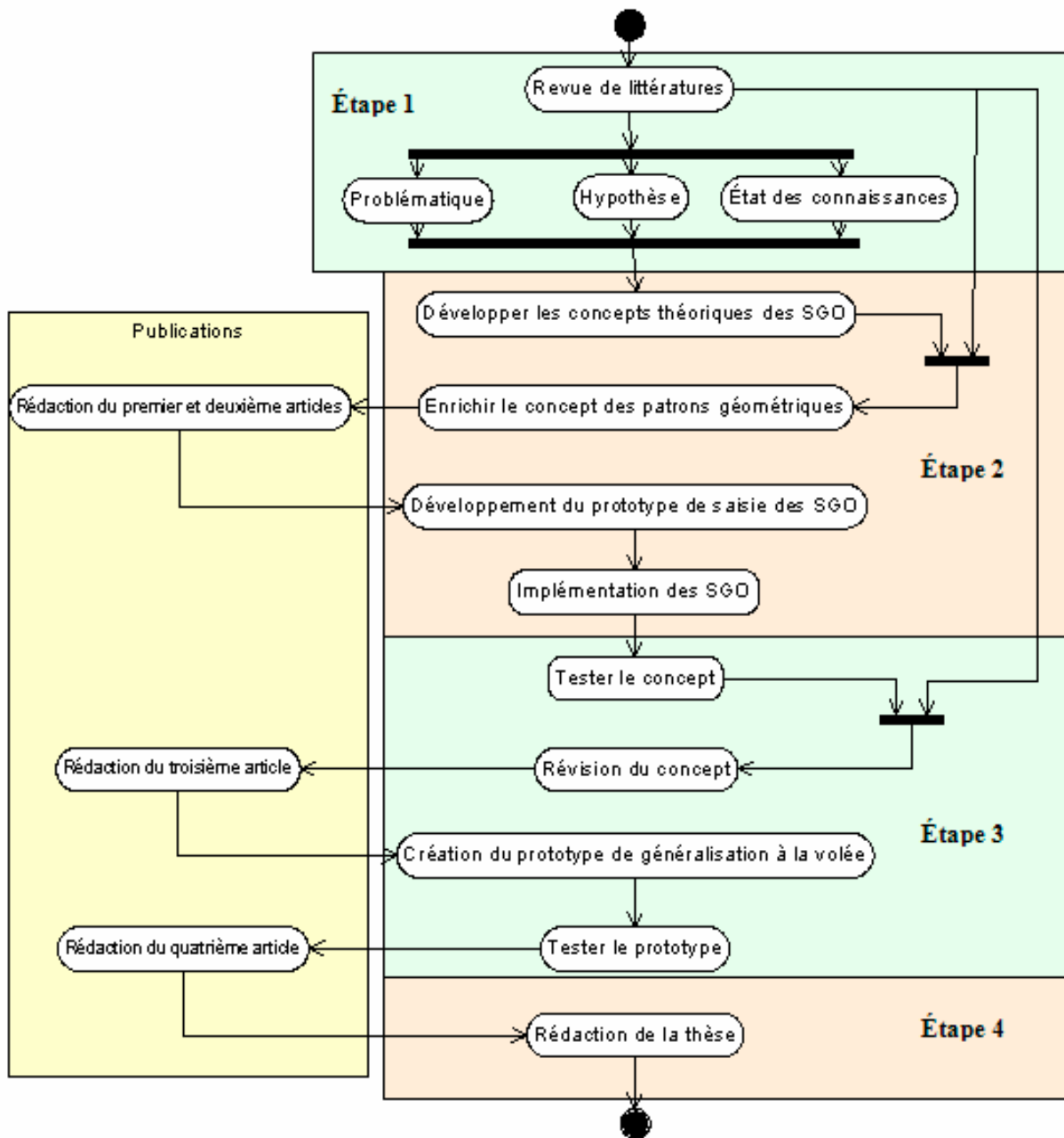


Figure 1- Les différentes étapes de la méthode de recherche

1.5 Présentation de la thèse

La présente thèse est essentiellement composée d'articles. Quatre articles dont un publié dans la revue *Geomatica* en 2005, deux soumis à des revues scientifiques à comité de lecture (*Cartographica* et *GeoInformatica*) et un accepté et présenté à une conférence (*International Cartographic Conference* à Coruna en Espagne). La présente thèse est constituée de six chapitres et une partie annexes:

Le **deuxième chapitre** présente une revue de littérature des travaux existants dans les domaines reliés à la thèse comme la généralisation cartographique automatique, la cartographie en ligne, la généralisation à la volée, et la représentation-multiple.

Le **troisième chapitre** présente l'enrichissement du concept des patrons géométriques et leur application au cas des bâtiments.

Le **quatrième chapitre** présente le concept des SGO. Ce chapitre introduit en même temps certaines parties (les patrons géométriques) qui ont été développées dans le précédent chapitre de la thèse.

Quant au **cinquième chapitre**, il présente le prototype multi-agent dédié à la généralisation à la volée des cartes à l'aide des SGO.

Le **sixième chapitre** conclut la thèse. Ce chapitre discute aussi des résultats obtenus lors de cette présente thèse et présente de même les limites de l'approche et des défis qui restent à relever.

Et finalement, dans la partie **annexes** de la thèse sera inclus le deuxième article présenté à une conférence qui porte sur la méthodologie à suivre lors de la création des patrons géométriques (annexe 1). L'annexe 2 présente un tableau comparatif des différentes approches multi-agents de généralisations comparables à la notre.

Remarque générale sur les articles intégrés dans la thèse : *Comme nous l'avons mentionné ci-haut, la présente thèse est essentiellement composée d'articles présentés à différentes revues, à des périodes différentes. Donc, comme les différents articles sont liés à*

la même problématique, des redondances peuvent être notées au niveau des différentes parties de la thèse. Par exemple, la mise en contexte des articles reprend en partie des éléments présentés dans la partie introduction de la thèse. Pour guider le lecteur, une section remarque spécifiant explicitement le principal message véhiculé est présentée au début de chaque chapitre s'appuyant sur un article.

1.6 Références

- Bédard Y., Bernier E. et Devillers R., "La métastructure VUEL et la gestion des représentations multiples". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p. 149-162.
- Bernier E., *Utilisation de la Représentation Multiple comme Support à la Génération de Vues de Bases de Données Géospatiales dans un Contexte SOLAP*. Mémoire de Maîtrise, Université Laval, 2002, 89 p.
- Bernier E. and Bédard Y., "A data warahouse strategy for on-demand multi-scale mapping". Mackaness W., Ruas A. and Sarjakoski T. (Eds.), *Generalization of Geographic Information: Cartographic Modelling and Applications*, 2007, p. 177-198.
- Cardenas A., *Utilisation des Patrons Géométriques comme Support à la Généralisation Automatique*. Mémoire de Maîtrise, Université Laval, 2004, 110 p.
- Cecconi A., Weibel R. and Barrault M., "Improving automated generalisation for on-demand Web mapping by multiscale database". *Proceeding of Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada, 2002.
- Devogele T., Trevisan J. et Raynal L., "Processus de constitution d'une base de données multi-échelles". *Revue Internationale de Géomatique*, 6(2-3), 1996, p. 249-263.
- Didier M. et Bourveyron C., *Guide Économique et Méthodologique des SIG*. Paris, Hermès, 1993, 330p.
- GeoTools,. The open source java GIS toolkit. Disponible sur <http://geotools.codehaus.org/Home>, 2006.
- JADE, Java Agent Development Framework. Disponible sur <http://jade.tilab.com/>, 2006.
- JTS, JTS Topology Suite. Disponible sur <http://www.vividsolutions.com/jts/jtshome.htm>, 2006.
- Kilpelainen T., "Maintenance of multiple representation database for topographic data". *The Cartographic Journal*, 37(2), 2000, p. 101-107.
- Kraak M.-J. and Brown A. *Web Cartography: Developments and prospects*. Londres, Taylor and Francis, 2001.
- Newell A. *Unified Theories of Cognition*. Cambridge MA, Harvard University Press, 1990, 549 p.

- Peterson M. P., "Trends in Internet map use, a second look. In touch the past visualize the future". *Proceeding of 19th International Cartographic Conference*, Ottawa, Canada, 1999.
- Internet World Stats, World, Internet Users and Population Stats. Disponible sur <http://www.internetworldstats.com/stats.htm>, 2006.
- Timpf S. and Frank A.U., "A multi-scale data structure for cartographic objects". *Proceeding of 17th International Cartographic Conference*, Barcelone, Spain, 1995.
- Vangenot C., "Représentation multi-résolution, concepts pour la description de bases de données avec multi-représentation". *Revue Internationale de Géomatique*, 8(1-2), 1998, p. 121-147.

Chapitre 2 Revue de littérature

Les travaux de la présente thèse nécessitent la compréhension de plusieurs concepts comme la généralisation cartographique ou la représentation multiple. Dans ce chapitre, nous présentons une synthèse de la littérature traitant les différents concepts liés à la présente thèse. Ainsi, en premier lieu, nous présentons les différents travaux réalisés dans le domaine de la généralisation cartographique tout en mettant l'accent sur l'automatisation de ce processus. En deuxième temps, nous présentons la représentation multiple qui est souvent utilisée pour stocker les résultats de la généralisation cartographique à défaut d'avoir des méthodes de généralisation à la volée fiables. Nous présentons ensuite le concept des cartes sur demande qui sont des cartes créées selon les exigences des utilisateurs. Dans la quatrième partie de ce chapitre, nous présentons le concept de la généralisation à la volée, mettant l'accent sur les différentes approches existantes. Et en conclusion un constat sur les différents concepts évoqués est présenté.

2.1 La généralisation cartographique

2.1.1. Définition et objectifs de la généralisation

La généralisation est l'un des processus de traitement de l'information les plus utilisés dans la vie quotidienne. Sans généralisation nos observations sur le monde qui nous entoure seront isolées dans l'espace et le temps (McMaster et Shea 1992). En cartographie, la généralisation est fréquemment utilisée dans la chaîne de production des données.

Selon Cuenin (1972), la généralisation cartographique est l'opération qui, par sélection, schématisation et harmonisation, reconstitue sur une carte la réalité de la surface terrestre représentée dans ses traits essentiels en fonction du but de la carte, de son thème, de son échelle et des particularités de la région cartographiée. C'est donc une opération d'abstraction qui permet de créer des données plus simplifiées tout en conservant l'information essentielle véhiculée par la carte.

L'objectif principal de la généralisation est d'améliorer la lisibilité de la carte afin de faciliter la lecture de l'information véhiculée (Weibel et Dutton 1999). Grâce au processus

de généralisation, le principal message véhiculé par la carte est mis en relief tandis que les messages superflus sont éliminés. La généralisation cartographique est influencée par plusieurs facteurs comme l'échelle de visualisation, l'objectif de la carte ou la spécificité du territoire cartographié. Mais, l'échelle de visualisation est le facteur qui influence le plus le degré de généralisation d'une carte.

La généralisation cartographique est l'un des aspects de la production cartographique le plus intellectuellement et techniquement difficile (Shea 1991). La complexité et le caractère subjectif de la généralisation sont connus dès les premiers écrits sur la généralisation cartographique. Ainsi, dès le début du vingtième siècle, le cartographe allemand Max Eckert disait ceci :

«In generalizing lies the difficulty of scientific map-making, for it no longer allows the cartographer to rely merely on subjective facts but requires him to interpret them subjectively. To be sure the selection of the subject matter is controlled by considerations regarding its suitability and value, but the manner in which this material is to be rendered graphically depends on personal and subjective feeling» (Eckert et Joerg 1908).

D'autre part, la généralisation cartographique est un processus hautement holistique. Lors d'un processus de généralisation manuelle, le cartographe a une vision globale de la carte, il ne regarde pas les objets de la carte d'une manière isolée, mais dans leur contexte tout en considérant à la fois leurs caractéristiques internes (ex. la grandeur, la sinuosité) et leurs relations avec les autres objets de la carte.

2.1.2. L'automatisation de la généralisation

2.1.2.1. La généralisation automatique

En cartographie numérique, la généralisation est considérée comme un processus qui permet de faire la transition entre différents modèles représentant une partie du monde réel en diminuant les détails, tout en maximisant le contenu de l'information en fonction d'une application donnée (Weibel et Dutton 1999). Selon Grünreich (1995), il existe deux types de généralisations : 1) la généralisation objet qui permet de modéliser les objets du monde

réel (sous forme du modèle primaire ou DLM- digital landscape model); 2) et la généralisation cartographique utilisée pour toutes les applications dans un but de visualisation (elle permet de générer le DCM-digital cartographic model). D'autres auteurs comme Weibel et Dutton (1999) proposent un troisième type de généralisation, la généralisation de modèle qui permet de simplifier le modèle primaire. La généralisation de modèle n'est jamais utilisée dans un but de visualisation, mais elle sert principalement à réduire le volume des données, d'accélérer le traitement des données ou de support à la généralisation cartographique (processus de pré-généralisation). La figure 2 proposée initialement par Grünreich (1995) et adaptée par Weibel et Dutton (1999) montre les différentes transitions entre les modèles.

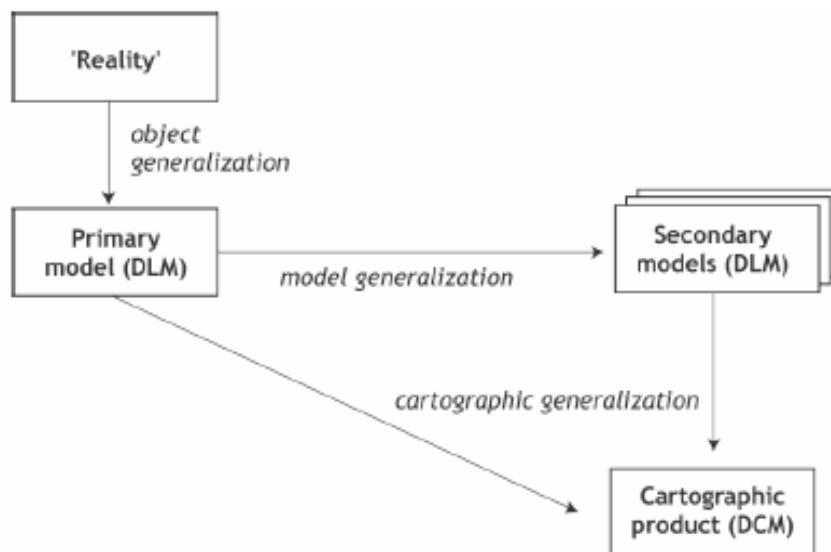


Figure 2- La généralisation comme séquence de modèles (après Grünreich 1985)

Les premières tentatives d'automatisation de la généralisation remontent aux années 60 (Perkal 1966; Topfer et Pillewizer 1966). Selon Bittenfield et McMaster (1991), les recherches sur l'automatisation de la généralisation ont été réalisées en trois étapes. Durant la première étape (1960 – 1975), les travaux se sont concentrés sur le développement des algorithmes de généralisation et particulièrement ceux destinés à la simplification des objets linéaires. Après cette période, l'évaluation des résultats générés par les algorithmes

était devenue une nécessité. C'est pourquoi la deuxième époque (débutant dans les années 70) dans le développement de la généralisation cartographique a été occupée par la création des méthodes et des outils permettant d'évaluer les conséquences de la généralisation (ex. Mailing 1989 ; Müller 1986). Quant à la troisième étape, elle a été axée sur la formalisation des connaissances sur la généralisation cartographique, principalement à l'aide des systèmes experts (ex. Armstrong 1991).

Selon McMaster (1991), pour développer un système de généralisation entièrement automatisé, trois problèmes complexes doivent être résolus : 1) un cadre conceptuel; 2) les procédures ou les opérations de généralisation doivent être créées; 3) et les connaissances cartographiques doivent être formalisées. Ces constats sont encore valides car ils sont constamment repris dans plusieurs approches récentes de généralisation. Nous reprendrons ces différents points dans les prochaines sections.

2.1.2.2. La conceptualisation de la généralisation

Les premiers écrits sur la généralisation cartographique émanent du cartographe allemand Max Eckert (1908). Ce dernier avait décrit le processus de la généralisation comme étant un travail artistique basé sur des principes scientifiques. Mais, c'est à partir des années 60 que les premiers modèles conceptuels de généralisation ont vu le jour. Les modèles fournissent une base pour la compréhension des aspects complexes et corrélés du processus de généralisation (Shea 1991).

Le premier modèle conceptuel de généralisation fût proposé par le cartographe polonais, Ratajski (1967). Ce modèle identifie deux types de généralisation : 1) La généralisation quantitative qui consiste en une diminution progressive du contenu de la carte suite à une réduction d'échelle; 2) et la généralisation qualitative qui consiste en un passage des notions élémentaires à des notions plus générales. Ce modèle représente la généralisation cartographique sous forme d'un triangle dont la base représente la capacité maximale de la carte et le sommet représente les limites de cette capacité. Quand la capacité de la carte tend vers ses limites appelées points de généralisation, un changement de méthode de représentation devient nécessaire (exemple des bâtiments individuels sont agrégés pour former des pâtés de maisons).

Les premiers modèles dédiés spécifiquement à la généralisation automatique ont seulement vu le jour dans les années 70 — 80 (Morrison 1974; Nickerson et Freeman 1986; Brassel et Weibel 1988; McMaster et Shea 1988; Shea et McMaster 1989). Parmi tous ces modèles, celui développé par Brassel et Weibel (1988) est considéré comme le plus complet (McMaster 1991). Il reste encore un des meilleurs modèles. Ce modèle identifie cinq processus de généralisation, qui sont : 1) l'identification des structures spatiales qui permet d'identifier les objets, les groupes d'objets ou les relations entre les objets; 2) l'identification des processus pour choisir les opérations de généralisation nécessaires; 3) la modélisation des processus pour compiler les procédures qui seront appliquées; 4) l'exécution des processus qui est la phase de généralisation proprement dite, durant laquelle les différentes transformations sont exécutées; 5) et la visualisation des données généralisées.

Une année plus tard, Shea et McMaster (1989) ont proposé un modèle qui a largement inspiré plusieurs chercheurs. Ce modèle identifie trois considérations pour la généralisation: (1) intrinsèque, ou pourquoi nous généralisons; (2) l'évaluation de la situation ou quand généraliser; (3) et les transformations spatiales et sémantiques ou comment généraliser. Les modèles de généralisation créés dans les années 80 ont surtout permis le développement des systèmes de généralisation à base de systèmes experts.

Tout récemment, des modèles s'inspirant de la technologie multi-agent et basés sur l'autonomie des objets géographiques ont vu le jour (ex. Ruas (1999), Jabeur (2005)). Dans ces types de modèles, une autonomie est donnée aux objets de la carte afin de leur permettre de choisir eux-mêmes les transformations nécessaires lors de la généralisation. Dans un tel modèle, plusieurs niveaux d'analyse sont le plus souvent explicités. Par exemple, dans les modèles proposés par Jabeur (2005) et Ruas (1999) on distingue deux niveaux d'analyse : le niveau des objets individuels et le niveau des groupes d'objets. Tous ces nouveaux modèles s'inspirent grandement de ceux développés dans les années 80-90 (ex. Brassel et Weibel 1988).

2.1.2.3. Les opérateurs et les algorithmes de généralisation

La généralisation est un processus très complexe. C'est pourquoi, dès le début des recherches sur ce processus le problème a été décomposé en plusieurs sous-problèmes afin de faciliter son automatiser. Cette décomposition de la généralisation est matérialisée par la définition de plusieurs opérateurs. Les opérateurs de généralisation représentent des transformations types et récurrentes utilisées pendant le processus de généralisation (Ruas 2002).

À ce jour, plusieurs opérateurs de généralisation ont été proposés (Brassel 1985; Buttenfield 1985; Nickerson et Freeman 1986; Beard 1987; Delucia et Black 1987; McMaster 1989). Une description complète des opérateurs de généralisation est présentée par Shea et McMaster (1989). Ils ont défini une douzaine d'opérateurs de généralisation dont deux destinés à la transformation des attributs des objets de la carte (classification et symbolisation).

Selon les auteurs, la définition et le nombre des opérateurs peuvent différer. Malgré ces différences, on retrouve chez presque tous les auteurs les opérateurs suivants : simplification, sélection, élimination, déplacements, agrégation, exagération. Selon Ruas (2002), le manque de définitions officielles, la diversité des objectifs de recherche et la diversité des langues de publication constituent les principales sources des divergences quant aux termes utilisés pour qualifier les opérateurs de généralisation.

Si les opérateurs de généralisation désignent des transformations abstraites lors de la généralisation, les algorithmes de généralisation quant à eux permettent d'exécuter ces transformations. Ceci implique que, contrairement aux algorithmes, les opérateurs de généralisation sont indépendants du modèle des données (ex. vecteur ou matriciel) (Cecconi 2003).

Depuis plusieurs années, les recherches sur la création des algorithmes de généralisation cartographique ont été en grande partie concentrées sur les objets linéaires. Ceci n'est pas étonnant si on sait que quatre-vingt pour cent des objets cartographiques sur des cartes topographiques à moyenne échelle sont linéaires (Müller 1991). En plus, les recherches sur

les objets linéaires profitent aussi aux objets surfaciques car selon les modèles actuels de données en mode vectoriel, les polygones sont composés de lignes.

Les algorithmes de généralisation peuvent être divisés en deux groupes : les algorithmes dits indépendants qui s'appliquent sur un objet et modifient sa géométrie (ex. les algorithmes de simplification), et les algorithmes contextuels qui traitent un ensemble d'objets (ex. les algorithmes de déplacements) (Ruas 2002). D'autre part, si certains algorithmes sont génériques du fait qu'ils ne tiennent pas compte de la sémantique des objets (ex. algorithmes de simplification de lignes), d'autres sont très spécialisés et ne peuvent être appliqués qu'à des objets spécifiques (ex. algorithme de déplacement des bâtiments).

De toutes les opérations de généralisation cartographique, la simplification des objets linéaires en mode vectoriel a eu le plus d'intérêt pour les chercheurs. La simplification est une opération qui permet de réduire les détails d'un objet cartographique. Du début des recherches sur l'automatisation à nos jours, plusieurs algorithmes de simplification ont été proposés. Nous pouvons citer entre autres : Lang 1969; Douglas et Peucker 1973; Reumann et Witkam 1974; Dougenik 1980; Opheim 1982; Jenk 1989; Cromley 1991; VisValingam et Whyatt 1993; Wang et Müller 1998. Selon McMaster (1987), les algorithmes de simplification de lignes peuvent être divisés en cinq groupes : 1) *Independent Point Algorithms* (qui ne tiennent compte d'aucune relation entre les points de l'objet) ; 2) *Local Processing Routines* (qui tiennent compte du voisinage immédiat pour déterminer l'importance d'un point) ; 3) *Unconstrained Extended Local Processing Routine*. En plus du voisinage immédiat, les algorithmes de cette catégorie évaluent aussi une portion de la ligne; 4) *Constrained Extended Local Processing Routine*, similaires à ceux du groupe précédent sauf qu'ils comportent des restrictions par rapport à la zone de recherche; 5) et les *Global Routines* qui considèrent la ligne en entier. C'est le cas de l'algorithme Douglas-Peucker (Douglas et Peucker 1973).

Une autre opération non moins importante dans le processus de généralisation cartographique est la **sélection**. Cette opération permet de sélectionner et d'inclure dans une carte les objets pertinents selon les objectifs de la carte qui sera générée. L'intérêt accordé à la sélection s'est amplifié ces dernières années avec l'arrivée de la généralisation à la volée.

Malgré ce relatif intérêt que portent certains chercheurs sur cette opération, toutes les recherches se résument pratiquement à une structuration hiérarchisée des données sous forme d'arbre pour permettre l'accès lors de la sélection. Bien que les problèmes de représentation des objets après des changements d'échelle soient bien connus depuis longtemps, peu de travaux ont été effectués sur la manière de quantifier ce phénomène. L'un des rares travaux dans ce domaine est celui présenté par Topfer et Pillewizer (1966). Ces derniers ont présenté les principes de sélection avec une formule qui permet de calculer le nombre d'objets qui seront sélectionnés lors de la généralisation en tenant compte de l'échelle et du nombre d'objets de la carte source et de l'échelle de la carte dérivée. Malheureusement, cette formule ne permet pas de dire quel objet sera sélectionné.

Malgré son importance évidente, le *déplacement* lors de la généralisation cartographique reste sous-étudié. En cartographie numérique, le déplacement est un processus très complexe qui demande beaucoup de temps pour son accomplissement. Ce délaissement est en partie dû à la complexité de l'opération de déplacement, car le déplacement est appliqué sur un groupe d'objets et la résolution d'un conflit peut engendrer d'autres conflits plus sévères. En déplaçant un objet en conflit, toutes les caractéristiques spatiales de son environnement (métrique, géométrique, topologique, <<gestaltique>>) sont modifiées (Barrault et Bader 2002). Donc, le problème réside non pas au niveau du déplacement même, mais au niveau des conséquences qu'engendre une telle manipulation. Selon Barrault et Bader (2002), le traitement d'un conflit par déplacement peut soit se faire localement, mais en prévoyant l'impact sur le voisinage (une approche locale), soit en intégrant dans une même résolution l'ensemble des objets potentiellement impliqués et leurs conflits (une approche globale). Malgré ce délaissement, plusieurs travaux ont été réalisés ces dernières années (Nickerson 1988; Mackaness 1994; Ruas 1998; Bader 2001, Jabeur 2006).

Il faut aussi noter qu'il existe d'autres opérations de généralisation non moins importantes, telles que l'agrégation, le lissage, la typification, collapse, l'exagération, l'élargissement, etc. Les différents opérateurs de généralisation et les principaux algorithmes pour ces opérations sont présentés dans McMaster et Shea (1992).

Les algorithmes de généralisation permettent généralement de réaliser une seule opération de généralisation (exemple la simplification). Or, pour automatiser le processus de généralisation, plusieurs algorithmes doivent être mis ensemble dans une même structure logique (McMaster 1989). Ainsi, pour généraliser chaque objet ou situation de la carte, il est important de bien choisir les algorithmes et leurs paramètres et la séquence de ces algorithmes. Pour cela, les connaissances des experts du domaine doivent être formalisées, c.-à-d. traduites en règles interprétables par des machines.

2.1.3. Les différentes approches de généralisation cartographique

2.1.3.1. L'approche séquentielle

Cette approche est basée exclusivement sur l'utilisation séquentielle des algorithmes de généralisation. Ainsi, chaque opération de généralisation (ex. la simplification) est appliquée sur toutes les occurrences d'une classe d'objets sans tenir compte de la spécificité géométrique et spatiale de chaque objet. Dans une telle approche, la généralisation est traitée comme un ensemble de procédures appliquées indépendamment les unes des autres, ce qui a l'inconvénient de ne pas tenir compte des facteurs importants que sont l'environnement des objets à généraliser, l'ordre d'application des algorithmes et les objectifs spécifiques de la généralisation (Scholl *et al.* 1996).

2.1.3.2. Les approches à base de connaissances

En effet, la complexité de la généralisation cartographique fait que l'approche algorithmique est mal appropriée, d'où la nécessité d'une approche globale qui fait appel à un "raisonnement intellectuel". Ainsi, pour orchestrer le processus de généralisation (ex. le choix des algorithmes, leur séquence, et les paramètres à appliquer), les approches à base de connaissances sont souvent utilisées.

Les approches de généralisation à base de connaissance englobent les systèmes experts à base de règles (Armstrong 1991, McMaster 1991, Nickerson et Freeman 1986), les systèmes à base de contraintes (Regnauld 1998; Weibel et Dutton 1998; Ruas 1999), les systèmes multi-agent (Ruas 1999, Lamy et al. 1999, Jabeur 2005) ainsi que des techniques

d'apprentissage automatique telles que les réseaux de neurones (Allouche et Moulin 2001) et les algorithmes génétiques (Wilson et al. 2003).

2.1.3.2.1. L'approche basée sur les systèmes experts à base de règles

Les systèmes experts sont des programmes informatiques qui manipulent les connaissances symboliques et des heuristiques afin de simuler l'expert humain dans la manière dont il résout les problèmes de la vie réelle (Weiss et Kulikowski 1984). Les systèmes experts sont composés de trois parties : 1) la base de faits; 2) la base de règles; 3) et un moteur d'inférence qui permet de prendre une décision à partir de la base de faits et la base de règles. Dans un système expert de généralisation les connaissances de généralisation sont généralement présentées sous formes de règles. Ces règles sont représentées sous forme d'heuristique IF-THEN. Dans un tel système, les algorithmes sont utilisés pour réaliser la mécanique de la généralisation (ex. simplification), par contre le système expert est réservé aux décisions de haut niveau (quand et comment généraliser) (João 1998). Des systèmes de généralisation comme STRATÈGE développé par l'IGN France ou CHANGE développé par l'université de Hannover utilisent de telles logiques.

2.1.3.2.2. L'approche à base de contraintes

Comme nous l'avons mentionné ci-haut, dans un système à base de règles, la généralisation est guidée par des règles basées sur des connaissances des experts du domaine. Malheureusement, les systèmes à base de règles sont très rigides car chaque règle impose une action particulière (Beard 1991) contrairement aux systèmes à base de contraintes. Pour développer des systèmes plus flexibles, Beard (1991) propose de remplacer les règles par des contraintes. Ces contraintes sont des règles que les données doivent suivre. Les systèmes à base de contraintes représentent donc une alternative aux systèmes à base de règles (Beard 1991).

Dans un système à base de contraintes, les contraintes sont explicitement formalisées (Weibel et Dutton 1998) afin de guider le processus de généralisation. Ces contraintes permettent de prendre en compte des facteurs qui influencent la généralisation comme l'échelle, les contraintes graphiques (ex. les seuils minimaux de perception) et le but de la

carte générée. Lors d'un processus de généralisation, toutes les contraintes doivent être satisfaites et pour cela différentes opérations de généralisation peuvent être appliquées. Ces dernières années, plusieurs travaux ont été consacrés à l'utilisation des contraintes dans un processus de généralisation (Beard 1999, Regnauld 1998; Weibel et Dutton 1998; Ruas 1999).

Les contraintes proposées par Beard (1991) sont organisées en quatre types : les contraintes graphiques qui émanent des limitations liées aux supports d'affichage et à l'acuité visuelle; les contraintes structurales qui permettent de préserver les relations spatiales et sémantiques; les contraintes d'application, qui dépendent de l'objectif du produit; et les contraintes procédurales, qui contrôlent l'ordre et les interactions entre les algorithmes ainsi que l'ordre de satisfaction des contraintes. Quant à Weibel et Dutton (1998), ils proposent une classification basée sur cinq catégories de contraintes, qui sont : les contraintes graphiques, les contraintes topologiques, les contraintes structurales, les contraintes d'esthétisme et les contraintes de processus. Quant à Ruas (1999), elle propose deux groupes de contraintes : les contraintes de généralisation : celles qui incitent une situation à se généraliser (les contraintes de réduction et les contraintes graphiques) et les contraintes de maintien qui incitent une situation à préserver au mieux son information initiale (importance sémantique, cohérence, précision, forme).

Dans ses travaux, Regnauld (1998) propose quatre groupes de contraintes hiérarchisées. Ce sont les contraintes de lisibilité (perception, séparation, densité), les contraintes de respect de la forme, les contraintes d'organisation spatiale et les contraintes d'harmonie globale (répartition globale, proportion entre les tailles). La hiérarchie reflète la différence de niveaux d'abstraction auxquels ces contraintes apparaissent. Par exemple, les contraintes du premier groupe concernent un ou deux objets, tandis que celles du dernier groupe (quatrième groupe) prennent en compte la carte dans sa totalité.

L'un des problèmes les plus importants auxquels est confrontée cette approche est que l'application d'une opération de généralisation pour résoudre la violation d'une contrainte peut générer de nouveaux conflits plus importants que ceux qu'on vient de résoudre. D'où la nécessité de bien choisir l'ordre de satisfaction des contraintes. Le choix de cet ordre est un processus très complexe.

2.1.3.2.3. *L'approche multi-agent*

Un agent est un programme informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu (Jennings *et al.* 1998). L'approche agent est à la connexion de plusieurs domaines, en particulier de l'intelligence artificielle, des systèmes informatiques distribués et du génie logiciel (Chaib-draa *et al.* 2001). Selon Chaib-draa *et al.* (2001), les systèmes multi-agents sont bien indiqués pour représenter des problèmes qui ont plusieurs solutions. C'est le cas de la généralisation. Par exemple, le caractère subjectif de la généralisation fait qu'une même carte sera généralisée différemment par deux cartographes.

Ces dernières années plusieurs travaux dédiés à l'utilisation de la technologie multi-agent dans le processus de généralisation ont vu le jour (Baeijs *et al.* (1995), Lamy *et al.* 1999, Li *et al.* 2002, Duchêne 2003, Jabeur *et al.* (2005)). Dans une telle approche, une autonomie est donnée aux objets cartographiques afin de leur permettre de choisir les outils nécessaires pour leur généralisation.

L'approche agent a eu un grand écho avec le projet européen AGENT (Lamy *et al.* 1999). Dans le contexte du projet AGENT, les objets géographiques deviennent des agents géographiques. Il peut s'agir d'objets initialement présents dans la base de données à généraliser, ou d'objets créés à partir de ceux-ci, par regroupement (p. ex. un groupe de bâtiments) ou par division (ex. une portion de route) (Duchêne et Regnaud 2002). L'approche comprend deux principaux types d'agents : les agents micro (ex. un bâtiment) et les agents macro qui représentent le niveau supérieur de la hiérarchie (ex. un îlot). Les opérations de généralisation sont guidées par des contraintes. Dans cette approche, la carte est partitionnée sur la base du réseau routier (ex. les îlots). Pour chaque unité, un agent méso est créé pour contrôler toutes les interactions à l'intérieur de cette unité (ex. îlot) car aucune interaction transversale à l'intérieur de l'unité n'est prévue (par exemple entre les agents représentant les bâtiments d'un îlot). Quand on sait que le nombre d'objets dans un îlot peut être relativement élevé, on peut facilement imaginer qu'avec un tel système il est difficile d'atteindre les performances qu'exige la généralisation à la volée. D'autre part, une telle structure est inefficace en dehors des villes. Pour adopter l'approche au contexte des zones rurale, Duchêne (2003) propose des interactions transversales aux agents micro.

Dans ses travaux de doctorat, Jabeur (2006) propose un système multi-agent de génération des cartes à la volée basé sur l'utilisation combinée de la représentation multiple et des algorithmes de généralisation. Le système permet entre autres de résoudre le problème d'encombrement. Le système utilise une approche de génération progressive des cartes par couches d'intérêts créées selon les priorités de l'utilisateur. Comme dans la majorité des approches multi-agent de généralisation, un agent logiciel est attribué à chaque objet et à chaque groupe d'objets de la carte. Le regroupement des objets est exclusivement basé sur l'existence des agrégats qui proviennent de la représentation multiple. Ceci a comme conséquence, le nombre de niveaux hiérarchiques des données et les opérations de contexte (ex. l'agrégation) sont conditionnés par le nombre d'échelles contenues dans la base de données à représentation multiple. Bien que basée sur la représentation multiple, l'approche n'utilise malheureusement pas la simplification pour raffiner les données extraites de la base de données à représentation multiple comme c'est le cas dans des approches combinées de généralisation (ex. Cecconi 2002). Ce raffinement est d'autant plus important lorsque la base de données comporte un nombre réduit d'échelles.

2.1.3.3. Les approches à base de connaissances et l'acquisition des connaissances de généralisation

En théorie, une approche à base de connaissances requiert que nos connaissances du processus de généralisation soient formalisées sous une forme interprétable par les programmes informatiques. Selon Armstrong (1991), trois types de connaissances sont nécessaires pour l'implantation d'un système de généralisation à base de connaissances : les connaissances géométriques, les connaissances procédurales et les connaissances structurelles.

Pour formaliser les connaissances géométriques et spatiales, des mesures sur les objets sont nécessaires. Ces mesures sont utilisées d'une part, dans le but d'analyser une situation afin de choisir les meilleurs outils qui seront utilisés pour généraliser les objets mesurés et d'autre part, pour évaluer la qualité de la généralisation (les conséquences de la généralisation). Shea et McMaster (1989) ont énuméré sept mesures qui aideront dans le processus de généralisation. Ce sont la densité, la distribution, la longueur et la sinuosité, la forme, la distance, Gestalt et les mesures abstraites.

Selon Müller *et al.* (1995b), les connaissances de généralisation peuvent être acquises à partir de trois sources : 1) des manuscrits et des spécifications des agences cartographiques; 2) des cartes existantes manuellement ou en utilisant des techniques d'intelligence artificielle comme les algorithmes génétiques ou les réseaux de neurones (par exemple, Allouche et Moulin (2005) utilisent les réseaux de neurones de type Kohonen pour détecter les zones de haute densité sur une carte) ; 3) et des experts du domaine. Malheureusement, les caractères subjectif et holistique de la généralisation cartographique rendent la formalisation des connaissances des experts très difficile. En effet, lors de la généralisation, le cartographe prend en compte l'objet à généraliser en entier, ainsi que son environnement et applique plusieurs opérations à la fois sans toujours être capable de les distinguer explicitement. Or pour les programmes informatiques, la généralisation cartographique est une application séquentielle de plusieurs opérations. Cette difficulté de passer des connaissances des cartographes souvent floues à des règles claires et interprétables par l'ordinateur constitue l'un des handicaps majeurs qui freinent l'automatisation de la généralisation cartographique. Cette situation est l'une des raisons qui ont poussé les chercheurs à se tourner vers des approches dites interactives (aussi appelées assistées ou intelligence amplifiée).

2.1.3.4. L'approche interactive ou intelligence amplifiée

La difficulté de développer des solutions entièrement automatisées et la déception de la communauté vis-à-vis des progrès réalisés afin de formaliser les connaissances de généralisation ont amené quelques chercheurs à tourner leurs efforts vers des techniques de généralisation interactive (Müller *et al.* 1995b). Dans les années 90, certains chercheurs comme Spiess (1995) pensaient même qu'une solution 100% automatisée est une illusion complète. L'approche interactive permet la division des tâches de généralisation entre l'homme et la machine. Ainsi, lors de la généralisation, les tâches de bas niveau (ex. la simplification) sont exécutées par le système, tandis que celles de haut niveau (ex. le choix de l'objet qui sera généralisé ou le choix des paramètres qui seront appliqués lors d'une opération de généralisation) sont accomplies par l'humain. Weibel et Buttenfield (1998) furent les premiers à suggérer une telle approche.

Cette approche de généralisation a fait naître plusieurs systèmes commerciaux de généralisation interactive. Ainsi, Intergraph fut le premier producteur à proposer une solution commerciale basée sur une telle approche avec le logiciel MGE Map Generalizer en 1993 (qui a évolué pour donner naissance à DynaGEN). Deux ans plus tard, Laser-scan proposait le LAMPS2 qui est un système orienté-objet avec une gestion de la topologie. De son côté, ESRI a intégré des fonctions de généralisation dans ses outils sans pour autant créer des logiciels spécifiquement destinées à la généralisation cartographique (ex. plusieurs algorithmes de généralisation sont intégrés dans ArcGis à travers le Toolbox).

En plus de l'intervention humaine, Weibel *et al.* (1995) suggèrent aussi l'introduction de l'intelligence artificielle (ex. les algorithmes génétiques) dans les systèmes de généralisation interactive afin de pallier les problèmes posés par des approches traditionnelles à base de connaissance (ex. les systèmes experts). Ainsi, des techniques d'intelligence artificielle peuvent être utilisées pour extraire automatiquement des connaissances de généralisation qui faciliteront le processus. Dans un tel système, on peut aussi imaginer que les actions réalisées par l'expert humain peuvent être automatiquement enregistrées par le système afin de peupler une base de connaissances. Il est quand même important de noter que dans certaines littératures, il existe une certaine différence entre l'approche de généralisation interactive (ex. DynaGEN) et l'approche dite intelligence amplifiée. En effet, contrairement à l'approche interactive, l'approche dite intelligence amplifiée est supportée par différents outils (autres que les algorithmes de généralisation) qui permettent d'assister le cartographe (ex. la triangulation). Malheureusement, de nos jours, l'approche dite intelligence amplifiée n'a jamais été implémentée.

Même si dans certains cas, les algorithmes peuvent être enchaînés automatiquement, l'intervention humaine est plus que nécessaire dans une telle approche, car il est indispensable de spécifier les paramètres utilisés par les algorithmes et de vérifier les résultats. D'autre part, les systèmes de généralisation interactive tels qu'implémentés ne sont dotés d'aucun outil permettant de guider l'utilisateur sur le choix des opérations à appliquer, ni sur les algorithmes ou les paramètres à utiliser pour réaliser une opération de généralisation. En plus, dans un tel système, la détection et la résolution des conflits est

réalisée grâce à un processus d'édition interactive. Tout cela fait que pour utiliser un tel système, l'utilisateur doit être un expert du domaine.

2.2. La représentation multiple

Comme la généralisation cartographique est actuellement un processus complexe qui demande souvent l'intervention d'un expert, les cartes générées grâce à ce processus sont souvent stockées dans une base de données à représentation multiple pour des éventuelles utilisations. Dans la présente section, nous présentons le concept de la représentation multiple.

2.2.1. Définition et concepts de la représentation multiple

Plusieurs phénomènes géographiques sont intimement liés aux échelles et lorsque le niveau d'analyse change (ex. quand on passe du niveau local à une échelle régionale), de nouvelles échelles de visualisation du même phénomène sont nécessaires. C'est pourquoi il est nécessaire de modéliser l'information géographique dans différents niveaux d'abstraction (Müller *et al.* 1995b). Idéalement, pour avoir plusieurs représentations d'un même phénomène, la solution serait d'avoir une base de données à grande échelle à partir de laquelle on peut générer les cartes dérivées selon le besoin en appliquant des opérations de généralisation. Malheureusement, la généralisation reste encore une tâche longue et laborieuse. C'est pourquoi les différentes échelles dérivées sont stockées dans une base de données à représentation multiple. La représentation multiple est le stockage de plusieurs représentations d'un même objet dans une même base de données spatiales (Kilpeläine 2000; Rigaux 1994; Vangenot 1998). Selon João (1998), tant que les problèmes de la généralisation automatique ne seront pas résolus, les utilisateurs continueront à utiliser la représentation multiple.

Le plus souvent, la représentation multiple est définie à tort, comme étant uniquement la représentation multiéchelles. Cette dernière est un sous-groupe de la représentation multiple qui est un concept plus vaste. Selon Devogele *et al.* (2002), il existe deux grands groupes de représentations multiples : 1) les représentations multiples à échelle constante qui regroupent les représentations multimodes (ex. raster ou vectoriel), les représentations

multicouches et les représentations multipoints de vues applicatif; 2) et la représentation multiéchelle.

2.2.2. Les structures de gestion des représentations multiples

Les premiers travaux sur les représentations multiples ont commencé dans les années 80 sous l'initiative du centre NCGIA (National Center for Geographic Information and Analysis). Depuis cette période, de nombreuses solutions permettant de stocker plusieurs représentations d'un même territoire dans une même base de données ont vu le jour (Jones 1991; Van Oosterom 1992; Timpf et Frank 1995; Devogele *et al.* 1996; Martel 1999; Kilpelainen 2000; Bédard *et al.* 2002; Bernier 2002). La majorité de ces recherches se penchent sur la manière de gérer les représentations multiples et particulièrement les représentations multiéchelles (Bernier 2002). Ainsi, plusieurs structures de gestion de représentation multiples ont été développées ces dernières années.

2.2.2.1. La structure multicouche

Pour gérer les multireprésentations, la structure la plus répandue est la structure multicouche (layer-based). Dans une structure multicouche, chaque représentation est stockée dans une couche différente sans aucun lien explicite entre les différentes couches. Présente dans presque tous les SIG, cette structure permet une bonne navigation thématique en contrôlant les thèmes qui seront visualisés. En introduisant quelques règles (ex. ne pas afficher la couche X à partir de l'échelle Y), cette structure permet aussi de gérer la représentation multiéchelles. Cette logique est implémentée actuellement dans plusieurs SIG (ex. MapInfo). Même si cette structure est très facile à mettre en œuvre, elle s'avère inadaptée pour certaines applications (ex. les SOLAP). En effet, la rigidité d'une telle solution ne permet pas une adaptation des règles jusqu'à un niveau occurrentiel et de façon transparente.

2.2.2.2. La structure hiérarchique

Pour gérer la représentation multiéchelles, certains auteurs proposent l'utilisation des structures hiérarchiques (Timpf et Frank 1995; Jones 1991). Dans une telle approche, les

différentes représentations sont interliées. La structure hiérarchique permet de faciliter l'utilisation des zooms intelligents. Par exemple, Timpf et Frank (1995) suggèrent d'utiliser des structures de données hiérarchiques pour stocker des représentations multiples si ces représentations ne varient que par leur échelle.

2.2.2.3. Les arbres hiérarchiques ou représentation multiple implicite

Dans une telle structure, la représentation multiple n'est pas explicite car les différentes représentations de l'objet ne sont pas stockées. La géométrie initiale de l'objet est tout simplement structurée sous forme d'arbre hiérarchique. Ainsi, un index est attribué à chaque objet ou à chaque vertex d'un objet et selon l'échelle de visualisation on procède à la sélection des objets ou une partie de l'objet selon leur importance dans la structure. Par exemple, Van Oosterom et Schenkelaars (1995) proposent la combinaison de deux structures : le Binary line Generalization (BLG-tree) pour la simplification des objets et le Reactive-tree pour supporter la sélection des objets selon leur importance.

2.2.2.4. L'estampillage

Vangenot (1998) présente une approche appelée estampillage qui consiste à définir la géométrie comme un attribut spatial à n paramètres. On accède à ces paramètres selon la résolution. Elle définit en plus l'estampillage des associations et l'estampillage des attributs. L'estampillage d'attributs permet d'associer plusieurs valeurs à un même attribut et d'associer à chaque valeur la résolution correspondante. Quant à l'estampillage d'associations, il permet d'associer une résolution à chaque instance d'une association.

2.2.2.5. Les vuels

Dans ses travaux, Martel (1999) présente trois grandes catégories de multiplicités : la multiplicité géométrique, la multiplicité graphique et la multiplicité sémantique. Pour supporter ces multiplicités, Bédard et al. (2002) proposent une métastructure appelée Vuel (figure 3). Le Vuel a pour but de gérer plusieurs types de multiplicités autant multiéchelles, qu'uni-échelle et de permettre le déploiement de systèmes plus flexibles. Un Vuel représente la composante élémentaire d'une vue de la base de données de la même manière

qu'un pixel (picture element) représente la composante élémentaire d'une image numérique.

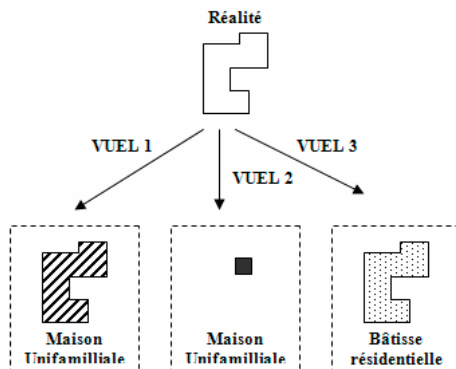


Figure 3- Trois vues servant à définir sémantiquement ou à représenter géométriquement et graphiquement une même réalité (Bédard *et al.* 2002)

2.2.2.6. Les patrons géométriques

Pour générer différentes représentations d'un même territoire, Bernier (2002) propose une méthode basée sur l'utilisation des patrons de représentations multiples. Des patrons qui correspondent à différentes représentations possibles d'un même objet sont stockés dans une base de données. Ainsi, au lieu de stocker plusieurs représentations de chaque objet, seule la géométrie initiale est stockée. À cette représentation initiale de l'objet, on associe des patrons. Ainsi, pour générer une représentation simplifiée de l'objet, on le fait remplacer par un patron qui lui correspond le mieux. Pour une bonne cohérence, on procède à certaines manipulations telles que la rotation, l'étirement, etc. Comme plusieurs objets cartographiques peuvent avoir un même patron, ceci permet de diminuer les problèmes de la redondance. La principale idée qui consiste à utiliser les patrons lors de la généralisation cartographique a été reprise dans le cadre du projet GEMURE financé par GEOIDE. Ainsi, Cardenas (2004) aborde l'aspect théorique des patrons géométriques pour permettre l'introduction de ces derniers dans un processus de généralisation automatique du réseau routier. Il définit le patron géométrique comme étant un objet géométrique avec les caractéristiques géométriques typiques de base qui sont représentatives d'un grand nombre d'occurrences d'une classe d'objets cartographique ou d'un type de primitives géométriques

et qui peut s'adapter à la géométrie de ces occurrences à différentes échelles en étant réutilisé plusieurs fois.

2.2.3. Les problèmes de la représentation multiple

Selon Devogele *et al.* (2002), la conception d'une base de données à représentation multiple pose trois types de problèmes : 1) la mise en relation des abstractions; 2) la mise en relation des instances des différentes représentations; 3) et la définition d'un processus d'appariement des instances. D'autre part, la représentation multiple est reliée aux problèmes suivants :

- La redondance à cause du fait que plusieurs représentations de chaque objet sont stockées dans la même base de données. Il faut noter que dans le cas des patrons, cette redondance est réduite au strict minimum car un patron peut être utilisé par plusieurs occurrences d'objets ;
- Le manque de flexibilité car les échelles sont prédéfinies. Ceci a pour conséquence qu'il n'est pas toujours possible d'avoir le niveau d'abstraction souhaité;
- La difficulté de maintenir la cohérence de la base de données;
- Coûts élevés pour la constitution, le maintien et la mise à jour d'une telle base de données;
- Elle hérite aussi des problèmes liés à la généralisation cartographique (ex. processus long et coûteux). En effet, le plus souvent, les différentes représentations des objets de la base de données sont générées grâce à la généralisation cartographique.

2.3 La cartographie sur demande

2.3.1 La carte sur demande

Le développement technologique des dernières années a révolutionné tous les domaines des technologies de l'information et plus particulièrement celui de l'information géographique. Ces changements ont bouleversé non seulement les outils de production cartographique,

mais aussi la manière d'utiliser l'information sur le territoire. Si dans le passé la cartographie se limitait pratiquement à la production des cartes statiques en série, la révolution de ces dernières années a facilité la multiplication d'un autre type de cartes : la carte sur demande

La carte sur demande est une carte générée selon les exigences ponctuelles de chaque utilisateur par opposition aux cartes générées en grande quantité pour des besoins génériques (ex. cartes topographiques) ou pour de larges groupes (ex. cartes routières touristiques). La principale caractéristique de la carte sur demande est que la carte générée doit être conforme aux exigences de l'utilisateur, exigences vis-à-vis de l'échelle, des thèmes à visualiser, des objets à mettre en évidence ou à filtrer, de la symbologie, du temps nécessaire à la production, etc. Cette carte peut être générée manuellement ou automatiquement, sur des supports papier ou électronique, avec ou sans généralisation, instantanément ou après un délai notable, pourvu que « ...la carte soit produite en fonction de l'ensemble des besoins et désirs des utilisateurs » (Bernier *et al.* 2003). Plusieurs applications comme la cartographie en ligne ou les SOLAP peuvent être considérées comme des applications de cartographie sur demande.

2.3.2. Exemples d'applications de cartographie sur demande

2.3.2.1. La cartographie en ligne

Le développement fulgurant d'Internet a complètement changé la manière de concevoir et de distribuer les cartes. Il n'y a pas si longtemps, les cartes étaient exclusivement distribuées en format papier. Avec des millions d'utilisateurs à travers le monde, Internet a permis de rendre accessibles au grand public les données géographiques jadis destinées à une poignée d'experts. Selon les statistiques de l'Internet World Stats (2006), le nombre d'internautes a dépassé le milliard en 2006, ce qui constitue plus de 16 % de la population mondiale. Ce développement d'Internet d'une part, et l'engouement envers les données géographiques d'autre part, ont facilité l'apparition d'une multitude de sites proposant des contenus à caractère cartographique. Actuellement, le nombre des sites donnant accès aux données cartographiques tourne autour de dizaines de milliers (Peterson 2003b). Certains

sites de cartographie en ligne comme Map24 ou MapQuest génèrent des millions de cartes par jour. À titre d'exemple, en 2004, le site MapQuest générait plus de 400 millions de cartes par mois.

Malgré le nombre phénoménal des cartes générées grâce aux services de cartographie en ligne, actuellement, le potentiel du Web vis-à-vis des données cartographiques est mal utilisé (Van Elzakker et Koussoulakou 1997). La majorité des sujets qu'on trouve sur le web ont des composantes spatiales, hélas, ces sujets ne sont pas présentés de la bonne manière : c'est le cas par exemple des sites à contenus touristiques qui contiennent le plus souvent des simples descriptions textuelles et des photographies (Van Elzakker 2001).

Bien avant l'introduction du Web, les cartes étaient distribuées à travers Internet grâce au protocole FTP. Par contre, les fichiers téléchargés nécessitaient des traitements particuliers (ex. désarchivage) et des outils de visualisation adaptés, ce qui limitait leur utilisation. Mais, l'apparition des fureteurs, des formats plus adaptés à la visualisation des données graphiques sur le web (ex. SVG) et des langages spécifiques aux données géographiques (ex. GML) a grandement favorisé le développement de la cartographie sur le Web. En plus, Internet permet non seulement de distribuer les cartes à un large public, mais aussi d'incorporer interaction et animation dans l'affichage (Peterson 2003a).

Du point de vue production cartographique, le Web présente plusieurs avantages. Contrairement aux cartes traditionnelles (cartes papier), le Web offre l'accessibilité et l'actualité (Van Elzakker 2001). Une fois publiées sur le web, les données peuvent être consultées de presque tous les coins du monde et ne sont désormais plus restreintes aux spécialistes, mais peuvent être consultées par tout internaute, aussi néophyte soit-il (Bernier et al. 2003). De plus, les mises à jour sont répercutées automatiquement sur le Web contrairement aux cartes papier qui nécessitent une réédition, processus très long et coûteux.

À ses premiers balbutiements, la cartographie sur le web était limitée à l'affichage de cartes statiques de format matriciel (Bernier et al. 2003). Pour remédier à ces lacunes, les cartes en format vectoriel et interactives ont vu le jour. Selon Kraak (2001), les cartes sur le web peuvent être divisées en deux groupes : les cartes statiques et les cartes dynamiques.

Chacune des cartes de ces deux groupes peut être interactive ou non interactive (view only). Actuellement on trouve de plus en plus d'applications interactives. Cependant, la majorité de ces applications sont très rigides en terme de personnalisation. Néanmoins, ces dernières années quelques applications plus flexibles en termes de définition des besoins des utilisateurs ont vu le jour (Gbei et al. 2003; Bernier et al. 2005).

Pour efficacement prendre en compte les besoins des utilisateurs, Bernier *et al.* (2005) ont développé UMapIT. Cette application permettant la création des cartes sur demande est basée sur la structure Vuel. L'architecture de l'application est montrée sur la figure 4.

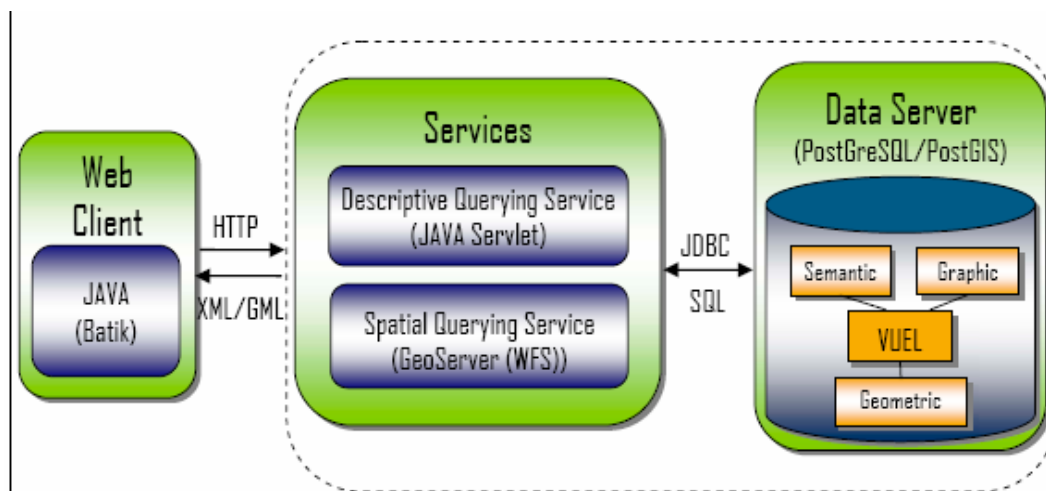


Figure 4- Architecture d'UMapIT (Bernier et al. 2005)

De leurs côtés, Gbei *et al.* (2003) ont développé un système de cartographie sur le Web appelé SIGERT. Ce système d'information récréotouristique permet à l'utilisateur de créer des cartes selon ses besoins. SIGERT est basé sur la technologie multi-agent et permet aussi de résoudre les problèmes d'encombrements (Jabeur 2006) qui surviennent suite à une diminution de l'échelle de visualisation grâce à des opérations de généralisation à la volée. De plus, le système est capable de générer des données sur des appareils à taille réduite comme les PDA et les téléphones cellulaires (Boulekruche 2005).

Bien qu'un grand pas ait été accompli ces dernières années en vue d'améliorer les outils et les contenus des cartes sur demande sur le Web, il n'en demeure pas moins que la majorité des applications sont actuellement incapables de générer des données à des échelles

arbitraires. Pour cela, ces applications doivent intégrer entre autres des outils efficaces de généralisation à la volée.

2.3.2.2. Les SOLAP

Le SOLAP (Spatial OnLine Analytical Processing) est une plate-forme visuelle supportant l'exploration et l'analyse spatio-temporelle faciles et rapides des données selon une approche multidimensionnelle à plusieurs niveaux d'agrégation via un affichage cartographique, tabulaire ou en diagramme statistique (Bédard 1997). Le SOLAP est un mariage des outils SIG et des outils OLAP. L'OLAP (OnLine Analytical Processing) est une catégorie de logiciels axés sur l'exploration et l'analyse rapide des données selon une approche multidimensionnelle à plusieurs niveaux d'agrégation (Caron 1998). L'OLAP permet l'exploration des données lors d'une prise de décision. Pour aider les décideurs qui ne sont pas nécessairement des spécialistes dans le domaine des données spatiales ou informatique, ces outils (les SOLAP et le OLAP) doivent être très rapides et faciles d'utilisation. En effet, l'utilisateur de ces technologies doit être capable d'interroger les données sans pour autant maîtriser les langages de requête comme le SQL ou sans connaître la structure des données sous-jacente (Marchand *et al.* 2001). Le SOLAP est basé sur une structure multidimensionnelle (ex. structures en étoile, en flocon ou mixtes). Le SOLAP est doté d'opérateurs comme le forage ou le pivot qui permettent à l'utilisateur de naviguer dans les données

Selon Bédard *et al.* (2005), il faut distinguer les *applications SOLAP* des *technologies SOLAP*. Une *technologie SOLAP* est une technologie générique construite spécialement pour offrir des fonctions SOLAP de base ou plus avancées sans nécessiter d'efforts de programmation. Une *application SOLAP* est une application métier qui fournit à l'utilisateur un certain nombre de fonctionnalités de type SOLAP et qui peut être construite soit avec la technologie SOLAP, soit avec des combinaisons de technologies non-SOLAP (ex. SIG et OLAP) et du code de programmation maison, ou soit avec d'autres technologies (ex. bibliothèques en Java). Les SOLAP utilisent des opérations (ex. zoom out) qui peuvent nécessiter une adaptation du niveau de détail des objets cartographiques représentés. Cette adaptation peut être possible grâce aux opérations de généralisation. Mais compte tenu du

caractère interactif des SOLAP, cette généralisation doit être à la volée. Donc les SOLAP peuvent éventuellement bénéficier des travaux de la présente thèse.

2.4. La généralisation à la volée

Comme nous l'avons précédemment mentionné, pour permettre aux utilisateurs de produire des cartes aux échelles désirées, l'idéal serait d'avoir une seule base de données à grande échelle et de dériver les plus petites échelles automatiquement selon les besoins grâce aux processus de généralisation cartographique. Mais, considérant l'aspect interactif des applications (la cartographie en ligne et les SOLAP) utilisées actuellement pour accéder aux données géographiques, tous les changements doivent être réalisés quasi instantanément. Donc, les changements d'échelle doivent être effectués en temps réel, ce qui exigerait une généralisation à la volée (on-the-fly generalization).

2.4.1. Définition

La généralisation cartographique à la volée peut être définie comme étant « la création, en temps réel, d'un produit cartographique approprié à son échelle et à son but, suite à la demande d'un utilisateur, et effectuée à partir d'une base de données à plus grande échelle » (Weibel *et al.* 2002). La généralisation à la volée doit générer des cartes conformes aux besoins des utilisateurs et tout cela sans aucun délai notable. En effet, selon plusieurs études (InfoWorld 1999; Wonnacott 2000) le temps d'attente sur le Web ne doit pas dépasser les 10 secondes, sans quoi, l'utilisateur quittera la page. Des récentes études démontrent que pour des applications comme le commerce en ligne, ce temps d'attente ne doit pas dépasser 4 secondes (Young et Smith 2006). D'autre part, les résultats de la généralisation (la carte généralisée) ne doivent pas être stockés car ceci introduirait de la redondance dans les données (Van Oosterom et Schenkelaars 1995).

2.4.2. Les différentes approches

En l'état actuel des connaissances, il existe trois grands axes de recherche dans le domaine de la généralisation à la volée : l'approche orientée-traitements, l'approche multi-représentations et l'approche combinée (Weibel *et al.* 2002).

2.4.2.1. L'approche orientée-traitements

Cette approche est basée exclusivement sur l'utilisation des algorithmes pour accomplir certaines opérations de généralisation. Comme la généralisation à la volée exige un temps de réponse quasi instantané, des méthodes basées sur des attributs précalculés sont parfois utilisées afin d'accélérer le processus. Ainsi, certains auteurs proposent d'utiliser l'algorithme de simplification Douglas et Peucker (1973) afin de structurer les vertex d'une ligne (Van Oosterom et Schenkelaars 1995 ; Edward et Kevin 2002). Ces vertex sont stockés dans une structure hiérarchisée. Lors d'un changement d'échelle, les vertex sont sélectionnés selon leur importance dans la structure.

La principale limitation d'une telle approche est qu'elle permet seulement de représenter des objets individuels qui ont été pré-généralisés indépendamment les uns des autres, ce qui a comme conséquence l'apparition des conflits entre des objets voisins lors de la diminution d'échelle (Jones *et al.* 2000). D'autre part, les méthodes basées sur des attributs précalculés dépendent le plus souvent d'un seul algorithme de simplification, l'algorithme Douglas et Peucker (1973). Par contre, une telle approche s'avère flexible et non redondante.

2.4.2.2. L'approche multi-représentation

L'approche orientée multi-représentation se base exclusivement sur une base de données à représentation multiple (cf. 2.2.). Cette approche repose essentiellement sur une structuration adéquate de la base de données spatiale (Weibel *et al.* 2002). Et lorsque l'utilisateur envoie sa requête, le système choisit l'échelle la plus proche de celle demandée. La majorité des applications de cartographie en ligne utilisent une telle approche. Dans, les sections précédentes, une revue des principaux travaux réalisés ces dernières années dans le domaine de la représentation multiple a été présentée. Un des inconvénients d'une telle approche est son manque de flexibilité car toutes les échelles sont fixées à l'avance, ce qui ne permet pas de satisfaire la majorité des utilisateurs en termes de personnalisation de la carte. D'autre part, elle hérite de tous les problèmes liés à la représentation multiple (voir section 2.2.3). Par contre, cette approche offre une grande performance en terme de temps de réponse car toutes les opérations de généralisation ont été réalisées à l'avance.

2.4.2.3. L'approche combinée

L'impossibilité d'automatiser entièrement la généralisation et les problèmes liés à la représentation multiple (difficulté de produire les données, la mise en correspondance entre les occurrences, les problèmes de la mise à jour, etc.) ont favorisé l'émergence d'une troisième approche : l'approche combinée (Cecconi *et al.* 2002; GiMoDig 2004; Jabeur 2006). Cette approche est basée sur la combinaison de l'approche orientée traitement et l'approche orientée multireprésentation.

Ainsi, Cecconi *et al.* (2002) proposent une approche basée sur la combinaison de la représentation multiple et de la généralisation à la volée. La base de données multiéchelles est utilisée en tant que base pour la généralisation des cartes et inclut un minimum de deux jeux de données représentant les objets à différents niveaux d'échelle ou de détail avec des liens de correspondance entre les différents niveaux. Ainsi, suite à une requête de l'utilisateur, on choisit l'échelle la plus appropriée, et en cas de besoin on raffine le résultat grâce à l'utilisation de certains algorithmes de généralisation.

Le projet européen GiMoDig (GiMoDig 2004) dont le but est de développer des méthodes pour une intégration et une généralisation en temps réel utilise une approche de généralisation à la volée semblable à celle proposée par Cecconi *et al.* (2002). La méthode adoptée dans ce projet consiste à utiliser la représentation multiple comme étant un processus de pré-généralisation et de pré-harmonisation.

Dans ses travaux de thèse, Jabeur (2006) présente aussi une approche combinant la représentation multiple et la généralisation à la volée. Cette approche est basée sur la technologie multi-agent (voir section 2.1.3.2.3).

Dans une telle approche, la généralisation est facilitée par la représentation multiple. En effet, lors de la généralisation cartographique, il est toujours préférable de passer par une échelle intermédiaire lors d'un grand changement d'échelle afin de faciliter le processus. Par exemple pour générer une carte à l'échelle 1 :250 000 à partir d'une carte à l'échelle 1 :20 000, il est souhaitable, sinon indispensable de générer d'abord une échelle intermédiaire entre les deux (par exemple 1 : 50 000 ou 1 :100000) et générer l'échelle 1 :250 000 à partir de cette dernière. L'avantage d'une telle approche est de minimiser les problèmes liés d'une

part à l'approche orienté-traitement (ex. le temps de traitement élevé) et d'autre part à l'approche basée sur la représentation multiple (ex. manque de flexibilité). Les travaux de la présente thèse s'inscrivent aussi dans cette catégorie.

2.5. Conclusion

Dans la présente section, nous avons présenté une revue des différents travaux en lien avec la présente thèse. Entre autres, nous avons passé en revue les principaux travaux dans les domaines de la généralisation cartographique, de la représentation multiple, des cartes sur demande, de la cartographie en ligne et de la généralisation à la volée. De cette revue des principaux travaux, les faits suivants ont été constatés :

- Le développement fulgurant d'Internet a permis la démocratisation des données géographiques. Malheureusement, les outils actuels qui permettent d'accéder à ces données ne permettent pas aux utilisateurs de personnaliser les données d'une manière flexible. Entre autres, ces outils ne permettent pas de générer des cartes à des échelles arbitraires;
- Pour générer des cartes à différentes échelles, un processus de généralisation est le plus souvent requis. Mais, compte tenu du caractère interactif des technologies utilisées pour accéder aux données, cette généralisation doit être réalisée à la volée;
- La généralisation à la volée exige un temps de réponse quasi instantané et une automatisation complète. Malheureusement, en l'état actuel, la généralisation cartographique reste encore une tâche de longue haleine qui nécessite fréquemment l'intervention humaine;
- Vu les problèmes liés à la généralisation automatique, les chercheurs se sont tournés vers la représentation multiple qui permet de conserver plusieurs représentations d'un même objet dans une même base de données. Elle est donc devenue une solution de remplacement à la généralisation cartographique. Par contre la représentation multiple présente plusieurs inconvénients, entre autres,

le manque de flexibilité (car les échelles sont fixes), les difficultés de produire les échelles nécessaires, la mise en correspondance des objets des différentes échelles, la propagation des mises à jour, et la redondance des données.

- Pour surmonter d'une part les problèmes liés à la généralisation cartographique et d'autre part ceux de la représentation multiple, les récents travaux tentent de combiner les deux approches afin de profiter des quelques avantages qu'elles présentent. Les travaux de la présente thèse s'inscrivent aussi dans ce courant.

2.6. Références

- Allouche M.K. et Moulin B., "Reconnaissance de patterns par réseaux de neurones : application à la généralisation cartographique". *Revue Internationale de Géomatique*, Vol. 11(2), p. 251-279, 2001.
- Armstrong M.P., "Knowledge classification and organization". Bittenfield B., McMaster R. and Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p.86-102.
- Bader M., *Energy Minimization Methods for Feature Displacement in Map generalization*. Thèse de Doctorat, Université de Zürich, Suisse, 2001, 201 p. Baeijs C., Demazeau Y. et Alvares L., "Application des systèmes multi-Agents à la généralisation cartographique". *Proceeding of 3èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-agents, AFCET&AFIA*, Chambery, 1995, p. 163-176.
- Barrault M. et Bader M., "Les algorithmes de déplacement". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p.287-302.
- Beard K.M., "Constraints on rule formation". Bittenfield B., McMaster R. and Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 121-135.
- Beard M.K., "How to survive on a single detailed database". *Proceeding of 8th International Symposium on Computer-Assisted Cartography*, Baltimor, Maryland, *AUTO-CARTO*, 1987, p. 211-220.
- Bédard Y., Spatial OLAP, Vidéoconférence. *Proceeding of 2ème Forum Annuel sur la R-D, Géomatique VI: Un Monde Accessible*, Montréal, 1997.
- Bédard Y., Bernier E. et Devillers R., "La métastructure VUEL et la gestion des représentations multiples". Ruas A. (Ed), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p. 149-162.

- Bédard Y., Proulx M.-J. et Rivest S., "Enrichissement du OLAP pour l'analyse géographique : exemples de réalisations et différentes possibilités technologiques". *Proceeding of Première Journée Francophone sur les Entrepôts de Données et l'Analyse en Ligne*, Lyon, Soumis en 2005.
- Bernier E., *Utilisation de la Représentation Multiple comme Support à la Génération de Vues de Bases de Données Géospatiales dans un Contexte SOLAP*. Mémoire de Maîtrise, Université Laval, 2002, 89 p.
- Bernier E., Bédard Y. and Hubert F., "UMapIT: an on-demand Web mapping tool based on a multiple representation database". *Proceeding of 8th ICA WORKSHOP on Generalisation and Multiple Representation*, A Coruña, Spain, 2005.
- Bernier E., Bédard Y. et Lambert M., "Cartographie sur demande sur le Web et bases de données multidimensionnelles. De la personnalisation par couche cartographique à la personnalisation par occurrences". *Revue Internationale de Géomatique*, Vol. 13(3), p. 339-359, 2003.
- Boulekrouche B., *Génération Cartographique à la Volée pour les Terminaux Mobiles de Type PDA*. Mémoire de Maîtrise, Université Laval, 2005.
- Brassel K.E., "Strategies and data models for computer-aided generalization". *International Yearbook of Cartography*, Vol. 25, p. 11-29, 1985.
- Brassel K.E. and Weibel R., "A review and conceptual framework of automated map generalization". *International Journal of Geographical Information Systems*, Vol. 2(3), p. 229-244, 1988.
- Buttenfield B., "Treatment of cartographic lines". *Cartographica*, Vol. 22(2), p. 1-26, 1985.
- Buttenfield B. and McMaster R., "Preface". Buttenfield B., McMaster R. and Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. xiii-xvi.
- Cardenas A., *Utilisation des Patrons Géométriques comme Support à la Généralisation Automatique*. Mémoire de Maîtrise, Université Laval, 2004, 110 p.
- Caron P. Y., *Étude du Potentiel de l' OLAP pour Supporter l'Analyse Spatio-temporelle*. Mémoire de Maîtrise, Université Laval, 1998, 132 p.
- Cecconi A., *Integration of Cartographic Generalization and Multi-scale Databases for Enhanced Web Mapping*. Thèse de Doctorat, Université de Zurich, 2003, 155 p.
- Cecconi A., Weibel R. et Barrault M., "Improving automated generalisation for on-demand Web mapping by multiscale Database". *Proceeding of Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada, 2002.
- Chaib-draa B., Jarras I. et Moulin B., "Systèmes multiagents : principes généraux et applications". Briot J.P. et Demazeau Y. (Eds.), *Agents et Systèmes Multiagents*, Paris, Hermès, 2001, p 27-60.
- Cromley R. G., "Hierarchical methods of line simplification". *Cartography and GIS*, Vol. 18(2), p. 125-131, 1991.

- Cuenin R., *Cartographie Générale*. Paris, Eyrolles, 1972, 324 p.
- Delucia A. and Black T., "A comprehensive approach to automatic feature generalization". *Proceeding of 13th International Cartographic Association Conference*, Morelia, Mexico, 1987.
- Devogele T., Badard T. et Libourel T., "La problématique de la représentation multiple". Ruas A. (Ed.), *Généralisation et Représentation Multiple*. Paris, Hermes science publications, 2002, p. 55-74.
- Devogele T., Trevisan J. et Raynal L., "Processus de constitution d'une base de données multi-échelles". *Revue Internationale de Géomatique*, Vol. 6(2-3), p. 249-263, 1996.
- Dougenik J., "Whirpool: A geometric processor for polygone coverage data". *Proceedings of AUTO-CARTO*, Vol. 4, p. 304-311, 1980.
- Douglas D.H. and Peucker T.K., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". *The Canadian Cartographer*, Vol. 10(2), p. 112-122, 1973.
- Duchêne C., "Coordinative agents for automated generalisation of rural areas". *Proceeding of Fifth Workshop on Progress in Automated Map Generalisation*, ICA, Commission on Map Generalisation, Paris, 2003.
- Duchêne C. et Regnauld N., "Le modèle AGENT". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Hermes science publications, 2002, p. 369-385.
- DynaGEN : user support. Disponible sur <http://imgssupport.intergraph.com/dynagen/>.
- Eckert M. and Joerg W., "On the nature of maps and map logic". *Bulletin of the American Geographical Society*, Vol. 40(6), p. 344-351, 1908.
- Edward P.F.C. and Kevin K.W.C., "On multi-scale display of geometric objects". *Data&Knowledge Engineering*, Vol. 40, p. 91-119, 2002.
- Gbei E., Moulin B., Cosma I., Jabeur N. et Delval N., "Conception d'un prototype de service Web géolocalisé appliqué à l'industrie récréotouristique". *Revue Internationale de Géomatique*, Vol. 13, p. 375-395, 2003.
- GiMoDig: Geospatial info-mobility service by real-time data-integration and generalisation. Disponible sur <http://gimodig.fgi.fi>.
- Gothic LAMPS2, Product Feature Overview. Disponible sur <http://www.laser-scan.com/products/lamps2.htm>.
- Grünreich D., "Development of computer-assisted generalization on the basis of cartographic model theory". Müller J.C., Lagrange J.P. and Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*, Bristol: Taylor & Francis, 1995, p. 47-55.
- Hardy P., Hayles M. and Rewell P., "Clarity- A new environment for generalisation using agents, java, XML and topology". *Proceeding of ICA Workshop on Generalisation and Multiple Representations*, Paris, 2003.

- InfoWorld, The speed of business: If your pages are slow, your customers will go. Disponible sur <http://www.infoworld.com/articles/op/xml/00/09/11/000911opsavvy.html>, 1999.
- Internet World Stats, World Internet Users and Population Stats. Disponible sur <http://www.internetworldstats.com/stats.htm>, 2006.
- Jabeur N., *A Multi-Agent System for On-The-Fly Web Map Generation and Spatial Conflict Resolution*. Thèse de Doctorat, Université Laval, 2006, 275 p.
- Jenk G.F., "Geographic logic in line generalization". *Cartographica*, Vol. 26(1), p. 27-43, 1989.
- Jennings N. R., Wooldridge M. and Sycara K., "A roadmap of agent research and development". *International Journal of Autonomous Agents and Multi-Agents Systems*, Vol. 1(1), p. 7-38, 1998.
- João E.M., *Causes and Consequences of Map Generalisation*. London, Bristol, PA, Taylor & Francis, 1998, 266 p.
- Jones C., "Database architecture for multi-scale GIS". *Auto-Carto*, Vol. 6(10), p. 1-14, 1991.
- Jones C., Abdelmoty A. I., Lonergan M. E., van der Poorten P. and Zhou S., "Multi-scale spatial database design for online generalisation". *Proceeding of 9th Spatial Data Handling Symposium*, Beijing, p. 7b.34-7b.44, 2000.
- Kilpelainen T., "Maintenance of multiple representation database for topographic data". *The Cartographic Journal*, Vol. 37(2), p. 101-107, 2000.
- Kraak M.J., "Setting and needs for web cartography". Kraak M.-J. and Brown A. (Eds.), *Web Cartography: Developments and Prospects*, Londres, Taylor & Francis, 2001, p. 1-7.
- Lamy S., Ruas A., Demazeau Y., Jackson M., Mackaness W. and Weibel R.. "The application of agents in automated map generalisation". *Proceeding of 19th International Cartographic Conference*, Ottawa, Vol. 2, p. 1225-1234, 1999.
- Lang T., "Rules for the robot draughtsmen". *The Geographical Magazine*, Vol. 42(1), p. 50-51, 1969.
- Li M., Zhou S. and Jones C., "Multi-agent systems for Web-based map information retrieval". *Proceeding of GIScience 2002: Geographic Information Science, Second International Conference, Boulder, CO, Lecture Notes in Computer Science*, Vol. 2478, p. 161-180, 2002.
- Mackaness W., "An algorithm for conflict identification and feature displacement in automated map generalization". *Cartography and GIS*, Vol. 21, p. 219-232, 1994.
- Mailing D., *Measurements from Maps- Principles and Methods of Cartometry*, Oxford, Pergamon Press, 1989, 450 p.
- Maozhen L., Sheng Z. and Jones C., "Multi-agent systems for Web-based map information". Egenhofer M.J. and Mark D.M. (Eds.), *GIScience*, Verlag Berlin, 2002, p. 161-180.

- Marchand P., Bédard Y. and Edwards G., "A hyper cube-based method for spatio-temporal exploration and analysis". *Soumis à Geoinformatica* en May 2001.
- Martel C., *Développement d'un Cadre Théorique pour la Gestion des Représentations Multiples dans les Bases de Données Spatiales*. Mémoire de Maîtrise, Université Laval, 1999, 189 p.
- McMaster R., "Automated line generalization". *Cartographica*, Vol. 24(2), p. 74-111, 1987.
- McMaster R., "The integration of simplification and smoothing algorithms in line generalization". *Cartographica*, Vol. 26(1), p. 101-121, 1989.
- McMaster R., "Conceptual framework for geographical knowledge". Buttenfield B., McMaster R. and Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 21-39.
- McMaster R. and Shea K.S., "Cartographic generalization in digital environment: A Framework for implementation in GIS". *Proceeding of GIS/LIS'88*, San Antonio, Texas, 1998, p. 240-249.
- McMaster R. and Shea K.S., "Generalization in digital cartography". *Association of American Geographers*, Washington, D.C., 1992.
- Morrison J.L., "A theoretical framework for cartographic generalization with emphasis on the process of symbolisation". *International Yearbook of Cartography*, Vol. 14, p. 115-127, 1974.
- Müller J. C., "Fractal dimension and inconsistencies in cartographic line representations". *The Cartographic Journal*, Vol. 23, 123-130.
- Müller J. C., "Generalization of spatial database". Maguire D., Goodchild M. and Rhind D. (eds.), *Geographical Information Systems Principles and Applications*, Vol. 1, Harlow, Logman, p. 457-475.
- Müller J. C., "Building knowledge thanks for rule based generalization". *Proceeding of International Cartographic Conference*, Bournemouth, 1991, p. 257-256.
- Müller J.C., Lagrange J. P. and Weibel R., "Editors' Preface". Müller J.C., Lagrange J.P. and Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. ix.
- Müller J.C., Weibel R., Lagrange J.P. and Salgé F., "Generalization: state of the art and issues". Müller J.C., Lagrange J.P. and Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. 3-18.
- Nickerson B.G., "Automatic cartographic generalization for linear features". *Cartographica* Vol. 25, p. 15-66, 1988.
- Nickerson B.G. and Freeman H., "Development of a rule-based system for automatic map generalization". *Proceeding of 2 th International Symposium on Spatial Data Handling*, Seattle, Washington, 1986, p. 537-556.
- Opheim H., "Fast Reduction of a Digitized Curve". *Geo-Processing*, Vol. 2, p. 33-40, 1982.

- Perkal J., "An attempt at objective generalization". *Michingam Inter-University Community of Mathematical Geographers*, Discussion paper 10, University of Michigan, 1966.
- Peterson M., "Maps and the Internet: An introduction". Peterson M. (Ed.), *Maps and the Internet*, London, Elsevier, 2003, p.1-16.
- Peterson M., "Preface". Peterson M. (Ed.), *Maps and the Internet*, London, Elsevier, 2003, p.ix-x.
- Ratajski L., "Phénomènes des points de généralisation". *International Yearbook of Cartography*, Vol. 7, p. 143-151, 1967.
- Regnault N., *Généralisation du Bâti: Structure Spatiale de Type Graphe et Représentation Cartographique*. Thèse de Doctorat, Université de Provence - Aix-Marseille, 1998, 191 p.
- Reumann K. et Witkam A.P.M., "Optimizing curve segmentation in computer graphics". *Proceeding of International Computing Symposium*, Amsterdam, 1974, p. 467-472
- Rigaux P., "La représentation multiple dans les systèmes d'information géographique". *Revue Internationale de Géomatique*, Vol. 4(2), p.137-164, 1994.
- Ruas A., "A method for building displacement in automated map generalisation". *International Journal of Geographical Information Science*, Vol. 12(8), p. 789-803, 1998.
- Ruas A., *Modèles de Généralisation de Données Géographiques à Base de Contraintes et d'Autonomie*. Thèse de Doctorat, Université de Marne-la-Vallée, 1999, 323 p.
- Ruas A., "Les problématiques de l'automatisation de la généralisation". Ruas, A. (Ed.), *Généralisation et Représentation Multiple*, Hermes science publications, 2002, p. 75-90.
- Scholl M., Voisard A., Peloux J.P., Raynal L. et Rigaux P., *SGBD Géographiques Spécificité*. Paris, International Thomson Publishing, France, 1996, 185 p.
- Shea K.S., "Design consideration for an artificially intelligent system". Battenfield B., McMaster R. and Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 3-20.
- Shea K.S. and McMaster R., "Cartographic generalization in a digital environment: when and how to generalize". *Proceeding of 9th International Symposium on Computer-Assisted Cartography, AUTO-CARTO 9*, Baltimore, Maryland, 1989, p. 56-67
- Spiess E., "The need of generalisation in a GIS environment". Müller J. C., Lagrange J. P. and Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. ix.
- Timpf S. and Frank A.U., "A multi-scale data structure for cartographic objects". *Proceeding of 17th International Cartographic Conference*, Barcelone, Spain, 1995.
- Topfer F. and Pillewizer W., "The principles of selection". *The Cartographic Journal*, Vol. 3(1), p. 10-16, 1966.

- Van Elzakker C.P.J.M., "Use of maps on the Web". Kraak M.-J. and Brown A. (Eds.), *Web Cartography: Developments and Prospects*. Londres, Taylor & Francis, 2001, p. 21-36.
- Van Elzakker C.P.J.M. and Koussoulakou A., "Maps and their use on the Internet". *Proceeding of 18th International Cartographic Conference, ICC'97*, Stockholm, p. 620-627, 1997.
- Van Oosterom P., "A storage structure for a multi-scale database: the reactive-tree". *Computer, Environment and Urban Systems*, Vol. 16, p. 239-247, 1992.
- Van Oosterom P. and Schenkelaars V., "The development of an interactive multi-scale GIS". *International Journal of Geographical Information Systems*, Vol. 9(5), p. 489-507, 1995.
- Vangenot C., "Représentation multi-résolutions, concepts pour la description des bases de données avec multi-représentations". *Revue Internationale de Géomatique*, Vol. 8(1-2), p. 121-147, 1998.
- VisValingam M. and Whyatt J.D., "Line generalisation by repeated elimination of points". *Cartographic Journal*, Vol. 30(1), p. 46-51, 1993.
- Wang Z. and Müller J.C., "Line generalization based on analysis of shape characteristics". *Cartography and GIS*, Vol. 25(1), p. 3-15, 1998.
- Weibel R., Bernier E., Bédard Y. et Cecconi A., "La généralisation à la volée". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p. 319-335.
- Weibel R. and Bittenfield B., "Map design for geographic information systems". *Proceeding of GIS/LIS*, San Antonio, Texas, 1988, p. 350-359.
- Weibel R. and Dutton G., "Constraint-based automated map generalization". *Proceeding of Spatial Data Handling*, Vancouver, Canada, 1998, p. 214-224.
- Weibel R. and Dutton G., "Generalising spatial data and dealing with multiple representations". Longley P.A., Goodchild M.F., Maguire D.J. and Rhind D.W. (Eds.), *Geographic Information Systems- Principles and Technical Issues*. John Wiley & Sons, 1999, p. 125-155.
- Weibel R., Keller S. and Reichenbacher T., "Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive system and computational intelligence". *Proceeding of COSIT '95: Spatial Information Theory - A Theoretical Basis of GIS, Lecture Notes in Computer Science 988*, Berlin, Springer-Verlag, 1995, p. 139-156
- Weiss S.M. and Kulikowski C.A., *A Practical Guide to Designing Expert Systems*. Rowman & Littlefield, Totowa, N.J. USA, 1984, 174 p.
- Wilson I.D., Ware M. and Ware A., "Reducing graphic conflict in scale reduced maps using a genetic algorithm". *Proceeding of ICA Workshop on Map Generalization*, Paris, 2003.

Young J., Smith S., Akamai and JupiterResearch Identify '4 Seconds' as the New Threshold of Acceptability for Retail Web Page Response Times. Disponible sur http://www.akamai.com/html/about/press/releases/2006/press_110606.html.

Zona Research Inc., The Economic Impacts of Unacceptable Web Site Download Speeds. Disponible sur http://www.webperf.net/info/wp_downloadspeed.pdf, 2006.

Chapitre 3 Introduction du concept des patrons géométriques

INTRODUCTION DU CONCEPT DE PATRON GÉOMÉTRIQUE ET APPLICATION AUX BÂTIMENTS AFIN DE FACILITER LEUR GÉNÉRALISATION CARTOGRAPHIQUE À LA VOLÉE

Auteurs : Sabo M.N., A. Cardenas, Y. Bédard, E. Bernier

(GEOMATICA Vol. 59 (3), 2005, p. 295-311)

Remarque : *Cet article présente le concept des patrons géométriques et leur application aux bâtiments. Les différents tests qui ont permis de déterminer la pertinence d'utiliser des patrons géométriques dans un processus de généralisation à la volée ont aussi été réalisés dans le présent article. Les patrons géométriques sont la composante principale des SGO. C'est pourquoi, les autres articles (Sections 4 et 5) qui traitent des SGO présentent aussi un résumé de la notion des patrons géométriques.*

Résumé de l'article : *Malgré d'importants progrès réalisés ces dernières années, la généralisation cartographique automatique reste encore une tâche difficile qui demande le plus souvent une intervention humaine. Pour cette raison, les solutions actuelles basées sur l'exploitation d'algorithmes ne peuvent pas à elles seules répondre aux exigences des nouvelles applications comme la cartographie sur le Web qui nécessitent une généralisation à la volée. Cet article présente le concept de patrons géométriques, qui sont des formes communes à plusieurs objets cartographiques, et propose leur utilisation au sein d'un processus de généralisation cartographique. Ainsi, au lieu de généraliser un objet cartographique (ex. : un bâtiment) par le biais d'algorithmes de généralisation afin d'assurer une bonne lisibilité de la carte, un patron géométrique est utilisé pour remplacer l'objet. Pour cela, le patron géométrique est ajusté à la géométrie de l'objet par l'intermédiaire d'opérations très simples comme la rotation et le déplacement. Un même*

patron peut servir à plusieurs centaines d'occurrences d'objets et seul un nombre minimum de paramètres distinguant chaque relation patron-objet est nécessaire. Les premiers tests démontrent que les patrons géométriques présentent une performance de quatre à vingt-sept fois supérieure à celle des approches de généralisation utilisant des algorithmes.

3.1. Introduction

Le développement technologique des dernières années a révolutionné tous les domaines des technologies de l'information en général et celui de l'information géographique en particulier. Ces changements ont bouleversé non seulement les outils de production cartographique, mais aussi la manière d'utiliser l'information sur le territoire. Si, dans le passé, la cartographie se limitait pratiquement à la production des cartes statiques en série, la révolution de ces dernières années a facilité la multiplication d'un autre type de cartes : la carte sur demande. La carte sur demande est une carte générée selon les exigences ponctuelles de chaque utilisateur par opposition aux cartes produites en grande quantité pour des besoins génériques. En fait, cette dernière est devenue l'objet d'une attention particulière ces dernières années, car l'apparition des nouvelles possibilités informatiques (ex. : services Web, interopérabilité) et géomatiques (ex. : suivi GPS en temps réel) permet d'entrevoir une fabrication simple, rapide et flexible de cartes personnalisées, et ceci, directement par le demandeur.

Internet est le média qui a le plus révolutionné la manière d'utiliser les cartes. En effet, avec l'avènement d'Internet et l'apparition de la cartographie sur le Web, les produits cartographiques qui étaient auparavant des produits destinés en général aux spécialistes sont devenus aujourd'hui des produits de consommation grand public. Ainsi, des sites de cartographie en ligne génèrent plusieurs millions de cartes par jour pour des utilisateurs situés partout sur la planète. À titre d'exemple, on peut citer le site de cartographie en ligne MapQuest, avec 28.5 millions de visiteurs et 400 millions de cartes personnalisées par mois [MapQuest 2004]. Si le nombre de cartes générées sur le Web est impressionnant, il n'en demeure pas moins que le nombre de sites proposant des contenus cartographiques reste relativement bas. En effet, une simple comparaison du nombre de sites qui composent le Web par rapport à ceux dédiés à la cartographie nous donne une idée de cette situation. Il est donc clair que du point de vue cartographique, le potentiel du Web n'est pas pleinement

exploité. Cette situation peut être imputable en partie à la relative jeunesse de la cartographie sur le Web d'une part et aux difficultés de produire les données cartographiques requises d'autre part. Cependant, avec l'apparition des serveurs et des données cartographiques gratuits, on peut s'attendre à une future explosion des sites à caractère cartographique.

À ses premiers balbutiements, la cartographie sur le Web était limitée à l'affichage des cartes statiques de format matriciel [Weibel *et al.* 2002]. Ces types d'applications sont caractérisés par un manque de flexibilité, car en général ils permettent seulement d'agrandir ou de réduire les images sans aucune possibilité de faire des requêtes spatiales ni d'adapter le contenu selon le goût des utilisateurs. Afin de contrer le manque d'interactivité des cartes en format matriciel, la cartographie vectorielle sur le Web a fait son apparition au milieu des années 1990 [Bernier *et al.* 2003]. Bien qu'elle présente une amélioration considérable par rapport à la cartographie matricielle, la cartographie vectorielle sur le Web telle qu'elle est déployée en général présente plusieurs limites qui réduisent son efficacité. Ainsi, les fonctions que comportent la majorité des applications de cartographie sur le Web ne permettent pas d'adapter les contenus des cartes selon les besoins réels des utilisateurs. Or, cette adaptation de contenu est plus que nécessaire lors d'un changement d'abstraction. Par exemple, lors d'une réduction d'échelle (cf. *zoom out*), la capacité de la carte à représenter son contenu peut être dépassée, ce qui a pour conséquence la détérioration de la lisibilité de la carte et l'apparition de conflits entre différents éléments de la carte (ex. : la superposition de certains objets). En cartographie conventionnelle, cette situation est remédiée en appliquant des opérations de généralisation pour simplifier le contenu de la carte et l'adapter à la nouvelle situation.

Pour la cartographie sur le Web, le processus de généralisation peut également être utilisé pour adapter le contenu des cartes selon l'échelle de visualisation. Mais, compte tenu de la nature interactive du Web, du nombre très élevé d'utilisateurs ayant des besoins différents de personnalisation et du fait que ces besoins ne sont pas connus à l'avance, il est impossible de fabriquer au préalable les différentes cartes que les utilisateurs aimeraient obtenir. Ceci implique que la généralisation et toutes les autres opérations qui affectent les cartes à afficher doivent être traitées à la volée.

La généralisation cartographique à la volée est définie comme étant « la création, en temps réel, d'un produit cartographique approprié à son échelle et à son but, suite à la demande d'un utilisateur, et effectuée à partir d'une base de données à plus grande échelle » [Weibel *et al.* 2002]. Contrairement à la généralisation conventionnelle qui est principalement assistée, la généralisation à la volée exige, d'un côté, un temps de réponse quasi instantané et de l'autre qu'il n'y ait aucune intervention humaine lors du processus. En effet, lors d'une généralisation à la volée, le temps d'attente doit être compatible avec la bande cognitive de Newell [1990] qui est inférieure à dix secondes, temps au-delà duquel les utilisateurs ont tendance à quitter la page Web sans attendre le résultat de leur requête.

Depuis plus de deux décennies, plusieurs recherches ont été dédiées à l'automatisation de la généralisation cartographique. Ces recherches ont permis de définir plusieurs opérations ainsi que des algorithmes pour automatiser ces opérations. Plusieurs algorithmes de simplification existent et a priori, tous ces algorithmes pourraient être utilisés lors d'une généralisation à la volée [Jenk 1989; Lang 1969; VisValingam et Whyatt 1993; Wang et Müller 1998]. Cependant, la contrainte temps qu'impose la généralisation à la volée entrave l'utilisation de certains d'entre eux. Pour ces raisons, seules des opérations faciles à réaliser et pour lesquelles il existe des algorithmes performants sont exécutées à la volée (Ex. la simplification à l'aide de l'algorithme Douglas et Peucker [1973]). La plupart des solutions de généralisation à la volée basées sur l'utilisation d'algorithmes de généralisation sont fondées sur l'algorithme de simplification Douglas et Peucker [1973], car il supporte les attributs précalculés, accélérant ainsi les traitements. Pour supporter la simplification et la sélection en même temps lors de la généralisation à la volée, Van Oosterom et Schenkelaars [1995] proposent la combinaison de deux structures : le Binary line Generalization (BLG-tree) et le Reactive-tree. Le BLG-tree est une structure qui stocke les calculs faits à partir de l'algorithme de simplification Douglas et Peucker [1973]. Le Reactive-tree a les mêmes propriétés que le R-tree et permet la sélection des objets selon leur importance. Cependant, d'un côté, cette approche est tributaire des progrès réalisés par la généralisation automatique et de l'autre, selon certaines recherches ([Sester et Brenne 2004]), cet algorithme n'est pas approprié à la généralisation des bâtiments.

L'automatisation de la généralisation est encore dans sa phase de recherche, car de nos jours, il n'existe pas de solutions entièrement automatisées pour bien généraliser une carte à des échelles arbitraires.

Pour contourner ce problème, la représentation multiple est devenue une alternative de choix. Cette approche est basée sur la conservation de plusieurs représentations d'un même objet au sein d'une même base de données. Ces dernières années, plusieurs travaux sur la représentation multiple ont été réalisés. On peut citer, entre autres, Martel [1999], Bédard *et al.* [2002], Devogele *et al.* [2002], Vangenot [1998] ou Timpf [1998]. Souvent, ces différentes représentations sont obtenues grâce à la généralisation manuelle ou semi-automatique, puis elles sont stockées sous forme de couches à raison d'une couche par échelle. Mais, selon Devogele *et al.* [1996], la création d'une base de données multi-échelle pose trois types de problèmes : la mise en correspondance entre différentes abstractions, la mise en correspondance entre différentes instances de représentation d'un même objet et la définition d'un processus d'appariement géométrique des instances. De plus, les bases de données à représentation multiple possèdent d'autres inconvénients, tels que la redondance de données et la difficulté de produire les différentes échelles nécessaires à leur constitution. En l'absence d'un processus d'appariement efficace, le maintien à jour des différentes représentations d'un même territoire devient une tâche difficile et coûteuse. En plus, l'approche utilisant la représentation multiple est souvent caractérisée par de grands sauts de contenus entre deux échelles consécutives. Ceci empêche parfois les utilisateurs d'avoir des données qui correspondent exactement à leurs besoins. Pour surmonter ce problème et beaucoup d'autres qui minent les deux approches décrites ci-haut, une autre solution dite « approche combinée » a vu le jour.

L'approche combinée exploite à la fois les algorithmes de généralisation et la représentation multiple. Dans la solution proposée par Cecconi *et al.* [2002], une base de données multi-échelle est utilisée en tant que base pour la généralisation des cartes et inclut un minimum de deux jeux de données représentant les objets à différents niveaux d'échelle ou de détail, avec des liens de correspondance entre ces niveaux. Ainsi, suite à une requête de l'utilisateur, on choisit l'échelle la plus appropriée et en cas de besoin on raffine le résultat grâce à l'utilisation d'algorithmes de généralisation. Le projet Européen GiMoDig

[GiMoDig 2004], dont le but est de développer des méthodes pour une intégration et une généralisation en temps réel, utilise une approche de généralisation à la volée semblable à celle proposée par Cecconi *et al.* [2002]. La méthode utilisée dans ce projet consiste à utiliser la représentation multiple comme processus de pré-généralisation et de pré-harmonisation.

La combinaison des algorithmes de généralisation et de la représentation multiple minimise l'effort de généralisation, car la transition entre l'échelle source et celle demandée par l'utilisateur est moins importante. Par contre, pour que l'effort de généralisation soit suffisamment minimisé, la base de données multi-échelle doit comprendre plusieurs échelles, ce qui nous ramène encore aux problèmes de la représentation multiple. Enfin, ces solutions utilisent des raisonnements qui sont les mêmes pour toutes les occurrences d'une classe d'objets (ex. : tous les bâtiments) même si sur le plan individuel le choix de l'échelle de départ pour la généralisation n'est pas le plus approprié, l'algorithme retenu pour la classe d'objets n'est pas le plus efficace pour la géométrie de l'occurrence, etc.

Pour améliorer l'utilisation combinée des algorithmes de généralisation et de la représentation multiple, le projet **GEMURE** (Généralisation multi-échelle et représentation multiple pour la production et la diffusion efficace de données cartographiques, dans le cadre du Réseau de centres d'excellence **GEOIDE**) a été initié en 2002. L'objectif de ce projet était de développer des nouvelles méthodes et outils combinant des opérations de généralisation cartographique et de la représentation multiple pour faciliter la généralisation dans un contexte de cartographie sur demande. L'approche utilisée dans le projet GEMURE est basée sur une architecture multi-tiers. En premier lieu, un entrepôt de données multi-échelle puis des comptoirs de données (datamarts) spécialisés par thème et plage d'échelles sont produits. L'entrepôt de données repose essentiellement sur la représentation multiple. Plusieurs approches de recherche sont abordées et testées au sein de cette architecture. Parmi ces approches, l'une exploite le concept des patrons géométriques. Ces patrons sont des objets avec des formes génériques capables de remplacer plusieurs objets de la carte lors d'un processus de généralisation.

Dans le cadre du projet GEMURE, le concept théorique des patrons géométriques du réseau routier a été défini. Dans cet article, nous proposons d'enrichir ce concept pour

l'appliquer au cas des bâtiments et de le tester avec des données réelles. L'article commence par un survol des différentes notions générales de patrons. Par la suite, nous présentons la méthodologie permettant d'appliquer le concept des patrons aux bâtiments. Les résultats des tests réalisés sont ensuite présentés. En conclusion, nous présentons les perspectives et les défis restants à relever.

3.2. Les différentes notions de patron

3.2.1. Les patrons dans la vie de tous les jours

Dans la vie de tous les jours, il est possible de remarquer une récurrence dans le temps et/ou dans l'espace, de plusieurs phénomènes. Quel que soit le domaine d'application, la récurrence est à la base de l'idée des patrons. Tout notre environnement social, culturel, économique, biologique, génétique, psychologique, éducatif, etc., se développe à partir de patrons qui nous permettent d'établir toute action, circonstance et conséquence de notre vie. Par exemple, les patrons génétiques qui déterminent les caractéristiques physiques dans une famille, les patrons culturels et religieux que nous inculquent nos parents, les patrons éducatifs mis en oeuvre par les institutions où nous effectuons nos études, les patrons utilisés pour la production de vêtements en série, etc. Un patron est donc un modèle que nous pouvons suivre pour effectuer quelque chose.

Malgré la diversité des domaines d'application des patrons, peu de travaux sont dédiés à ce concept. L'un des premiers travaux majeurs consacrés aux patrons est celui réalisé par l'architecte Alexander *et al.* [1977]. Il définit le patron comme une solution à un problème récurrent pour lequel chaque patron décrit le problème ainsi que la solution, de telle façon que cette solution peut être utilisée plusieurs fois sans pour autant le faire deux fois de la même manière.

3.2.2. Les patrons informatiques

Suite aux travaux d'Alexander *et al.* [1977], plusieurs chercheurs (ex. [Coad 1992 ; Gamma *et al.* 1995]) ont emprunté le concept de patrons pour l'appliquer au domaine de l'informatique. Ainsi, Gamma *et al.* [1995] proposent 23 motifs de conception dont chacun

permet de résoudre un problème récurrent de la programmation-objet. Les patrons dans le domaine de l'informatique sont le résultat de l'évolution des algorithmes, des structures de données et des objets. Ils se présentent comme des moyens simples, rapides et efficaces de simplifier les processus informatiques grâce à l'application d'une solution récurrente telle quelle ou légèrement modifiée. Bien que ceux-ci ne soient pas utilisés comme une structure visible, ils sont déployés de façon implicite. Malgré le fait qu'il existe une multitude de types de patron pour la conception de logiciels, il n'existe pas de cadre formel pour leur utilisation.

3.2.3. Les patrons en géomatique

Quoique le lien entre l'informatique et la géomatique actuelle soit très étroit, le concept des patrons est très peu utilisé dans le domaine de la production des données géographiques. Les patrons en géomatique servent surtout à la conception des logiciels dédiés à ce domaine. Par contre, l'abondance des phénomènes récurrents dans notre environnement géographique et leur caractère hiérarchique devrait faciliter l'utilisation des patrons dans le domaine de la géomatique en général et celui de la généralisation cartographique en particulier. De même, des méthodes de caricature utilisant des gabarits en forme de lettres de l'alphabet (comparables à des patrons) sont utilisées depuis des années avec succès par l'agence cartographique Danoise pour représenter les fermes à l'échelle 1 : 50 000 [Regnauld 2002]. À l'aide d'algorithmes d'appariement de patrons simples, Rainsford et Mackaness [2002] proposent d'automatiser la sélection de ces gabarits.

Parmi les rares travaux traitant de l'utilisation des patrons dans un processus de généralisation cartographique, notons ceux réalisés par Mackaness et Edwards [2002]. En se basant sur les travaux de Zacks et Tversky [2001] traitant de la structure des événements utilisée en psychologie, ils ont mené des réflexions sur l'importance d'introduire le concept des patrons dans un processus de généralisation cartographique à cause du caractère multi-échelle de ces derniers.

3.2.4. Les patrons géométriques

S'inspirant des travaux sur les patrons précédemment décrits, Cardenas [2004] définit les concepts théoriques des patrons géométriques qui sont utilisés pour appuyer la généralisation automatique. Il définit un *patron géométrique* comme étant un objet géométrique avec les caractéristiques géométriques typiques représentatives d'un grand nombre d'occurrences d'un phénomène cartographique ou d'une primitive géométrique. Une *primitive géométrique* est l'élément géométrique atomique servant à construire un objet cartographique ou un patron géométrique. Ces patrons sont destinés à être utilisés à différentes échelles. Ainsi, Cardenas [2004] propose lors d'une diminution d'échelle, de remplacer l'objet de la carte par un patron (donc, une représentation simplifiée) au lieu de le généraliser à l'aide des algorithmes de généralisation. Les patrons sont préalablement définis et stockés dans une base de données. Ils sont ensuite ajustés par de simples opérations (la rotation, le déplacement et changement de grandeur) pour les adapter à la géométrie des objets à remplacer. L'auteur distingue deux types de patrons géométriques : les patrons géométriques simples composés d'une ou plusieurs instances d'une même primitive et les patrons géométriques complexes qui sont des agrégats de patrons simples. Bien que ces travaux soient réalisés dans une perspective de généralisation des éléments du réseau routier, le concept général des patrons décrit par Cardenas [2004] est applicable aux bâtiments. Mais pour cela, un enrichissement de ces concepts s'avère nécessaire, car de par leur géométrie et la manière de les généraliser, les bâtiments diffèrent beaucoup des éléments du réseau routier. Théoriquement, les patrons géométriques présentent deux grands avantages : minimiser le volume de la base de données (un patron géométrique est utilisé pour plusieurs occurrences et à différentes échelles) et accélérer le processus de généralisation. Donc les présents travaux proposent d'adapter les concepts des patrons aux bâtiments, de les enrichir et de les tester sur des données réelles. Les enrichissements nécessaires seront apportés tout au long de la partie méthodologie (section 3.3) et les résultats des tests seront présentés à la section 3.4.

3.3. Enrichissement et application du concept de patrons géométriques à la cartographie et à la généralisation des bâtiments

Cette section présente les enrichissements du concept proposé, ainsi que la méthodologie suivie pour utiliser les patrons géométriques dans un processus de généralisation de bâtiments. Certaines étapes sont préalables à l'emploi des patrons géométriques lors de la généralisation des bâtiments. Celles-ci se divisent en deux grandes phases : 1) la création de la base de données contenant les patrons géométriques et 2) l'association des patrons aux occurrences de bâtiments. Ces deux grandes phases sont présentées et détaillées aux sections suivantes. Elles ont été testées et validées dans un contexte réel avec un jeu de données provenant de la ville de Québec et comprenant plus de 2 800 bâtiments. À la fin de cette section, un diagramme résumant les différentes étapes de cette méthodologie est présenté.

3.3.1. Construction de la base de données de patrons géométriques

La création de la base de données des patrons géométriques se divise en 5 grandes étapes : 1) la détection des différents patrons de forme à partir de l'analyse d'une carte existante, 2) la sélection des primitives et la classification des patrons de forme, 3) la sélection du patron commun de forme le plus détaillé pour chaque type de primitive, 4) la construction du patron géométrique initial, et enfin 5) la généralisation du patron géométrique initial pour générer les patrons géométriques dérivés et le peuplement de la base de données des patrons. Dans la suite de l'article, le terme « patron de forme » sera employé pour les patrons détectés soit dans notre environnement physique, soit sur une carte. Quant au terme « patron géométrique », il fera référence aux patrons construits et stockés dans la base de données et qui seront utilisés lors d'un processus de généralisation.

3.3.1.1. L'analyse des cartes existantes et détection des patrons de forme

Le but de cette phase est de déceler les différents patrons de forme des bâtiments à partir du matériel cartographique existant (cartes, photographies aériennes, etc.). Dans le cadre de

nos travaux, une carte de la ville de Québec à l'échelle 1K a été utilisée. Parmi tous les jeux de données à notre disposition, celui-ci représentait l'échelle la plus grande. À partir de cette carte, les différents patrons potentiels ont été identifiés en analysant visuellement les différents bâtiments tout en prêtant une attention particulière à la répétitivité de leur forme. Pour faciliter la détection, des cartes aux échelles 20K et 50K de la même zone ont aussi été utilisées. Ces cartes supplémentaires ont été nécessaires, car l'échelle de la carte initiale (1K) était très grande et certains patrons de bâtiments n'apparaissaient qu'à des échelles plus petites. Les patrons de forme détectés sont représentés sous forme de croquis faits à la main. Une fois les différents patrons recensés, un raffinement a été effectué pour éliminer les patrons les moins représentatifs, c'est-à-dire ceux qui ne s'appliquent qu'à très peu d'occurrences. Évidemment, il n'y a pas de limite au nombre de patrons retenus. Cependant, il ne faut pas oublier l'essence même de l'approche par patron qui est de simplifier la tâche de généralisation et de diminuer le volume de la base de données. Un trop grand nombre de patrons ne ferait que complexifier les processus de création et d'utilisation des patrons.

3.3.1.2. Choix des primitives et classification des patrons de forme

L'élément de base pour la construction d'un patron géométrique est la primitive géométrique. Ici, il faut noter la différence entre la primitive géométrique et la primitive de patron. La primitive géométrique est l'élément de base qui sert à la construction d'un patron géométrique. Quant aux primitives d'un patron, elles constituent les différentes parties d'un patron de forme. Les patrons de forme recensés lors de l'analyse des cartes existantes (cf. 3.3.1.1) sont minutieusement analysés pour détecter les différentes primitives de patron qui les composent. Pour cela, chaque patron de forme doit être considéré comme étant une juxtaposition de plusieurs formes primaires comme des rectangles, des cercles ou des triangles. Par exemple, le patron d'un bâtiment peut être considéré comme un assemblage de plusieurs rectangles. La conception des objets comme un arrangement de parties est actuellement partagée par plusieurs chercheurs dans le domaine de la reconnaissance de formes [Fortin et Rousseau 1992].

L'étape de détermination des primitives est capitale, car elle influence directement la forme des patrons géométriques qui composeront la future base de données. Cette étape est guidée par la recherche d'un équilibre entre, d'un côté, le souci d'avoir un nombre restreint de primitives, donc de patrons géométriques et, de l'autre côté, le souci de représenter adéquatement les différentes occurrences de bâtiments.

Pour éviter des confusions lors de la manipulation des futurs patrons géométriques, une primitive ne doit pas pouvoir être obtenue par modification d'une autre. Par exemple, un carré peut facilement être obtenu en modifiant la grandeur d'un rectangle et vice versa. Pour éviter cette situation, une seule des deux formes géométriques serait conservée comme primitive de patron; le rectangle, le carré, ou encore une forme plus générique comme un parallélogramme droit.

Une fois les primitives déterminées, on procède à un regroupement des patrons de forme détectés lors de la phase d'analyse (cf.3.3.1.1) sur la base de la forme des primitives retenues. Par exemple, tous les patrons de bâtiments à angles droits formeront un seul groupe, celui des patrons formés à partir des instances d'un parallélogramme droit.

3.3.1.3. La sélection du patron commun le plus détaillé

Pour chaque groupe de patrons, son patron commun le plus détaillé est déterminé. *Le patron commun le plus détaillé* est le patron de forme qui possède le plus grand nombre d'instances de primitive et à partir duquel on peut construire tous les autres patrons du groupe (c.-à-d. formés par la même primitive) seulement en retirant certaines primitives. Le patron commun le plus détaillé sert de modèle lors de la construction du patron géométrique initial (cf. 3.3.1.4). Il est en quelque sorte l'union des formes de tous les patrons de forme du même groupe, ce qui permet de construire un seul patron géométrique initial à partir duquel on va générer tous les autres patrons géométriques du groupe. La figure 5 montre un groupe de patrons formés à partir d'une même primitive où le patron encerclé est le patron commun le plus détaillé de ce groupe. Le choix du patron commun le plus détaillé est une phase cruciale, car la grandeur et la finesse de la base de données, ainsi que la qualité des données générées en dépendent.

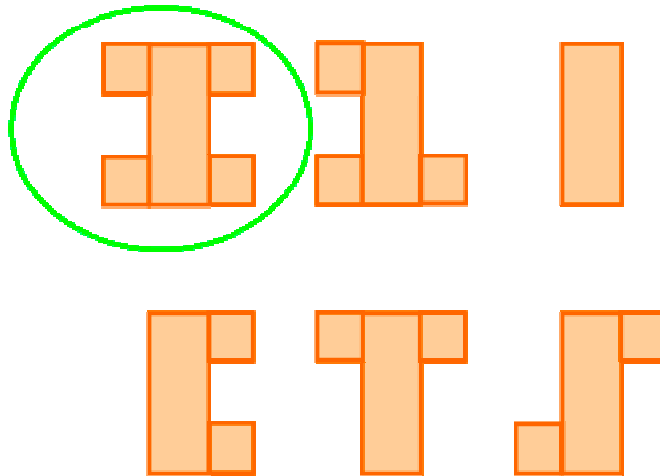


Figure 5- Un groupe de patrons et leur patron commun le plus détaillé

Pour remédier aux erreurs pouvant se glisser lors de la phase de détection des patrons (par exemple l'omission de certains patrons), le nombre des instances de primitive du patron commun le plus détaillé peut être renforcé. Par exemple, si notre patron commun le plus détaillé est composé de 4 instances de la même primitive, un autre patron à 6 instances peut être choisi pour avoir une plus grande marge de manœuvre lors de la création des patrons géométriques dérivés. La majoration du nombre des instances de primitive permet de couvrir un territoire plus vaste que celui sur lequel les analyses ont été réalisées et de limiter le nombre d'itérations lors de la création de la base de données de patrons. Par contre, une majoration substantielle du nombre d'instances d'une primitive dans un patron peut considérablement augmenter la taille de la base de données.

3.3.1.4. La construction du patron géométrique initial

Pour chaque patron commun le plus détaillé on détermine le nombre d'instances de primitives de même que la manière dont ces dernières sont connectées. À défaut d'outils spécialisés pour la construction des patrons, les outils habituels de saisie de données vectorielles sont ensuite utilisés pour créer le patron géométrique initial qui est une reproduction du patron commun le plus détaillé. Ce patron géométrique initial ne se doit pas d'être une copie conforme du patron commun le plus détaillé parce que, d'un côté, le patron géométrique se doit générique et de l'autre, les opérations qui permettent de l'ajuster

au besoin doivent être simples. En premier moment, les patrons initiaux simples sont créés. Ces derniers sont des patrons initiaux formés par des instances d'une seule primitive.

La construction du patron initial simple des bâtiments a été réalisée en deux étapes :

- La création de la primitive principale. La *primitive principale* est la primitive centrale autour de laquelle seront rattachées les autres instances de primitive, *les primitives secondaires du patron*;
- La création des primitives secondaires. Ces dernières sont des primitives de patron qui sont greffées à la *primitive principale*. Dans un patron de bâtiment, il n'y a pas de limite au nombre de primitives secondaires rattachées à une primitive principale. Dans le cas d'un patron simple, la primitive principale et les primitives secondaires sont des instances d'une même primitive.

Pour assurer un bon degré de maniabilité, chaque primitive secondaire d'un patron géométrique est dotée des paramètres suivants :

- *Des points d'ancrage* qui sont les points d'accrochage de la primitive secondaire par rapport à la primitive principale;
- *La grandeur de la primitive*. Pour une primitive rectangulaire, la grandeur est déterminée par la longueur et la largeur;
- *L'orientation de la primitive* qui est caractérisée par l'angle formé par la primitive principale et ladite primitive. Pour la majorité des bâtiments, cet angle est égal à 90^0 .

Le but recherché lors de l'utilisation des patrons géométriques n'est pas d'adapter ces derniers pour avoir une géométrie exacte de l'objet à remplacer, mais de faire en sorte que les proportionnalités entre les différentes primitives soient préservées afin d'assurer une certaine consistance visuelle. Pour cela, des coordonnées exactes de chaque point d'ancrage ou des grandeurs (longueur, largeur) exactes de chaque primitive ne sont pas nécessaires. Les coordonnées de chaque point d'ancrage sont des coordonnées relatives par rapport au

côté de la primitive principale à laquelle elles sont ancrées. Les grandeurs doivent être présentées proportionnellement à la grandeur de la plus grande primitive. La figure 6 montre comment les différents paramètres (points d'ancrage, grandeur et angle de connexion) ont été déterminés. Une grille a été utilisée pour montrer les différentes coordonnées relatives pour chaque primitive secondaire. Les paramètres prélevés sont montrés dans le tableau 1. La première colonne du tableau présente les primitives secondaires par leur numéro. Les autres colonnes représentent les différents paramètres. Par exemple, le premier point d'ancrage de la primitive numéro 1 est 6 unités de mesure, donc à 6/10 du haut de la primitive principale. Sa longueur est la moitié de la plus longue primitive (primitive numéro 2). Le signe négatif est utilisé pour distinguer des primitives secondaires opposées.

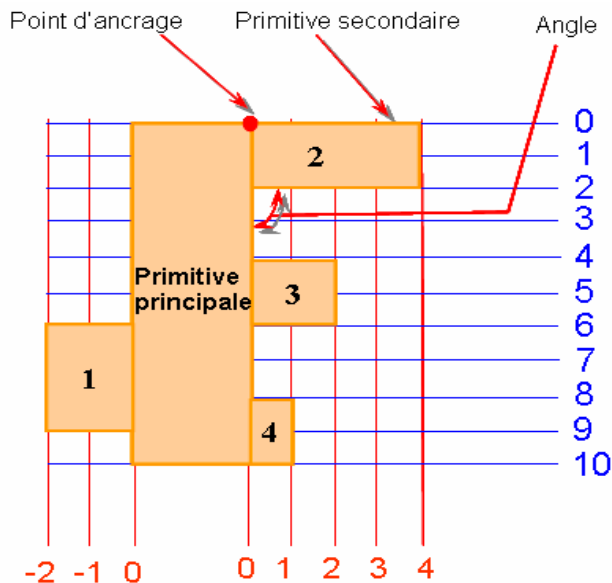


Figure 6- Patron géométrique d'un bâtiment

Tableau 1- Paramètres d'un patron géométrique de bâtiments

Primitive secondaire	Points d'ancrage	Longueur	Angle
1	6,9	-2	90°
2	0,2	4	90°
3	4,6	2	90°
4	8,10	1	90°

Les patrons géométriques simples permettent de représenter la majorité des bâtiments, surtout des bâtiments simples comme des maisons. Par contre, certains bâtiments (ex. plusieurs édifices administratifs) possèdent des formes très complexes que les patrons géométriques simples sont incapables de représenter. Donc, pour représenter ces types de bâtiments avec une fidélité géométrique suffisante, des patrons géométriques avec des formes plus complexes sont nécessaires. Les patrons complexes sont créés de deux façons : 1) en combinant plusieurs primitives (les instances de plusieurs primitives), ou 2) en combinant des patrons géométriques déjà existants. Ces derniers peuvent être des patrons simples ou complexes. Tous les concepts de patron géométrique sont résumés dans un diagramme de classe UML sur la figure 7.

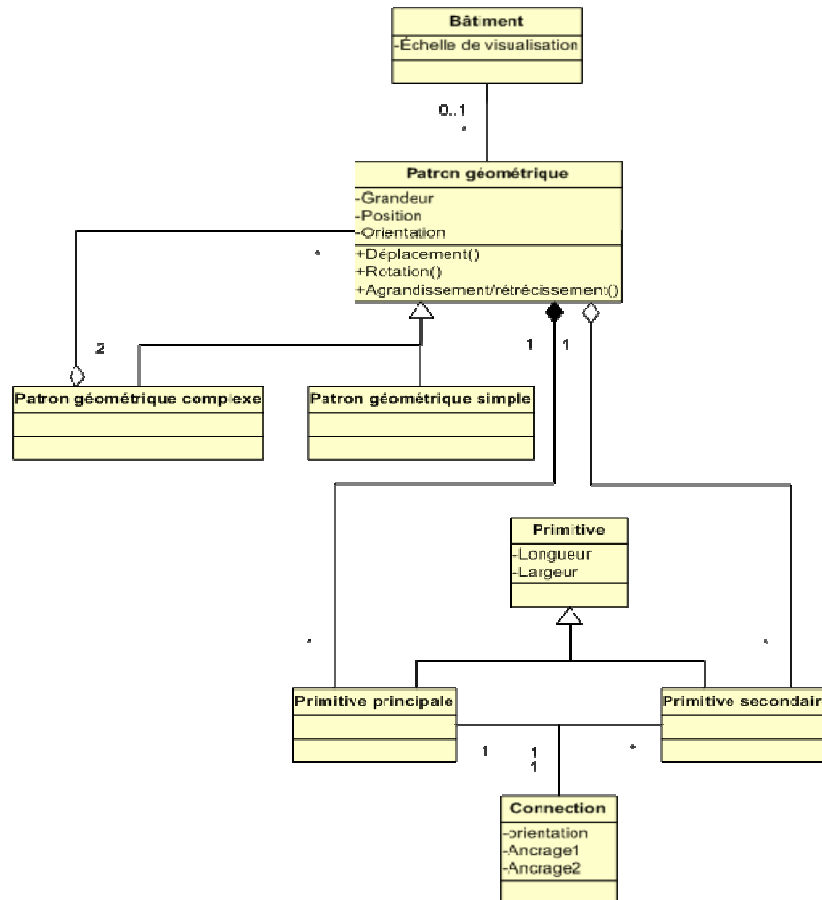


Figure 7- Diagramme de classe UML représentant le patron géométrique

3.3.1.5. La création des patrons géométriques dérivés et peuplement de la base de données des patrons géométriques

Il existe deux opérations pour simplifier un patron géométrique en diminuant le nombre des instances de primitive. Ce sont :

- L'élimination des primitives secondaires;
- Et la combinaison des primitives secondaires.

Cette phase consiste à appliquer des opérations de généralisation afin de générer les différentes variantes simplifiées d'un même patron géométrique initial. La structure en composants (juxtaposition de primitives) des patrons assure une généralisation contrôlée, donc un résultat prévisible. Lors de cette phase, les opérations de simplification des patrons

géométriques et particulièrement « l'élimination des primitives secondaires » seront utilisées pour générer les patrons dérivés. En appliquant récursivement cette opération, on simplifie le patron le plus détaillé jusqu'à retrouver la forme de la primitive géométrique ayant servi à la construction du patron initial (c.-à-d. la primitive principale). Par exemple, si nous avons un patron géométrique initial formé par des instances d'une primitive rectangulaire, l'opération élimination de primitive est appliquée récursivement pour éliminer toutes les primitives secondaires (cf. figure 11). À chaque itération, on élimine une seule primitive à la fois. Chaque patron géométrique généré est appelé patron géométrique dérivé. Une fois tous les patrons géométriques dérivés générés, tous les patrons redondants sont éliminés. Tous les patrons géométriques pouvant être obtenus à l'aide de la rotation ou du miroitement d'un autre patron géométrique sont également considérés comme redondants.

Chaque patron géométrique détaillé et ses différents patrons dérivés sont stockés dans une base de données sous forme d'arbre hiérarchique dont la profondeur détermine le nombre de primitives et par conséquent le niveau de détail du patron. Cette structure arborescente du patron facilite la navigation à travers différents niveaux d'abstraction d'un même patron. La structure hiérarchisée est souvent utilisée pour stocker les données géographiques afin de minimiser la redondance et de faciliter l'accès [Van Oosterom et Schenkelaars 1995].

3.3.2. Saisie des patrons géométriques

Cette phase consiste à sélectionner des patrons géométriques à partir de la base de données créée à l'étape précédente (cf. 3.3.1.5) et de les associer aux occurrences de bâtiments. Ainsi, pour chaque bâtiment sur la carte, des patrons géométriques sont sélectionnés afin que chacun corresponde à la forme de l'objet à un degré d'abstraction donné. Mais en réalité, un seul patron sera associé à l'objet, celui dont sa géométrie se rapproche plus de celle de l'objet. La structure hiérarchique des patrons stockés dans la base de données facilite l'accès aux autres niveaux d'un patron. Quand on associe un patron à un bâtiment de la carte, on détermine en même temps l'échelle à partir de laquelle ce patron est utilisé, de même que les paramètres qui permettent d'ajuster les patrons à la géométrie de cet objet, comme l'emplacement, la grandeur ou l'orientation de l'objet. Toutes ces informations

(l'identifiant du patron, l'échelle à partir de laquelle le patron est utilisé, les paramètres qui permettent d'ajuster le patron à la géométrie de l'objet) sont stockées dans une base de données de paramètres utilisée lors de la généralisation.

La sélection du patron correspondant à la géométrie du bâtiment est réalisée par un opérateur humain en comparant visuellement la géométrie du bâtiment à celle des différents patrons géométriques de la base de données. Ainsi, pour chaque bâtiment, on stocke l'identifiant du patron retenu.

L'échelle à partir de laquelle un bâtiment est remplacé par son patron géométrique est déterminée de la façon suivante : on détermine interactivement le détail minimal (E) qui sera simplifié pour que cet objet prenne la forme de son patron géométrique. Pour cela, n'importe quel outil GIS possédant des fonctions de mesure de distance peut être utilisé. Dans notre cas, ArcGIS (ESRI) a été utilisé. Le dénominateur de l'échelle (K) est déterminé par l'équation suivante :

$$K = (E * 1000) / \delta \quad \text{Équation 1;}$$

Où δ est le seuil de perception utilisé en généralisation cartographique manuelle, il est d'environ 0.3 mm. La valeur de E a été multipliée par 1000 pour la convertir en mm.

Par exemple, soit un bâtiment qui se rapproche de la forme 'L' (figure 8) et dont un détail d'une longueur $E = 1.5$ m doit être enlevé pour que ce bâtiment ait la forme 'L'. Le dénominateur de l'échelle (K), est calculé à partir de l'équation 1 :

$$K = 1.5 * 1000 / 0.3 = 5000.$$

Donc, à l'échelle 1 : 5000, ce bâtiment est remplacé par le patron en forme 'L'.

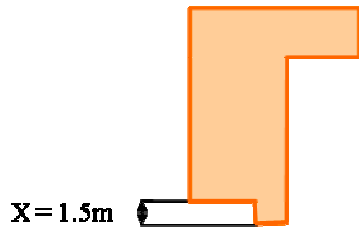


Figure 8- Mesure du détail minimal d'un bâtiment

Les paramètres (la position, l'angle et la grandeur) qui sont utilisés pour ajuster le patron sont extraits en appariant interactivement le patron au bâtiment. Ainsi, on prélève la position, l'angle de rotation et la valeur de l'extension du patron.

Parfois, pour précisément adapter un patron à la géométrie d'un objet de la carte, les opérations sur le patron en entier sont insuffisantes. Des opérations qui s'appliquent aux primitives (*les opérations d'ajustement*) sont alors nécessaires. Ces opérations permettent d'ajuster finement un patron géométrique en changeant les paramètres (ex. la grandeur) de ses primitives. Ces opérations sont : le changement de grandeur d'une primitive (allongement/raccourcissement, élargissement/rétrécissement), le déplacement d'une primitive et le changement d'orientation d'une primitive. Dans le cas où ces opérations sont utilisées, les paramètres ayant servi à l'ajustement des primitives sont extraits et stockés pour une éventuelle utilisation lors d'un processus de généralisation à la volée.

3.3.3. La généralisation des bâtiments à l'aide des patrons géométriques

Lors d'un processus de généralisation, au lieu d'appliquer des algorithmes de généralisation à un objet cartographique, on prend tout simplement son patron géométrique et on réalise quelques opérations (ex. rotation, déplacement et agrandissement/rétrécissement) pour l'ajuster à la géométrie de l'objet. Les paramètres utilisés pour faire ces ajustements sont ceux prélevés lors de la phase de saisie. Les relations hiérarchiques existant entre différents patrons géométriques ainsi que les opérations de généralisation permettent de généraliser des objets cartographiques à des échelles arbitraires.

Si l'opération « simplification des patrons » (l'élimination des primitives secondaires) a été appliquée pour générer les patrons géométriques dérivés formant en partie la base de données des patrons, les opérations d'ajustement quant à elles sont surtout utilisées à la volée, lors d'un processus de généralisation. Par exemple, si la taille d'une primitive secondaire est trop petite, et que ce détail est très important, il est nécessaire de le conserver. Si la longueur de la primitive est trop courte, l'opération « allongement de primitive secondaire » est utilisée. Mais si la largeur de la primitive ne répond pas au seuil exigé, cette dernière est élargie. Par contre, l'élargissement d'une primitive secondaire engendre parfois un problème de proximité entre différentes primitives. Ce problème, qui est une violation de la contrainte de la distance minimale entre différents objets, est résolu soit en combinant les primitives secondaires créant ce conflit, soit en déplaçant certaines primitives secondaires. C'est souvent le cas en généralisation cartographique. Lors de la généralisation cartographique, le conflit de proximité est l'un des plus importants et difficiles à corriger. Sa correction nécessite le plus souvent le déplacement ou la fusion des objets, voire une élimination de l'objet. Dans le cas des bâtiments, pour éviter ces opérations, l'érosion, qui est une opération pour couper une partie de l'objet afin de gagner de l'espace, est souvent utilisée. Pour les patrons géométriques, cette opération se réalise facilement et de plusieurs manières, soit par un raccourcissement des primitives secondaires, soit par l'élimination d'une ou plusieurs primitives, évitant ainsi l'élimination de l'objet en entier.

Pour assurer l'orthogonalité des côtés des bâtiments pendant la généralisation, l'opération « équarrissage » est généralement appliquée. Dans le cas des patrons, cette opération n'est pas nécessaire, car la majorité des opérations de généralisation des patrons préservent cette orthogonalité.

Les principales étapes de la généralisation des bâtiments à l'aide des patrons géométriques sont résumées sur le diagramme représenté à la figure 9.

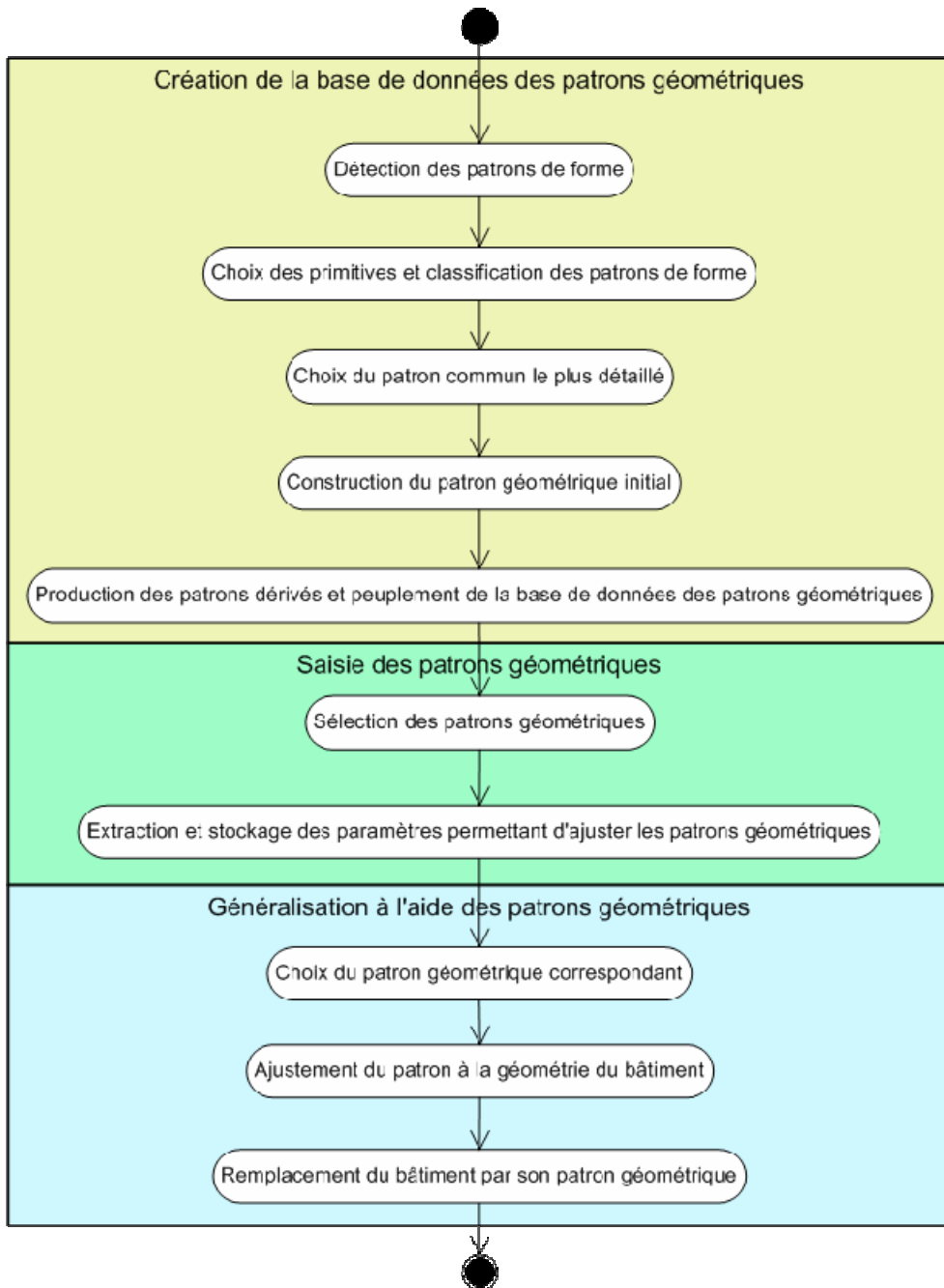


Figure 9- Les principales étapes de la généralisation des bâtiments à l'aide des patrons géométriques

3.4. Les résultats des tests

Pour améliorer considérablement le processus de généralisation à l'aide des patrons géométriques, un nombre élevé de bâtiments doit être remplacé par un nombre restreint de patrons géométriques avec un minimum de manipulations. Pour vérifier le bénéfice apporté par les patrons géométriques, deux groupes de tests ont été réalisés dans le cadre de cette recherche. Des tests de performance et des tests pour vérifier le taux de remplacement des bâtiments par des patrons géométriques à différentes échelles ont été réalisés. Pour cela, la base de données des patrons géométriques a été préalablement créée. Dans cette partie, après une brève description du matériel et des données utilisés, les résultats des différents tests seront présentés.

3.4.1. Données et matériel utilisés

Les tests ont été réalisés avec des données de format Shapefile (ESRI) provenant de Défense Canada, l'un des partenaires du projet GEMURE. Ces données représentent les bâtiments de la ville de Québec à l'échelle 1k. Cette échelle a été choisie parce que contrairement aux cartes à plus petites échelles, l'échelle 1k n'a pas subi des opérations de généralisation comme l'agrégation ou l'élimination de bâtiments. Les données test couvrent une superficie d'environ 5 km² et comprennent 2844 bâtiments, tous représentés par des polygones. Les tests de performance ont été réalisés sur l'ensemble des 2844 bâtiments. Mais, pour les autres travaux qui nécessitent une plus grande intervention humaine (création des patrons et vérification du taux de remplacement des bâtiments par des patrons), une portion de la zone test composée de 365 bâtiments représentatifs a été choisie. La figure 10 montre un extrait de la zone test. Tous les tests ont été réalisés sur un Pentium 4 CPU 2.4 GHZ avec une mémoire vive de 3.6 Gigaoctets.



Figure 10- Extrait des bâtiments de la zone test

3.4.2. Résultats de la création de la base de données de patrons géométriques

Après une analyse visuelle des bâtiments de la portion de la zone test (365 bâtiments), un recensement sommaire des principaux patrons de formes a été réalisé. Il ressort de l'examen de ces différents patrons que le parallélogramme à angles droits (rectangle) est la primitive de patron la plus fréquemment rencontrée, et le patron commun le plus détaillé est un patron à quatre extensions (primitives). Pour permettre une éventuelle extension de la zone d'utilisation des patrons à construire, le nombre de primitives a été majoré de deux. Le patron initial construit est ainsi un patron formé de six instances d'une primitive rectangulaire. Pour la construction du patron géométrique initial, le logiciel ArcGIS a été utilisé. La construction s'est faite interactivement en juxtaposant plusieurs instances d'un rectangle. Après la simplification manuelle du patron initial, 63 patrons géométriques dérivés ont été générés (cf. 3.3.1.). Une fois les patrons redondants éliminés, l'arbre résultant est un arbre à 24 patrons géométriques simples (figure 11).

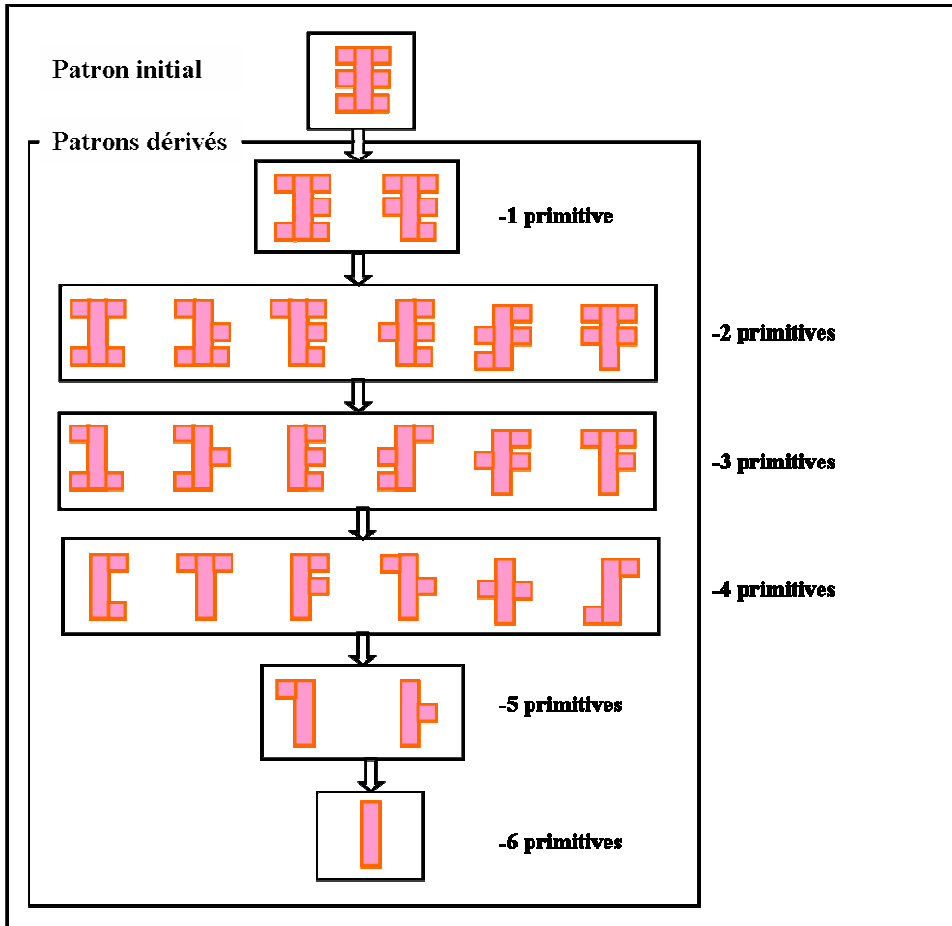


Figure 11- Arbre hiérarchique des patrons géométriques

3.4.3. Vérification du taux de remplacement des bâtiments par des patrons géométriques

Ce test consiste à vérifier la proportion des bâtiments de la zone test remplaçables par des patrons géométriques selon les échelles de visualisation. Ainsi, pour chaque bâtiment, on a déterminé tous les patrons géométriques qui lui correspondent et l'échelle à laquelle ces patrons géométriques sont utilisés (cf. 3.3.2). Pour cela, chaque patron choisi est ajusté manuellement et le détail minimal est mesuré. Pour chaque bâtiment, tous les couples de valeurs (numéro du patron géométrique, détail minimal) sont stockés dans un fichier créé à cette fin. Le nombre de couples de valeurs d'un bâtiment correspond au nombre de patrons géométriques qui lui sont associés. L'échelle à laquelle chaque patron remplacera le bâtiment auquel il est associé a été calculée à partir de la valeur du détail minimal en utilisant l'équation 1 (cf. 3.3.2). Les statistiques suivantes ont été extraites : le taux des

bâtiments remplacés selon les échelles (figure 12); la fréquence d'utilisation des différents patrons géométriques (figure 13) et la fréquence d'utilisation des patrons géométriques selon le nombre d'instances de primitives secondaires (figure 14). ArcGIS a été utilisé pour la réalisation de ce test.

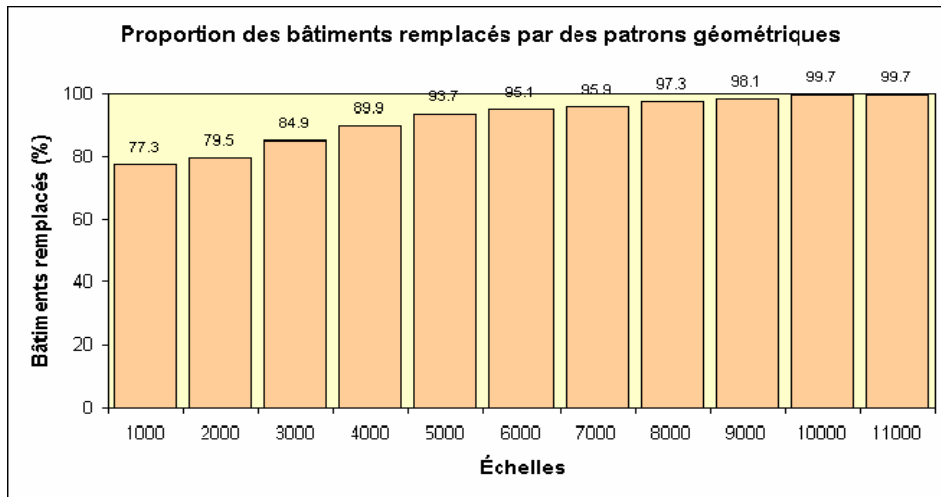


Figure 12- Taux des bâtiments remplacés par des patrons géométriques selon les échelles

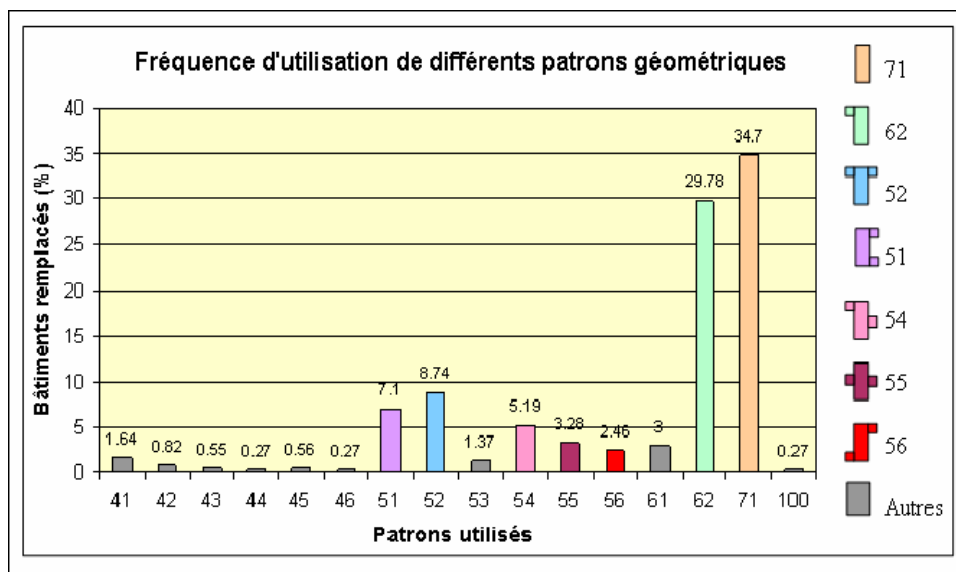


Figure 13- Taux des bâtiments remplacés par type de patron géométrique

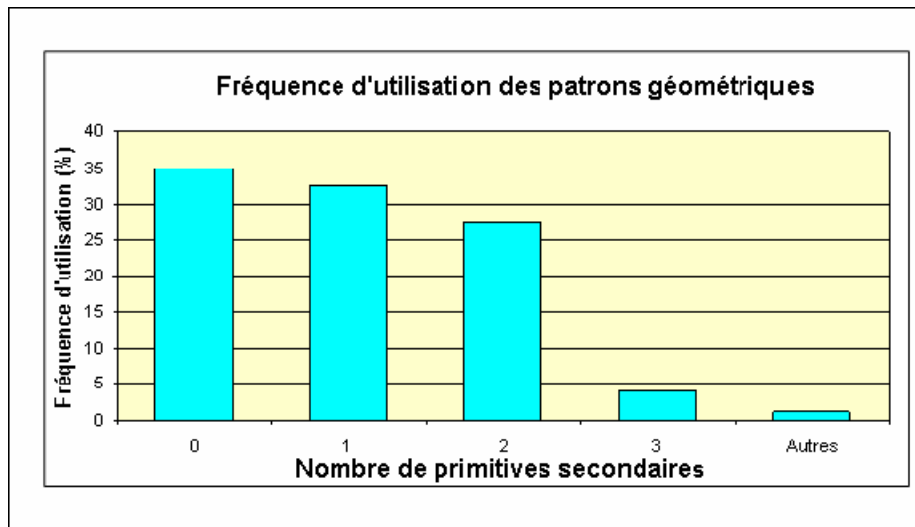


Figure 14- Fréquence d'utilisation des patrons géométriques selon le nombre d'instances de primitives secondaires

Les résultats révèlent qu'à l'échelle 1K, plusieurs bâtiments de la zone sont remplaçables par des patrons géométriques (plus de 70 % de bâtiments). Ce taux augmente avec la diminution de l'échelle pour se stabiliser à plus de 99 % aux environs de l'échelle 10K. En effet, certains patrons n'apparaissent qu'à des échelles plus réduites. La stagnation est due au fait qu'un bâtiment de forme circulaire se trouve dans la zone test. Du moment où notre base de données de patrons ne possède pas de patrons de cette forme, il est impossible donc de remplacer ce bâtiment par un patron géométrique.

La figure 13 peut aussi servir de guide dans le travail d'optimisation de la base de données de patrons géométriques afin de faciliter l'accès aux patrons géométriques les plus utilisés. Sur les 24 patrons de la base de données, seulement 16 ont été utilisés pour réaliser tous les remplacements. Tous les patrons utilisés ont un nombre de primitives secondaires inférieur à 4. Les patrons géométriques les plus simplifiés (avec moins de primitives secondaires) présentent une grande fréquence d'utilisation, car ils sont utilisés par la majorité des bâtiments à un certain niveau d'abstraction. Par exemple, à une certaine échelle, presque tous les bâtiments deviennent des rectangles.

Les bâtiments les plus difficiles à remplacer par des patrons géométriques simples sont dans la catégorie des bâtiments publics. Ceci est dû à la complexité de leur géométrie. Mais, la majorité des bâtiments publics sont remplaçables par des patrons géométriques simples. Après une analyse sommaire d'une zone plus étendue que la zone test, il ressort que seulement environ 5 % des bâtiments publics sont irremplaçables par des patrons simples à une grande échelle, en l'absence de patrons complexes dans notre base de données. Ces bâtiments représentent moins de 1 % de l'ensemble des bâtiments de la zone test. Ce taux peut être considérablement diminué grâce à l'utilisation de patrons complexes (combinaison de plusieurs patrons simples).

3.4.4. Tests de performance des patrons

En plus de la diminution de la redondance, l'intérêt principal de l'utilisation des patrons géométriques dans un processus de généralisation est de faciliter la création des cartes à des échelles arbitraires, et ce, sans délai notable. Pour cela, des tests pour comparer la performance de la généralisation à l'aide des patrons géométriques et celle à l'aide des algorithmes traditionnels ont été réalisés. Les tests réalisés sont divisés en deux groupes :

- Les tests de performance de la généralisation des bâtiments individuels;
- Les tests de performance de la généralisation des groupes de bâtiments.

Pour les deux groupes de tests, le logiciel FME Professional de la firme Safe Software Inc. a été utilisé [Safe Software 2005]. Cet outil, destiné à la transformation des données géographiques, renferme plusieurs algorithmes permettant de généraliser des objets individuels ou groupe objets. De même, il offre des fonctions nécessaires à l'ajustement des patrons géométriques (rotation, déplacement, changement de grandeur) et génère le temps de réponse de chaque opération.

3.4.4.1. La généralisation des bâtiments individuels

Premièrement, dans FME, les 2844 bâtiments de la zone test ont été simplifiés automatiquement avec deux algorithmes différents, LineGeneralizer et AreaGeneralizer. Pour chacun des deux algorithmes, différents paramètres ont été appliqués. Ces paramètres

permettent de générer des données correspondantes à des échelles précises (10k, 50k, 100k et 200k). Ainsi, pour chaque échelle, le seuil du détail minimal des bâtiments est calculé en utilisant l'équation 1. Connaissant ce seuil pour une échelle donnée, le paramètre de chaque algorithme est déterminé expérimentalement de la sorte qu'il permette d'éliminer les détails en dessous du seuil calculé pour cette échelle. Deuxièmement, une simulation pour remplacer autant de bâtiments (2844) par des patrons géométriques a été réalisée. Les paramètres nécessaires à l'ajustement de chaque patron géométrique (la position, l'angle de rotation et la grandeur) ont été préalablement stockés dans la base de données des bâtiments. La proportion des patrons utilisés pour remplacer ces bâtiments à différentes échelles est montrée sur la figure 15. Chacun des tests a été réalisé au minimum trois fois. Le temps moyen selon les échelles pour chaque opération est montré sur la figure 16.

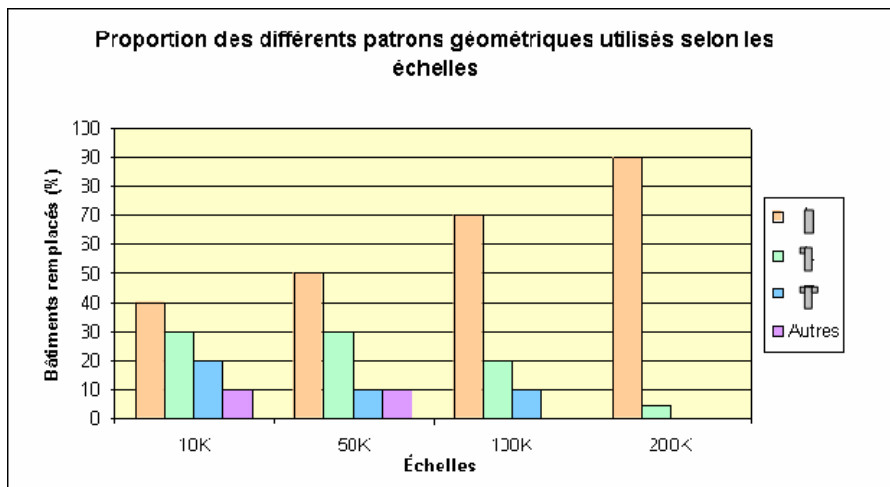


Figure 15- Proportion des différents patrons géométriques utilisés pour remplacer 2844 bâtiments

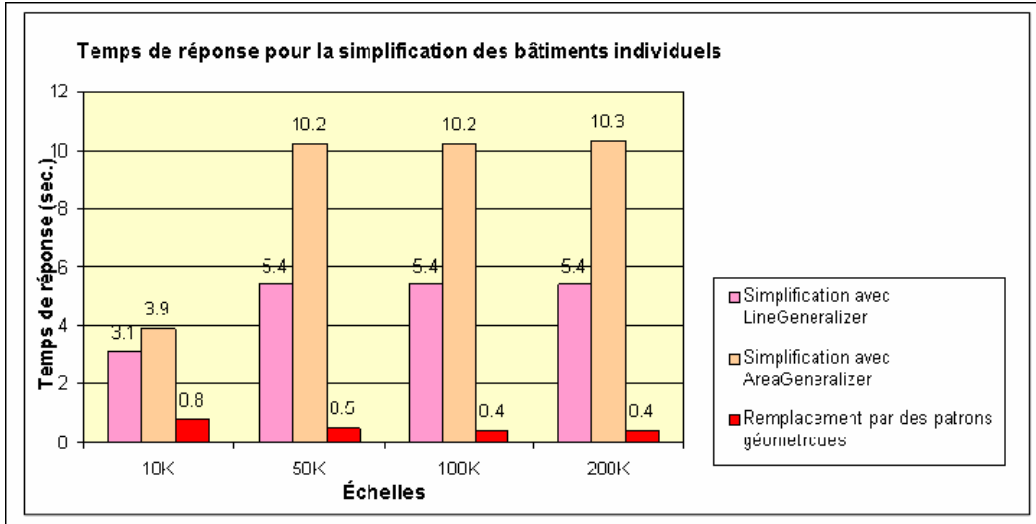


Figure 16- Temps de réponse moyen (sec.) pour la généralisation des bâtiments individuels

3.4.4.2. La généralisation des groupes de bâtiments

La même démarche que celle utilisée lors de la simplification des bâtiments individuels a été suivie pour la généralisation des groupes de bâtiments. Les deux seules différences sont, d'une part, que les algorithmes de simplification ont été appliqués après une agrégation des bâtiments individuels et, d'autre part, que les patrons géométriques ont été utilisés pour remplacer 635 groupes de bâtiments. L'agrégation des bâtiments génère 635 groupes de bâtiments. Les temps moyens selon les échelles pour chaque opération sont montrés à la figure 17.

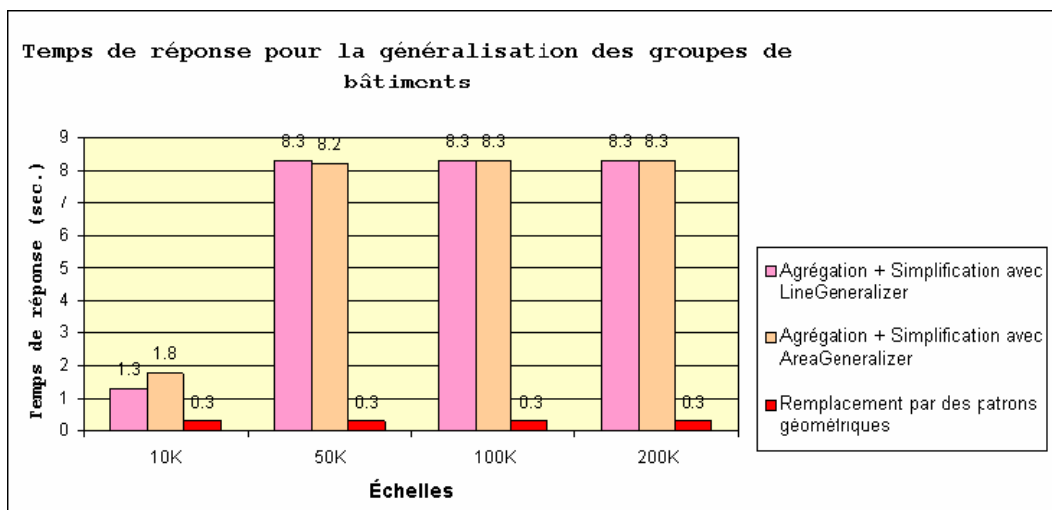


Figure 17- Temps de réponse moyen (sec.) pour la généralisation des groupes de bâtiments

Des exemples de bâtiments généralisés avec différentes approches de généralisation (à l'aide des algorithmes de généralisation comparée à l'utilisation des patrons géométriques) sont présentés à la figure 18. Sur la figure, les bâtiments simplifiés à l'aide des algorithmes de généralisation (18-c et 18-d) représentent les meilleurs résultats que nous avons obtenus en appliquant les paramètres permettant d'atteindre le même niveau de simplification que celui des patrons géométriques (18-b).

Contrairement à la simplification à l'aide des algorithmes de généralisation où le temps de traitement augmente avec la diminution de l'échelle, le temps nécessaire à l'ajustement des patrons géométriques décroît lors de la diminution de l'échelle. Pour les patrons géométriques, cette croissance de la performance est due au fait que la complexité géométrique des patrons utilisés décroît avec la diminution de l'échelle. Par exemple, pour remplacer le même nombre de bâtiments, plus de cinq patrons géométriques ont été utilisés à l'échelle 50K contre seulement deux à l'échelle 200K. De plus, parmi tous les patrons de la base de données, ces deux derniers présentent la forme la moins complexe. Pour les algorithmes de simplification, la performance diminue, car le nombre des points à simplifier augmente avec la diminution de l'échelle (ex. à l'échelle 10K, environ 11000 points sont simplifiés contre 15 000 au 200K). Ceci entraîne un gain de performance de l'approche patrons géométrique par rapport aux approches traditionnelles de généralisation allant de 5 à 27 fois.

Pour les méthodes de généralisation testées (algorithmes de généralisation et remplacement par des patrons géométriques), on note une stagnation du temps de réponse à partir d'un certain niveau d'abstraction. Pour l'approche par patrons géométriques, cette stagnation est due au fait qu'à partir d'une certaine échelle, la majorité des objets (bâtiments individuels ou agrégats de bâtiments) ont atteint leur forme la plus simplifiée, ce qui entraîne l'utilisation d'un même patron malgré la diminution de l'échelle. Pour l'approche par algorithmes de généralisation, la stagnation commence quand la majorité des bâtiments ont atteint leur niveau de simplification maximale. Par exemple, quand une polyligne d'un polygone devient une ligne droite, elle a atteint son niveau de simplification maximale.

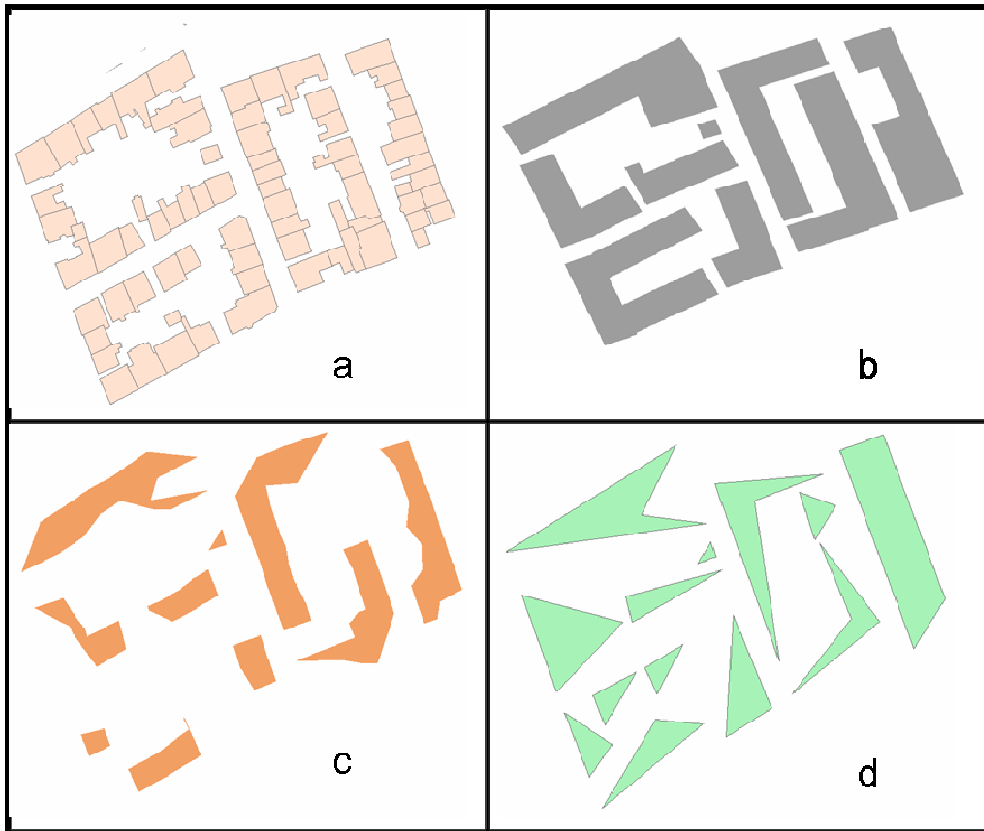


Figure 18- Exemples de généralisation: a) données initiales; b) utilisation des patrons géométriques; c) agrégation et simplification avec Area Generalizer; d) agrégation et simplification avec Line Generalizer.

Lors de la simplification des bâtiments individuels, le temps de réponse de LineGeneralizer est presque deux fois inférieur à celui du AreaGeneralizer. Par contre, la qualité des données générées par ce dernier est meilleure à celle des données obtenues après la simplification à l'aide de LineGeneralizer. Ceci s'explique par le fait que l'algorithme utilisé dans AreaGeneralizer vérifie la topologie. Cette vérification de topologie entraîne un délai supplémentaire. Par contre, lors de la généralisation des données agrégées (groupes de bâtiments), le temps de réponse pour les deux méthodes de simplification est approximativement le même. Avec un nombre d'objets relativement bas (635 agrégats contre 2844 bâtiments), la vérification de la topologie devient plus simple, ce qui explique cette ressemblance de performance.

3.5. Conclusion

Dans cet article, nous avons présenté une méthodologie pour appliquer le concept des patrons géométriques à la généralisation des bâtiments. Des tests de performance et une vérification du taux de bâtiments remplaçables par des patrons géométriques s'avèrent très concluants. D'une part, la simplicité des opérations nécessaires pour ajuster les patrons géométriques assure un gain de performance pouvant atteindre un facteur de 27 par rapport au temps de réponse des méthodes traditionnelles de généralisation. D'autre part, un grand nombre de bâtiments sont remplacés par des patrons géométriques avec un degré de fidélité acceptable (plus de 70 % des bâtiments de la zone test sont remplacés par des patrons géométriques à l'échelle 1K). Conséquemment, nous sommes d'avis que les patrons géométriques démontrent un fort potentiel pour la généralisation à la volée qui exige un temps de réponse quasi instantané.

De plus, l'utilisation des patrons géométriques peut faciliter les transferts des données à travers le réseau. En effet, dans une architecture client/serveur, tous les patrons géométriques nécessaires à la généralisation de la carte demandée par l'utilisateur peuvent être transférés une seule fois vers le poste de l'utilisateur. Ainsi, lors de la généralisation, au lieu de transférer les résultats de la généralisation (environ 2844 bâtiments généralisés dans notre cas), seuls les paramètres nécessaires à l'ajustement du patron géométrique de chaque bâtiment sont transférés vers le client. Ceci diminue considérablement le volume des données transférées du serveur vers le client. En effet, l'espace disque nécessaire pour stocker dans une base de données à représentation multiple les différentes représentations simplifiées d'un bâtiment à 16 vertex est de 7 fois supérieur à celui nécessaire pour stocker son patron géométrique et les paramètres nécessaires à son ajustement. Et si on considère qu'un patron géométrique est utilisé par plusieurs instances de bâtiments, le gain en espace de stockage devient encore plus appréciable.

Même si un grand nombre de bâtiments de la zone test a pu être remplacé par des patrons géométriques, il ne faut pas perdre de vue que certaines formes de bâtiments ou certaines classes d'objets comme le réseau hydrographique se prêtent moins à l'utilisation des patrons. Par exemple, pour des bâtiments aux formes très complexes et non redondantes, il est difficile de trouver des patrons géométriques simples à manipuler. Bien que la

combinaison des patrons géométriques simples afin de former des patrons plus complexes soit possible, nous pensons que le nombre de combinaisons doit être limité. Un nombre élevé de combinaisons rend les patrons moins flexibles et moins performants.

Pour les bâtiments n'ayant ni patrons géométriques simples ni patrons géométriques complexes pour les remplacer, l'utilisation d'algorithmes de généralisation traditionnels s'avère nécessaire. Les algorithmes de généralisation sont utilisés sous forme de patrons de traitements qui sont des groupes récurrents d'algorithmes qui servent à généraliser des objets cartographiques de caractéristiques géométriques et spatiales similaires. Ces patrons de traitements sont intégrés à l'intérieur des SGO (objets auto-généralisants). Les patrons de traitements, ainsi que les SGO seront traités dans les chapitres subséquents de la thèse.

Les patrons stockés dans la base de données peuvent aussi servir lors d'une saisie multi-échelle. Ainsi, au lieu de saisir un objet à différentes échelles, on peut tout simplement saisir sa géométrie détaillée et associer des patrons géométriques pour les géométries simplifiées. Dans le cadre du projet GEMURE, des travaux intégrant les patrons géométriques dans un processus de saisie multi-échelle ont déjà débutés.

À première vue, la saisie des patrons géométriques, à savoir la sélection des différents patrons pour chaque objet cartographique, l'extraction des paramètres nécessaires à l'ajustement des patrons à la géométrie des objets qu'ils doivent remplacer apparaît comme une tâche ardue. Mais en réalité, ce travail se fait une seule fois et est utilisé plusieurs fois. De plus, les techniques de reconnaissance des formes peuvent éventuellement être utilisées pour automatiser le processus. Pour le moment, dans le cadre de cette recherche, des outils interactifs de saisie des patrons géométriques de bâtiments sont en cours de développement. Ces outils permettront de considérablement faciliter les processus de création et de saisie des patrons géométriques.

3.6. Remerciements

Les auteurs tiennent à remercier le Réseau canadien de centres d'excellence GEOIDE qui a rendu possible la réalisation de ces travaux dans le cadre du projet GEMURE, ainsi que les

partenaires suivants: Recherche et développement pour la défense Canada, Ressources naturelles Canada, Ressources naturelles et Faune Québec, Intergraph.

3.7. Références

- Alexander C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel. 1977. *A pattern language: towns, buildings, construction*. New York, Oxford University Press, 1171 p.
- Bédard Y., E. Bernier et R. Devillers. 2002. La métastructure VUEL et la gestion des représentations multiples. Dans Ruas A. (Dir.) : *Généralisation et représentation multiple*. Paris, Hermes science publications, pp. 149-162.
- Bernier E., Y. Bédard et M. Lambert. 2003. Cartographie sur demande sur le Web et bases de données multidimensionnelles. De la personnalisation par couche cartographique à la personnalisation par occurrences. *Revue Internationale de Géomatique*, 13(3), pp. 339-359.
- Cardenas A. 2004. Utilisation Des Patrons Géométriques Comme Support À La Généralisation Automatique. Mémoire de maîtrise non-publié, Faculté de foresterie et géomatique, Université Laval, Québec, 110 p.
- Cecconi A., R. Weibel and M. Barrault. 2002. Improving automated generalisation for on-demand web mapping by multiscale database. *Proceeding of joint International Symposium on Geospatial Theory, Processing and Applications*, July 8-11, Ottawa, Canada, pp. 1-9.
- Coad P. 1992. Object-Oriented Patterns. *Communications of the ACM*, 35(9), ACM Pres, New York, USA, pp. 152-159.
- Devogele T., T. Badard et T. Libourel. 2002. La problématique de la représentation multiple. Dans Ruas A. (Dir.) : *Généralisation et représentation multiple*. Paris, Hermes science publications, pp. 55-74.
- Devogele T., J. Trevisan et L. Raynal. 1996. Processus De Constitution D'une Base De Données Multi-Échelles. *Revue Internationale de Géomatique*, 6(2-3), pp. 249-263.
- Douglas D. H. and T. K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2), pp. 112-122.
- Fortin C. et R. Rousseau. 1992. *Psychologie Cognitive : Une Approche de Traitement de L'information*. Sillery, Presses de l'université du Québec, 434 p.
- Gamma E., R. Helm, R. Johnson and J. Vlissides. 1995. *Design patterns: elements of reusable object-oriented software*. Reading, Mass. Don Mills, Ont., Addison-Wesley, xv, 395 p.
- GiMoDig. 2004. Geospatial info-mobility service by real-time data-integration and generalisation. <http://gimodig.fgi.fi> (Accédé le: 12-07-2004).

- Jenk G. F. 1989. Geographic logic in line generalization. *Cartographica*, 26(1), pp. 27-43.
- Lang T. 1969. Rules for the robot draughtsmen. *The Geographical Magazine*, 42(1), pp. 50-51.
- Mackaness W.A. and G. Edwards. 2002. The Importance of Modelling Pattern and Structure in Automated Map Generalisation. *Join ISPRS/ICA workshop on multi-scale representations of spatial data*, July 7-8, Ottawa, Canada.
- MapQuest. 2004. Advertise with us. <http://www.mapquest.com/features/main.adp?page=advertise> (Accédé le: 09-09-2004).
- Martel C. 1999. Développement d'un cadre théorique pour la gestion des représentations multiples dans les bases de données spatiales. Mémoire de maîtrise non-publié, Département des Sciences géomatiques, Université Laval, Québec, 139 p.
- Newell A. 1990. *Unified theories of cognition*. Cambridge MA, Harvard University Press, 549 p.
- Rainsford D. and W.A. Mackaness. 2002. Template Matching in Support of Generalisation of Rural Buildings. Dans Richardson D. et P. Van Oosterom (Dir.): *Advances in Spatial Data Handling 10th International Symposium on Spatial Data Handling*. Springer, Berlin, pp. 137-152.
- Regnauld N. 2002. Généralisation Des Bâtiments. Dans Ruas A. (Dir.): *Généralisation et représentation multiple*. Paris, Hermes science publications, pp. 257-272.
- Safe Software. 2005. <http://www.safe.com/>. (Accédé le: 12-04-2005).
- Sester M. and C. Brenne. 2004. Continuous Generalization for Fast and Smooth Visualization on Small Displays. Dans Fisher P. (Dir.): *Developments in Spatial Data Handling- 11th International Symposium on Spatial Data Handling*. Springer Verlag, pp. 469-480.
- Timpf S. 1998. Hierarchical structures in map series. PhD. thesis, Department of Geoinformation, Technical University of Vienna, Vienna, 124 p.
- Van Oosterom P. and V. Schenkelaars. 1995. The development of an interactive multi-scale GIS. *International Journal of Geographical Information Systems*, 9(5), pp. 489-507.
- Vangenot C. 1998. Représentation multi-résolution, concepts pour la description de bases de données avec multi-représentation. *Revue Internationale de Géomatique*, 8(1-2), pp. 121-147.
- VisValingam M. and J. D. Whyatt. 1993. Line generalisation by repeated elimination of points. *Cartographic Journal*, 30(1), pp. 46-51.
- Wang Z. and J. C. Müller. 1998. Line generalization based on analysis of shape characteristics. *Cartography and GIS*, 25(1), pp. 3-15.
- Weibel R., E. Bernier, Y. Bédard et A. Cecconi. 2002. La généralisation à la volée. Dans Ruas A. (Dir.): *Généralisation et représentation multiple*. Paris, Hermes science publications, pp. 319-335.

Zacks J. and B. Tversky. 2001. Event Structure in Perception and Conception. *Psychological Bulletin*, 127(1), pp. 3-21.

Chapitre 4 Les objets auto-généralisants (SGO)

INTEGRATING GEOMETRIC PATTERNS WITH GENERALIZATION ALGORITHMS: TOWARDS SELF-GENERALIZING OBJECTS AND ON-THE- FLY MAP GENERALIZATION

Auteurs: Mamane Nouri SABO, Yvan BÉDARD, Bernard MOULIN, Eveline BERNIER

(Sera soumis à la revue Cartographica)

Remarque : *Cet article introduit le concept théorique des SGO (Section 4.3). Dans les Sections 4.1 et 4.2, l'article reprend certaines notions qui ont été présentées dans les chapitres introduction (Section 1) et revue de littérature (Section 2). D'autre part, l'article introduit brièvement le système multi-agent de généralisation à la volée qui sera présenté en détail dans l'article subséquent (Section 5).*

Résumé de l'article : *La généralisation cartographique est un processus complexe qui demande parfois l'intervention humaine. Afin de supporter un tel processus à la volée, nous proposons une approche de généralisation qui se base sur les SGO (Objets Auto-Généralisants ou Self Generalizing Objects) qui encapsulent à la fois des patrons géométriques (qui sont des formes communes à plusieurs objets cartographiques), des algorithmes de généralisation et des contraintes d'intégrité spatiales. Lors d'un processus d'enrichissement de la base de données, un SGO est créé et associé à chaque objet ou groupe d'objets de la carte. Les SGO créés sont ensuite transformés en agents logiciels (agents SGO) dans un système multi-agent de généralisation à la volée. Les agents SGO sont dotés de comportements qui leurs permettent de coordonner le processus de généralisation. Dans cet article, nous présenterons le concept des SGO et les prototypes (un prototype pour la création et l'enrichissement des SGO et un autre pour la généralisation à la volée, utilisant les SGO créés) développés pour supporter cette approche.*

Abstract:

Map generalization is a complex task which requires sometimes human intervention. In order to support such a process on-the-fly, we propose a generalization approach based on Self-Generalizing Objects (SGOs) which encapsulate geometric patterns (forms common to several cartographic features), generalization algorithms, and spatial integrity constraints. During a database enrichment process, a SGO is created and associated with a cartographic feature, or a group of features. Then, each created SGO is transformed into a software agent (SGO agent) in a multi-agent on-the-fly map generalization system. SGO agents are equipped with behaviours which enable them to coordinate the generalization process. In this paper, we present the concept of SGO and two prototypes that we developed in order to support this approach: a prototype for the SGOs' creation and another for the on-the-fly map generalization (which uses the created SGOs).

4.1. Introduction

In the past, map production was a cartographer's responsibility because of the specific skills required to create maps. Users received the maps created in advance, generally produced in a series. However, the technological developments of the last decade have generated a new type of media for spatial data dissemination, the Web. In addition, there are new applications such as Web mapping and spatial database querying. With the emergence of these new applications, cartographic data has become more accessible to the general public and can be better adapted to the users' needs.

This democratization of cartographic data has resulted in new users' requirements for visualization tools and geographic data. Therefore, beyond a simple static visualization, users expect flexible querying of the data and more powerful map personalization tools. More and more, they want to define parameters such as map scale, symbology, map content, and cartographic features that must be highlighted in order to get maps better adapted to their needs. However, considering the increasing number of web map users, it is impossible to create maps in advance in order to satisfy the needs of all users. Moreover, the interactive characteristics of web mapping require that all transformations be carried out quasi-instantaneously. Indeed, according to Newell (1990), a response time of 0.1 to 10

seconds is needed for the performance of cognitive tasks in order to maintain a specific train of thought. Similarly, a recent study set 4 seconds as the new threshold of acceptability for on-line shopping (Young and Smith 2006).

The ideal solution for helping users to create maps with a specific content and at a desired scale would be to have a single, large-scale database and to derive maps at smaller scale according to users' needs using automatic cartographic generalization processes. However, in order to derive maps of a smaller scale according to the users' needs, automatic cartographic generalization processes is often deployed. In conventional cartography, map generalization is responsible for reducing map complexity through a scale reduction process. This process emphasises the essential, while suppressing the unimportant; it maintains logical and unambiguous relations between map objects, and preserves the aesthetic quality (Weibel and Dutton 1999).

During the last decades, several projects have dealt with cartographic generalization (Douglas and Peucker 1973; Mackaness 1994; Lamy *et al.* 1999; Sarjakoski and Kilpeleinen 1999; Sester 2000; Weibel and Dutton 1998; etc.). At first, research on the automation of map generalization focused on the development of algorithms, with particular emphasis on algorithms for line simplification. During this earlier period, generalization operations were typically applied to all occurrences of an object class, no matter what the object's characteristics or context was. The expected power of the used algorithms was to include the necessary parameters for all possible geometries that may be encountered in any given context. Except in fully controlled and well-defined situations, this is an impossible objective. Map generalization is a complex process, requiring a coordinated usage of several operations. In order to determine the required algorithms, along with their sequences and corresponding parameters, knowledge-based approaches are used (Allouche and Moulin 2005). This approach includes, among others, rule-based approaches (Nickerson and Freeman 1986; Armstrong 1991; McMaster 1991), constraint-based approaches (Beard 1991; Weibel and Dutton 1998; Ruas 1999), and multi-agent approaches (Baeijs *et al.* 1995; Lamy *et al.* 1999; Li *et al.* 2002; Jabeur 2006, Jabeur *et al.* 2006). A knowledge-based approach requires that our knowledge of the generalization process be formalized into a chain of reasoning paths, each leading to a particular decision

or procedure for generalization to take place (Müller 1991). Unfortunately, cartographic knowledge is very difficult to formalize given that map generalization is a subjective and holistic process. It applies to an infinite number of situations, and the quality of the result depends upon the objective of the final map. Consequently, no two cases are alike, and when one organisation develops a solution, it is typically to generalize a specific dataset (e.g., 1:20,000 topographic map) into another very specific dataset (e.g., 1:100,000 topographic map).

Despite the significant progress that has been made over the past few years, automatic cartographic generalization remains a difficult task, still requiring human intervention (Sabo *et al.* 2005). For applications such as Web mapping, that require extremely fast response times, today's alternative is to use a multiple representations database to simulate on-the-fly map generalization. In its simplest, but most common form, a multi-representations database (MRDB) is a spatial database used to store various representations of the same territory (Devogele *et al.* 1996; Weibel and Dutton 1999), at different levels of abstraction (e.g., various scales). In its most complete form (although rarely implemented), Bédard *et al.* (2002) present the MRDB as the storage unit for various geometry, semantics, and graphics of the same object. Thus, according to the user's request, the most suitable representations are selected from the multi-representations database for the desired scale.

There are several types of map-based Web sites that allow users to zoom in and out of a particular region. This is usually based upon simple enlargement/reduction of the presented map. When the enlargement/reduction ratio reaches a large number (e.g. 5X), the represented map is replaced by another map, which may differ significantly in the degree of its generalization. In such an approach, map scales are predefined and do not allow users to get the data corresponding at any desired level of abstraction. This is why certain authors (e.g., Cecconi *et al.* 2002; GiMoDig 2004; Jabeur 2006) propose the refinement, if necessary, of the data selected from a multi-representation database, thanks to a generalization process. Unfortunately, in such an approach, the main problems related, on the one hand, to cartographic generalization (e.g., knowledge formalization problems) and, on the other hand, to multiple representations (e.g., redundancy) are systematically inherited.

In order to allow users to generate maps on-the-fly and at arbitrary scales, we propose an approach based upon geometric patterns (shapes representative of several cartographic features), combined with sets of generalization algorithms (applicable to patterns and cartographic features) and spatial integrity constraints, all encapsulated in a single structure called a SGO (Self-Generalizing Object). This approach allows us to include additional human expertise in an efficient way at the level of individual cartographic features, which then leads to database enrichment that better supports automatic generalization. We begin this paper with an overview of Web mapping and on-the-fly generalization concepts. Then, we present the SGO concept and its underlying components: geometric patterns, generalization algorithms, spatial integrity constraints, and their combined usage. Next, we introduce two working prototypes: first, the *SGO's Creation Engine*, and secondly, the *On-the-fly Map Generalization System*. The latter is based upon a multi-agent approach and allows us to generate maps at arbitrary scales. Finally, we conclude by presenting the perspectives and the remaining challenges.

4.2. On-demand mapping and on-the-fly map generalization

4.2.1. On-demand mapping

Recent technological developments have revolutionized all the IT (information technology) fields, including the geographic information domain. These developments have had an impact not only on the tools and techniques of map production, but also on the ways that we use land information. Although in the past, cartography was almost exclusively limited to the mass production of static paper maps, recent technological developments have permitted the use of another type of map: on-demand maps.

An on-demand map is a map generated according to the specific users' requirements, in contrast to maps generated in mass quantities for the generic needs (e.g., topographic maps) of large groups. The user's requirements may concern the map's scale, its content, the objects to be highlighted or filtered, the symbols to be used, the time necessary for delivery, etc. This map may be created manually or automatically, on paper or using an electronic support, with or without generalization, and with or without delay. Therefore, the main goal

of on-demand cartography is to provide users with data having an adequate level of abstraction, a content and a graphic representation suited to their needs, while taking into account the specifications of each user's visualization tools and hardware constraints (e.g., screen size and network bandwidth).

Traditional cartography is not unfamiliar with on-demand maps. Indeed, there are several types of maps with low printing volumes in traditional cartography; those which are used in extremely specialized fields are typically generated on demand. The main difference between traditional and today's on-demand mapping is that traditional on-demand mapping tries to satisfy a small group of users from a specific field, whereas on-demand mapping from the Internet era tries to satisfy not only the needs of a selected group of people, but also those of individuals accessing the Web. Consequently, there are many more constraints that must be taken into account.

The digital solutions that are currently available typically use a unique and homogenous level of abstraction for their maps. Changing the level of abstraction generally means changing the map for another one created using a different scale. However, as manual cartography has already shown, the level of abstraction does not need to be uniform within a map. For example, in a street map, the elements of the road network are often more detailed than the elements of other features, such as those of the hydrographic network which are presented using a higher level of abstraction. Figure 19 illustrates such heterogeneousness. Indeed, except for topographic maps that try to find a certain balance in terms of levels of abstraction for all the different categories of map features, the other types of maps usually present heterogeneous levels of abstraction.

Recently, the number of Internet users exceeded the one billion mark (Internet World Stats 2006), which represents more than 16% of the world's population. This development of the Internet, along with the fervour for geographical data has facilitated the development of a multitude of cartographic Web sites. In 2003, the number of sites providing access to cartographic data was in the tens of thousands (Peterson 2003). Unfortunately, according to van Elzakker (2003), many subjects found on the Internet have a strong geospatial component, but are not always adequately presented. For example, many sites for tourism and travel contain only descriptive texts and photographs.



Figure 19- Quebec City's road map and bicycle route map with heterogeneous levels of abstraction.

The increasing number of Internet users accessing cartographic products drastically increases the number of requirements in terms of spatial data. These requirements are followed by technological constraints. Thus, even though the Internet has helped with the development and democratization of geographic data, it has also brought with it new production challenges with respect to technological issues (e.g., data transfer rates and map supports), and user issues (e.g., heterogeneous users and users without basic cartographic knowledge). Consequently, it is impossible to totally know in advance the user's context, or the kind of customization he will require.

The early Web mapping sites were based upon simple static maps, in the form of scanned images. Currently, several sites have more sophisticated applications with, for example, interactive maps and a routing capability. In spite of these apparent improvements, many of these applications are very limited in terms of a map's personalization. For instance, several sites do not even allow the user to change the symbology of the required map, and scales are predefined. In practice, most currently available web-based solutions offer simple functions such as zooming and the choice of the map's extension. Only rare applications, such as UMapIt (Bernier *et al.* 2003), provide users with intuitive tools to create highly personalized maps, and to efficiently navigate within the maps. In order to satisfy the needs of every user, it is imperative to find more flexible methods that will allow adapting the

existing geographic data to the user's requirements. In order to adapt the abstraction level, cartographic generalization is necessary. With an interactive and dynamic media such as the Internet, the concept of a "map on demand" has become synonymous with that of a "map generated in real time," given that all map creation processes, as well as generalization processes, must be performed instantaneously. Therefore, the map generalization process must take place on-the-fly.

4.2.2. On-the-fly map generalization

4.2.2.1. What is on-the-fly map generalization?

On-the-fly map generalization can be defined as the "creation, in real time and according to the user's request, of a cartographic product suited to its scale and to its purpose, from a larger-scale database" (Weibel et al. 2002). As previously mentioned, during the on-the-fly map generalization process, the processing time is a crucial factor given that this waiting time must be compatible with Newell's (1999) cognitive band, which is less than 10 seconds. Thus, to provide on-the-fly cartographic generation, the process must rely on fast, effective, and powerful methods. Unfortunately, cartographic generalization is known to be a complex and time consuming process. Besides, in order to produce maps suited to a user's requests, on-the-fly map generalization must also be flexible enough to take into account numerous factors.

One of the main problems associated with on-the-fly map generalization is to determine whether or not the generated map is suited to the user's needs. In conventional cartography, the validation of results, in accordance with the producer's specifications is done a posteriori by the map producer. However, in the case of on-the-fly map generalization, data producers have limited control over the content of resulting maps and, consequently, over the quality of the generalized data. What's more, most users of Web mapping applications have no or very limited knowledge about the map making process, and they cannot assess the quality of the resulting product (and thus, the quality of the map generalized on-the-fly). Consequently, on-demand map production tools in general, and on-the-fly map generalization processes in particular, must generate '*predictable*' and '*good quality*' results for users.

All these requirements (speed, complete automation, flexibility, predictability, and quality) are major challenges considering that the problem of conventional cartographic generalization is not yet entirely solved.

4.2.2.2. State of the art of on-the-fly map generalization

Currently, there are three main research orientations in the field of on-the-fly map generalization (Weibel *et al.* 2002). The first one concentrates on the use of generalization algorithms, whereas the second is devoted to the use of multiple representations. As for the third, it tries to combine the first two approaches (generalization algorithms and multiple representations).

4.2.2.2.1. Approach based upon generalization algorithms

This approach is exclusively based upon generalization algorithms. However, as previously mentioned, generalization is a complex and time-consuming process. Even if the technological developments of recent years have allowed for the creation of high-speed processors, it is still nevertheless necessary to find methods that can accelerate processes in order to support on-the-fly map generalization. This explains why certain authors propose methods based upon pre-computed attributes in order to accelerate the generalization process. For example, in order to facilitate the selection and simplification operations, Van Oosterom and Schenkelaars (1995) propose the combination of two structures: the Binary Line Generalization tree (BLG-tree) and the Reactive-tree. The BLG-tree is a structure that stores results computed by the Douglas-Peucker simplification algorithm (Douglas and Peucker 1973). The Reactive-tree is an index structure allowing the selection of map objects according to their importance.

Unfortunately, current on-the-fly map generalization methods are exclusively based upon generalization algorithms and limited to rather simple operations like "simplification" and "selection". In addition, these approaches do not take into account the local spatial environment of the map's objects. Furthermore, due to its complexity, generalization cannot be carried out with a simple sequential application of generalization algorithms without taking into account the objects' spatial neighbourhood. Finally, existing federative

generalization methods based upon artificial intelligence (e.g., the multi-agent approach) are very complex and require long treatments and adjustments, which currently limit their use for on-the-fly map generalization.

4.2.2.2. Multi-representations approach

As previously mentioned, the ideal solution to allow users to get data at the desired level of abstraction would be a unique database at the most detailed level of cartography from which we could automatically create products at smaller scales. However, given the current state of automatic map generalization, this is still impossible. Therefore, some researchers use multi-representations as an alternative solution. This approach proposes to store several pre-defined representations of a given object (usually at different scales) within the same database. The simplest representations are usually obtained from the manual or semi-automatic generalizations of the most detailed representations. Most of the time, this approach leads to a multi-scale database, including several map layers at different scales. Then, according to the user's request, the appropriate scale is selected. Throughout recent years, much effort has been devoted to manage multi-representations (Martel 1999; Bédard 2002; Devogele 2002; Vangenot 1998; Timpf 1998).

Hence, the various map scales used to create a multi-representations database are usually obtained using generalization operations. Moreover, in addition to inheriting some of the problems related to automatic generalization, multi-representations have three main types of problems according to Devogele (1996): mapping between various abstractions; mapping between various representations of the same object; and the definition of a matching process. In terms of personalization, multi-representations are extremely limited because all scales are predefined. Other important problems related to multiple representations are data redundancy as well as the difficulty to propagate data updates. All these problems limit the effective use of multi-representations for on-the-fly map generalization.

4.2.2.2.3. Approach combining generalization algorithms and multiple representations

The problems associated with automatic generalization and multiple representations have given rise to a third approach that combines these two approaches. In this combination, a multi-scale database is used and, depending on the user's request, the most appropriate scale is selected. When the selected scale does not match the user's requested scale, the selected data can be refined by applying simple generalization operations (e.g., selection or simplification). This is a two-step process where the system selects the representation that is closest to the user's desired map representation (based upon the closest map scale) and the generalization process is applied to this "closest" representation. Several authors (e.g., Cecconi *et al.* 2002; Jabeur 2006) use this approach. Its advantage is that it reduces the effort needed for generalization and improves the quality of the result because the smaller the difference between the initial map scale and the desired one, the easier the generalization process. However, to be truly efficient, this method must rely on a database that includes several scales, leading to the typical problems associated with multiple representations. On the contrary, keeping the number of map scales to a minimum leads to increased difficulty for automatic generalization. Then, to improve this third approach, it is necessary to develop new methods that minimize, as much as possible, the problems associated with automatic generalization and multiple representations. It is in this context that we propose an approach combining generalization algorithms and geometric patterns, a type of multi-representations where data redundancy is kept to a minimum.

4.3. Self-Generalizing Object- SGO

4.3.1. What is a SGO?

In order to support a process of map generalization on-the-fly, we propose an approach of generalization based upon a fast generalization method (geometric patterns) and database enrichment, all encapsulated in an object called SGO (Self-Generalizing Object). This enrichment that exploits human expertise allows us to both generate explicit spatial information (e.g., the alignment of buildings), which is implicit in the initial database, and to introduce into the database, the generalization knowledge (e.g., procedural knowledge) that is difficult to formalize.

A SGO, or Self-Generalizing Object, can be defined as an object based upon fast generalization methods and database enrichment, in addition to being associated with a cartographic feature in order to facilitate and accelerate the automatic generalization of this feature. To make a cartographic feature generalizable, the SGO uses three fundamental components: 1) *Geometric patterns* (shapes representative of several cartographic features) (see Section 4.3.2.1); 2) *Process patterns* (recurrent sets of generalization algorithms and their sequences) (see Section 4.3.2.2); 3) *Spatial integrity constraints* (see Section 4.3.2.3).

Thus, during a database enrichment phase, the Operator can create a SGO for every map feature and enrich it by specifying its components (e.g., geometric patterns, process patterns, and spatial integrity constraints). Once created and enriched, a SGO contains the necessary tools and data that enable it to generalize the objects with which it is associated. The process of generalization is initiated by the SGO in order to satisfy a violated integrity constraint. For that, SGOs are equipped with behaviours which enable them to coordinate the generalization process and to interact with other SGOs. To solve a violated constraint, a SGO uses geometric patterns and/or process patterns as generalization methods. Although the SGO's enrichment can be considered as additional work for cartographers, it is important to mention that this work is done only once, and created SGOs are used several times (i.e., every time map generalization occurs).

The difficulty in automating the generalization process, particularly the difficulty in formalizing generalization knowledge, has led some researchers to propose the generalization approach, allowing the division of tasks between humans and computers (Weibel 1991; Weibel and Buttenfield 1988). This division of tasks is achieved by using interactive generalization tools. By using these tools, the cartographer can: select the objects or groups of objects to be generalized; determine the generalization operations, algorithms, and parameters to be applied; assess the quality of the generalized data; and finally, detect the conflicts and eliminate them. Unfortunately, even today, the cartographer's work is partially lost because typically, only the generalized map is stored in the database, not the tasks that he performed (ex. chosen the sequence of generalization algorithms). Sometimes it is even necessary to redo all the tasks in order to generate a map of the same territory, but only using another scale. The philosophy underlying the SGO

approach is to either recover or reproduce the work necessary for map generalization, and to store it. Therefore, the SGO can prove as a good means allowing to recover the work completed by the experts during generalization process. Figure 20 demonstrates the philosophy underlying the SGO's concept.

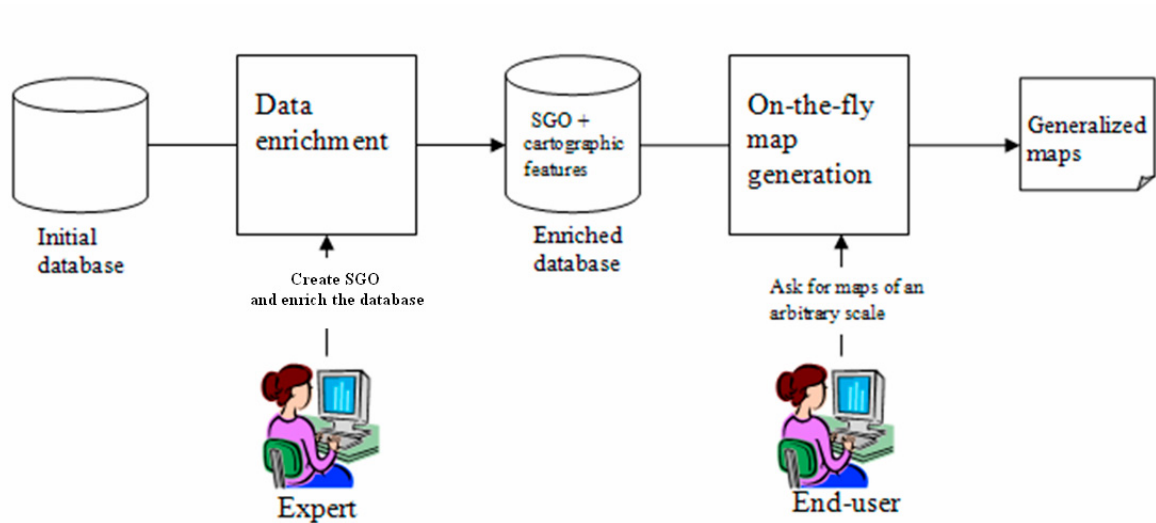


Figure 20- The SGO philosophy

4.3.2. SGO components

4.3.2.1. Geometric patterns

In any domain, the concept of pattern is related to the concept of recurrence. In our environment, several phenomena or objects have recurrent geometries. Consequently, when viewing the map, the eye discerns patterns of shape, orientation, connectedness, density, and distribution (Mackanness and Edwards 2002). For example, on a map, many buildings can have a similar shape. This similar shape represents a shape pattern (Figure 21, left picture). Thus, the geometric pattern (Figure 21, right picture) is a geometrical construction of a shape pattern (i.e., a geometrical construction of a recurring shape). The created geometric patterns are then stored in a geometric patterns database

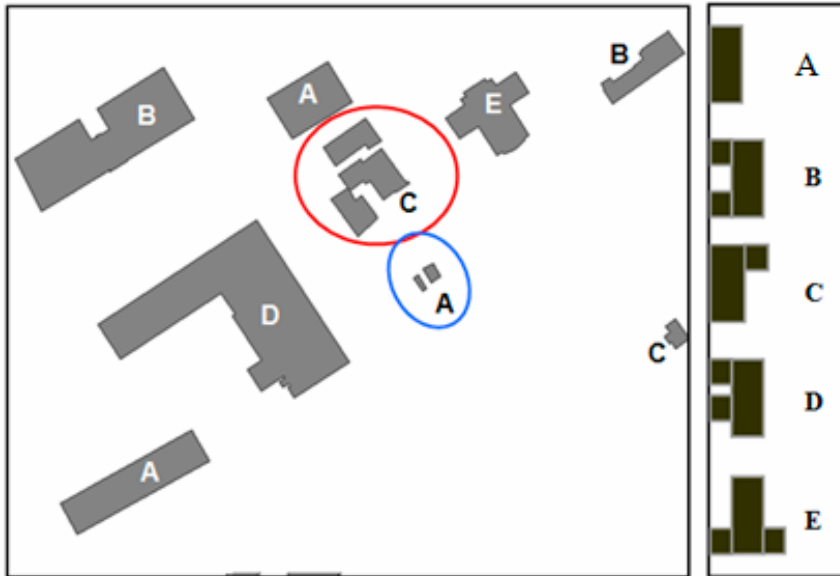


Figure 21- Detected patterns of shape on a map (left picture) and their corresponding geometric patterns (right picture).

The idea is to associate a geometric pattern with each cartographic feature (when it is possible). When a geometric pattern is associated with a cartographic feature, parameters like the feature's orientation, position, and size are defined and stored. Then, during the generalization process, instead of generalizing this feature using the traditional generalization algorithms, the feature is simply replaced by the associated geometric pattern (or its simplified form) according to the required level of abstraction. In order to match the geometric pattern to the geometry of the cartographic feature being replaced, the geometric pattern is adjusted using simple operations (rotation, positioning, and scaling) and stored parameters (e.g., position, size, and orientation). The use of a geometric pattern during a cartographic generalization process is illustrated in Figure 22.

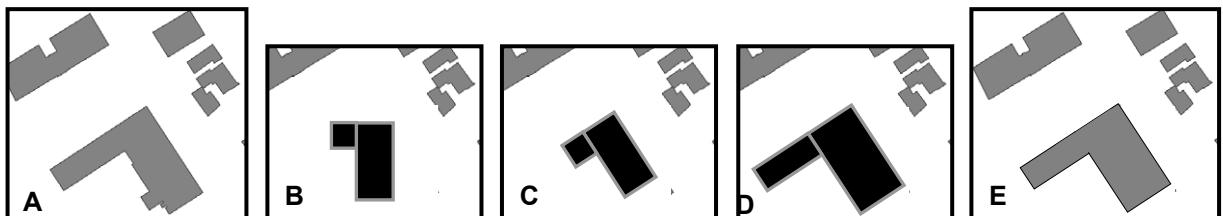


Figure 22- Use of a geometric pattern in a cartographic generalization process: A - Initial map; B – Initial feature replaced by a geometric pattern; C - Pattern's orientation; D - Pattern's scaling; E – Result

To make it flexible, a geometric pattern is composed of one or several primitives (basic elements that make up a geometric pattern). For example, geometric patterns of buildings can be composed of one or several connected primitives (e.g., juxtaposed rectangles). This flexibility facilitates the generalization of the geometric pattern, which in turn allows us to generate a pattern's simplified form. When generalizing a geometric pattern, the operation is under control thanks to geometric pattern structure (composed of connected primitives) and the coordination of the geometric pattern's adjustment operations ensured by the SGO. A geometric pattern is generalized using simple generalization operations. These operations are: elimination of a primitive, combination of primitives, and changing the size of a primitive. A UML class model of our data structure for a geometric pattern is illustrated in Figure 23. Figure 24 shows some geometric pattern's generalization operations. Finally, Figure 25 shows us a geometric pattern, along with its simplified forms.

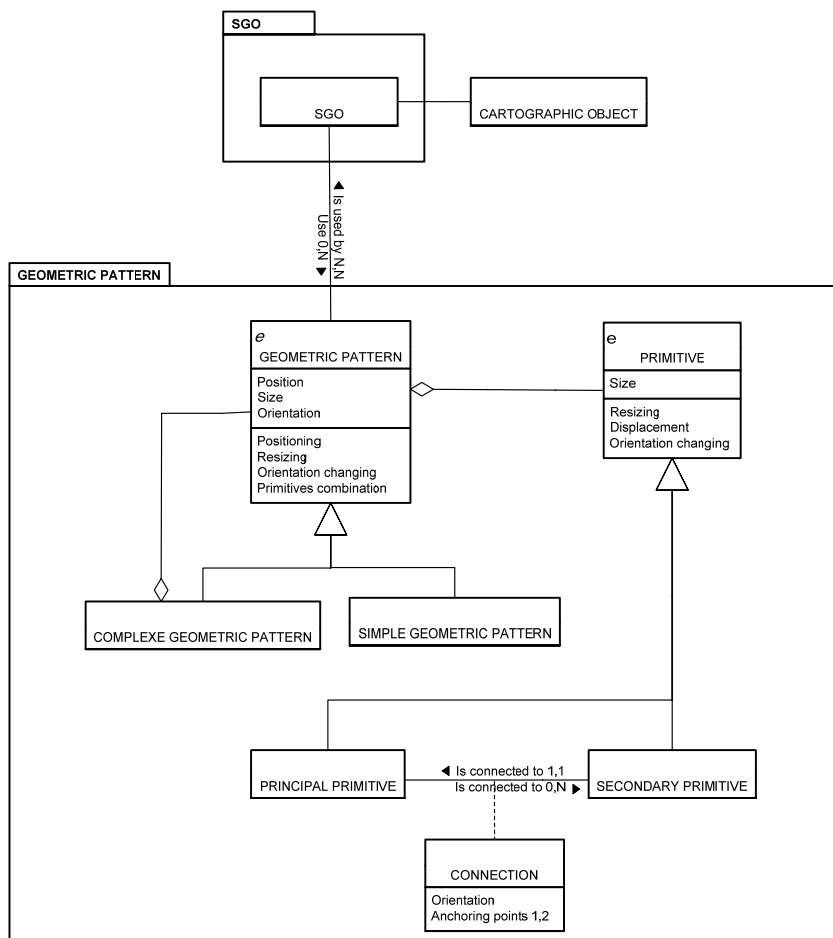


Figure 23- UML class diagram for the geometric pattern's data structure.

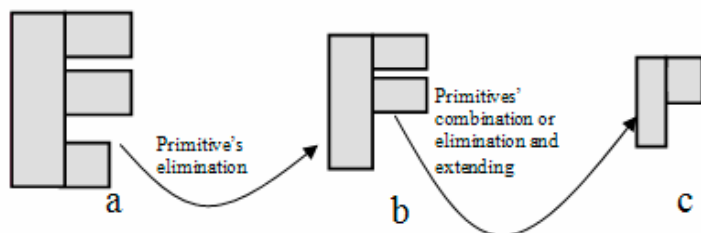


Figure 24- Geometric pattern's generalization operations.

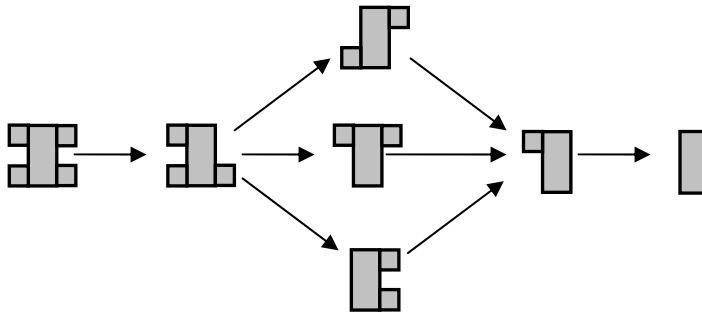


Figure 25- A geometric pattern and its different simplified forms obtained by iterative generalization (primitive's elimination) (Sabo *et al.* 2005a).

The concept of geometric pattern is not limited to buildings. It may be also applied to other classes of objects (e.g., road cloverleaves, cul-de-sacs, and silos). For instance, a highly sinuous road segment, such as those found in the Alps, can be generalized using a geometric pattern composed of several successive bends (Figure 26). This geometric pattern can be adjusted using the same operations as those used for geometric patterns of buildings.

The flexibility of geometric patterns and the simplicity of the associated operations (e.g., operations used to adjust or simplify a geometric pattern) allow us to simplify and accelerate the generalization process. Our experiments with a dataset from Quebec City have shown that one geometric pattern and its simplified forms allow replacing 2844 individual buildings or 365 groups of buildings appearing at several different scales. In addition, adjusting a geometric pattern is more than 20 times faster than simplification operations used in traditional automatic generalization. For more information about geometric patterns, see Sabo *et al.* (2005a, 2005b).

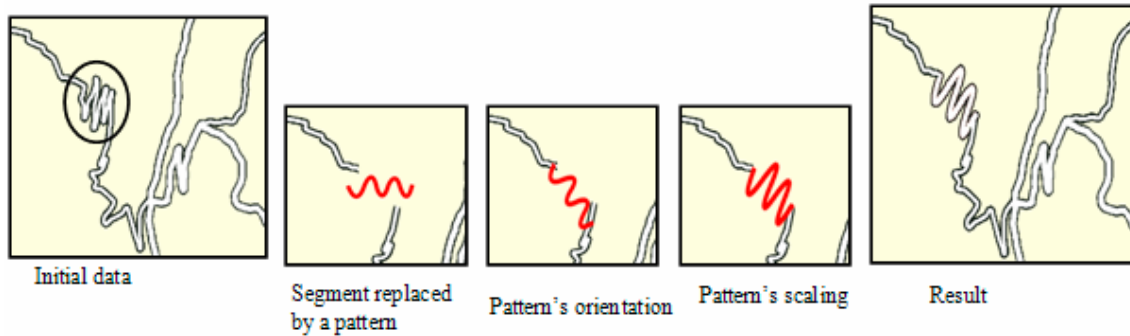


Figure 26- Generalization of a road segment using a geometric pattern (Weibel *et al.* 2002).

4.3.2.2. Process patterns

All map features do not lend to the use of geometric patterns in the same manner. For example, for features having complex, non-recurring shapes such as rivers, it is difficult, or even impossible, to find a shape pattern, consequently it is impossible to use a geometric pattern. In addition, certain patterns only appear on at small scale (e.g., at a very small scale, almost all buildings are represented by a rectangle). Therefore, the principal idea behind our approach is to use generalization algorithms in place of, or as a complement to, geometric patterns in these circumstances.

Under similar conditions, objects with similar geometric, semantic, and spatial context characteristics will be generalized in approximately the same way. Generalization of these objects would require the application of the same group of algorithms, and probably in the same sequence. Accordingly, the idea is to create, for each group of objects with similar characteristics, a set of generalization algorithms. For example, we will have different sets of algorithms for buildings having orthogonal angles, depending upon whether they are aligned or scattered. Contrary to the algorithms for scattered buildings, aligned buildings will require a set of algorithms including aggregation operation. We call such a set of generalization algorithms a "process pattern" since they can be repeated again and again with different parameters for different cartographic features. Thus, a process pattern is a recurrent group of generalization algorithms and sequences that is used to generalize several map features with similar characteristics. Like for geometric patterns, process patterns are also based upon the principle of recurrence. They typically apply to groups of

similar objects, and contrary to the more traditional algorithmic-based approach, they are not forced to apply to the complete map layer or to an entire object class.

Process patterns allow us to simplify the choice of generalization algorithms during the database enrichment phase, as well as to overcome the limits of current cartographic generalization with respect to the choice of generalization algorithms and their sequences. The automatic choice of algorithms and their sequences is a complex task whose results are sometimes dubious. Therefore, the choice can be made during a database enrichment process in order to continue benefiting from the cartographers' expertise. With the integration of several algorithms in a process pattern, we now simply select the one we want during the enrichment phase, instead of having to individually choose for each cartographic feature the algorithms necessary for each generalization operation.

Process patterns allow us to take into account the dependency that exists between various generalization algorithms, given the inter-dependence of many of their operations, as it is the case for simplification and smoothing operations. The simplification of certain linear objects, such as rivers, generally increases their angularity, which is not acceptable for this kind of object. To correct this situation, objects are smoothed. Smoothing provides a nice aesthetic appearance by decreasing the angularity caused by the simplification operation. In order to be able to consider the dependency that exists between these generalization operations, they need to be integrated into the same process pattern, which can provide a select synchronization of the operations.

In some cases, it is important to break down complex generalization operations into more simple ones. This, in turn, allows a few of these simple operations to be executed during the database enrichment phase that precedes generalization. For example, displacement is composed of many simple operations including: the proximity calculation and conflict detection, calculation of the displacement vector, the displacement itself, as well as the evaluation of the displacement's consequences. By breaking down the displacement into several simple operations, we can determine such things as the proximity relationships and the direction of displacement in the event of a possible proximity conflict during the data enrichment phase. The other operations can be included in a process pattern and executed on-the-fly during the generalization process.

A process pattern can implement either one or several generalization operations at once. In the case where it uses only one, it is called a *Simple Process Pattern* (e.g., an isolated building that does not require a contextual generalization operation, such as aggregation, can only implement a simplification operation when other generalization operations applied to the individual features, such as exaggeration, are neglected). However, when the process pattern implements more than one generalization operation, it is called a *Complex Process Pattern* (e.g., where a process pattern of a building requires several operations such as simplification, exaggeration, and displacement for its generalization). *Complex Process Patterns* are composed of several *Simple Process Patterns*. Moreover, the latter can consist of either a *Single-method* or *Multi-methods*. In contrast to *Single-method* process patterns, those having *Multi-methods* make use of several algorithms for the same generalization operation. For example, certain simplification algorithms, such as Douglas&Peucker, are not suited for great scale changes. Instead they can easily be used in combination with another algorithm that is better suited to great scale changes. Thus, depending upon the importance of the scale change, either one or the other algorithm can be used. These distinctions make the use of algorithms easier and flexible.

A process pattern can be applied to a group of map features (e.g., aligned buildings) in order to complete a contextual generalization operation (e.g., aggregation). In this case, the process pattern is called *Group Process Pattern*. But if the process pattern is applied to a single feature, it is an *Individual Process Pattern*. The UML class model of Figure 27 shows the various types of process patterns.

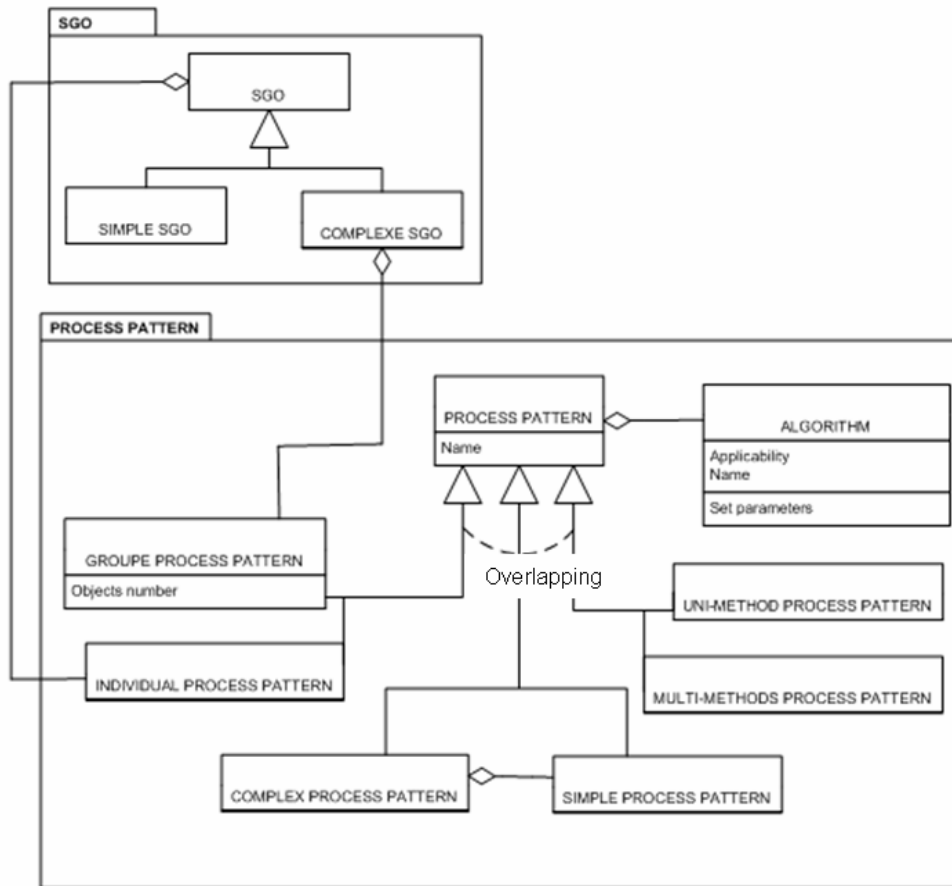


Figure 27- UML class model of the process pattern.

4.3.2.3. Spatial integrity constraints

According to João (1998), geometric data models for generalization are only effective if they can record spatial relationships between features. For several databases, these spatial relationships can be deduced through simple spatial queries. Unfortunately, certain important relationships are not explicit in spatial databases (e.g., building alignment). Not only are existing methods not able to adequately detect all the necessary spatial relationships, but this detection is also very complex and time-consuming sometimes, and usually requires a human operator to validate the results. This situation is not acceptable for an on-the-fly map generalization process; not only must the results be instantaneously available, but human intervention is not possible. For these many reasons, the relationships must be explicit and generated during the database enrichment process.

Detection of spatial relationships is a necessary condition, but is insufficient for a generalization process. In order to coordinate generalization processes, these spatial relationships must be expressed in the form of spatial integrity constraints. Moreover, these constraints must be satisfied during the generalization processes. SGO spatial integrity constraints are rules that must always be satisfied by a SGO during a generalization process, in order to ensure coherence and reliability of the generated map's features (e.g., aligned buildings must remain aligned after generalization). Therefore, during the creation of a SGO, the spatial relationships considered to be relevant for cartographic generalization can be defined and expressed in the form of spatial integrity constraints. The quantity and type of constraints to be considered depends upon the characteristics of the cartographic features. This task is only done once during the database enrichment phase rather than having to be repeated every time generalization is necessary.

For the purpose of generalization, we distinguish two separate types of spatial integrity constraints. The first category is called "*Individual Constraints*," which are applied to single map features (e.g., minimal size). Such a constraint is related to the nature of the feature to which it is being applied. Secondly, we have the "*Group Constraints*" that are applied to groups of map features (e.g., proximity constraints and alignment constraints). Contrary to an *Individual Constraint*, a *Group Constraint* applies to the relationships existing between several SGOs. We differentiate two types of *Group Constraints*:

- The *Binary Constraint* connects two SGOs (e.g., proximity constraint). During the generalization process, the violation of this kind of constraint is solved through direct interaction between the SGOs to which it applies.
- *N-ary Constraints* connect several SGOs (e.g., an alignment constraint or an inclusion constraint).

When generalization is guided by constraints, these constraints must be adapted to different situations. Map feature transformations generally affect several constraints. For example, the elimination of an object that is connected to another object by a proximity constraint involves the deactivation of this constraint. However, for certain constraints, the

elimination of one object does not necessarily mean the disappearance of the constraint, but rather its transformation. The constraint's transformation during the generalization process is an extremely complex process. For instance, if several buildings are aligned along a rectilinear road, and at a specific Level of Detail, the road disappears, what will happen to this constraint? First of all, this constraint specifies that the buildings be lined up at approximately the same distance from the road (because the road is rectilinear). Furthermore, if we look at this constraint as being only one, the disappearance of the road should have automatically caused the disappearance of the constraint, leading to the disappearance of the alignment constraint. However, if this constraint is considered to be two distinct constraints (alignment and proximity), the elimination of the road will only generate the disappearance of the proximity constraint. Thus, to ensure proper generalization, constraints must be sufficiently flexible to adapt to new situations. Therefore, the breakdown of complex constraints into several simple constraints is necessary. Thus, complex constraints will be an aggregate of simple constraints.

4.3.3. Types of SGOs

For objects that are well-suited for geometric patterns (e.g., simple buildings), the corresponding SGO may only contain geometric patterns. For objects that are not suited for geometric patterns (usually because their shape is not repetitive), such as hydrographic elements, the associated SGO can only contain process patterns. Finally, some map features can have a SGO that contains both geometric patterns and process patterns.

According to the number of objects with which it is associated, a SGO can be either simple or complex. A *Simple SGO* is associated with a single cartographic feature (e.g., a single building). In order to preserve the properties of a map's group of features (e.g., to maintain the alignment of selected buildings), and to facilitate the generalization operations applied to a group of features (e.g., aggregation), several SGOs can be grouped together to form a complex SGO. A *Complex SGO* can be defined as an aggregate of several SGOs (simple and/or complex), created to handle a *N-ary Constraint*. It is not necessary to create a complex SGO to handle a Binary constraint (e.g., proximity constraint).

The same SGO can be part of several *Complex* SGOs. Each time a SGO enters into the composition of a *Complex* SGO, its value increases to give it more importance when faced with an elimination of map features. During the generalization process, SGOs are selected or eliminated according to their importance. What’s more, a complex SGO can be a member of another complex SGO; there is no limit on number of SGO imbrications. Figure 28 shows the data structure to manage SGOs.

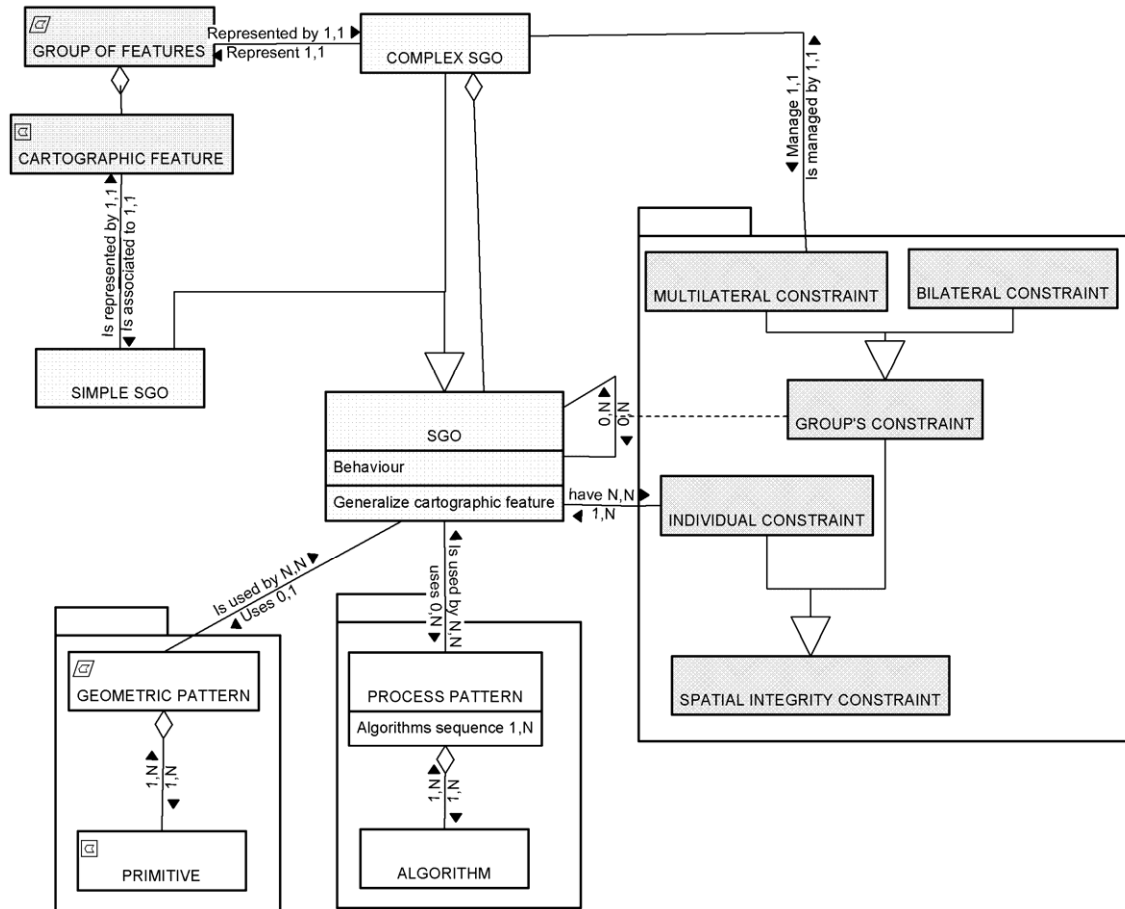


Figure 28- Simplified UML class diagram of the data structure to manage SGO.

4.3.4. Generalization process and the SGO

In the previous sections, we presented the static aspect of an SGO (i.e., a structure that permits the storage of all database enrichments). In this section, we present the dynamic aspect of an SGO (i.e., coordination of the generalization process). We will present the relationships and interaction between SGOs and SGO behaviour patterns.

4.3.4.1. Relationships and interaction between SGOs

In order to coordinate the generalization process, while taking into account the environment, SGOs have various types of relationships that allow them to interact during the generalization process. Depending upon the type of SGOs involved, there are three categories of relationships between them:

- *Vertical Relationship*: an asymmetrical relationship between a Complex SGO and its members. This relationship is asymmetrical because the Complex SGO has a hierarchical level higher than that of its members. This is a father-son relationship where interaction can be both to and from the Complex SGO.
- *Horizontal or Transversal Relationship*: a relationship between SGOs having the same hierarchical level (but it is not a father-son relationship). It is the kind of relationship that links SGO members belonging to the same Complex SGO. It can connect two Simple SGOs, two Complex SGOs, or a Complex and a Simple SGO. The only conditions are that the two SGOs are not in a father-son relationship and that they have the same hierarchical level. Thus, this kind of relation cannot connect a Complex SGO with its members.
- *Oblique Relationship*: a non-father-son relationship between SGOs from different hierarchical levels. This type of relationship can be specified during the data enrichment phase or generated automatically during the generalization process after the aggregation of several cartographic features by its Complex SGO. Following this aggregation, the Complex SGO inherits the relationship (e.g., proximity constraints) of its SGO members representing the aggregated objects. Therefore, the Complex SGO will be in relationship with SGOs of a lower hierarchical level.

4.3.4.2. Behaviour pattern

SGOs are linked to the exact geometry of a cartographic feature and include expert knowledge in order to facilitate the generalization of the associated feature. This enrichment permits the creation of a complete behaviour pattern that allows the SGO to

know how to generalize the associated cartographic feature at smaller scales. A SGO's behaviour pattern controls the order in which spatial integrity constraints are checked, the way in which a violated constraint is satisfied, and the interactions between various SGOs.

The main role of behaviour patterns is to coordinate generalization operations. They know which generalization method (geometric pattern vs. generalization algorithm) to use during a generalization process. The behaviour pattern also allows the SGO to decide which geometric pattern to use for a given scale and how to adjust or simplify the selected geometric pattern. The need to apply a given operation is controlled by the behaviour pattern. If there are several algorithms for the same operation, the behaviour pattern selects the most suitable one. The SGO's behaviour pattern also allows to determine the parameters of a generalization algorithm during the generalization process. Given that the sequence of the algorithms is previously defined in the process pattern, the behaviour pattern allows the SGO to change this predefined sequence if necessary.

The SGO behaviour pattern is determined by the type of the SGO (e.g. Simple or Complex), the type of the cartographic feature associated with the SGO (e.g. building feature) and the type of constraints which are applied to this SGO (e.g. alignment constraint). For a Simple SGO, the behaviour pattern allows to coordinate generalization operations applied to a single cartographic feature (e.g. simplification, displacement). For a Complex SGO, in addition to allowing generalization operations (e.g. aggregation), the behaviour pattern also enables the SGO to coordinate and/or facilitate the actions of its members in order to ensure the coherence of the group. For example, the Complex SGO of aligned buildings can facilitate the displacement by computing the axis of alignment. When a SGO (Simple or Complex) is member of a complex SGO, it has two distinct behaviours patterns. The first behaviour pattern which enables the SGO to solve its personal constraints and the second which allows it to respect the constraint imposed to all members of the group (e.g. alignment).

There are also general behaviours; common to almost all SGOs (e.g., all SGOs check their internal constraints, such as minimum size, before checking their external constraints, such as proximity). The more specific behaviours depend primarily upon the type of SGO constraint (e.g., behaviour pattern of aligned buildings).

4.4. Applying the SGO concept

To validate the SGO approach within the proposed framework, two prototypes were developed: (1) a *SGO Creation Engine*, and (2) an *On-the-fly Map Generalization System*. The two prototypes were developed in Java and use Open Source API. The *SGO Creation Engine* is devoted to the creation of different SGOs and their components (geometric patterns, process patterns, etc.), while the *On-the-fly Map Generalization System* is dedicated to on-the-fly map generalization based upon multi-agent technology, using the previously created SGOs as input. Figure 29 shows the simplified architecture of the developed prototypes.

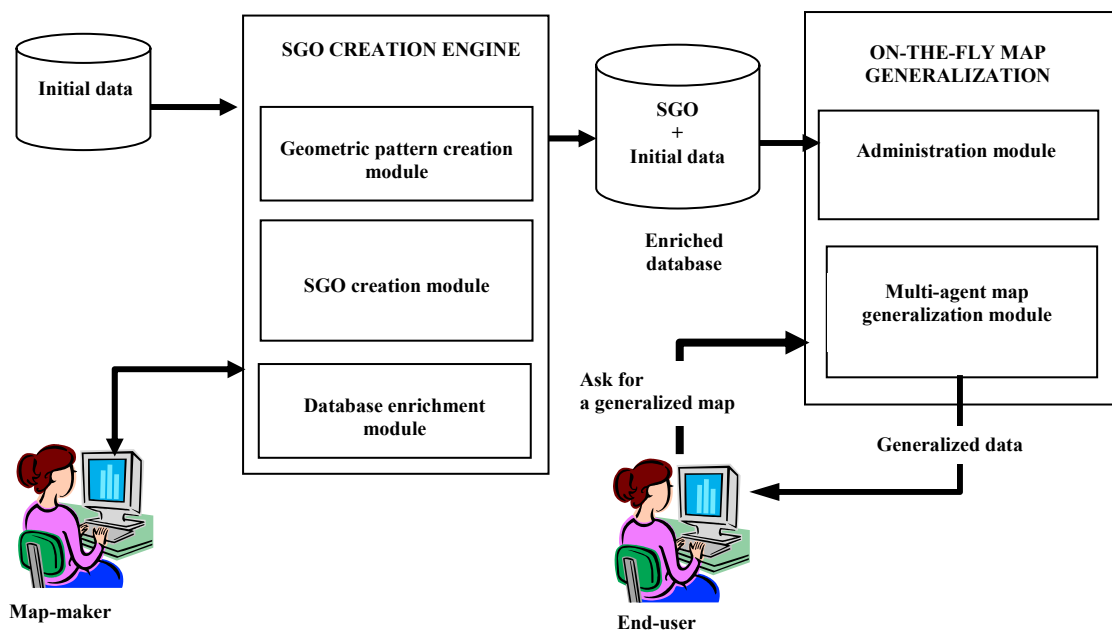


Figure 29- A simplified view of the architecture including both developed systems: the SGO Creation Engine and the On-the-fly Map Generalization System.

4.4.1. A prototype for the creation and enrichment of SGOs (The *SGO Creation Engine*)

This prototype is based upon the Java Topology Suite (JTS) library (JTS 2006). The JTS library is a Java API of 2D spatial predicates and functions using an explicit precision model and robust geometric algorithms. JTS implements the Simple Features Specification for SQL, published by the Open GIS Consortium. For the visualization of geographic data,

the prototype uses the GeoTools API (GeoTools 2006), which is an Open Source Java GIS toolkit for developing OpenGIS compliant solutions.

In order to test the prototype, we used data in a Shapefile format (ESRI) from the Ministry of National Defense (Canada). These data are on a scale of 1:1000 and cover a part of Quebec City. Two classes of objects, namely "buildings" and "roads", were used for the SGO creation. All occurrences of these two classes are represented by polygons.

With this prototype, the creation of SGOs is achieved in three main phases: creation of geometric and process pattern databases; importation of map features and creation of simple SGOs; and enrichment of created SGOs.

4.4.1.1. Creation of geometric and process pattern databases

During this phase, the geometric and process patterns are created in order to be used by the SGO. The prototype allows the automatic creation of each geometric pattern and its various simplified forms. As an input, only the number of the geometric pattern's primitives (i.e., number of juxtaposed rectangles) is used (cf. figure 25). Based on this number, specified by the Operator, the system creates the geometric pattern. After that, this geometric pattern is automatically simplified in an iterative way in order to create all its various possible simplified forms. The geometric pattern and its simplified forms are stored in the geometric pattern's database. The result is represented in a hierarchical tree of simplified shapes (derived from the pattern, see Figure 30-D), where we can choose the desired geometry having the required level of detail.

This phase also allows us to interactively form the various process patterns. A process pattern is created by simply selecting which generalization operations are to be included from a list box. The order of the selected operations determines the sequence in which they are to be applied during a generalization process (of course this order can be changed). For each selected operation, a list of available algorithms is presented. From this list, algorithms which will be used to carry out each generalization operation are selected. Finally, the completed process pattern is then added to a list of available patterns.

4.4.1.2. Importation of a map's features and the creation of simple SGOs

The SGO creation phase is fully automated. This prototype allows us to import buildings and road-network objects from our initial data, which is in a Shapefile format. For each imported feature, a corresponding SGO is created. A Simple SGO (a SGO of a single cartographic feature) is created for each building feature. What's more, this automatically created simple SGO is not enriched (i.e., without a geometric pattern, a process pattern, or a spatial integrity constraint). The imported road-network features are polygonal and each street is made of polygonal segments and junctions. Thus, for each road element (segment or junction), a simple SGO is created and automatically stored in a SGO database.

4.4.1.3. Enrichment of created SGOs

During this phase, building and road-network SGOs (created during the previous phase) are enriched. During this phase, building Complex SGOs (e.g., a SGO representing a group of aligned buildings) are interactively created by grouping several SGOs together. For each created complex SGO, a spatial integrity constraint is defined, that allows us to preserve the created group during the generalization process. Currently, three Group constraints (i.e., applied to a group of map features) are implemented: the proximity constraint (i.e., between two buildings, or between a building and a street), the alignment constraint, and the block constraint (i.e., allows buildings to remain within the same block). However, created building SGOs (simple and complex) are completed by specifying their components (i.e., geometric patterns, process patterns, and spatial integrity constraints). Geometric patterns are associated with SGOs by a simple "drag & drop" operation. Furthermore, they are matched to cartographic feature geometry by clicking several control points on the cartographic feature. The system automatically extracts and stores in the SGO the parameters (i.e., feature's position, size, and orientation) that allow us to match the geometric pattern to the geometry of the cartographic feature to be replaced. These parameters will be used later on by the SGO during the on-the-fly map generalization process in order to replace the cartographic feature by its geometric pattern. This phase also allows to associate a process pattern with a specific SGO by interactively selecting the needed process pattern from the list created during the first phase and including it in the given SGO.

Contrary to the enrichment of building SGOs, the enrichment of road-network SGOs is based upon an automatic process. During the enrichment of the road-network SGOs, segments and junctions imported from the previous phase are automatically reconnected in order to recreate the road network. Based upon simple road-network SGOs (i.e., SGOs representing segments and junctions), a complex SGO representing each road is individually created. Values are automatically allotted to each road SGO according to their importance in the network (e.g., number of connections and width). What's more, the automatically allotted parameters can be changed interactively if necessary.

The figure 30 illustrates the interface of the developed SGO's Creation Prototype. This prototype allows us to create and enrich SGOs in a three step process. Thereafter, the SGOs are used in an On-the-fly Map Generalization System which will be described in the next section. The UML sequence diagram of the enrichment of database is illustrated on the figure 31

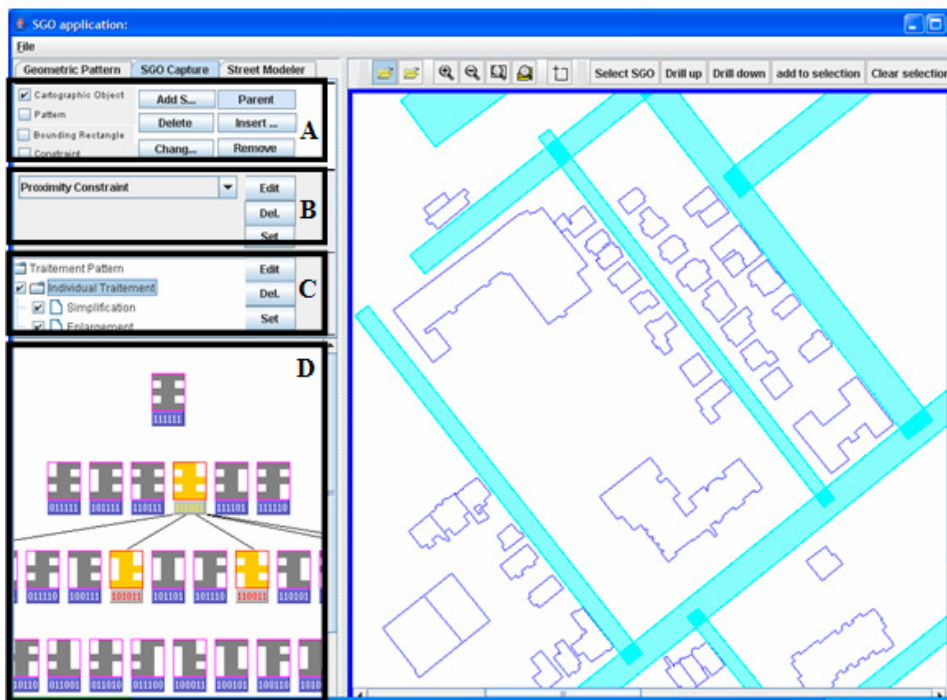


Figure 30- User interface of the SGO creation prototype. A- Complex SGO creation panel; B- Constraint specification panel; C- Process pattern specification panel; D- Geometric pattern (on top of the hierarchy) and its simplified forms.

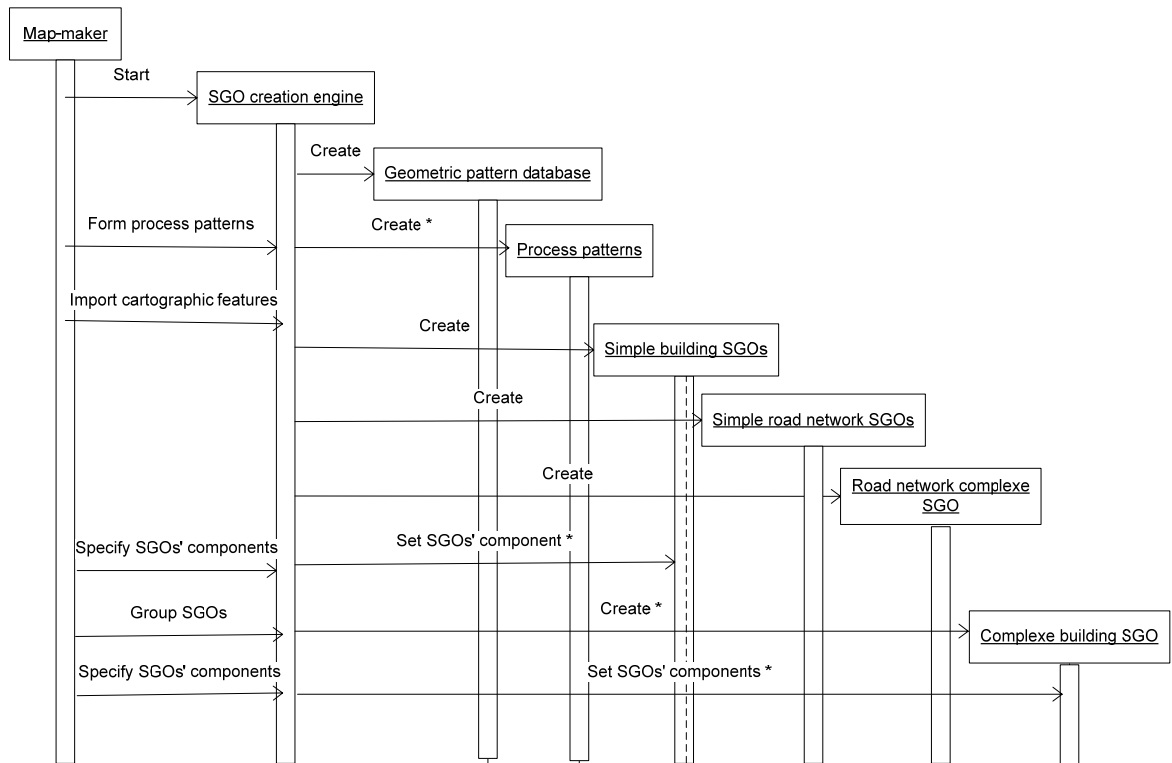


Figure 31 UML sequence diagram of the enrichment of database (the stars represent interactive processes)

4.4.2. On-the-fly Map Generalization System

Like the previous one, this prototype is developed in Java. Its visualization tool is based upon the GeoTools library. The multi-agent module for this prototype is developed using the JADE environment (JADE 2006). JADE (Java Agent Development Framework) is a software framework implemented in Java. It simplifies the implementation of multi-agent systems through a middleware implementing the FIPA (Foundation of Intelligent Physical Agents) specifications (FIPA 2006). This prototype supports on-the-fly production of maps at arbitrary scales. As an input, the prototype uses the SGOs that were created using the first prototype (see Section 4.4.1.). The present prototype is composed of two main modules: administration and on-the-fly map generation.

4.4.2.1. The Administration Module

This module imports the SGO's file and allows the cartographer to specify constraint thresholds (e.g., minimum separation between two map symbols). Currently, five constraint thresholds (minimum object size, minimum segment length, maximum object orientation deviation, proximity, and maximum object displacement) can be specified using the administration module. During the specification, the system checks the compatibility of each value and emits a visual signal (i.e., color change) in the case of an incompatibility. The multi-agent module is launched using this administrative module. During the launching process, the system agent in charge of the multi-agent module initialization is created.

4.4.2.2. The On-the-fly Map Generation Module

This module is composed of three sub-modules:

- The *Coordination Sub-Module* contains the *System Agent* responsible for the initialization of the other sub-modules. During the initialization, this agent creates all the other agents of the system. For each imported SGO, a corresponding SGO agent is created, and a behaviour pattern is automatically associated with it depending upon its constraints. The *System Agent* also allows the user to interact with the system (e.g., this agent intercepts the user's requests and extracts the needed parameters, such as the required map's scale). During a scale change, the *System Agent* converts all constraint thresholds for the requested scale. The converted thresholds are sent to SGO agents.
- The *Visualization Sub-Module* contains the *Drawing Agent*, and the User Interface that includes the navigation tools. The *Drawing Agent* is responsible for drawing the generalized data.
- The *Map Generalization Sub-Module* is in charge of the map generalization. This sub-module contains several SGO agents which communicate with each other in order to generalize the selected data. Generalized data are sent to the *System Agent* that transfers it to the *Drawing Agent*. Figure 32 illustrates the interface of the developed on-the-fly multi-agent map generalization prototype.

Using this prototype, a user (even if he has no knowledge about cartographic generalization) can generate maps at arbitrary scales. The user has no parameter to specify, given that all the parameters were specified by the system administrator. The system allows the quasi-instantaneous generalization of data. For example, the average time necessary to generalize a hundred map features is about 300 milliseconds. In terms of response time, this is compatible with the requirement for on-the-fly map generalization.

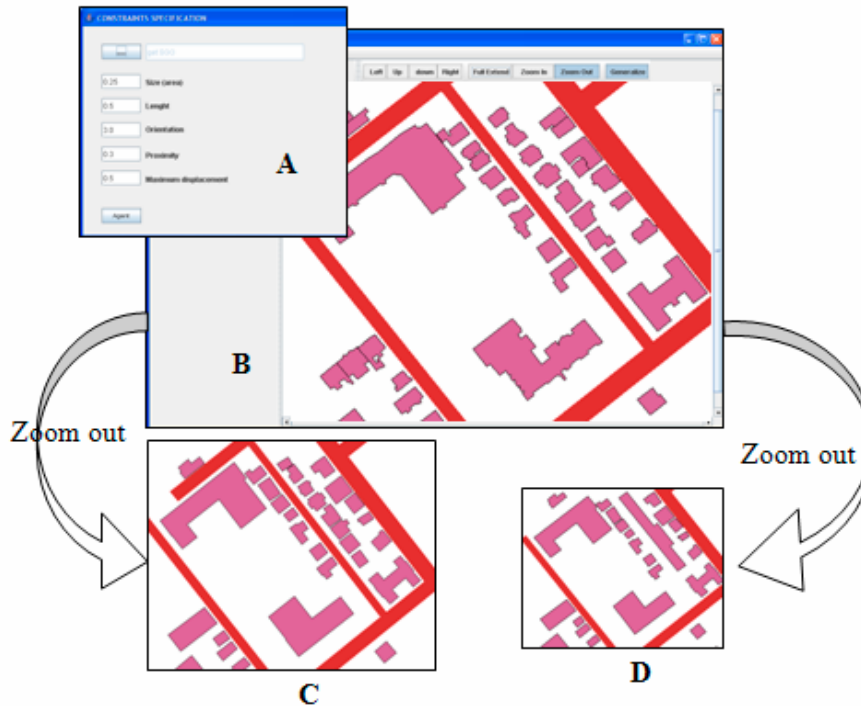


Figure 32- Interface of the administration (A) and map generation modules of the on-the-fly, multi-agent map-generalization prototype, and generalized data of different scales (C, D).

4.5. Conclusion

In this paper, we presented the SGO concept which integrates geometric patterns, process patterns, spatial integrity constraints, and behaviour patterns. The existence of two generalization methods, namely the use of geometric patterns and generalization algorithms in order to generalize cartographic features or geometric patterns provide a good flexibility of the generalization process. This approach is also based upon a process of enrichment that allows to make explicit, information which are implicit in the initial database (e.g., a building's alignment), and to introduce specific generalization knowledge (e.g., the choice

of the most appropriate generalization algorithms) into the data. This enrichment process overcomes many limitations of the current automatic generalization process, such as the problem of formalizing expert knowledge. In order to validate our SGO, two prototypes were developed. The first is needed to create a SGO database, while the second, using multi-agent technology, allows the creation of on-the-fly maps at arbitrary scales.

In addition to facilitating the generalization process, SGOs also speed up the whole process and allows on-the-fly map generalization. This significant reduction in processing time is due on one hand to the database enrichment phase, during which the process patterns and integrity constraints are defined and, on the other hand to the simplicity of the operations that are needed in order to adjust and simplify the geometric pattern.

Thanks to the geometric patterns, the use of SGOs can facilitate the data transfer through the Internet network. Indeed, in a client/server architecture, geometric patterns can be stored on the client side (as the use of fonts in a word processing application). Thus, instead of transferring a generalized cartographic feature from the server to the client side, only parameters (feature's position, size, orientation) allowing to replace it by its geometric pattern can be transferred. Using these parameters, the geometric pattern can be adjusted on the client side. The reduction of the traffic flow through the network, especially for on-line cartography applications, is of great importance given that the data transfer rate is known as being one of the main limitations for such applications (Bertolotto 2001, Buttenfield 2002).

At first sight, the effort required to create SGOs, which implies the choice of geometric patterns, process patterns, and the creation of spatial integrity constraints for each cartographic feature or group of features, may be perceived as the weak point in this approach. However, this work is done only once and is re-used several times, leading to significant overall savings in time. Moreover, the SGO creation tool greatly simplifies this task. Although many of these operations are interactive for the moment, some may eventually be automated. For example, Rainsford and Mackaness (2002) proposed a method for the automatic recognition of alphanumeric templates similar to some of our geometric patterns. In addition, certain operations, such as the choice of algorithms, are already performed during a conventional interactive generalization process. In this case, the

task will consist of creating a mechanism to record this work and to store it in a SGO database by coupling the developed system with an existing generalization system.

4.6. Acknowledgements

The authors wish to acknowledge contribution to this work by: the Canadian Network for Centers of Excellence in Geomatics (GEOIDE) which made possible the realization of this research within the framework of GEMURE and MUSCAMAGS projects; Canada NSERC Industrial Research Chair in Geospatial Database for Decision-Support; and the following partners: Research and Development Canada Defense, Natural Resources Canada, Natural resources, Fauna and Park Quebec and Intergraph.

4.7. References

- Allouche M.K. and B. Moulin. 2005. "Amalgamation in cartographic generalization using Kohonen's feature nets." *International Journal of Geographical Information Science* 19(8-9): 899-914.
- Armstrong M.P. 1991. "Knowledge classification and organization." In *Map Generalization: Making Rules for Knowledge Representation*, eds. Buttenfield B.P., R. McMaster and H. Freeman. Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley. 86-102.
- Baeijs C., Y. Demazeau and L. Alvares. 1995. "Application des systèmes multi-agent à la généralisation cartographique." In *Proceedings of 3èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-agents, AFCET&AFIA*, Chambéry. 163-176.
- Beard K.M. 1991. "Constraints on rule formation." In *Map Generalization: Making Rules for Knowledge Representation*, eds. Buttenfield B.P., R. McMaster and H. Freeman. Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley. 121-135.
- Bédard Y., E. Bernier and R. Devillers. 2002. "La métastructure VUEL et la gestion des représentations multiples." In *Généralisation et Représentation Multiple*, ed. Ruas A., Paris: Hermes science publications. 149-162.
- Bernier E., Y. Bédard and M. Lambert. 2003. "Cartographie sur demande sur le Web et bases de données multidimensionnelles. De la personnalisation par couche cartographique à la personnalisation par occurrences." *Revue Internationale de Géomatique* 13(3): 339-359.
- Bertolotto M. and M. Egenhofer. 2001. "Progressive transmission of vector map data over World Wide Web." *Geoinformatica* 5 (4): 345-373.

- Buttenfield B.P. 2002. "Transmitting vector geospatial data across the Internet. *GIScience 2002.*" *Boulder, CO, USA Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg. 51-64.
- Buttenfield B.P. and R. McMaster. 1991. "Preface." In *Map Generalization: Making Rules for Knowledge Representation*, eds. Buttenfield B.P., R. McMaster and H. Freeman. Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley. xiii-xvi.
- Cecconi A., R. Weibel and M. Barrault. 2002. "Improving automated generalisation for on-demand Web mapping by multiscale database." In *Proceeding of Symposium on Geospatial Theory, Processing and Applications*. Ottawa, Canada.
- Devogele T., T. Badard and T. Libourel. 2002. "La problématique de la représentation multiple." In *Généralisation et Représentation Multiple*, ed. Ruas A., Paris: Hermes science publications. 55-74.
- Devogele T., J. Trevisan and L. Raynal. 1996. "Processus de constitution d'une base de données multi-échelles." *Revue Internationale de Géomatique* 6(2-3): 249-263.
- Douglas D.H. and K.P. Thomas. 1973. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." *The Canadian Cartographer* 10(2): 112-122.
- FIFA. 2006. The Foundation for Intelligent Physical Agents. Available at <http://www.fipa.org/>.
- GeoTools. 2006. The open source java GIS toolkit. Available at <http://geotools.codehaus.org/Home>.
- GiMoDig. 2004. Geospatial info-mobility service by real-time data-integration and generalisation. Available at <http://gimodig.fgi.fi>.
- Jabeur N. 2006. *A Multi-agent System for On-the-fly Web Map Generation and Spatial Conflict Resolution*. PhD thesis, Université Laval. 275 p.
- Jabeur N., B. Boulekrouche and B. Moulin. 2006. "Using multi-agent systems to improve real time mobile map generation", In *Advances in Artificial Intelligence*, eds. Lamontagne L. and M. Marchand, Springer Verlag LNAI 4013. 37-48.
- JADE. 2006. Java Agent Development Framework. Available at <http://jade.tilab.com/>
- João E.M. 1998. *Causes and Consequences of Map Generalisation, Research Monographs in Geographic Information Systems*. London, Bristol, PA: Taylor & Francis. 266 p.
- Jones B. and M. Ware. 2005. "Map generalization in the Web age." *International Journal of Geographical Information Science* 19 (8-9): 859-870.
- Lamy S., A. Ruas, Y. Demazeau, M. Jackson, W.A. Mackaness and R. Weibel. 1999. "The application of agents in automated map generalisation." In *Proceedings of 19th International Cartographic Conference*. Ottawa. 2: 1225-1234.
- Li M., S. Zhou and B. Jones. 2002. "Multi-agent systems for Web-based map information retrieval." In *Proceedings of GIScience 2002: Geographic Information Science*,

- Second International Conference, Boulder, CO, Lecture Notes in Computer Science* 2478: 161–180.
- Mackaness W.A. 1994. "An algorithm for conflict identification and feature displacement in automated map generalization." *Cartography and Geographic Information Systems* 21(4): 219-232.
- Mackaness W.A. and Edwards G. 2002. "The Importance of modeling pattern and structure in automated Map generalization." In *Proceedings of Joint ISPRS/ICA Workshop on Multi-scale Representations of Spatial Data*, July 7-8, Ottawa, Canada.
- Martel C. 1999. *Développement d'un Cadre Théorique pour la Gestion des Représentations Multiples dans les Bases de Données Spatiales*. M.Sc. thesis, Sciences géomatiques, Laval, Québec. 139 p.
- McMaster R. 1991. "Conceptual framework for geographical knowledge." In *Map Generalization: Making Rules for Knowledge Representation*, eds. Buttenfield B.P., R. McMaster and H. Freeman. Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley. 21-39.
- Müller J.C. 1991. "Building knowledge thanks for rule based generalization." In *Proceedings of International Cartographic Conference*, Bernemouth. 257-256.
- Newell A. 1990. *Unified Theories of Cognition*. Harvard University Press, Cambridge MA. 549 p.
- Nickerson B.G. and H. Freeman. 1986. "Development of a rule-based system for automatic map generalization." In *Proceedings of the 2th International Symposium on Spatial Data Handling*, Seattle, Washington. 537-556.
- Peterson M. 2003. "Preface." In *Maps and the Internet* ed. Peterson M. London: Elsevier. ix-x.
- Rainsford D. and W.A. Mackaness. 2002. "Template matching in support of generalization of rural buildings." In *Proceedings of Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada.
- Ruas A. 1999. *Modèles de Généralisation de Données Géographiques à Base de Contraintes et d'Autonomie*. Doctorat thesis, Sciences de l'Information Géographique, Université de Marne-la-Vallée. 323 p.
- Sabo M.N., Y. Bédard and E. Bernier E. 2005. "Methodology for developing a database of geometric patterns to better support on-the-fly map generalization." In *Proceedings of International Cartographic Conference*, 9-16 July, Coruna, Spain.
- Sabo M.N., A. Cardenas, Y. Bédard and E. Bernier. 2005. "Introduction du concept de patrons géométriques et application aux bâtiments afin de faciliter leur généralisation cartographique à la volée." *Geomatica* 59(3): 295-311.
- Sarjakoski T. and T. Kilpelainen. 1999. "Holistic cartographic generalization by least squares adjustment for large data sets." In *Proceedings of 19th International Cartographic Conference*, Ottawa, Canada. 1091–1098.
- Sester M. 2000. "Generalization based on least squares adjustment." *International Archives of Photogrammetry and Remote Sensing* 33 (B4/3): 931-938.

- Internet World Stats. 2006. *World Internet Users and Population Stats*. Available from <http://www.internetworldstats.com/stats.htm>.
- Timpf S. 1998. *Hierarchical Structures in Map Series*. Doctorat thesis, Departement of Geoinformation, Technical University of Vienna, Vienna. 124 p.
- Van Elzakker C. 2001. "Use of maps on the Web." In *Web Cartography: Developments and Prospects*, eds. Kraak M.J. and A. Brown. London: Taylor & Francis. 21-36.
- Van Oosterom P. and V. Schenkelaars. 1995. "The development of an interactive multi-scale GIS." *International Journal of Geographical Information Systems* 9(5): 489-507.
- Vangenot C. 1998. "Représentation multi-résolution, concepts pour la description de bases de données avec multi-représentation." *Revue Internationale de Géomatique* 8(1-2) : 121-147.
- JTS. 2006. JTS Topology Suite. Available at <http://www.vividsolutions.com/jts/jtshome.htm>.
- Weibel R. 1991. "Amplified intelligence and rule-based systems." In *Map Generalization: Making Rules for Knowledge Representation*, eds. Buttenfield B.P., R. McMaster and H. Freeman. Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley. 172-186.
- Weibel R., E. Bernier, Y. Bédard and A. Cecconi. 2002. "La généralisation à la volée." In *Généralisation et Représentation Multiple*, ed. Ruas A., Paris: Hermes science publications. 319-335.
- Weibel R. and B.P. Buttenfield. 1988. "Map design for geographic information systems." In *Proceedings of GIS/LIS*, 30 November 2 December, San Antonio, Texas. 350-359.
- Weibel R. and G. Dutton. 1998. "Constraints-based automated map generalization." In *Proceedings of the 8th International Symposium on Spatial Data Handling*, Vancouver. 214-224.
- Weibel R. and G. Dutton. 1999. "Generalising spatial data and dealing with multiple representations." In *Generalising Spatial Data and Dealing with Multiple Representations*, eds. Longley P.A., M.F. Goodchild, D.J. Maguire and D.W. Rhind. John Wiley & Sons. 125-155.
- Wonnacott L. 2000. The speed of business: If your pages are slow, your customers will go. Available at <http://www.infoworld.com/articles/op/xml/00/09/11/000911opsavvy.html>.
- Young J. and S. Smith. 2006. Akamai and Jupiter Research Identify '4 seconds' as the New Threshold of Acceptability for Retail Web Page Response Times. Available at http://www.akamai.com/html/about/press/releases/2006/press_110606.html.
- Zona Research Inc. 1999. The Economic Impacts of Unacceptable Web Site Download Speeds. Available at http://www.webperf.net/info/wp_downloadspeed.pdf.

Chapitre 5 La généralisation à la volée à l'aide des objets auto-généralisants (SGO)

ON-THE-FLY MAP GENERALIZATION SYSTEM USING SELF-GENERALIZING OBJECTS (SGO) AND MULTI-AGENT TECHNOLOGY

Auteurs : Mamane Nouri SABO, Yvan BÉDARD, Bernard MOULIN, Eveline BERNIER³

(Sera soumis à Geoinformatica)

Remarque : *Cet article présente en détail le système multi-agent de généralisation à la volée (Section 5.3) qui a été brièvement abordé au chapitre précédent (Section 4.4.3). Comme ce système est basé sur l'approche SGO, l'article aborde brièvement ce concept déjà introduit au précédent chapitre (Section 4.3).*

Résumé de l'article: *Cet article propose un système multi-agent de généralisation à la volée qui permet de générer des cartes à des échelles arbitraires. Ce système est basé sur une approche qui encapsule à la fois des patrons géométriques (qui sont des formes communes à plusieurs objets cartographiques), des algorithmes de généralisation, et des contraintes d'intégrité spatiales dans un même objet appelé objet auto-généralisant (Self Generalizing Object- SGO). Un SGO est créé et associé à un objet cartographique, ou un groupe d'objets, lors d'une phase l'enrichissement afin de faciliter et d'accélérer leur généralisation. Puis, chaque SGO créé est transformé en agent logiciel dans le système multi-agent développé, afin de leur donner de l'autonomie. Des comportements complets sont automatiquement associés à chaque agent. Ces comportements permettent aux agents de savoir comment généraliser les objets, ou les groupes d'objets de la carte.*

Abstract:

This paper proposes a multi-agent, on-the-fly map generalization system which permits the generation of maps of arbitrary scales. This system is based upon an approach encapsulating geometric patterns, which are forms common to several cartographic features, generalization algorithms, and spatial integrity constraints within the same object called a Self-Generalizing Object (SGO). A SGO is created and associated with a cartographic feature, or a group of features, during an enhancement phase in order to facilitate and to accelerate their generalization. Then, in order to become autonomous, each created SGO is transformed into a software agent in the developed, multi-agent system. Complete behaviors are automatically associated with each agent. These behaviors allow agents to know how to generalize the features, or the groups of objects on the map.

Keywords:

On-the-fly map generalization, Self-Generalizing Object, geometric pattern, generalization algorithm, spatial-integrity constraint, multi-agent system, multiple representations.

5.1. Introduction

Geographical Phenomena are scale-dependent and numerous applications in the climate, water resources, agriculture, forestry, transportation, and land and urban planning sectors require changing Levels of Detail (LoD) (Muller et al. 1995). Each application requires a specific level of detail for the data that is sufficient, adequate, and thus optimal. Having too much detail results in losing sight of important objects, whereas too little information is not helpful enough (Sester 2005). Whatever the level of detail, maps must remain readable, thus, certain details must be removed in order to adapt the map's contents, but the message must be effective. The generalization process is usually necessary to adapt the map's contents.

In conventional cartography, map generalization is responsible for reducing a map's complexity through a scale reduction process, emphasizing the essential while suppressing the unimportant, maintaining logical and unambiguous relationships between the map's objects, while preserving the aesthetic quality (Weibel and Dutton 1999). During the past

three decades, several important research projects have been devoted to automatic cartographic generalization (Douglas and Peucker 1973, McMaster 1987, João 1998, McMaster 1991, Muller et al. 1995, Beard 1991, Ruas 1998, Brassel and Weibel 1988, Armstrong 1991, Lamy et al. 1999, Harrie and Sarjakoski 2002, Sester 2005, Allouche and Moulin 2005, Jabeur 2006, etc.).

For a long time, research on cartographic generalization concentrated on the creation of algorithms in order to produce generalization operations in a fully automatic manner. During this period, proposed solutions usually relied upon a class-based approach whereby an operation affected all the occurrences of an object's class (e.g. all the buildings), regardless of their individual characteristics. Although the algorithms constitute the building blocks of the automatic map generalization, they alone cannot provide for the full automation of this process. For that, a mechanism which permits the coordination of the generalization process (e.g. choice of the appropriate algorithms) with the context and characteristics of the map's objects is necessary.

In order to determine the required algorithms, as well as their sequences and corresponding parameters, knowledge-based approaches and specifically expert systems have been used (Allouche and Moulin 2005). Throughout the last few decades, several research studies have been devoted to knowledge-based generalization approaches (Armstrong 1991, Beard 1991, Battenfield 1991, McMaster 1991, Nickerson and Freeman 1986, and Weibel 1991). In theory, a knowledge-based approach requires that our knowledge of the generalization process be arranged into a sequence of reasoning paths, each leading up to a particular decision or procedure for generalization to take place (Müller 1991). Generalization knowledge can be obtained directly from the cartographer or extracted from existing maps (for example, one such method involves the use of artificial intelligence techniques, such as genetic algorithms and neural networks). In such a system, knowledge is typically formulated into rules. Unfortunately, however, the subjective and holistic characteristics of the cartographic generalization make the rules formalization very complex.

According to Beard (1991), rule-based systems are rigid because rules are bound to particular actions. To make the systems more flexible, Beard (1991) proposed to replace rules by constraints, which allow us to take into account generalization factors such as the

map's purpose, scale, and graphic limits. To date, several research studies have been dedicated to the constraint-based generalization approach (e.g. Beard 1991, Weibel and Dutton 1998, Ruas 1999), in which constraints are explicitly formalized (Weibel and Dutton 1998) so that they may be able to handle possible conflicts during the generalization process. What's more, all constraints must be satisfied, or their violation resolved. To resolve a constraint violation, generalization operations are applied to the data. Nevertheless, the application of such an operation can, sometimes, generate conflicts that are even more severe. This is why, in such an approach, algorithms and their sequences must be chosen so as to minimize conflict. Unfortunately, however, finding a strategy that minimizes conflict generation is still very difficult and costly (Allouche and Moulin 2005).

The difficulty in developing fully automated solutions leads some researchers to turn their efforts towards techniques of interactive generalization (Müller et al. 1995). The interactive approach allows the generalization tasks to be distributed between the human operator and the computer. Thus, the low level tasks (e.g. simplification) are carried out by the system, while those that are high level (e.g. the choice of generalization algorithms and their sequence) are performed by the human operator (Weibel 1991). For the majority of researchers in this field, the interactive approach is an alternative while waiting for the creation of a fully automated generalization system. In the beginning, researchers expected to gradually extract and formalize cartographers' knowledge. However, because the problem of formalizing generalization knowledge is not yet solved, intensive human intervention is unfortunately needed for the interactive approach.

Recently, the development of software agents in computer science has engendered the emergence of generalization approaches based upon multi-agent systems (Baeijs et al. 1995, Lamy et al. 1999, Li et al. 2002, Duchêne 2003, Jabeur 2006, Jabeur et al. 2006). In such approaches, cartographic features are represented by software agents in order to make them autonomous. Thus, various software agents, representing cartographic features, communicate with each other during the generalization process in order to solve their conflicts. Unfortunately, on-the-fly multi-agent-based generalization approaches are not intended to carry out on-the-fly map generation (Jabeur et al. 2006).

With the last decade's technological developments, and the Internet in particular, geographic data has become more and more accessible to a wider range of users. This has also led to the emergence of new map-on-demand applications like Web-mapping and SOLAP (Spatial On-line Analytical Processing). An on-demand map is a map generated according to the specific requirements of the user. SOLAP can be defined as a visual platform especially built to support rapid and easy spatio-temporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays (Bédard 1997). For more information about SOLAP, see Stefanovic et al. (2000) and Rivest et al. (2001).

According to Jones (2005), there are several types of public access Web-mapping sites that allow a user to zoom in and out of a particular region. But at present, this is usually based upon stepping around independent, pre-generalized datasets which may differ markedly in the degree of their generalization. It would be advantageous to be able to change the level of detail on such systems in a smooth and progressive manner, rather than having to make the quantum-leap changes that often characterize current approaches (Jones and Ware 2005). However, in order to do that, a process of generalization is necessary. But, according to the interactive nature of these new map-on-demand applications (e.g. Web-mapping), generalization process must be on-the-fly.

On-the-fly map generalization is the creation, in real time, of a cartographic product adapted to its scale and purpose, following a user's request and generated from a large scale database (Weibel et al. 2002). Thus, this process is subjected to a temporal constraint. According to Newell (1990), in order to maintain a train of thought, a response time between 0.1 and 10 seconds is needed to perform cognitive tasks. In addition, on-the-fly map generalization does not produce a second data set, as this would introduce redundant data (Van Oosterom and Schenkelaars, 1995).

Currently, only a partial solution exists for deriving new data, on-the-fly, from a single detailed database. The lack of a complete solution is due to the complexity of the generalization process. Existing solutions are generally limited to one or two generalization tasks, i.e. simplification and/or selection. However, generalization should not be limited to a simple operation, but must also take into account other factors, such as the object's

surrounding environment. Unfortunately, automatic generalization approaches which take into account the object's context (e.g. multi-agent-based generalization approaches) are very complex. In their initial form, they cannot provide at the same time the performance, the flexibility and the degree of automation required by on-the-fly map generalization.

As we previously mentioned, cartographic generalization is extremely time consuming process, given that it is generally done manually or using a semi-automatic approach. Since it is currently impossible to create on-the-fly maps for arbitrary scales starting from a single, large-scale map, several authors propose to store the results of a generalization process (generalized maps) in a multiple representations database (MRDB) (Bédard et al. 2002, Devogele et al. 1996, Kilpelainen 2000, Timpf and Frank 1995, Vangenot 1998). A multiple representations database permits to store several representations of the same territory, in the same database. Although various representations of the same map's object can be linked together, this is not currently the case in several applications given that only maps at different scales are linked. In addition, in order to change the level of abstraction, one map is usually totally replaced by another at a different scale. The MRDB approach is associated with several problems such as data redundancy, updates' propagation, and linking the representations of a same object. Furthermore, the MRDB approach inherits the problems related to the cartographic generalization since the maps, which make it up, are generally derived using a generalization process.

Considering the problems of cartographic generalization (e.g. complexity and incomplete automation) on the one hand, and of multiple representations (e.g. redundancy, data update propagation) on the other hand, some researchers have recently proposed a third means which tries to minimize the problems related to both approaches (Cecconi et al. 2002; GiMoDig 2004, Jabeur et al. 2006). For example, a solution proposed by Cecconi et al. (2002) is based upon a multi-scale database that includes a minimum of two different scales of the same territory. These scales are linked together. Following a user's request, the most suitable scale is selected, and if necessary, the result is refined with the help of generalization operations. The combination of generalization algorithms and multiple representations usually minimizes the effort of generalization when the difference between the scales of the requested map and the source map is very small. However, in order to

significantly facilitate the generalization process, the multi-scale database must include several scales, which still brings back the problems of multiple representations (e.g. redundancy).

In order to support an on-the-fly map generalization process which requires a quasi-instantaneous response time, no human intervention, and a minimum of data redundancy, we recently proposed a generalization approach based upon the concept of Self-Generalizing Objects (SGO), which integrate three main fundamental elements: geometric patterns (shapes common to several cartographic features), generalization algorithms, and spatial integrity constraints (Sabo et al. 2007). To validate this concept, we also developed an experimental prototype based upon multi-agent technology. This prototype permits the transformation of SGOs into software agents in order to generate maps at arbitrary scale, on-the-fly. The goal of this paper is to present this prototype, as well as the tests that we conducted. We begin with a short overview of the various concepts related to Self-Generalizing Objects (SGO). We describe the multi-agent, on-the-fly map generalization prototype (i.e. architecture and implementation) that we developed, in addition to our experiments and results. Finally, we conclude by discussing the challenges which remain to be tackled.

5.2. Self-Generalizing Objects (SGO)

5.2.1. Concept of SGO

As mentioned in the introduction, the key motivation for this study is the proposal of a generalization approach that can meet the main requirements of on-the-fly map generalization (i.e. quasi-instantaneous response time and complete automation). Thus, in addition to being faster, this map generalization approach must also overcome the main limitations to actual automatic map generalization (e.g. problems of choosing algorithms and their sequence which will result in human intervention). To cope with these requirements, our approach (Self-Generalizing Object-SGO) is based upon data enrichment.

A SGO is a structure associated with a cartographic feature, which is enriched during an enrichment phase in order to facilitate and accelerate automatic generalization of the cartographic feature. A Self-Generalizing Object (SGO) is composed of three main components:

- Geometric patterns (shapes common to several cartographic features);
- Process patterns (a set of generalization algorithms and their sequence);
- Spatial integrity constraints.

For each map's object, a SGO is created, as well as a link between the two. According to the map's object characteristics (e.g. geometric, semantic and spatial), its SGO is enriched by specifying its geometric pattern, process pattern, and spatial integrity constraints. A simplified UML class model of a SGO is shown in figure 33. This data enrichment allows the acceleration of the generalization process as certain operations are carried out in advance (e.g. spatial integrity constraints like alignment or proximity are defined during the data enrichment phase), while it also overcomes certain limitations of map generalization (e.g. the choice of generalization algorithms and their sequence). It is obvious that this data enrichment is an additional task, but it is done only once, whereas the results are used several times and allow the generation of arbitrarily-scaled maps. In the next section, we describe each component of a SGO.

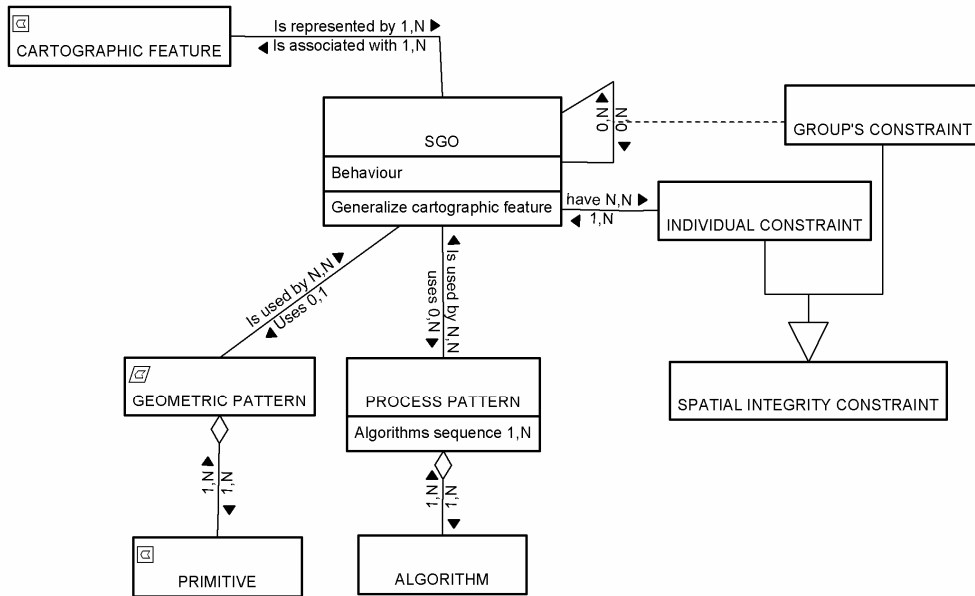


Figure 33- Simplified UML class model of a SGO associated with a cartographic feature

5.2.2. SGO components

5.2.2.1. Geometric patterns

Our entire social, cultural, economic, and physical environment develops by means of patterns. In any domain, the recurrence is the idea behind patterns. According to the architect, Alexander et al. (1977), a pattern is a solution to a contextual problem, whereby each pattern describes a problem which occurs over and over again in the environment, and then describes the core of the solution to the problem, in such a way that the solution can be used a million times over, without ever doing it the same way twice.

On a map, several shapes may be recurring. For instance, several buildings on a map may have a common shape (e.g. an 'E' shape). Here, this common shape can be considered as a repeated pattern. According to this common shape (pattern), we can construct a geometric pattern; therefore, the geometric pattern is a geometrical construction of a recurring form. To make them more flexible, geometric patterns are composed of primitives, which are the atomic geometric elements of the geometric pattern. For example, in order to construct the geometric patterns of buildings, we can juxtapose several rectangles (Figure 34(a)). These

created geometric patterns are then stored in a geometric patterns database. Thus, for each map's object for which we can find a corresponding pattern, we will have an associated geometric pattern, selected from the patterns database.

When a geometric pattern is associated with a map's object during the SGO creation phase, parameters like the object's orientation, position, and size are defined and stored. After the map's scale changes, instead of generalizing the object by using the generalization algorithms, the SGO can replace the object's geometry with a geometric pattern (a simplified representation). In order to fit the replaced geometry, the geometric pattern must be adjusted. This adjustment is carried out using simple operations like rotation, positioning or scaling, and stored parameters (object's orientation, position, and object's size). The simplicity of these operations provides for a high level of efficiency using the SGO (e.g. 0.4 seconds to replace 2244 buildings by geometric patterns as opposed to 10.8 seconds to simplify the same number of buildings using the generalization algorithm, i.e. 27 times faster).

If, at the time a geometric pattern is replacing a map's object, the pattern geometry presents details below the threshold of perception (e.g. a pattern's primitive is too small), it may be simplified. This can be done because geometric patterns are equipped with simple generalization operations (elimination of a primitive, combination of primitives, or changing the size of a primitive). For example, according to the required scale, a geometric pattern with the 'E' shape can be simplified to take the 'F' form, an 'L' shape, or an 'I' shape (Figures 34(b), 34(c)). For more information about buildings' geometric patterns, see Sabo et al. (2005).

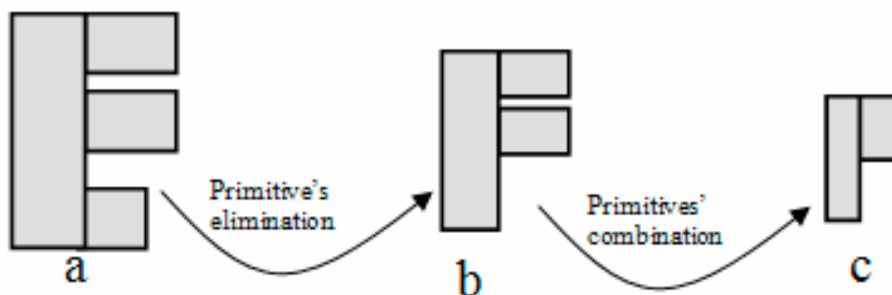


Figure 34- A geometric pattern and its generalization (Sabo et al. 2007).

5.2.2.2. Process pattern

Though the use of geometric patterns during the generalization process has a very high potential, this method does not allow the complete generalization of all object classes. In fact, geometric patterns cannot be used at all by some object classes. For example, this is the case with rivers that have irregular, non-repeatable complex shapes. Moreover, numerous other objects can only be used under specific conditions (e.g. at very large scales, buildings with extremely complex shapes cannot be replaced by geometric patterns). Besides, there are already algorithms that work quite well for certain cartographic objects under distinct geometric conditions. The main idea is to use generalization algorithms as a complement to geometric patterns.

Normally, with existing automatic generalization systems, algorithms are represented in the form of methods for an entire class of objects (e.g. all buildings are equipped with the same generalization algorithms). Nevertheless, all the objects, even in the same class, should not always be generalized using the same algorithm (e.g. the simplification algorithm that would be used to generalize a building with orthogonal sides will not be the same as for a non-orthogonal building). Consequently, algorithms must be chosen according to the characteristics of the map's objects. However, the automatic choice of generalization algorithms and their sequences is a very complex and time-consuming task, the results of which are uncertain. The best way to overcome these limitations is, in the enrichment phase, to specify the most appropriate generalization algorithms, as well as their sequences, for each object or group of objects on the map.

The choice of generalization algorithms and their sequences depends to a large extent on the semantic, geometric, and spatial characteristics of the objects, as well as the required degree of generalization. Furthermore, on a map, several objects can have almost the same characteristics; accordingly, they will have common generalization algorithms. As a result, we will observe a repetition of certain groups of algorithms which can be used during the generalization of this map. Each repeating algorithm group makes up a process pattern. Therefore, instead of choosing each individual generalization algorithm for each object's occurrence (one or more algorithms for each generalization's operation), we simply select a process pattern (a group of algorithms) that was created beforehand and which is suitable

for several objects with similar characteristics. Hence, the basic philosophy of the process pattern is the same as that which governed the concepts of geometric patterns, the recurrence.

5.2.2.3. Spatial integrity constraints

According to João (1998), geometric data models for generalization are only effective if they can record spatial relationships between features. Unfortunately, important relationships from the cartographic generalization point of view (e.g. the building alignment) are not explicit in the database. In addition, it is known that the detection of some of these spatial relationships (e.g. proximity detection using Delauney triangulation) is a complex and costly operation, and for others, current algorithms do not always give good results (e.g. building alignment detection). For these reasons, the solution is to detect these relationships during the data enrichment process.

Accordingly, in our SGO approach, the spatial relationships considered to be relevant for cartographic generalization can be defined during the data enrichment phase. The quantity and types of constraints to be considered depends upon the characteristics of the map's objects, as well as the needs and objectives of the data producer. For each considered relationship, a spatial integrity constraint can be created. Spatial integrity constraints are rules that must always be satisfied by a SGO during a generalization process in order to ensure logical and unambiguous relationships between map's objects, in addition to preserving the aesthetic quality and legibility of the generated map.

Constraints can be applied to a single cartographic feature (e.g. minimal size), or to a group of cartographic features (e.g. alignment). We differentiate between two types of spatial integrity constraints: *Individual constraints* (e.g. minimal size constraint) and *Group constraints* (applied to the relationships existing between several SGOs). Group constraints can be *Binary* (when they connect a couple of SGOs; e.g. the proximity constraint), or *N-ary* (when they connect several SGOs; e.g. alignment or inclusion constraints).

5.2.2.4. Behavior pattern

SGOs are linked to the exact geometry of an object in order to create a complete behavior pattern that allows the feature to know how to generalize itself using a combination of geometric patterns, generalization algorithms, and spatial integrity constraints. The SGO behavior patterns correspond, in a certain way, to the procedural constraints described by Beard (1991). According to Beard (1991), procedural constraints control the order and interaction of procedures or operations, as well as the order in which constraints are satisfied. In the case of the SGO, the behavior pattern controls the order in which spatial integrity constraints are checked, the selection of necessary operations and algorithms from the process pattern, the interactions between various SGOs, in addition to the contents of the exchanged messages between SGOs at the time of the generalization process. The behavior pattern allows the SGO to find the way a cartographic feature must be generalized, using geometric patterns or generalization algorithms.

What's more, the behavior pattern can be general or specific. The general behavior pattern is common to all SGOs, it is the behavior which forces a SGO to check its internal constraints before checking its proximity constraints. A specific behavior pattern is conditioned by a particular spatial integrity constraint (e.g. the behavior which allows a SGO to check the alignment constraint and to solve a conflict when this constraint is violated). In our prototype, behavior patterns are not specified during the SGO creation phase. They are automatically created during the initialization of the on-the-fly map generalization system.

5.2.3. Types of SGOs

Depending upon the complexity of the object's shape, the associated SGOs can be: 1) *SGOs with geometric patterns only*. This is the case of SGOs representing the majority of buildings; 2) *SGOs with geometric patterns and process patterns* (a set of generalization algorithms). For instance, a building with a complex shape, at a large scale, may use a generalization algorithm for its first generalization (to transform the very detailed polygon into a simplified polygon), while a later generalization may use a geometric pattern (to replace the detailed building by a geometric pattern in a smaller scale); and 3) *SGOs with*

process patterns only. These SGOs are used for objects with particular geometries (e.g. rivers) for which we cannot find any prevalent characteristic and consequently any pattern.

According to the number of cartographic features for which they are created, the SGOs can be *simple* or *complex*. A *simple SGO* is a SGO created for a single map's object (e.g. a single building). Thus, a *complex SGO* is a SGO created for a group of map's objects in order to handle *N-ary constraints* (e.g. alignment constraint). As a SGO is created for each object of a map, a complex SGO can be considered as being an aggregate of SGOs. The same SGO can be part of several complex SGOs. There is no limit to the number of Complex SGOs in which a SGO can participate. Each time that a SGO becomes part of a Complex SGO, it obtains an additional weight. The weight determines the importance of a SGO regarding generalization process. Then again, a Complex SGO can be part of another Complex SGO. In theory, there is no limit to the number of imbrications of SGOs.

In addition, with regards to handling generalization constraints, a complex SGO is able to manage the data common to all the group members (e.g. object's density, the average distance between objects), in order to better carry out the contextual generalization (e.g. aggregation, elimination). The necessity of categorizing groups of map's objects has been largely documented during the past decade (e.g. Ruas 1999; Lamy et al. 1999, Duchêne 2003, Jabeur 2006). In our approach, the creation of a *N-ary constraint* (see 5.2.2.3) systematically engenders the creation of a *Complex SGO*.

5.2.4. Creation of Self-generalizing objects' (SGOs')

5.2.4.1. SGOs' creation prototype and data used

To create the database of SGOs, we developed a prototype (*SGOs' Creation Prototype*) based upon objects of the JTS library (The Java Topology Suite). The Java Topology Suite is a Java API of 2D spatial predicates, and functions using an explicit precision model and robust geometric algorithms (JTS 2006). JTS implements the Simple Features Specification for SQL, published by the Open GIS Consortium. For geographic data visualization, the SGOs' creation prototype uses the GeoTools API (GeoTools 2006). This prototype allows us to: 1) automatically create the geometric patterns' database; 2) automatically create

simple SGOs from imported cartographic features; and 3) interactively enrich the created SGOs. This enrichment enables us to create complex SGOs, to associate geometric patterns and process patterns with created SGOs, and to specify spatial integrity constraints. The created SGOs are serialized in a file which is then used by the On-the-fly Map Generalization System (see Section 5.3). To populate the SGOs' database, data in the format of Shapefile (ESRI) from the Ministry of National Defense (Canada) has been used. This data is at the scale 1K, partially cover the territory of Quebec City. Two map layers, the building and the street layers, were used for the SGOs' creation. All the map's objects in these two layers are represented by polygons.

5.2.4.2. Buildings' SGOs creation

For buildings, Shapefile data is used as an input. During the importation, the system transforms each building feature into a Simple SGO without a geometric pattern, a process pattern, or a spatial integrity constraint. Interactively, using the SGOs' creation prototype, the data capture Operator creates Complex SGOs by grouping several SGOs. Geometric patterns, spatial integrity constraints, and process patterns (a set of generalization algorithms) can be associated with the created SGOs (simple or complex). When linking geometric patterns to a SGO, all parameters (e.g. the map's object's position, orientation, and size) which were used for the geometric pattern adjustment are automatically extracted and stored within the SGO. These parameters will be used later on, during the on-the-fly map generalization process. Currently, three inter-SGO constraints are implemented: 1) the proximity constraint (between two buildings or between a building and a street); 2) the alignment constraint, and 3) the block constraint (which allows buildings to remain in a block).

5.2.4.3. Road network SGOs' creation

The road network features are in a polygonal structure, because the input data is provided at a very large scale (1K). Each street is represented by several polygons (i.e. many segments and junctions per street). The imported polygons from a Shapefile are automatically classified into segments and road junctions. The decision to classify a polygon as a segment or as a junction is primarily based upon the number and the type of connections, as well as

the size of the polygon. Segments and junctions are automatically connected. Weights are allotted automatically to each street according to their importance (depending on the street length, width, number of connections, etc.) in the network. The Operator also has the possibility of changing automatically selected parameters (e.g. to change a street's weights). The streets' weights let us know which street will be selected at a given scale during the generalization process. In the system, all segments and junctions are managed like Simple SGOs, whereas the streets are managed like Complex SGOs, consisting of several Simple SGOs.

5.3. Applying the SGO approach to on-the-fly map generalization

In Section 5.2, we presented the concept of SGO, which encompasses geometric patterns, process patterns, as well as spatial integrity constraints. SGOs are created for each object or group of objects, of a map. Created and enriched SGOs are injected into an on-the-fly multi-agent map generalization system in which they are transformed into software agents in order to give them autonomy and communication capabilities. In this section, we describe the architecture of this multi-agent map generalization system, as well as its implementation.

5.3.1. System architecture

The prototype is developed in Java and is based upon several open source libraries. It is composed of two main modules: the administration module and the on-the-fly map generation module. The general architecture of the system is shown in Figure 35. In this section, we describe the two modules that compose this architecture.

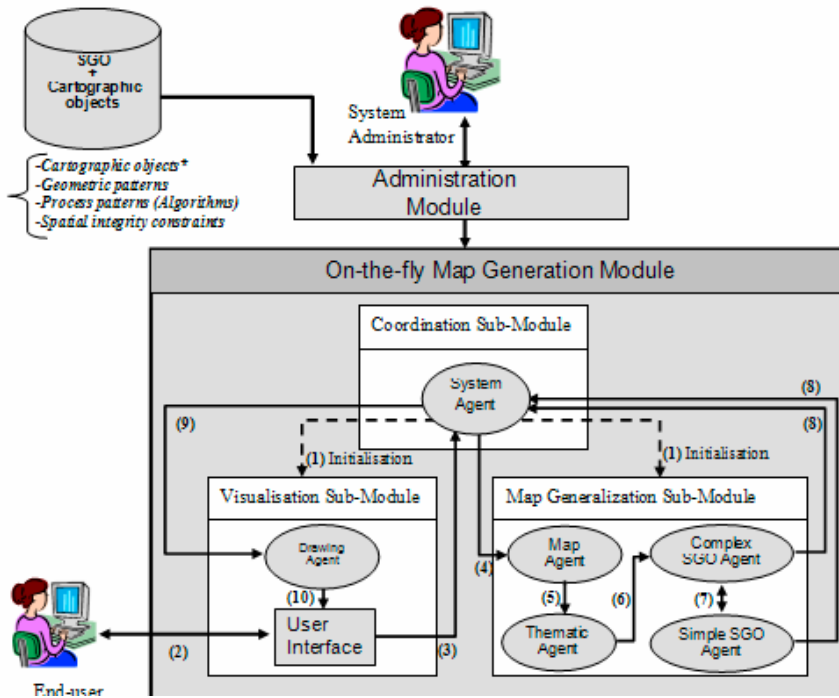


Figure 35- Overview of the on-the-fly map generation system architecture (numbers in parentheses indicate the sequence of interactions in the system).

**In the current architecture, since cartographic objects occupy very little disk space (few kilobits), we decided to integrate them into the SGOs' database.*

5.3.1.1. Administration Module

This module validates the SGOs' file, produced during the SGOs' creation phase (see Section 5.2.4), and then imports it. It also authorizes the specification of the thresholds of each spatial integrity constraint used by SGOs (e.g. minimum separation between two map symbols used for the proximity constraint). Currently, five constraints' thresholds (object's minimum size, segment's minimum length, object's maximum orientation deviation, proximity, and object's maximum displacement) can be specified using the administration module. When specifying thresholds, the system checks the validity of the input values using the default values encoded in the system. The default values are parameters usually used for cartographic generalization (e.g. a 0.2 mm threshold of perception must be respected between two map symbols). Currently, the default values cannot be modified by the user.

Separating this module and the generalization module enables us to limit access to the constraint parameters, and relieves end-users from configuring the system. The parameters' configuration requires knowledge about the generalization process, which is not the case of every user. The on-the-fly map generation module is started from this module.

5.3.1.2. On-the-fly Map Generation Module

This module, also developed in Java, is based upon the JADE (Java Agent Development Framework) platform (Jade 2006), and on objects of the JTS library (JTS 2006). JADE is a software framework, fully implemented in Java that simplifies the implementation of multi-agent systems through a middleware. JADE implements FIPA's (Foundation of Intelligent Physical Agents) specifications. The On-the-fly Map Generation Module is composed of three sub-modules:

- The *Coordination Sub-Module*. This sub-module contains the *System Agent*, which is the first agent created when starting the On-the-fly Map Generation Module. The Coordination Sub-Module is responsible for the initialization, which is entirely automated, of other sub-modules in the On-the-fly Map Generation Module. This sub-module is also responsible for the geometric patterns database creation. The geometric patterns, which are used by the SGOs are stored in binary format (the geometry is not stored), and their geometry is generated on-the-fly during the generalization process. But, if necessary (e.g. when the represented territory contains a large number of SGOs using geometric patterns), these geometric patterns geometry can be generated during the system initialization in order to accelerate the generalization process;
- The *Visualization Sub-Module*. This sub-module contains the *User Interface* and the *Drawing Agent*. The User Interface allows users to interact with the system; among other things, it allows them to activate and deactivate map layers, to select a part of the territory to be generalized. In addition to being equipped with navigation tools, such as zoom in, and zoom out, this visualization interface is based upon the GeoTools API, which is an Open

Source Java GIS toolkit for developing OpenGIS compliant solutions. The user interface is shown in Figure 36. The Drawing agent is responsible for drawing the generated data.

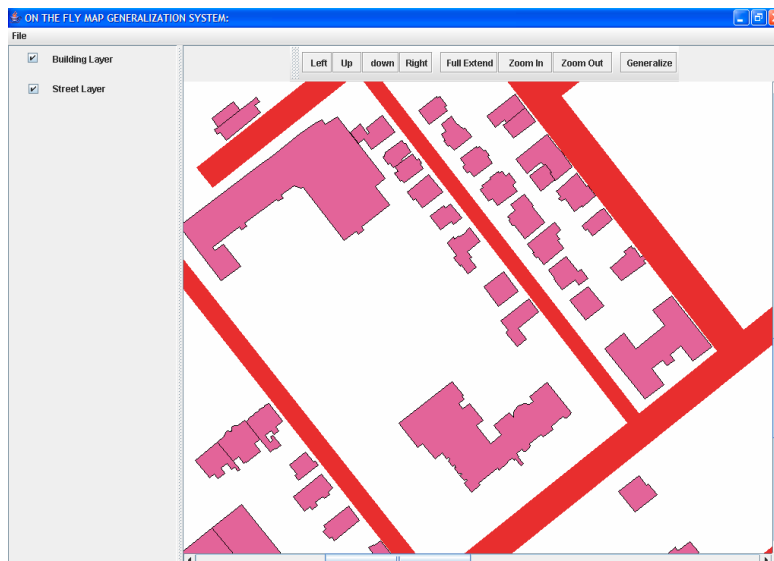


Figure 36- Multi-agent On-the-fly Map Generation System interface

- The *Map Generalization Sub-Module*. This sub-module is in charge of the map's generalization; it is made up exclusively of various software agents (*Map Agent, Thematic Agents, and Simple and Complex SGO Agents*) that interact with each other. The various agents of this sub-module are described in Sections 5.3.2 and 5.4. The interactions between the systems' modules is shown in Figure 37.

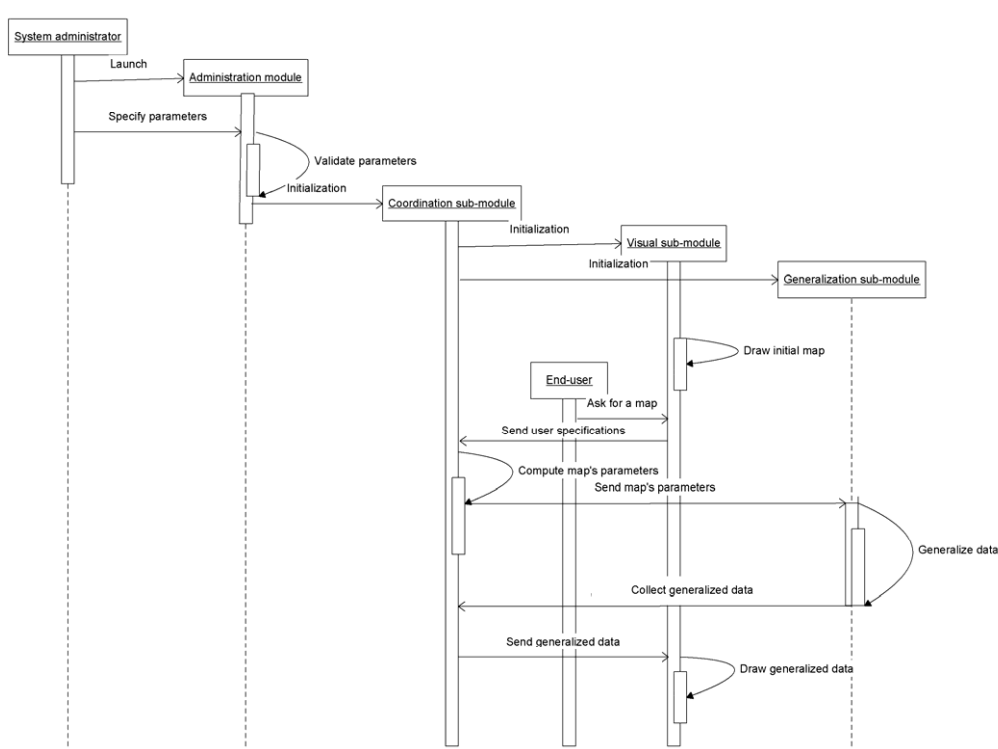


Figure 37- UML sequence diagram of system modules

5.3.2. Multi-agent architecture

The choice of multi-agent technology to develop our system is motivated by the fact that SGOs have a hierarchical structure which is well-suited for the creation of a multi-agent system. According to Schillo et al. (2001), agents are well-suited to the exploration of dynamic phenomena in which the interaction of individual entities can be appropriate at a micro-level, while the emergence of behavioral patterns can be observed at a macro-level. In addition, "multi-agent systems are ideal to represent problems having multiple methods of resolution, multiple prospects and/or multiple solutions" (Chaib-draa et al. 2001) as map generalization. In this section, we describe the different software agents contained in the various sub-modules of the system, all of which were developed using the Jade library.

5.3.2.1. The Coordination Sub-Module's agent: System Agent

As we previously mentioned, the *System Agent* initializes the visualization and map generalization sub-modules. During the initialization phase, the system agent creates all the other agents of the system: the agent of the visualization sub-module (the *Drawing Agent*) and all agents of the map generalization sub-module (*Map Agent*, *Thematic Agents*, and all *SGO Agents*). Thus, for each SGO contained in the SGO database file, an equivalent SGO agent is created. According to the characteristics of each SGO (its constraints, its generalization methods, etc.), a behavior pattern is automatically linked to it, and it is this behavior that determines the type of agent (see Section 5.4). Besides being responsible for the system initialization, the *System Agent* is also used as an interface between the user and the map generalization sub-module dedicated to the generalization process. It intercepts users' requests and extracts the parameters necessary for map creation, such as the map's extension, as well as the selected map layers. Based upon the user's request, the *System Agent* computes the scale of the required map and converts the constraints' thresholds to the requested map's scale. Once all of these thresholds are converted, the *System Agent* sends them to the *Map Agent* contained in the map generalization sub-module. Furthermore, the *System Agent* also associates a time constraint with the sent thresholds, which indicates the maximum time allotted to all SGO Agents for the generalization tasks. This time is computed by the system agent according to the number of selected objects. Once the data are generalized by SGO Agents, results are collected by the system agent that sends them to the *Drawing Agent*. When the user decides to stop the system, the system agent releases the platform by destroying the agents created during the initialization. The UML state diagram for this agent's behavior is shown in Figure 38.

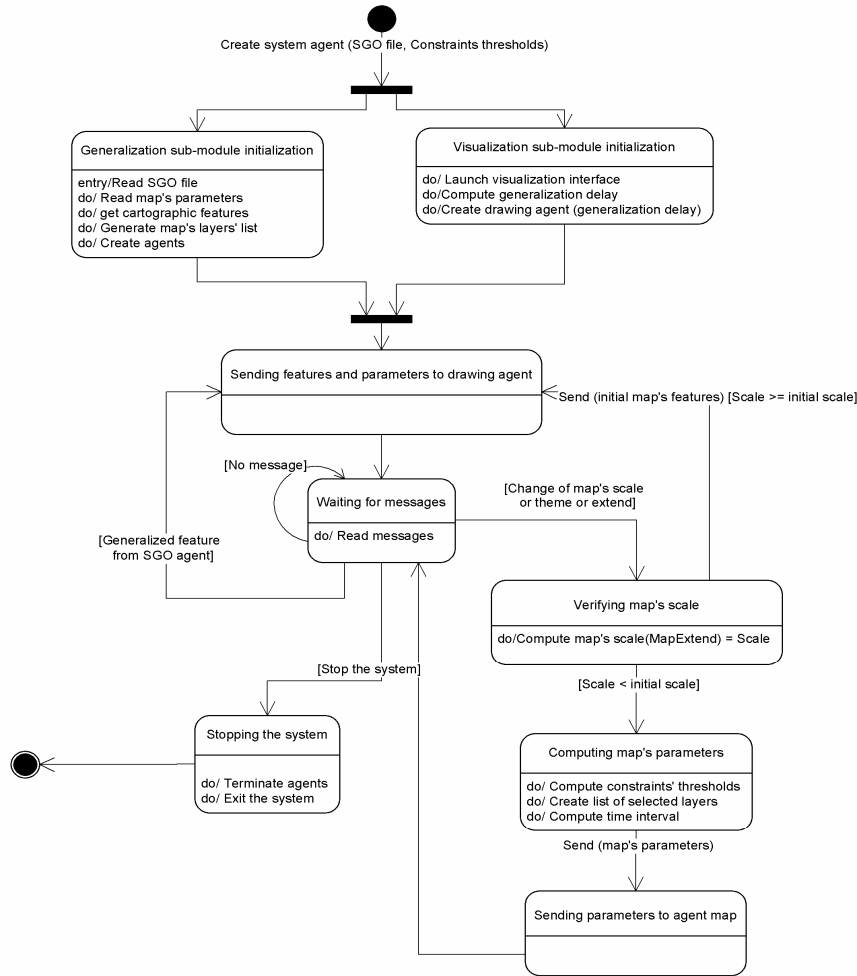


Figure 38- UML state diagram of the System Agent's Behavior

5.3.2.2. The Map Generalization Sub-Module's agents

The *Map Generalization Sub-Module* contains one *Map Agent*, several *Thematic Agents*, and several *SGO Agents (Complex and Simple)*. In this section, we will describe each agent contained in this sub-module.

5.3.2.2.1. Map Agent

The system contains only one *Map Agent* which is responsible for the management of all map layers. In addition to managing the map's layers, it also transfers the information that it receives from the System agent (e.g. an order to an unselected map's layer to deactivate its members) to *Thematic Agents* (see Section 5.3.2.2.2). Moreover, it also verifies the interdependency of the map's different layers, in order to be sure that a *SGO agent* does not

wait for information from an agent whose layer is unselected. For example, if a layer is not selected by the user, the Map Agent verifies whether or not other map layers are dependent upon the former. In such a case, the Map Agent can send a message to the Thematic Agent, responsible for the selected layer, to inform it that the dependent layer is unselected, and that it should not try to interact with the members of the deactivated layer. The list of interdependencies between the various layers is created during the initialization of the system by the System Agent. What's more, according to the list of selected layers by the user, the Map Agent selects the agents to which the information is sent. The UML state diagram of this agent's behavior is shown in Figure 39.

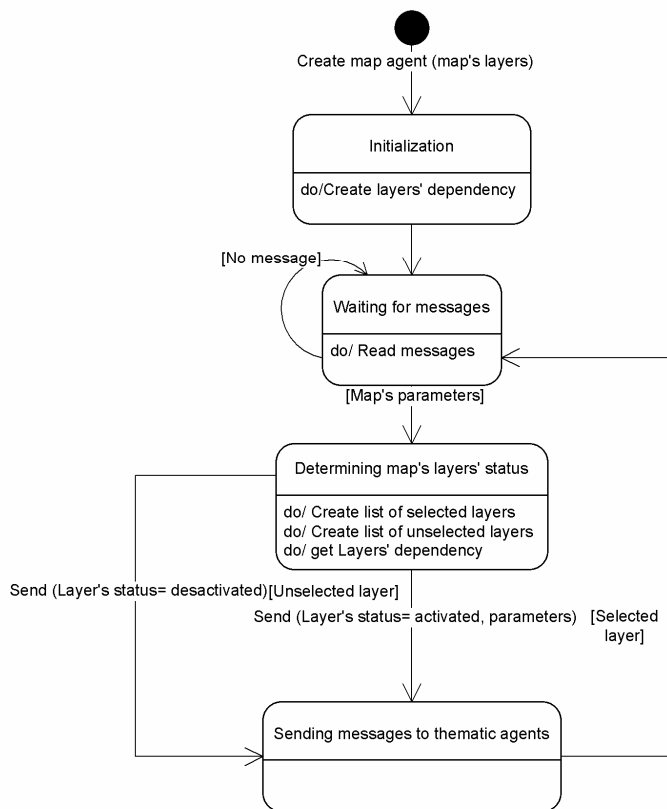


Figure 39- UML state diagram of the Map Agent's Behavior

5.3.2.2.2. *Thematic Agent*

The *Thematic Agent* is responsible for the management of the objects' classes. The main roles of this agent are to perform pre-selection tasks, and to relay the information to the SGO agents representing the layers' objects. There are as many *Thematic Agents* as classes of objects in the system. The *Thematic Agent* implements some properties of the complex

SGO agents, such as the ability to index its members and filter messages. The UML state diagram of this agent's behavior is shown in Figure 40.

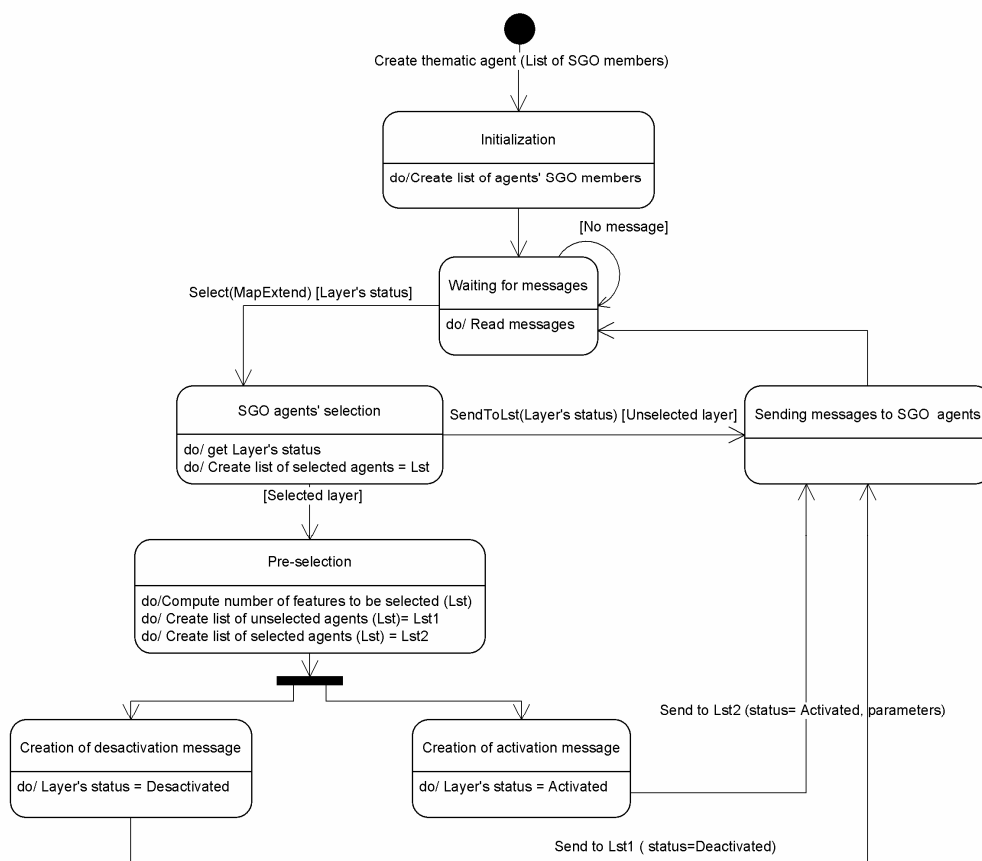


Figure 40- UML state diagram of the Thematic Agent's Behavior

5.3.2.2.3. SGO agents

A SGO agent is a software agent created from a SGO in order to autonomously interact with other surrounding SGO agents. The SGO is the essential component of an SGO agent, and provides the agent with the necessary data (the link with cartographic features, spatial integrity constraints) and generalization capability (geometric patterns and process patterns) to achieve their goals (to generalize the objects that they represent).

In the current system, for each imported SGO (simple and complex SGOs), there is an equivalent SGO agent that is created. Because SGO agents are extensions of the Jade platform agents, they inherit the properties of the Jade agents as for example

communications protocol. We differentiate two groups of SGO agents: Simple SGO agents based upon Simple SGOs, and Complex SGO agents based upon complex SGOs. As we previously mentioned, according to the SGO constraints, a specific behavior pattern is associated with the created SGO agent. This associated behavior pattern then determines the type of the SGO agent (see Sections 5.4.1 and 5.4.2). The SGO agents' behaviors are generated as an extension of Jade agent's behaviors. Jade provides several behaviors that facilitate the implementation of agents' coordination.

5.3.2.2.4. The role of a SGO agent

The main role of a SGO agent is to generalize the represented cartographic feature, in order to adapt it to the level of detail requested by the user. For this reason, it is the SGO agent that is responsible for the satisfaction of its constraints. In the event of a conflict (a constraint violation), the SGO agent must apply the proper generalization operations in order to solve the disagreement. To carry out generalization operations, a SGO agent can use geometric patterns, and/or process patterns (i.e. a set of generalization algorithms). During the generalization process, the SGO agent gives priority to the use of geometric patterns because it is the fastest alternative. Consequently, the SGO agent is responsible for verifying the order of constraint violations, conflict resolution, adjustments made to the geometric pattern, applying necessary generalization operations, and interaction with other SGO agents. This is made possible thanks to the SGO agent behavior pattern. When generalizing the cartographic feature, SGO agents send back the results (i.e. generalized data) to the system agent.

In order to facilitate the contextual generalization, such as aggregation, one of the main roles of complex SGO agents is to monitor the group's cohesion. For this purpose, a complex SGO agent can play the role of a facilitator (mediator) for the SGO members of the group. It does not solve the conflicts for its members, but it does make sure that the solution is coherent and that the time limit for data generalization is respected by its members. If its members cannot solve the conflict themselves, they can notify the complex SGO agent about their trouble. In such a case, the complex SGO can then use other methods to solve the problem. For instance, if the agents belonging to the same complex SGO cannot solve their proximity conflicts because they are lacking free space, then they

can ask the complex SGO agent for assistance, which may decide to carry out an aggregation by using a tool such as a geometric pattern. The advantage of using a complex SGO agent to supervise certain generalization operations is primarily due to the fact that, contrary to a simple SGO agent, it has a higher level of perception. Moreover, the complex SGO possesses information about the group necessary for generalization, which facilitates the decision-making process. For example, in the case of aligned buildings, it can compute the space necessary to represent all members; and knowing the requested scale, it can determine if elimination is necessary. In such a case, it would choose aggregation.

In a complex SGO agent, its members are categorized spatially in order to help with communication. The complex SGO agent has access to this catalog, which enables it to send targeted messages according to the positions of the members. This is extremely important at the time of a scale change, or when it is not necessary to generalize the entire map, but rather, only a specific territory selected by the user. In this case, a complex SGO agent only sends a notification message to the agents located in the selected area.

In addition, each SGO agent (simple or complex) has a memory that enables it to remember the conditions under which each operation is carried out (e.g. under which scale aggregation was carried out). Therefore, this memory allows the SGO to anticipate various actions. Contrary to SGOs, in certain multi-agent approaches (e.g. Jabeur 2006) various states of an agent are memorized, which creates data redundancy. For instance, thanks to our memory mechanism, if a complex SGO carried out an aggregation at a scale of 20k, it would not have to wait for the request of its members to carry out an aggregation if a similar, but lower scale situation occurs (e.g. 50k). What's more, the scale related to a given action can be refined. For example, if aggregations were carried out on the scales 20k and 10k, the new scale related to this operation would be 10k. This permits the continuous enrichment of the SGO, in addition to the constant improvement of its performance.

5.3.2.2.5. SGO agent perception

In order to solve the relational conflicts which can be materialized, SGO agents must interact by exchanging messages. The Delaunay, triangulation-based approach, or area-based approach, is frequently used for determining the neighbors of a map's object.

Nonetheless, the majority of multi-agent generalization approaches use a predefined area to demarcate the agent's field of perception. For example, Duchêne (2003) uses an area delimited by a circle with a predefined radius around each agent; hence, the agent's field of perception. In his work, Jabeur (2005) uses a buffer around each map's object in order to delimit the field of perception of its agent. In the area-based approach, the area must be defined according to the scale of the map which will be generated. Thus, for on-the-fly map generalization, whose scale is arbitrary, the use of a predefined area intended to delimit the agent's perception cannot guarantee a good view of the agent's neighbor at any scale. The choice of an excessively small area results in the loss of certain neighbors despite their close proximity. In contrast, a too large area results in taking into account relations which are not relevant to the generalization process.

To overcome these problems in our approach, the perception of the SGO agent is based upon the constraint of proximity. In other words, each SGO agent identifies the other SGO agents through its proximity constraint, which is defined interactively during the enrichment phase (of course Delaunay triangulation can also be used). In addition, for each object regarding its constraints of proximity, a vector of potential displacement is automatically given; this vector determines the direction of displacement in the event of a proximity conflict.

In more of the perception related to the proximity constraint which is common to all SGO agents, complex SGO agents also recognize all the members of the group that they represent. This is due to the fact that only complex SGO agents have access to the register of all its members. In order to obtain the address of a member with which it does not have a direct connection (i.e. it is not linked by a proximity constraint), an agent must demand it from another agent in its immediate vicinity, or from its complex SGO agent.

5.3.2.2.6. Interactions and communication between SGO agents

During the resolution of proximity constraints, the agents must communicate in order to find the best solution. During these interactions, an agent can request a service (e.g. to ask another agent to move), send information. Each SGO agent can communicate with agents

that it recognizes and with its complex SGO agent. A complex SGO agent can also communicate with its SGO agent members. Depending upon the type of the involved agents and the kinds of relationships that link them, we can distinguish three categories of interactions between SGO agents:

- Vertical interactions. They are interactions that take place between a complex SGO agent and its members. This interaction is asymmetrical because the interlocutors belong to various hierarchical levels; it is a kind of father-son relationship. In these types of interactions, a complex SGO agent has more power than its members, which sometimes enables it to give orders. During the interactions, waves of information can be ascending (from a member to its complex SGO agent) or descending (from the Complex SGO agent to its members). One of the most important roles of descending communication is the diffusion of information. For instance, when the scale changes, the agent, which is on top of the hierarchy, can send this information to its children, and they in turn, can send the information to their own children, and so on. In this way, the information can be sent out in a scattered manner to all the concerned agents. The advantage of such an approach for the dissemination of information is that complex SGO agents have access to a spatially indexed list of all their members; they can next use their ‘filtering power’ to choose which members are directly concerned by the information at hand, and a targeted message can then be sent to them.
- Horizontal or transversal interactions. They are interactions between two SGO agents which are not involved in a father-son relationship, but that are on the same hierarchical level (e.g. belonging to the same complex SGO). When these agents belong to the same group, no supervision is necessary; in fact, they only have to inform the complex SGO agent about the results of their interactions so that the group’s knowledge (e.g. the average and minimal distance between aligned buildings) can be up-to-date. Nevertheless, during this update, if the complex SGO agent realizes that the action damages the group (e.g. in a complex SGO of aligned buildings, when a SGO member

moves so far away from the axis of alignment), it can ask the concerned members to cancel the operation, or to correct the proposed action. However, when the interaction takes place with a SGO agent outside of the group, this interaction must be overseen by the complex SGO agent in order to avoid actions that could possibly prove to be harmful to the group.

- Oblique interactions. They are interactions between SGO agents that come from various hierarchical levels and that are not related by father-son type relationships. In other multi-agent systems (e.g. Jabeur 2006, Lamy et al. 1999), oblique interaction is nonexistent. This kind of communication allows, for example, a simple SGO agent to interact with a complex SGO agent (the simple SGO and the complex SGO are not related by father-son relationships), even if they come from different hierarchical levels (relation R2 in figure 41). These exchanges are allowed between SGO agents linked by a spatial relationship (e.g. proximity) which may be either specified during the enrichment phase, or generated during the generalization process. This kind of contact can occur during the generalization process when a complex SGO agent aggregates several objects together. For example, two simple building SGOs (e.g. SGOs A and B in figure 41) from different complexes SGOs can be linked by a spatial relationship (e.g. proximity constraint). During the generalization process, one of the complex SGOs can be aggregated (e.g. the complex SGO containing SGO A). In such case, the complex SGO containing SGO A inherits the relationships (e.g. proximity constraints) of its SGO members (the relationships of the SGO A). Thus, the complex SGO agent (agent representing the complex SGO containing SGO A) will interact with a simple SGO agent with a lower hierarchical level (the agent of the simple SGO B). Thus the inherited transversal interactions (relation R1 in figure 41) become oblique (relation R2 in figure 41).

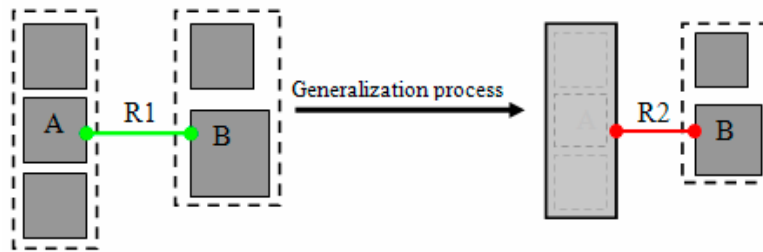


Figure 41- Transformation of a transversal (R1) interaction to oblique (R2) during a generalization process

5.3.2.3. The visualization Sub-Module's agent: Drawing Agent

The main role of this agent is to draw the data generated by the map's generalization sub-module (i.e. by SGO agents). The *Drawing Agent* does not receive this data directly from the map generalization sub-module's SGO agents, but rather, it receives the data via the *System Agent*. The *Drawing Agent* enables the generation of the map's layers by linking each generalized data with the suitable map style. To this end, each SGO agent attaches information about its layer of origin when sending the generalized data. The *Drawing Agent* then uses this information to determine which style should be applied to the received data. Furthermore, this agent is also able to draw the received data at regular time intervals, specified by the agent system that computes this interval by using the density of the map's objects, in addition to the time assigned for the map generalization. Indeed, the drawing operation is a time consuming process and drawing each received data individually would have to harm considerably the system performance. When the allotted time expires, the agent draws all the received data. If no data was received, it simply draws the initial data, that from which the map is generalized. The UML state diagram of this agent's behavior is shown in Figure 42.

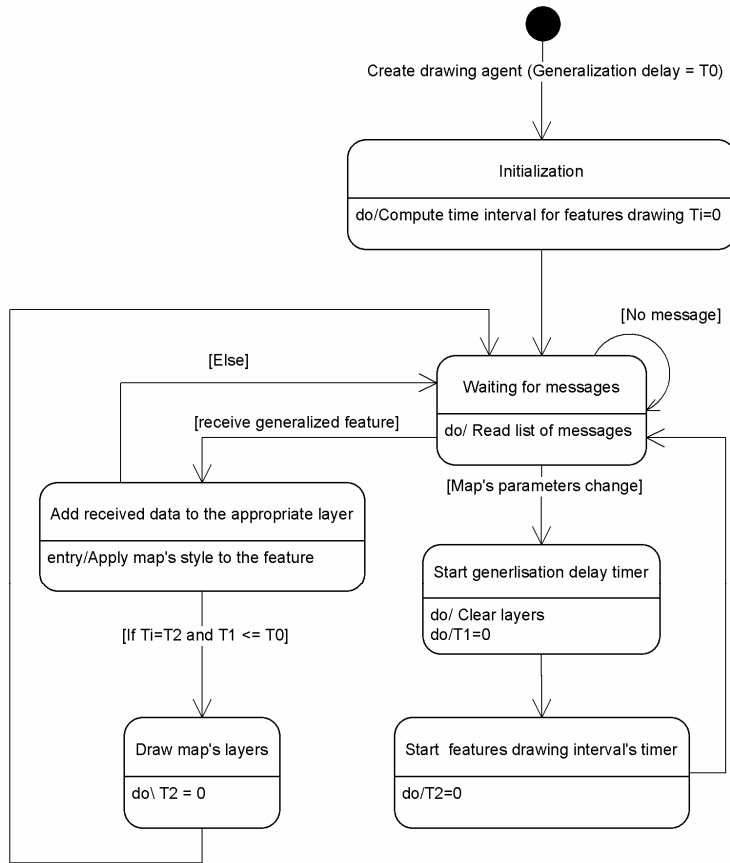


Figure 42- UML state diagram of Drawing Agent's Behavior

5.4. Implementation of generalization sub-module's agents

Agents of this sub-module present a hierarchical architecture. At the highest hierarchical level the Map Agent is located, whereas the lowest level is populated by simple SGO agents (e.g. a single building's SGO agent). In the intermediate levels, there are Thematic Agents and several complex SGO agents. Currently, our prototype contains two map layers: the road network and the building layers. Thus, two Thematic Agents are implemented in the system: the road network Thematic Agent and the buildings' Thematic Agent. In this section, we present the implementation of all agents found in each map layer; however, the Map Agent will not be described here because this agent was already presented in Section 5.3.2.2.1, and its implementation does not depend upon the number or the type of objects' classes that are used in the system.

5.4.1. Implementation of buildings' agents

5.4.1.1. Implementation of buildings' Thematic Agent

In order to minimize the number of interactions between several SGO agents during the generalization process (to improve the performance), this agent carries out the pre-selection. This operation allows the deactivation of very small building SGO agents (which cannot be represented at the required scale). Furthermore, only scattered buildings (which cannot be aggregated) are affected by the pre-selection operation. The pre-selection threshold is selected by taking into account the fact that small buildings can be exaggerated. We experimentally determine this threshold depending upon the density of the map's objects. What's more, the final selection of each map's object is the result of transactions between several SGO agents.

5.4.1.2. Implementation of buildings' SGO agents

During the creation of buildings' SGO agents from the SGO database, a behavior pattern is associated with every created agent. This behavior depends upon the *N-ary Constraint* of the SGO (Section 5.2.2.3.). Currently, three *N-ary Constraints* have been implemented in the system; they are the alignment constraint, block constraint, and proximity constraint, between buildings, and between buildings and streets. In order to get distinct behaviors according to the constraints currently implemented in the system, three types of SGO agents have been formed. These agents are the SGO agents of aligned buildings, the SGO agents of buildings in a block, and the SGO agents of scattered buildings. In the following sections, we will describe the implementations of these various agents. A simplified UML class model of buildings' SGO agents is presented in Figure 43.

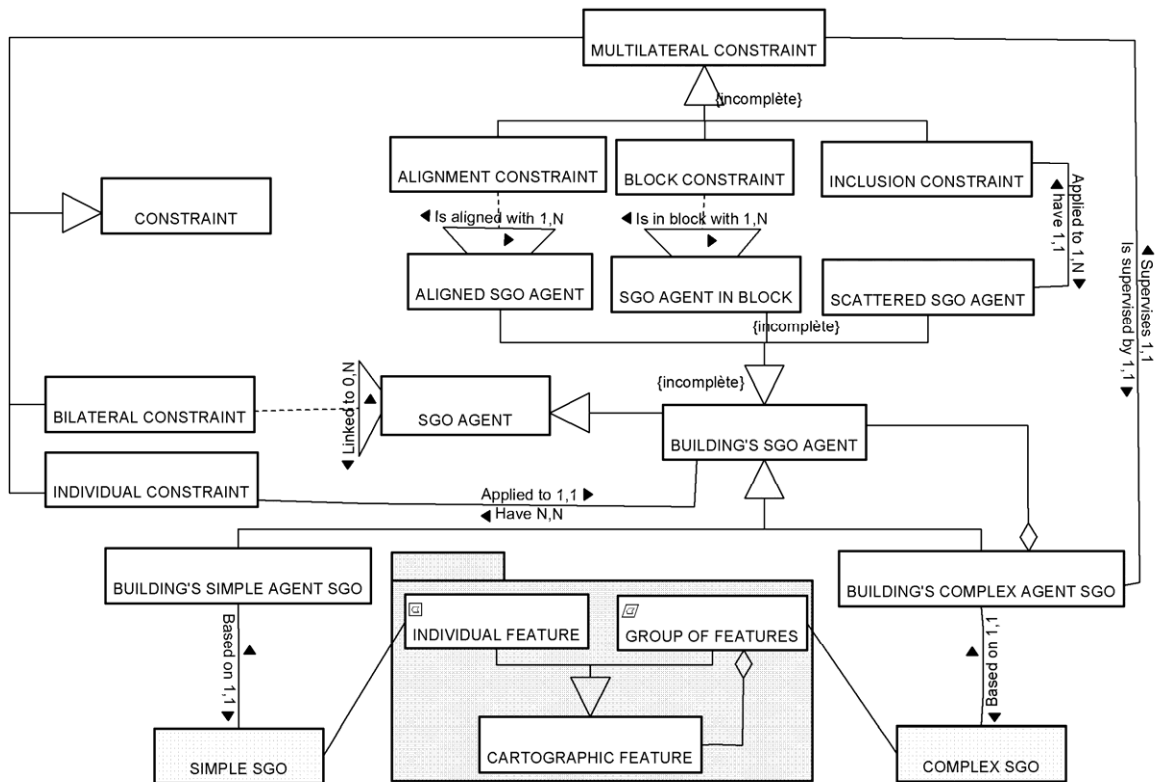


Figure 43- Simplified UML class model of Buildings' SGO Agents

5.4.1.2.1. SGO agents of aligned buildings

The role of the alignment constraint is to maintain the members of the Complex SGO aligned during the generalization process. The SGO agent members of this Complex SGO are equipped with the following generalization operations: simplification by means of the geometric patterns and/or the process patterns, exaggeration and enlargement by means of geometric patterns, displacement, and elimination (auto-deactivation). These generalization operations are common to all SGO agents of buildings. The member can be a complex or a simple SGO agent. In addition, the complex SGO agent of aligned buildings can perform aggregation. It also possesses information about the cartographic features represented by the SGO agent members, such as the minimum and average size of the buildings, the average and minimal distances between buildings, and the axis of buildings' alignment. This information allows the Complex SGO agent to make specific decisions (e.g. to apply the aggregation operation) and/or to guide its members during the generalization process.

When a scale changes, the complex SGO agent must decide whether or not to go ahead with an aggregation, or to inform its members to start the generalization process. The complex SGO agent decides to pursue aggregation when it notices that several of its SGO agent members are too small to be represented at this scale. When aggregation is not necessary, the complex SGO informs its members to begin the generalization process. When they receive this notification, SGO agent members first verify their internal constraints, such as the minimum size constraint and the minimum detail constraint (where the violation can trigger simplification). To solve the conflicts due to the violation of the internal constraints, the SGO agent member uses the geometric patterns or the generalization algorithms (process patterns).

Once the internal constraints are satisfied, the members solve their proximity constraints. For this purpose, agents located at the extremities of the alignment solve their conflicts first with the external SGO agents (agents that are not members of the group). Once the extremities have solved any possible conflicts (if there is a violation of the constraint), they inform the complex SGO agent which recomputes the new axis of alignment (if the agents situated at the extremities moved), and checks the deviation of this axis compared to its initial direction. If the deviation respects the maximum deviation threshold, the complex SGO agent asks the other intermediate SGO agents to adjust their positions according to the new axis (the parameters of the new axis are sent to the intermediate SGO agents). Otherwise, the complex SGO can cancel the operation. When receiving the order, intermediate SGO agents also solve their proximity conflicts, respecting the new alignment axis. If, for any reason, the intermediate members cannot respect the axis of alignment (e.g. an external SGO agent requires a member of the group to move), they inform the complex SGO agent that then decides which action should be undertaken (e.g. to ask all the members of the group to move, or to carry out an aggregation using a geometric pattern). A UML state diagram of a complex SGO agent of aligned buildings is shown in Figure 44. For SGO agent members (aligned SGO agents) to solve proximity conflicts, they must have the same behavior as scattered SGO agents (see Section 5.4.1.2.3). The only difference is that during displacements, aligned SGO agents must respect the axis of alignment.

5.4.1.2.2. SGO agents of buildings in block

The block SGO agents have a rather simple behavior that allows all the members of the group to synchronize their actions so that they can always stay together. All the actions of the SGO agent members of the group are collective. Indeed, a member cannot move individually. If a displacement is necessary, it is the complex SGO agent that computes the vector and the intensity of the displacement, and orders its execution by the members. For this type of grouping, aggregation is frequently used and the complex SGO agent decides if it needs to use a geometric pattern or a process pattern. However, following an aggregation, according to the required scale, the aggregated buildings' geometry can require other generalization operations such as simplification. In such a case, all operations applied to the aggregated geometry are realized by the complex SGO (it becomes responsible for the aggregate). For all types of building SGO agents, it is the complex SGO agent that is responsible for all procedures relevant to the aggregates (i.e. the result of aggregation).

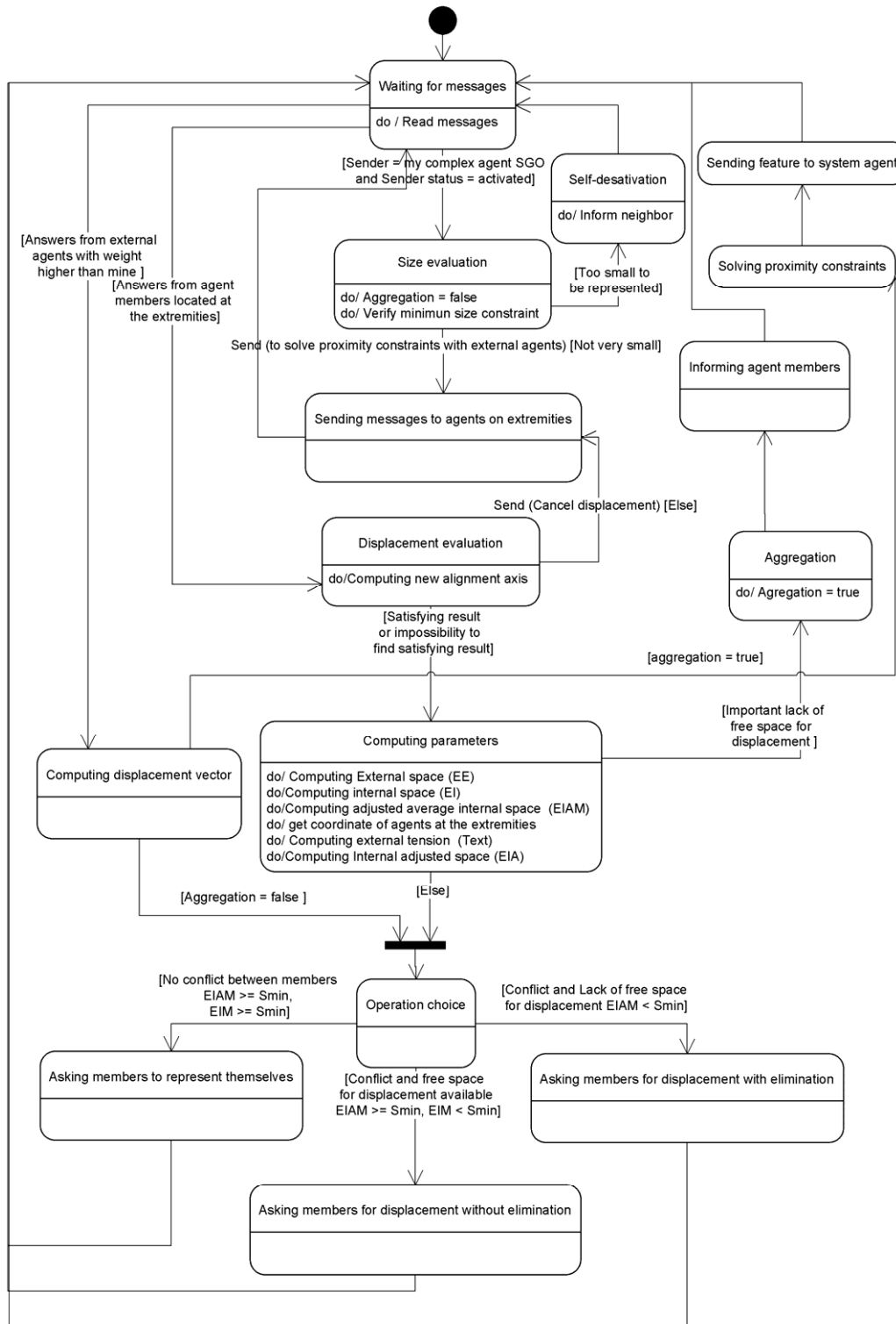


Figure 44- UML state diagram of complex SGO agent for aligned buildings

5.4.1.2.3. SGO agents of scattered buildings

According to the system's current implementation, scattered SGO agents are agents that are neither aligned nor in a block. The complex SGO agent of scattered buildings only plays the roles of the container and distributor of information to its members (e.g. notify for a change of scale). The generalization procedures are only completed by the SGO agent members. In addition to satisfying their internal constraints, the SGO agent members of this group have the responsibility for taking care of the satisfaction of the proximity constraints, which bond them to other SGO agents.

In order to solve their proximity constraints, these agents carry out either a displacement or the self-deactivation when displacement is impossible. The displacement of scattered agents is based upon the following principles:

- 1) Prior to checking its proximity constraints, the agent must wait for the other agents (i.e. agents with which it has proximity constraints and which have more weight than it is), including street agents that check their constraints and inform the scattered SGO agent of their new coordinates in the event of a displacement. During this waiting time, agents can check their internal constraints (e.g. to carry out simplification if necessary). In approaches without latency (e.g. Jabeur 2006), objects with the lowest priority must constantly check for and solve new possible conflicts due to displacement or exaggerations of neighbors' objects having a higher priority. Moreover, the multiple checks can harm the system's performance.
- 2) Each proximity constraint must have one or two responsible agents. The responsibility is based upon the importance of the object (i.e. its weight). In a proximity relationship, where the two members have different weights, the least important agent becomes responsible for the constraint; therefore, the least important agent must try to fulfill the constraint. In other words, the non-satisfaction of the constraint can lead to the deactivation of the agent in charge of the constraint. However, if two agents have similar weights, the responsibility for the proximity constraint is assumed jointly. Furthermore, in the case of a displacement, each of the two agents must move half way of the necessary distance.

3) If an agent is unable to move, it can ask its neighbor to move away, including the agent with which it is in conflict; this approach helps promote the propagation of displacements.

4) Following a request for assistance, a SGO agent can either refuse (if it is not capable of moving), fully accept the request (move the complete distance requested by the neighbor), or partially accept the request (move a shorter distance than what was requested by the neighbor).

5) If it is impossible to fulfill a constraint, the agent responsible for the constraint must be deactivated. When two agents have equal weight (i.e. co-responsibility), the agent which has the largest number of conflicts is deactivated. In practice, it is always preferable to find the agent that is responsible of the constraint (i.e. based upon their weights and the number of conflicts with which they are involved).

6) For any resultant displacements, an agent must inform the entire neighborhood of its current position, as well as its current situation (i.e. activated or deactivated), so that these neighbors can update their knowledge.

As in other multi-agent generalization approaches (e.g. Jabeur (2005)), in our approach, the resolution of spatial conflicts is achieved through interactions between SGO agent members in addition to the intervention of its complex SGO agent (but, for scattered buildings, the complex SGO agent's intervention is not necessary). However, the heuristic, knowledge and interactions used in our approach differ considerably from those used in the other approaches.

In our approach, displacement is facilitated by the data enrichment process during which explicit information was created and stored (e.g. agent responsible of each proximity constraint, the potential displacement vector or the minimum distance between two features). Furthermore, during the generalization process, this information is updated on-the-fly. The UML state diagram of a simple scattered building SGO agent is presented in Figure 45, and its complex SGO agent in Figure 46.

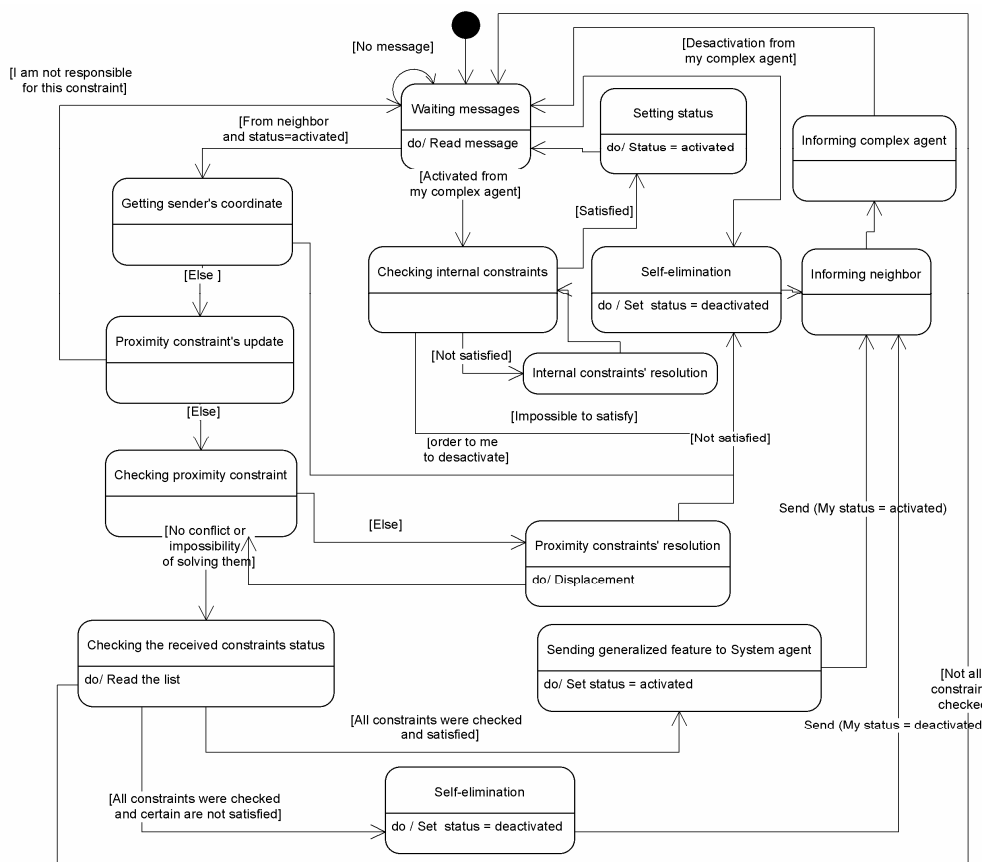


Figure 45- UML state diagram of a simple scattered building SGO agent

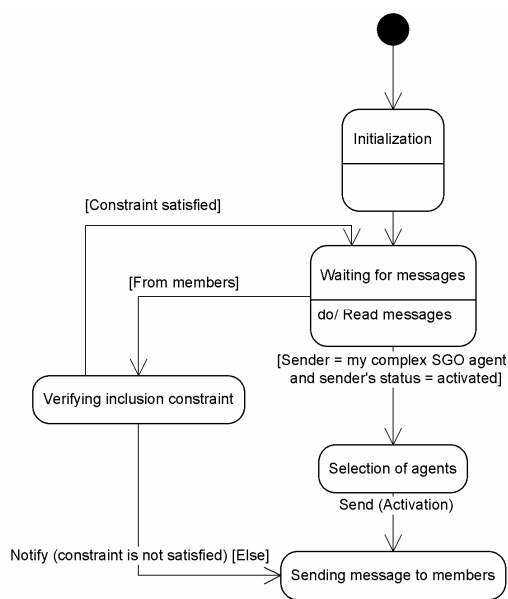


Figure 46- UML state diagram of complex scattered building SGO agent

5.4.2. Implementation of road network agents

5.4.2.1. Implementation of road network Thematic Agent

During the generalization, the road network Thematic Agent does a preliminary selection of the streets that will be represented, at a given scale, based upon their importance. In our system, streets are structured using the principle of good continuation, proposed by Thomson and Richardson (1999). In order to determine the number of objects that will be selected during the scale reduction, the road network thematic agent uses the formula proposed by Topfer and Pillewizer (1961). This formula gives us the approximate number of objects that will be selected at a given scale. Unfortunately, it does not allow us to know which object must be eliminated. For that, the number of objects computed by this formula is increased by a given margin (e.g. 25% more). This margin lets us be sure that we do not eliminate important objects. What's more, the final selection of streets emerges from the interaction of SGO agents in the network.

5.4.2.2. Implementation of road network SGO agents

In the case of the road network, each street is represented by a complex SGO agent, whereas the segments and junctions that make it up are represented by simple SGO agents. Currently, three constraints have been implemented: the minimum length segment constraint, the minimum street width constraint, and the connectivity constraint. The minimum length constraint also allows us to preserve the minimum building block side. The minimum width constraint guarantees a minimum separation of two nearby blocks of buildings. When taking into account the type of road network (e.g. star-like network), complex SGO agents may be added. For example, if the classification of the road network, in either star-like or grid-like networks, must be taken into account, we can create a complex SGO agent for every type of network. This is necessary because the measurements that allow us to determine whether or not two neighboring buildings' blocks must be aggregated are different, depending upon whether it is a star-like or grid-like network. Consequently, the behavior pattern for these SGO agents must be dissimilar. For a grid-like network, such as the one that is implemented in the current system, if a street segment does not satisfy the minimum length constraint, one of its junctions (the least important) is deactivated. This means that all streets connected to this specific street via the deactivated

junction must also be deactivated. As building blocks are limited by streets, the street deactivation gives the visual impression that adjacent building blocks are aggregated. However, to avoid the deactivation of important streets, the constraint verification process is done through propagation. The agent for the most important street verifies its constraints. Next, it sends a message to those street agents to which it is connected so that they can verify their constraints as well. For this purpose, all streets must be connected to the network. If the network becomes disconnected, a complex SGO agent must be created for every sub-network. In the current system, the mapped zone is only represented by a grid-like network. Here, a street is deactivated if: 1) its width is less than the indicated threshold; or 2) if one of the street segments that it intersects does not satisfy its minimum length constraint. No matter what the SGO agent decides (activation or deactivation) during the generalization, it must inform its neighboring agents (the streets to which it is connected, and the building SGO agents which are along it) about its decision. The UML state diagram of the road segment SGO agent is shown in Figure 47, and the road SGO agent (complex SGO agent) in Figure 48. In the current implementation, the role of the street junction SGO agent is to provide the SGO agents representing street segments the list of the connected streets.

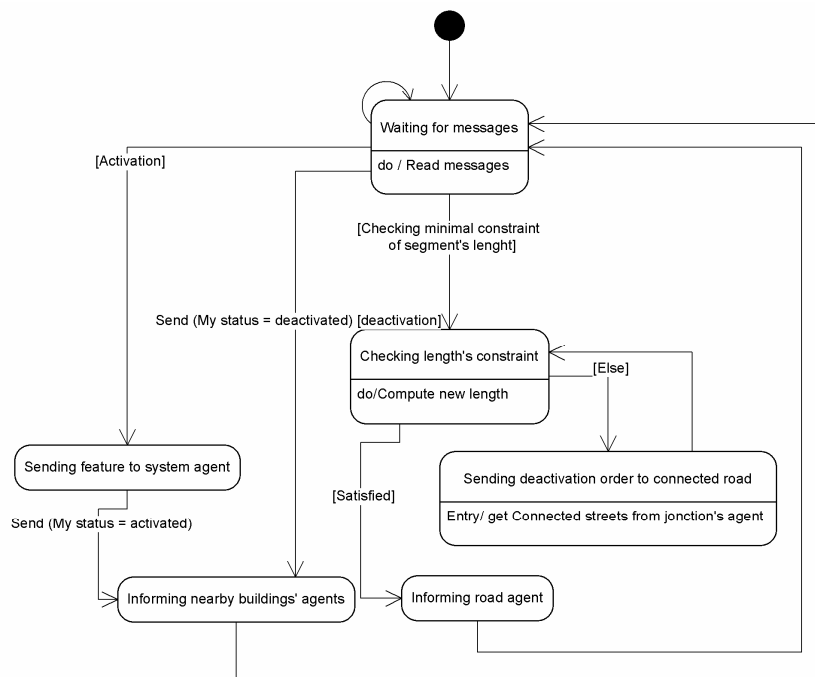


Figure 47- UML state diagram of a road segment SGO agent (simple SGO agent)

agents (in our case, the road network thematic agent and the building's thematic agent). Each thematic agent selects its member SGO agents that are concerned by the information (i.e. those that are included in the territory selected by the user), and sends them a message so that they can begin the generalization process. According to its behavior, each agent verifies its integrity constraints, and in the event of a constraint violation, it uses the appropriate operation to solve the problem. Once it generalizes its data, each SGO agent sends the results (generalized geometry) to the system agent that, in turn, sends it (i.e. received data) to the drawing agent in charge of applying the map style to the generalized data, and drawing it. The following UML diagram (Figure 49) shows a simplified variant of the interactions between the various SGO agents. The figures in the Table 2 show the generalized data generated by the developed system.

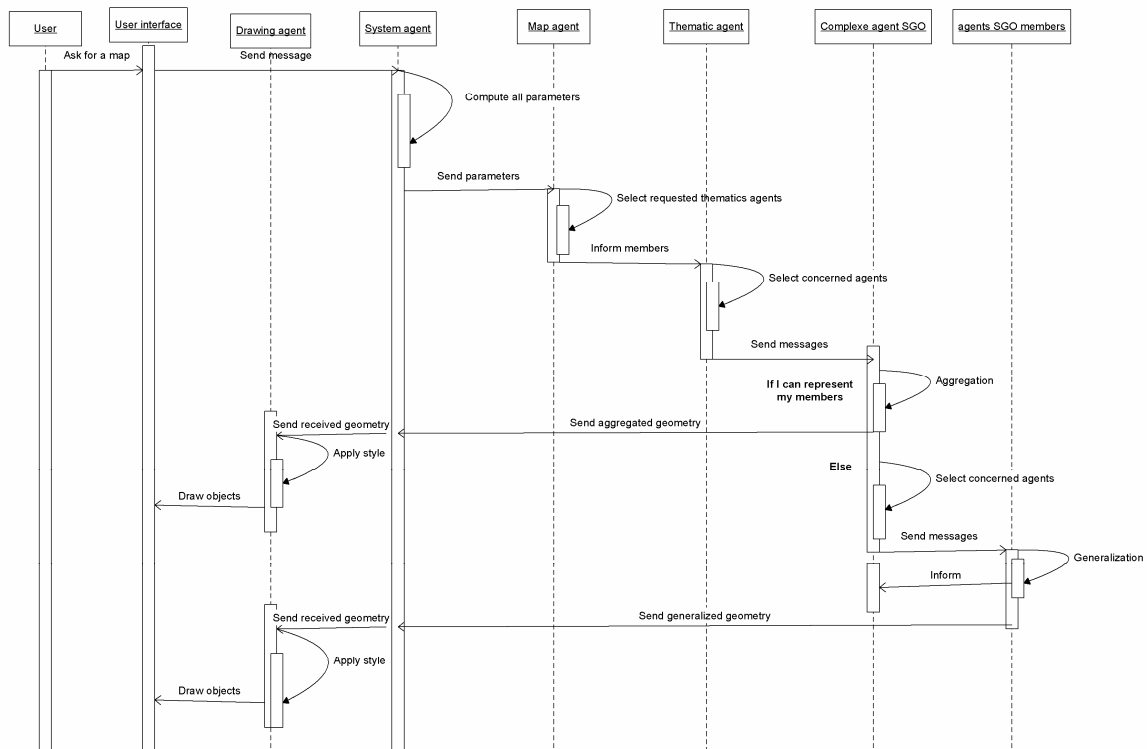


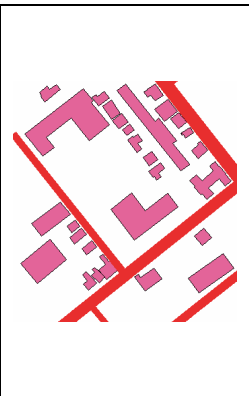



Figure 49- Simplified UML sequence diagram of agents' interactions

The prototype was tested on a Pentium 4 CPU 2.8 GHz with 512 megabytes of RAM (Random Access Memory) under Microsoft Windows XP family edition and Service Pack

2. Figure 50 demonstrates the necessary time (in milliseconds) for generalizing at different map scales (a hundred objects). All map scales are generated from the initial map (1K). The figure shows that when the scale decreases, the time necessary to generalize the map decreases as well. This allows us to achieve a high level of system performance, even when the number of objects to be generalized increases. A decreased generalization process time is due to the fact that the aggregation decreases the number of objects which are generalized.

Tableau 2- Generalization results using the prototype. The scales are those that were used at the time of generalization, and the scales of the images can differ slightly from those of the generated maps.

				
Initial map (1 :1000)	1 : 1582	1 : 2550	1 : 4100	1 : 27176

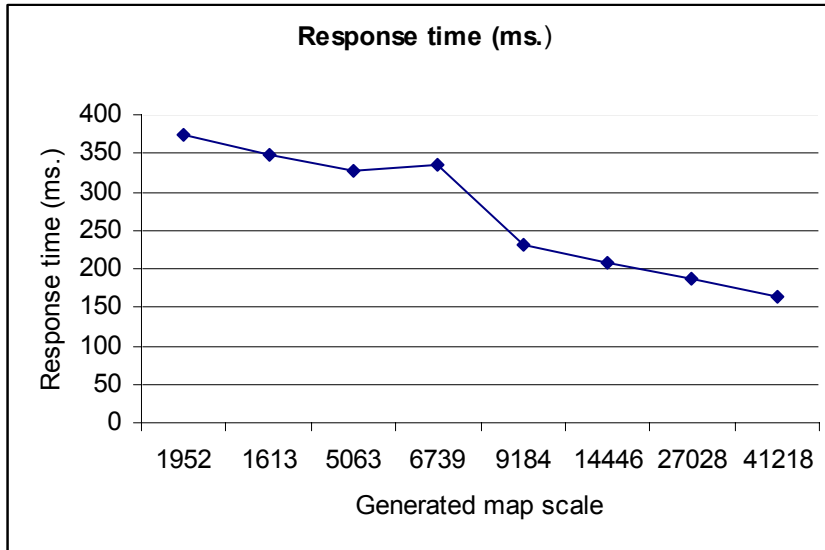


Figure 50- Times necessary for generating various map scales (all scales are generated from the 1K scale. These arbitrary scales were computed according to the map extension selected by the user.).

5.5. Conclusion and Discussion

In this paper, we presented a multi-agent system which allows users to automatically generate data adapted to the required level of abstraction, thanks to the on-the-fly map generalization process. The architecture of the system was presented, in addition to its main modules, and various agents that make it up. The approach used in this prototype is based upon self-generalizing objects (SGOs). Each SGO encapsulates in the same object the geometric patterns, generalization algorithms in the form of process patterns, and spatial integrity constraints. Contrary to the majority of on-the-fly map generalization approaches, the SGO approach uses the geometric patterns (which are forms common to several map's features) in the place of the traditional multiple representations. The use of geometric patterns reduces considerably the data redundancy because the same geometric pattern can be used by several map's features. Besides facilitating the map generalization process, SGOs allow its acceleration. This acceleration is due to the simplicity of the operations carried out to adjust the geometric pattern, as well as database enrichment. During this enrichment, generalization knowledge, such as the choice of algorithms and their sequence, or the creation of spatial relationships, is introduced into the SGO database. Moreover, the

SGO approach demonstrates a good way to perpetuate the expert's cartographic knowledge since this knowledge is stored and used several times.

The tests carried out using the prototype enabled us to confirm the feasibility of a multi-agent based, on-the-fly map generalization approach, able to efficiently take into account the objects' environment, thanks to the use of the spatial integrity constraints. Furthermore, the results of these tests are very encouraging according to the performance of the developed system and the quality of the generated data. These results allow to consider this approach and the developed system fully compatible with the main requirements of an on-the-fly map generalization (i.e. quasi-instantaneous response time, and no human intervention during the generalization process). Indeed, the time required for map generalization (e.g. for a hundred objects) is in conformity with the cognitive Newell (1990) upper bound for an acceptable response time, which is less than 10 seconds. Beyond this amount of time, users typically leave the Web page, without awaiting the results of their request. However, several aspects of this study can be improved. Currently, the developed system implements only two classes' objects of restricted area (building and road network). Thus, the tests can be extended to other map objects (e.g. hydrography or land layer). In addition, to deploy the test on a large area, it would be desirable to automate as much as possible the creation of the SGO. All of these aspects make up the future challenges for the approach, things to improve and new tools to be developed during future research.

5.6. Acknowledgments

The authors are grateful to the Canadian Network of excellence, GEOIDE, which made this study within the framework of GEMURE and MUSCAMAGS projects possible. We are also grateful to the Canada NSERC Industrial Research Chair in Geospatial Database for Decision-Support and our following partners: Research and Development Defense Canada, Natural Resources Canada, Ressources Naturelles et Faune Quebec, and Intergraph.

5.7. References

- C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel. *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press: New York, 1977, 1171 p.
- M. Allouche and B. Moulin. "Amalgamation in cartographic generalization using Kohonen's feature nets", *International Journal of Geographical Information Science*, Vol. 19(8-9): 899-914, 2005.
- M.P. Armstrong. "Knowledge classification and organization", In B.P. Buttenfield, R. McMaster and H. Freeman (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, pp. 86-102.
- C. Baeijs, Y. Demazeau and L. Alvares. "Application des systèmes multi-agent à la généralisation cartographique", In *Proceedings of the 3èmes Journées Francophones sur l'intelligence Artificielle Distribuée et les Systèmes Multi-agents, AFCET&AFIA*, Chambéry, 1995.
- K.M. Beard. "Constraints on rule formation", In B.P. Buttenfield, R. McMaster and H. Freeman (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, pp. 121-135.
- Y. Bédard. Spatial OLAP, Vidéoconférence. 2^{ème} Forum annuel sur la R-D, Géomatique VI: Un monde accessible, 13-14 novembre, Montréal, 1997.
- Y. Bédard, E. Bernier and R. Devillers. "La métastructure VUEL et la gestion des représentations multiples", In A. Ruas (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, pp. 149-162.
- K.E. Brassel and R. Weibel. "A review and conceptual framework of automated map generalization", *International Journal of Geographical Information Systems*, Vol. 2(3):229-244, 1988.
- B.P. Buttenfield. "A rule for describing line feature geometry", In B.P. Buttenfield, R. McMaster and H. Freeman (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, pp. 150-171.
- A. Cecconi, R. Weibel and M. Barrault. "Improving automated generalisation for on-demand Web mapping by multiscale Database", In *Proceedings of the Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada, 2002.
- B. Chaib-draa, I. Jarras and B. Moulin. "Systèmes multiagents: principes généraux et applications", In J.P. Briot and Y. Demazeau (Eds.), *Principes et Architecture des Systèmes Multi-agents*, Paris, Hermès, 2001, pp. 27-70.
- T. Devogele, J. Trevisan and L. Raynal. "Processus de constitution d'une base de données multi-échelles", *Revue Internationale de Géomatique*, Vol. 6(2-3):249-263, 1996.

- D.H. Douglas and T.K. Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *The Canadian Cartographer*, Vol. 10(2):112-122, 1973.
- C. Duchêne. "Coordinative agents for automated generalisation of rural areas", In *Proceedings of the Fifth Workshop on Progress in Automated Map Generalisation, ICA, Commission on Map Generalisation*, Paris, 2003.
- GeoTools, 2006, The open source java GIS toolkit. From <http://geotools.codehaus.org/Home>.
- GiMoDig, 2004, Geospatial info-mobility service by real-time data-integration and generalisation. From <http://gimodig.fgi.fi>.
- L. Harrie and T. Sajakoski. "Simultaneous graphic generalization of vector data sets", *GeoInformatica*, Vol. 6(3): 233-261, 2002.
- N. Jabeur. *A Multi-agent System for On-the-fly Web Map Generation and Spatial Conflict Resolution*. PhD thesis, Université Laval, 2006, 275 p.
- N. Jabeur, B. Boulekrouche and B. Moulin. "Using multi-agent systems to improve real time mobile map generation", In L. Lamontagne and M. Marchand (Eds.), *Advances in Artificial Intelligence*, Springer Verlag LNAI 4013, 2006, p. 37-48.
- Jade, 2006, Java Agent Development Framework. From <http://jade.tilab.com/>.
- E.M. João. *Causes and Consequences of Map Generalisation*. Taylor & Francis: London, Bristol, PA, 1998, pp. 266.
- C.B. Jones and M. Ware. "Map generalization in the Web age", *International Journal of Geographical Information Science*, Vol. 19 (8-9):859-870, 2005.
- JTS, 2006, The JTS Topology Suite. From <http://www.vividsolutions.com/jts/jtshome.htm>.
- T. Kilpelainen. "Maintenance of multiple representation database for topographic data", *The Cartographic Journal*, Vol. 37(2):101-107, 2000.
- S. Lamy, A. Ruas, Y. Demazeau, M. Jackson, W.A. Mackaness and R. Weibel. "The application of agents in automated map generalisation", In *Proceedings of the 19th International Cartographic Conference*, Ottawa, Vol. 2:1225-1234, 1999.
- M. Li, S. Zhou and C.B. Jones. "Multi-agent systems for Web-based map information retrieval", In *Proceedings of the GIScience 2002: Geographic Information Science, Second International Conference, Boulder, CO, Lecture Notes in Computer Science*, Vol. 2478:161-180, 2002.
- R. McMaster. "Automated line generalization", *Cartographica*, Vol. 24(2):74-111, 1987.
- R. McMaster. 1991, "Conceptual framework for geographical knowledge", In B.P. Buttenfield, R. McMaster and H. Freeman (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, pp. 21-39.
- J.C. Muller. "Building knowledge thanks for rule based generalization", In *Proceedings of the International Cartographic Conference*, Bournemouth, pp. 257-256, 1991.

- J.C. Muller, R. Weibel, J. P. Lagrange and F. Salgé. "Generalization: state of the art and issues" In J.C. Muller, J.P. Lagrange and R. Weibel (Eds.), *GIS and Generalization: Methodology and Practice*, Taylor & Francis: Bristol, 1995, pp. 3-18.
- A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge MA, 1990, 549 p.
- B.G. Nickerson and H. Freeman. 1986, "Development of a rule-based system for automatic map generalization", In *Proceedings of the 2th International symposium on spatial data handling*, Seattle, Washington, 1986.
- S. Rivest, Y. Bédard and P. Marchand. "Toward better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing (SOLAP)", *Geomatica*, Vol. 55(4):539-555, 2001.
- A. Ruas. "A method for building displacement in automated map generalisation", *International Journal of Geographical Information Science*, Vol. 12(8):789-803, 1998.
- A. Ruas. *Modèles de Généralisation de Données Géographiques à Base de Contraintes et d'Autonomie*. Ph.D. thesis, Sciences de l'Information Géographique, Université de Marne-la-Vallée, 1999, 323 p.
- M.N. Sabo, Y. Bédard, E. Bernier and B. Moulin. "Integrating geometric patterns with generalization algorithms: Towards Self-Generalizing Objects and On-the-fly map generalization", Submitted to *Cartographica*, 2007.
- M.N. Sabo, A. Cardenas, Y. Bédard and E. Bernier. "Introduction du concept de patrons géométriques et application aux bâtiments afin de faciliter leur généralisation cartographique à la volée", *Geomatica*, Vol. 59(3):295-311, 2005.
- M. Schillo, K. Fischer and C. Klein. "The micro-macro link in DAI and sociology", In *Proceedings of the Multiagent-Based Simulation: Second International Workshop on Multiagent-Based Simulation*, Springer-Verlag: Boston MA, USA, 2001, pp. 133-148.
- M. Sester. "Optimization approaches for generalization and data abstraction", *International Journal of Geographical Information Science*, Vol. 19 (8-9):871-897, 2005.
- N. Stefanovic, J. Han and K. Koperski. "Object-based selective materialization for efficient implementation of spatial data cubes", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12(6):938-958, 2000.
- S. Timpf and A.U. Frank. "A multi-scale data structure for cartographic objects", In *Proceedings of the 17th International Cartographic Conference*, Barcelone, Spain, 1995.
- R.C. Thomson and D.E. Richardson. "The 'Good Continuation' principle of perceptual organization applied to the generalization of road networks". In *Proceedings of the 19th International Cartographic Conference*, Ottawa, 1999, pp. 1215–1223.
- F. Topfer and W. Pillewizer. "The principles of selection", *The Cartographic Journal*, Vol. 3(1):10-16, 1966.

- P. Van Oosterom and V. Schenkelaars. "The development of an interactive multi-scale GIS", *International Journal of Geographical Information Systems*, Vol. 9(5):489-507, 1995.
- C. Vangenot. "Représentation multi-résolutions, concepts pour la description de bases de données avec multi-représentations", *Revue Internationale de Géomatique*, Vol. 8(1-2):121-147, 1998.
- R. Weibel, E. Bernier, Y. Bédard and A. Cecconi. "La généralisation à la volée", In A. Ruas (Ed.), *Généralisation et Représentation Multiple*, Hermes science publications : Paris, 2002, pp. 319-335.
- R. Weibel and G. Dutton. "Generalizing spatial data and dealing with multiple representations", In P.A. Longley, M.F. Goodchild, D.J. Maguire and D.W. Rhind (Eds.), *Geographic Information Systems- Principles and Technical Issues*, John Wiley & Sons, 1999, pp. 125-155.
- R. Weibel and G. Dutton. "Constraint-based automated map generalization", In *Proceedings of the Spatial Data Handling*, Vancouver, Canada, 1998, pp. 214-224.
- R. Weibel. "Amplified intelligence and rule-based systems". In B.P. Buttenfield, R. McMaster and H. Freeman (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, pp. 172-186.

Conclusion

6.1 Sommaire

Dans cette thèse, nous avons présenté le concept des SGO et son application. Les SGO permettent d'encapsuler des patrons géométriques, des algorithmes de généralisation et des contraintes d'intégrité spatiale lors d'une phase d'enrichissement. Les SGO créés sont par la suite transformés en agents logiciels dans un système de généralisation à la volée. Ce système est utilisé par les utilisateurs pour générer des cartes à des échelles arbitraires. De par sa nature, le SGO se prête facilement à la navigation SOLAP, car on peut envisager de forer sur un objet en exploitant les relations hiérarchiques explicites entre les différents SGO et les relations de proximité entre des SGO voisins.

Le chapitre 1 de la thèse présente le contexte général dans lequel ces travaux ont été réalisés. De même, ce chapitre aborde les différentes problématiques de la thèse, à savoir les problèmes engendrés par la démocratisation des données spatiales et l'interactivité des nouvelles applications utilisées pour accéder à ces données. Une méthodologie qui nous a permis de mener à bien cette thèse a aussi été présentée.

Dans le chapitre 2, une revue de littérature des concepts étroitement liés à la présente thèse a été réalisée. Dans un premier temps, nous avons abordé la généralisation cartographique de même que les différents travaux majeurs qui ont été réalisés en vue de son automatisation. Par la suite, les différents travaux liés la représentation multiple ont été abordés. En effet, cette dernière est devenue une solution de rechange face aux problèmes liés à l'automatisation du processus de généralisation. Nous avons ensuite présenté les notions de la cartographie sur demande et quelques exemples d'applications (la cartographie en ligne et les SOLAP). Et enfin, nous avons passé en revue les différentes approches qui sont actuellement utilisées pour réaliser la généralisation à la volée.

Le chapitre 3 introduit le concept des patrons géométriques et leur application aux bâtiments afin de faciliter leur généralisation cartographique à la volée. Les patrons géométriques sont des formes communes à plusieurs objets de la carte. Ces patrons géométriques sont utilisés pour remplacer des objets de la carte au lieu de les généraliser à

l'aide des algorithmes conventionnels de généralisation. Les différentes opérations nécessaires pour ajuster les patrons géométriques, de même que les différentes étapes pour la généralisation des bâtiments à l'aide des patrons géométriques ont été présentées dans ce chapitre. Enfin, plusieurs tests comme la performance ou le nombre de bâtiments qui peuvent être remplacés par des patrons géométriques ont été réalisés afin de vérifier la pertinence de l'utilisation d'une telle approche.

Au chapitre 4, nous avons présenté les concepts théoriques des objets auto-généralisants (SGO). Les SGO permettent d'encapsuler dans un même objet les algorithmes de généralisation sous forme de patrons de traitement, des patrons géométriques et des contraintes d'intégrité spatiales. Les SGO permettent d'introduire dans les données les connaissances de généralisation et de les exploiter ultérieurement lors d'un processus de généralisation, y compris lors d'un processus de généralisation à la volée. Ces connaissances de généralisation sont introduites lors d'un processus d'enrichissement des données. L'utilisation des patrons géométriques accompagnée d'un enrichissement des données permet d'atteindre une grande performance lors de l'utilisation des SGO dans un processus de généralisation. Les différents types de SGO et les différentes composantes des SGO ont été aussi présentés dans ce chapitre. En plus, un bref aperçu de l'outil qui permet de créer et d'enrichir les SGO a été présenté.

Enfin, le chapitre 5 décrit le prototype de généralisation à la volée. Basé sur le concept des SGO, ce prototype permet de généraliser à la volée (quasi instantanément et sans aucune intervention humaine) les données sélectionnées par l'utilisateur. Le prototype est développé en Java en se basant sur plusieurs bibliothèques Open Source. Pour la généralisation de la carte demandée par l'utilisateur, le système utilise comme entrée les SGO créés lors d'une phase de saisie. Lors de l'initialisation du système, pour chaque SGO, un agent logiciel équivalent est créé. Et à chaque agent, des comportements sont automatiquement associés. Ces comportements permettent aux agents de déterminer les moyens qui leur permettent de généraliser l'objet qu'ils représentent. Pour générer les données demandées, les différents agents du système interagissent entre eux. L'architecture et les différents agents du système, de même que les différentes interactions entre d'une part, les différentes

composantes logicielles du système et d'autre part, entre les différents agents ont été présentés. Pour tester le système, des données réelles de la ville de Québec ont été utilisées.

6.2 Discussions

Cette thèse présente une approche de généralisation novatrice basée sur les objets auto-généralisants (SGO). Les SGO sont créés lors d'un processus de pré-généralisation (enrichissement) et sont par la suite injectés dans un système de généralisation afin de générer des cartes à des échelles arbitraires. Pour supporter une telle approche, deux prototypes ont été développés, un pour la création et l'enrichissement des SGO et un deuxième utilisant la technologie multi-agents permettant de généraliser les cartes à la volée. Dès lors qu'ils sont injectés dans un système de généralisation, les SGO peuvent être utilisés par les utilisateurs pour générer des cartes à des échelles arbitraires. Pour cela, aucune connaissance cartographique n'est requise des utilisateurs finaux du système. Pour valider l'approche présentée et tester les prototypes développés (*SGO Creation Engine* et *On-the-fly Map Generalization System*), des données réelles ont été utilisées. Les tests se sont avérés très concluants de par la performance de l'approche et des systèmes développés pour supporter cette approche et la qualité des données générées. En effet, le temps moyen nécessaire pour généraliser nos cartes (une centaine d'objets à des échelles qui varient entre 1K et 50K) est de l'ordre de 300 millisecondes.

En ce sens, nous pensons que l'hypothèse principale de la thèse qui est que «l'intégration d'algorithmes de généralisation, de patrons géométriques et de contraintes d'intégrité spatiales pour créer des objets auto-généralisant (SGO) peut supporter un processus de généralisation à la volée» est vérifiée. L'objectif principal de la thèse était de «développer un nouveau concept appelé SGO capable de supporter un processus de généralisation à la volée, puis d'en tester le potentiel grâce à des données réelles». Nous pensons aussi que cet objectif a été atteint. En effet, le temps nécessaire à la généralisation des cartes à l'aide de l'approche développée est compatible avec la bande cognitive de Newell (1990) qui est inférieure à dix secondes, temps au-delà duquel les utilisateurs ont tendance à ne plus attendre les résultats de leur requête. Cependant, les tests ont été réalisés avec un nombre très restreint de données. À première vue, ce nombre paraît insuffisant, mais lors d'un

processus de visualisation interactive, les utilisateurs visualisent souvent des petites portions successives de territoires (donc un nombre réduit d'objets à la fois) surtout quand on sait que beaucoup de ces utilisateurs accèdent de plus en plus aux données grâce à des appareils de taille réduite, de type PDA (donc, visualisation d'un nombre encore plus réduit d'objets à la fois). Par contre, pour des médias avec des grandes surfaces d'affichages (ex. ordinateurs de bureau), l'étendue des territoires visualisée devient plus importante, par conséquent, le nombre d'objets à généraliser devient plus important. C'est pourquoi, comme perspective de recherche, un déploiement avec un plus grand nombre d'objets est envisagé. Mais, les tests déjà réalisés lors de cette recherche ont démontré que la performance du système développé augmente avec la diminution de l'échelle de visualisation. Ce constat laisse à croire que même avec un nombre élevé d'objets à généraliser, la performance ne sera pas considérablement affectée surtout quand on sait que le système actuel peut être optimisé davantage et d'autre part la performance de l'ordinateur sur lequel les présents tests ont été réalisés laisse à désirer. Du point de vue de la généralisation cartographique, l'analyse visuelle et la comparaison des résultats vis-à-vis des données généralisées à l'aide des algorithmes habituels de généralisation (figures 18-c et 18-d) ont montré que la qualité esthétique des cartes générées par le système est satisfaisante. D'autre part, les principales contraintes de lisibilité (ex. contrainte de séparation minimale entre les objets) utilisée lors d'un processus de généralisation ont été respectées.

Grâce à ses principaux composants (patrons géométriques, patrons de traitements et les contraintes), les SGO permettent d'introduire les connaissances des experts dans les objets lors d'un processus d'enrichissement sans faire recourt aux outils formels de représentation des connaissances (ex. les graphes conceptuels). Lors de ce processus d'enrichissement, des connaissances procédurales (ex. le choix des algorithmes et leur enchaînement), structurales (ex. les contraintes d'intégrités) sont créées et pérennisées dans la base de données. Cette pérennisation des connaissances de généralisation présente un grand intérêt pour des organisations qui produisent des données multiéchelles en utilisant des processus de généralisation. En effet, dans les approches traditionnelles de généralisation (qui sont le plus souvent semi-manuelles), l'essentiel du travail (ex. le choix des algorithmes ou le choix des objets auxquels on veut appliquer une opération donnée) qui conduit à la

généralisation de la carte est perdu et lorsqu'on veut générer une autre carte du même territoire, tout ce travail doit être repris. Mais, avec les SGO, la matérialisation de ces connaissances pour chaque SGO ou groupe de SGO permet sa réutilisation. D'autre part, cette introduction des connaissances a priori peut grandement faciliter la formalisation des connaissances de généralisation. En effet, Lors d'un processus de généralisation, des connaissances structurales (ex. alignement) et procédurales (ex. le choix des algorithmes et leur séquence) sont utilisées afin de prendre la décision sur la manière de généraliser chaque objet ou situation de la carte. Or il est connu depuis plusieurs années que la formalisation de ces connaissances est une tâche complexe dont les résultats sont souvent incertains. En effet, plusieurs chercheurs considèrent le problème de formalisation des connaissances des cartographes comme étant une des entraves majeures qui empêchent l'automatisation complète de la généralisation cartographique. Dans une approche de généralisation utilisant les SGO, ces connaissances structurales et procédurales sont caractérisées par l'opérateur lors d'une phase d'enrichissement précédant la généralisation. Ceci simplifie grandement la tâche.

Les SGO ont une structure hiérarchique dont chaque niveau correspond à un niveau d'abstraction donné. D'autre part, les SGO permettent de générer des données à des échelles arbitraires. Il est donc clair que les SGO peuvent faciliter la création des bases de données à représentation multiple, dans lesquelles les différentes représentations d'un même objet peuvent être inter-liées. En plus, les liens explicites qui existent entre les différents niveaux d'abstraction, de même que les outils de généralisation inclus dans les SGO peuvent éventuellement faciliter une propagation des mises à jour dans une telle base de données à représentation multiples.

Ces dernières années, plusieurs modèles multi-agents ont été proposés dans le but d'automatiser le processus de généralisation cartographique (ex. Ruas 1999, Jabeur 2006). Le modèle proposé par Ruas (1999) dans le cadre du projet AGENT est basé sur une structure de type quartier — îlot gérée par l'agent Méso qui manage d'une manière centralisée toutes les contraintes. Ce modèle est orienté sur la production cartographique traditionnelle (la production des cartes topographiques en zone urbaine). Quant au modèle proposé par Jabeur (2006) dans le cadre du projet GEMURE, il se base exclusivement sur

l'existence des agrégats qui proviennent de la représentation multiple. Contrairement à ces deux approches, la structure des SGO présentée dans cette thèse se base sur les contraintes d'intégrité spatiales (des relations qui doivent être préservées lors du processus de généralisation) qui sont détectées lors d'une phase d'enrichissement des données. Une telle structure n'est ni restreinte aux zones urbaines, ni conditionnée par la représentation multiple.

Pour supporter un processus de généralisation à la volée (processus quasi instantané), plusieurs travaux récents proposent des approches basées sur l'utilisation de la représentation multiple comme support à la généralisation (Jabeur 2006, Cecconi 2003, GiMoDig 2004). Contrairement à ces approches, notre approche se base sur l'utilisation des patrons géométriques et un processus d'enrichissement de la base de données. Les patrons géométriques permettent de minimiser la redondance des données tout en accélérant le processus de généralisation. Quant à l'enrichissement, en plus d'accélérer le processus (car certaines opérations sont réalisées à l'avance), il permet de surmonter certaines limitations actuelles de la généralisation cartographique (ex. le choix des algorithmes). Dans la majorité des approches de généralisation (ex. Ruas 1999, Jabeur 2006), le choix des algorithmes se fait à la volée. Contrairement à ces approches, dans la nôtre, ce choix est réalisé au préalable lors de la phase d'enrichissement grâce aux patrons de traitement, en profitant de l'expertise humaine sans pour autant compromettre la flexibilité du système. Les principales distinctions entre notre approche et d'autres approches de généralisation semblables à la nôtre sont présentées à l'annexe 2 de la thèse.

Même si cette approche est générique, l'intensité de l'enrichissement introduit dans les données peut être considérablement influencée par les besoins du producteur des données (ex. l'organisation qui va créer les SGO). En effet, tous les producteurs n'ont pas les mêmes besoins en terme de la qualité des données recherchée ou de la plage d'échelles à couvrir. Par exemple, un producteur de données topographiques va plus chercher la fidélité géométrique des données qui seront générées contrairement au producteur des cartes touristiques qui privilégiera probablement la fidélité sémantique. C'est pourquoi, avant l'enrichissement des données, il serait souhaitable de se poser les questions sur les types d'usages éventuels du système et sur la plage d'échelle des cartes qui seront générées. À

défaut de répondre à ces questions, des avertissements spécifiant les limites du système peuvent éventuellement être intégrés dans le système, surtout s'il est destiné au grand public.

À première vue, l'effort nécessaire pour la création et l'enrichissement des SGO peut être considéré comme étant le point faible de notre approche. Par contre, nous pensons que le bénéfice qu'on peut tirer de l'expertise humaine grâce à l'enrichissement (ex. l'amélioration de la qualité des données générées et la performance) justifie un tel investissement. D'autre part, cet enrichissement est nécessaire une seule fois et le résultat peut être utilisé pour générer des cartes à différentes échelles. En plus, certaines tâches (ex. l'ajustement des patrons) peuvent éventuellement être automatisées.

6.3 Conclusions

Les résultats de cette thèse nous ont permis de tirer les conclusions suivantes :

- Il est possible de mettre au point des méthodes et des outils efficaces capables de générer des données à des échelles arbitraires en utilisant des approches de généralisation simples et performantes (ex. les patrons géométriques) et un enrichissement adéquat des données. En effet, l'approche développée satisfait les critères que nous avons fixés, à savoir, un temps de réponse quasi instantané sans intervention humaine;
- L'utilisation combinée des patrons géométriques et des algorithmes de généralisation permet la création des systèmes de généralisation performants tout en minimisant les problèmes de redondance des données liés aux approches de généralisation à la volée basées sur la représentation multiple traditionnelle. En effet, dans notre approche des données mono échelle sont utilisées. À ces données on associe des patrons géométriques, mais comme un même patron peut être associé à plusieurs occurrences d'objets, ceci réduit énormément la redondance;
- Contrairement à la plupart des approches de généralisation à la volée utilisant les algorithmes de généralisation qui se caractérisent par une application séquentielle

des algorithmes sans aucune prise en compte du contexte des objets, les SGO permettent de générer à la volée des données de qualité esthétique satisfaisante (voir chapitre 3 figure 18 et chapitre 5 tableau 2);

- Le fait de généraliser les patrons géométriques donne plus de contrôle sur les résultats de la généralisation, donc des résultats « prévisibles ». Ce contrôle est rendu possible grâce d'une part à la structure en composants (une juxtaposition de primitives) des patrons géométriques et d'autre part, à la simplicité des algorithmes utilisés pour généraliser ces patrons (ex. la simplification est réalisée en enlevant tout simplement une primitive) et à la supervision de toutes ces opérations par les SGO.

6.4 Limitations de l'approche et perspectives de recherche

L'approche et les prototypes présentés dans cette thèse permettent de générer des cartes à des échelles arbitraires. Toutefois, l'approche présente aussi quelques limitations, c'est pourquoi plusieurs aspects de la thèse mériteraient d'être approfondis :

- Une prise en compte plus efficace des contraintes sémantiques. Actuellement, notre approche met l'emphase sur les contraintes spatiales. Mais, pour une généralisation à la volée plus efficace, les contraintes sémantiques doivent être efficacement prises en compte. Pour cela, comme perspective il serait important d'introduire des contraintes sémantiques dans les SGO (ex. pour une carte touristique, ne pas réaliser des agrégations entre des bâtiments abritant un musée et un commerce) pour guider certaines opérations de généralisation. On peut par exemple modéliser ces contraintes sous forme d'ontologies selon les types d'utilisation de la carte. Par exemple, on peut avoir une ontologie des contraintes sémantiques pour les cartes touristiques. Ainsi, lors de l'utilisation du système, l'ontologie la plus appropriée peut être choisie selon le type des cartes qui seront générées. En deuxième temps, on peut penser permettre à l'utilisateur de créer son propre profil, donc sa propre ontologie. Afin de prendre efficacement les besoins des utilisateurs en compte lors

de la modélisation de ces ontologies, les travaux du groupe de travail de l'Association Cartographique Internationale sur les besoins des utilisateurs (ICA Working Group on Use and User Issues 2007) peuvent grandement aider à préciser les principaux besoins qui peuvent être exprimés par les différents groupes d'utilisateurs;

- Les SGO sont aussi des moyens qui permettent de récupérer le travail réalisé en amont par le cartographe afin de généraliser une carte. Dans le cadre de cette recherche, un outil de saisie permettant la création des SGO a été développé. Comme perspective, cet outil de saisie peut être couplé à un système de généralisation interactive de type DynaGEN. Ceci permettrait de récupérer le travail des cartographes lors de la généralisation interactive afin de peupler la base de données des SGO ;
- Actuellement, le système de généralisation à la volée développé sert de preuve de concept et fut fait avec deux thèmes inter-reliés (les bâtiments et les rues) et testé avec des données réduites (couvrant une petite superficie). Il serait intéressant d'implémenter d'autres classes d'objets comme l'hydrographie et de réaliser des tests avec des données couvrant un grand territoire. Par contre, un déploiement à grande échelle (comportant plusieurs feuillets cartographiques) peut nécessiter un travail d'harmonisation entre certains feuillets quand on sait que tous les feuillets ne suivent pas toujours les mêmes spécifications du point de vue généralisation cartographique (par exemples, au Canada, les spécifications peuvent varier d'une province à une autre);
- Implémenter le système de généralisation à la volée dans un système de cartographie sur le Web (ex. UMapIT) ou dans un outil SOLAP. Dans cette perspective, il est important d'incorporer des méthodes de transmission des données plus efficaces et adaptées au contexte Web. Dans ce cas, nous pouvons tirer avantage de l'approche de génération progressive des cartes par couches d'intérêt proposée par Jabeur (2006) et aussi adresser le problème de mismatch agrégation/généralisation cartographique défini par Bédard et al. (2007).

- Bien que l'outil de saisie développé facilite énormément la création des SGO, il est important de noter que beaucoup d'opérations (ex. le choix et l'ajustement des patrons géométriques) restent encore interactives. Comme perspective, il serait important d'automatiser davantage ce processus de saisie. Par exemple dans leurs travaux, Rainsford et Mackaness (2002) ont proposé une approche qui permet d'apparier automatiquement des templates alphanumériques semblables à nos patrons géométriques. Une autre voie serait d'explorer l'utilisation des "squelettes" afin d'extraire la forme des objets pour lesquels on veut chercher un patron géométrique. L'analyse des squelettes peut nous renseigner sur la forme de l'objet et sur le patron qui convient le mieux. Dans ses travaux de thèse de Doctorat, actuellement en cours au département des sciences géomatiques de l'Université Laval, Benoît Frédérique a développé une approche paramétrique de saisie en mode représentations multiples basée sur la reconnaissance des formes des bâtiments. Une troisième voie serait d'explorer la possibilité d'utiliser les paramètres utilisés dans cette approche afin d'identifier et de paramétrer les patrons des bâtiments lors de la saisie.
- Actuellement, la simplification des patrons se fait en éliminant une primitive au complet. Il serait toutefois intéressant d'implémenter la généralisation à granularité variable (e.g. diminuer la longueur d'une primitive lors de la généralisation au lieu de l'éliminer complètement) et de tester les conséquences d'une telle approche sur la qualité et la perception des données généralisées.
- Actuellement, la mémorisation des actions réalisées par les agents SGO a été conservée de façon simple. Il serait toutefois intéressant d'explorer l'utilisation des techniques d'apprentissage automatique afin d'améliorer le processus de mémorisation des agents SGO.

6.5. Références

Bédard Y., Rivest S. et Proulx M.-J., "Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective".

ICA Working Group on Use and User Issues. Disponible sur <http://plone.itc.nl/icawguse>.
2007

Wrembel R. et Koncilia C. (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press (Idea Group), London, UK, 2007, p. 298-319.

Bibliographie

(Les références citées dans les chapitres de la thèse sont identifiées par un astérisque : *).

- *Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. et Angel S., *A Pattern Language: Towns, Buildings, Construction*. New York, Oxford University Press, 1977, 1171p.
- *Allouche M. et Moulin B., "Amalgamation in cartographic generalization using Kohonen's feature nets", *International Journal of Geographical Information Science*, Vol. 19(8-9), p. 899-914, 2005.
- *Allouche M. et Moulin B., "Reconnaissance de patterns par réseaux de neurones : application à la généralisation cartographique". *Revue Internationale de Géomatique*, Vol. 11(2), p. 251-279, 2001.
- *Armstrong M.P., "Knowledge classification and organization", Buttenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 86-102.
- Babitchev B.A., Babilov I.E., Nevskaya G.A. et Hudiakov E.I. *Rukavodstva Po Kartografitcheskim i Kartoizdatelskim Rabotam*. Moskva, Redakcioni-Izdatelski otdel, 1985.
- *Bader M., *Energy Minimization Methods for Feature Displacement in Map generalization*. Thèse de Doctorat, Université de Zürich, Suisse, 2001, 201 p.
- *Baeijs C., Demazeau Y. et Alvares L., "Application des systèmes multi-agent à la généralisation cartographique", In *Proceedings of the 3èmes Journées Francophones sur l'intelligence Artificielle Distribuée et les Systèmes Multi-agents, AFCE&AFIA*, Chambéry, 1995, p. 163-176.
- *Barrault M. et Bader M., "Les algorithmes de déplacement". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p.287-302.
- *Beard K.M., "Constraints on rule formation." Buttenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 121-135.
- *Beard M.K., "How to survive on a single detailed database". *Proceeding of 8th International Symposium on Computer-Assisted Cartography*, Baltimor, Maryland, *AUTO-CARTO*, 1987, p. 211-220.
- *Bédard Y., Bernier E. et Devillers R., "La métastructure VUEL et la gestion des représentations multiples". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002, p. 149-162.

- *Bédard Y., Proulx M.-J. et Rivest S., "Enrichissement du OLAP pour l'analyse géographique : exemples de réalisations et différentes possibilités technologiques". In *Proceedings of the Première Journée Francophone sur les Entrepôts de Données et l'Analyse en Ligne*, Lyon, Soumis en 2005.
- *Bédard Y., Rivest S. et Proulx M.-J., "Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective". Wrembel R. et Koncilia C. (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press (Idea Group), London, UK, 2007, p. 298-319.
- *Bédard Y., Spatial OLAP, Vidéoconférence. 2^{ème} Forum annuel sur la R-D, Géomatique VI: Un monde accessible, 13-14 novembre, Montréal, 1997.
- Bellifemine F., Poggi A. et Rimassa G., "Developing multi-agent systems with jade". In *Proceedings of the Proceedings Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, 2000.
- *Bernier E., Bédard Y. et Hubert F., "UMapIT: An On-Demand Web Mapping Tool Based On A Multiple Representation Database". In *Proceedings of the 8th ICA WORKSHOP on Generalisation and Multiple Representation*, A Coruña, Spain, 2005.
- *Bernier E., Bédard Y. et Lambert M., "Cartographie sur demande sur le Web et bases de données multidimensionnelles. De la personnalisation par couche cartographique à la personnalisation par occurrences". *Revue Internationale de Géomatique*, Vol. 13(3), p. 339-359, 2003.
- *Bernier E. et Bédard Y., "A data warahouse strategy for on-demand multi-scale mapping". Mackaness W., Ruas A. et Sarjakoski T. (Eds.), *Generalization of Geographic Information: Cartographic Modelling and Applications*, 2007, p. 177-198.
- *Bernier E., *Utilisation de la Représentation Multiple comme Support à la Génération de Vues de Bases de Données Géospatiales dans un Contexte SOLAP*. Mémoire de Maîtrise, Université Laval, 2002, 89 p.
- *Bertolotto M. et Egenhofer M., "Progressive Transmission of Vector Map Data over World Wide Web". *Geoinformatica*, Vol. 5 (4), p. 345-373, 2001.
- *Boulekrouche B., *Génération Cartographique à la Volée pour les Terminaux Mobiles de Type PDA*. Mémoire de Maîtrise, Université Laval, 2005.
- *Brassel K.E. et Weibel R., "A review and conceptual framework of automated map generalization". *International Journal of Geographical Information Systems*, Vol. 2(3), p. 229-244, 1988.
- *Brassel K.E., "Strategies and data models for computer-aided generalization". *International Yearbook of Cartography*, Vol. 25, p. 11-29, 1985.
- Brophy D.N., "An automatic methodology for linear generalization in thematic cartography". In *Proceedings of the Proceeding of the ASCM Conference, 33rd Meeting*, Washington D.C., 1973.

- *Buttenfield B. et McMaster R., "Preface". Buttenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. xiii-xvi.
- *Buttenfield B., "A rule for describing line feature geometry", Buttenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 150-171.
- *Buttenfield B., "Transmitting Vector Geospatial Data across the Internet". *GIScience 2002, Boulder, CO, USA Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg, 2002, p. 51-64.
- *Buttenfield B., "Treatment of cartographic lines". *Cartographica*, Vol. 22(2), p. 1-26, 1985.
- *Cardenas A. *Utilisation Des Patrons Géométriques comme Support à la Généralisation Automatique*. Mémoire de maîtrise, Université Laval, 2004, 110 p.
- *Caron P. Y., *Étude du Potential de l' OLAP pour Supporter l'Analyse Spatio-temporelle*. Mémoire de Maîtrise, Université Laval, 1998, 132 p.
- Cartwright W., "Maps on the Web". Peterson M. (Ed.), *Maps and the Internet*, London, Elsevier, 2003, p. 35-56.
- *Cecconi A., *Integration of Cartographic Generalization and Multi-scale Databases for Enhanced Web Mapping*. Thèse de Doctorat, Université de Zurich, 2003, 155 p.
- *Cecconi A., Weibel R. et Barrault M., "Improving automated generalisation for on-demand web mapping by multiscale database". *Proceeding of Joint International Symposium on Geospatial Theory, Processing and Applications*, July 8-11, Ottawa, Canada, 2002, p. 1-9.
- *Chaib-draa B., Jarras I. et Moulin B., "Systèmes multiagents: principes généraux et applications", In Briot J.P. et Demazeau Y. (Eds.), *Principes et Architecture des Systèmes Multi-agents*, Paris, Hermès, 2001, p. 27-70.
- Clarke K.C., *Analytical and computer cartography*. Prentice Hall, 1995, 288p.
- *Coad P. "Object-Oriented Patterns". *Communications of the ACM*, 35(9), ACM Pres, New York, USA, 1992, p. 152-159.
- *Cromley R. G., "Hierarchical methods of line simplification". *Cartography and GIS*, Vol. 18(2), p. 125-131, 1991.
- Cromley R.G. et Campell M.G., "A geometrically efficient bandwidth line simplification". In *Proceedings of the 4th International Symposium on Spatial Data Handling*, Zurich, Switzerland, 1990.
- *Cuenin R., *Cartographie Générale*. Paris, Eyrolles, 1972, 324 p.
- *Delucia A. et Black T., "A comprehensive approach to automatic feature generalization". *Proceeding of 13th International Cartographic Association Conference*, Morelia, Mexico, 1987.

- *Devogele T., Badard T. et Libourel T., "La problématique de la représentation multiple". Ruas A. (Ed.), *Généralisation et Représentation Multiple*. Paris, Hermes science publications, 2002, p. 55-74.
- *Devogele T., Trevisan J. et Raynal L., "Processus de constitution d'une base de données multi-échelles". *Revue Internationale de Géomatique*, Vol. 6(2-3), p. 249-263, 1996.
- *Didier M. et Bourveyron C., *Guide Économique et Méthodologique des SIG*. Paris, Hermès, 1993.
- *Dougenik J., "Whirpool: A geometric processor for polygone coverage data". *Proceedings of AUTO-CARTO*, Vol. 4, p. 304-311, 1980.
- *Douglas D.H. et Peucker T.K., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". *The Canadian Cartographer*, Vol. 10(2), p. 112-122, 1973.
- Duchêne C. "The CartACom model: a generalisation model for taking relational constraints into account". In *ICA Workshop on Generalisation and Multiple Representation*, Leicester, 2004.
- *Duchêne C. et Regnauld N., "Le modèle AGENT". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Hermes science publications, 2002, p. 369-385.
- *Duchêne C., "Coordinative agents for automated generalisation of rural areas". *Proceeding of Fifth Workshop on Progress in Automated Map Generalisation, ICA, Commission on Map Generalisation*, Paris, 2003.
- Dutton G., "Fractal enhancement of cartographic line detail". *The American Cartographer*, Vol. 8(1), 1981, p. 23-40.
- *DynaGEN : user support. Disponible sur <http://imgssupport.intergraph.com/dynagen/>.
- *Eckert M. et Joerg W., "On the Nature of Maps and Map Logic". *Bulletin of the American Geographical Society*, Vol. 40(6), p. 344-351, 1908.
- *Edward P.F.C. et Kevin K.W.C., "On multi-scale display of geometric objects". *Data&Knowledge Engineering*, Vol. 40, p. 91-119, 2002.
- Egenhofer M.J. et Herring J.R., "A mathematical framework for the definition of topological relationships". In *Proceedings of the 4th International Symposium on Spatial Data Handling*, Zurich, 1990.
- ESRI. *Automation of Map Generalization, The cutting-Edge Technology*. ESRI White Paper Series, Redlands, 1996.
- Ferber J., *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*. Harlow: Addison-Wesley, 1999, 528 p.
- *FIPA. 2006. The Foundation for Intelligent Physical Agents. Disponible sur <http://www.fipa.org/>.
- Fisher P.F. et Mackaness W.A., "Are cartographic expert systems possible?" In *Proceedings of the Auto-Carto 8*, Baltimore, 1984, p. 530-534,

- Fisher P.F., Mackaness W.A., Peacegood G. et Wilkinson G.G., "Artificial intelligence and expert system in geo-data processing". *Progress in Physical Geography*, Vol. 12, 1988, p. 371-388.
- Follin J.M. *Gestion Incrémentale de Données Multi-résolutions dans un Système Mobile de Visualisation d'Informations Géographiques*. Thèse de Doctorat, Université De La Rochelle, 2004, 172 p.
- *Fortin C. et Rousseau R. *Psychologie Cognitive : Une Approche de Traitement de L'information*. Sillery, Presses de l'université du Québec, 1992, 434 p.
- Frank R. et Ester M., "Evaluating the quality of generalized maps by using a quantitative map similarity measure". *13th ACM International Symposium of GIS*, November 12-13th, Bremen, Germany, submitted, 2005.
- Galanda M. et Weibel R., "An agent-based framework for polygonal subdivision generalization". Richardson D. et van Oosterom P. (Eds.), *Advances in Spatial Data Handling. 10th International Symposium on Spatial Data Handling*, Springer-Verlag Berlin Heidelberg, 2002, p. 121-136.
- *Gamma E., Helm R., Johnson R. et Vlissides J. *Design Patterns: Elements of Reusable Object-oriented Software*. Reading, Mass. Don Mills, Ont., Addison-Wesley, xv, 1995, 395 p.
- Gao W., Gong J. et Li Z., "Knowledge-based generalization on land-use data". In *Proceedings of the ICA Generalisation Workshop*, Paris, 2003.
- *Gbei E., Moulin B., Cosma I., Jabeur N. et Delval N., "Conception d'un prototype de service Web géolocalisé appliqué à l'industrie récréotouristique". *Revue Internationale de Géomatique*, Vol. 13, p. 375-395, 2003.
- *Geospatial info-mobility service by real-time data-integration and generalisation. Disponible sur <http://gimodig.fgi.fi>.
- *GeoTools, 2006, The open source java GIS toolkit. From <http://geotools.codehaus.org/Home>.
- *GiMoDig 2004. Geospatial info-mobility service by real-time data-integration and generalisation. Disponible sur <http://gimodig.fgi.fi>.
- *Gothic LAMPS2, Product Feature Overview. Disponible sur <http://www.laser-scan.com/products/lamps2.htm>.
- *Grünreich D., "Development of computer-assisted generalization on the basis of cartographic model theory". Müller J.C., Lagrange J.P. et Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*, Bristol: Taylor & Francis, 1995, p. 47-55.
- Hampe M., Anders K.H. et Sester M., "MRDB applications for data revision and real-time generalisation". In *Proceedings of the 21st International Cartographic Conference*, 2003.
- *Hardy P., Hayles M. et Rewell P., "Clarity- A new environment for generalisation using agents, java, XML and topology". *Proceeding of ICA Workshop on Generalisation and Multiple Representations*, Paris, 2003.

- Hardy P.G., "Map production from an active object database, using dynamic representation and automated generalization". *The Cartographic Journal*, Vol. 35 Number 2, 1998, p. 181-189.
- *Harrie L. et Sajakoski T., "Simultaneous graphic generalization of vector data sets", *GeoInformatica*, Vol. 6(3), p. 233-261, 2002.
- Harrie L., "The constraint method for solving spatial conflicts in cartographic generalization". *Cartography and Geographic Information Science*, Vol. 26(1), 1999. p. 55-69.
- Harrie L., Sarjakoski T. et Lehto L., "A mapping function for variable-scale maps in small-display cartography". *Journal of Geospatial Engineering*, Vol. 2(3), 2002, p. 111-123.
- *InfoWorld, The speed of business: If your pages are slow, your customers will go. Disponible sur <http://www.infoworld.com/articles/op/xml/00/09/11/000911opsavvy.html>.
- *ICA Working Group on Use and User Issues. Disponible sur <http://plone.itc.nl/icawguse>. 2007
- *Internet World Stats 2006. World, Internet Users and Population Stats. Disponible sur <http://www.internetworldstats.com/stats.htm>.
- *Jabeur N. *A Multi-agent System for On-the-fly Web Map Generation and Spatial Conflict Resolution*. Thèse de Doctorat, Université Laval, 2006, 275 p.
- *Jabeur N., Boulekrouche B. et Moulin B., "Using multi-agent systems to improve real time mobile map generation", Lamontagne L. et Marchand M. (Eds.), *Advances in Artificial Intelligence*, Springer Verlag LNAI 4013, 2006, p. 37-48.
- *JADE 2006. Java Agent Development Framework. Disponible sur <http://jade.tilab.com/>
- Jenk G.F., "Geographic logic in line generalization". *Cartographica*, Vol. 26(1), p. 27-43, 1989.
- *Jennings N. R., Wooldridge M. et Sycara K., "A roadmap of agent research and development". *International Journal of Autonomous Agents and Multi-Agents Systems*, Vol. 1(1), p. 7-38, 1998.
- *João E.M., *Causes and Consequences of Map Generalisation*. London, Bristol, PA, Taylor & Francis, 1998, 266 p.
- *Jones C. et Ware M., "Map generalization in the Web age", *International Journal of Geographical Information Science*, Vol. 19 (8-9), p. 859-870, 2005.
- *Jones C., "Database architecture for multi-scale GIS". *Auto-Carto*, Vol. 6(10), p. 1-14, 1991.
- *Jones C., Abdelmoty A. I., Lonergan M. E., van der Poorten P. et Zhou S., "Multi-Scale Spatial Database Design for Online Generalisation". *Proceeding of 9th Spatial Data Handling Symposium*, Beijing, p. 7b.34-7b.44, 2000.
- *JTS. 2006. JTS Topology Suite. Disponible sur <http://www.vividsolutions.com/jts/jtshome.htm>.

- Kettani D. et Moulin B., "A spatial model based on the notions of spatial conceptual map and of object's influence areas". Freska C. et Mark D.M. (Eds.), *Spatial Information Theory, Cognitive and Computational Foundations of Geographic Information Systems, Proceedings of the International Conference COSIT'99*, Springer Verlag LNCS 1661, 1999, p. 401-415.
- *Kilpelainen T., "Maintenance of multiple representation database for topographic data". *The Cartographic Journal*, 37(2), 2000, p. 101-107.
- Kraak M.-J. et A. Brown. *Web Cartography: Developments and prospects*. Londres, Taylor and Francis, 2001, 213 p.
- *Kraak M.J., "Setting and needs for web cartography". Kraak M.-J. et Brown A. (Eds.), *Web Cartography: Developments and Prospects*, Londres, Taylor & Francis, 2001, p. 1-7.
- *Lamy S., Ruas A., Demazeau Y., Jackson M., Mackaness W.A., et Weibel R., "The Application of Agents in Automated Map Generalisation." In *Proceedings of the 19th International Cartographic Conference*. Ottawa, 1999, Vol. 2, p. 1225-1234.
- *Lang T., "Rules for the robot draughtsmen". *The Geographical Magazine*, Vol. 42(1), p. 50-51, 1969.
- Laser-scan, 2003. *Gothic LAMPS2 Product Feature Overview*. Disponible sur <http://www.laser-scan.com/products/lamps2.htm>.
- Lehto L. et Kilpelainen T., "Generalizing XML-encoded spatial data on the Web". In *Proceedings of the 20th International Cartographic Conference*, 6–10 August, Beijing, China, 2001.
- *Li M., Zhou S. et Jones B., "Multi-agent systems for Web-based map information retrieval." *Proceeding of GIScience 2002: Geographic Information Science, Second International Conference, Boulder, CO, Lecture Notes in Computer Science*, 2002, Vol. 2478, p. 161–180.
- Lichtner W., "Computer-Assisted Processes of Cartographic Generalization in Topographic Maps". *Geo-Processing*, 1979, Vol. 1(1), p. 183-199.
- *Mackaness W.A. "An Algorithm for Conflict Identification and Feature Displacement in Automated Map Generalization." *Cartography and Geographic Information Systems*, Vol. 21(4), p. 219-232, 1994.
- *Mackaness W.A. et Edwards G., "The Importance of Modelling Pattern and Structure in Automated Map Generalisation". *Proceeding of Joint ISPRS/ICA Workshop on Multi-scale Representations of Spatial Data*, July 7-8, Ottawa, Canada, 2002.
- *Mailing D., *Measurements from Maps- Principles and Methods of Cartometry*, Oxford, Pergamon Press, 1989, 450 p.
- *Maozhen L., Sheng Z. et Jones C., "Multi-agent Systems for Web-Based Map Information". Egenhofer M.J. et Mark D.M. (Eds.), *GIScience*, Verlag Berlin, 2002, p. 161-180.
- *MapQuest. 2004. Advertise with us. Disponible sur <http://www.mapquest.com/features/main.adp?page=advertise>.

- *Marchand P., Bédard Y. et Edwards G., "A hyper cube-based method for spatio-temporal exploration and analysis". *Soumis à Geoinformatica* en May 2001.
- *Martel C. *Développement d'un Cadre Théorique pour la Gestion des Représentations Multiples dans les Bases de Données Spatiales*. Mémoire de maîtrise, Université Laval, 1999, 139 p.
- *McMaster R. "Conceptual framework for geographical knowledge." Battenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 21-39.
- *McMaster R. et Shea K.S., "Cartographic generalization in digital environment: A Framework for implementation in GIS". *Proceeding of GIS/LIS'88*, San Antonio, Texas, 1998, p. 240-249
- *McMaster R. et Shea K.S., "Generalization in digital cartography". *Association of American Geographers*, Washington, D.C., 1992.
- McMaster R., "A statistical analysis of mathematical measures for linear simplification". *The American Cartographer*, Vol. 13(2), 1986, p. 106.
- *McMaster R., "Automated line generalization", *Cartographica*, Vol. 24(2), p. 74-111, 1987.
- *McMaster R., "The integration of simplification and smoothing algorithms in line generalization". *Cartographica*, Vol. 26(1), p. 101-121, 1989.
- Monmonier M., "Regionalizing and matching features for interpolated displacement in automated generalisation of digital cartographic databases". *Cartographica*, Vol. 26(2), 1989, p. 21-39.
- *Morrison J.L., "A theoretical framework for cartographic generalization with emphasis on the process of symbolisation". *International Yearbook of Cartography*, Vol. 14, p. 115-127, 1974.
- *Müller J. C., "Building knowledge thanks for rule based generalization". *Proceeding of International Cartographic Conference*, Bournemouth, 1991, p. 257-256
- *Müller J. C., "Fractal dimension and inconsistencies in cartographic line representations". *The Cartographic Journal*, Vol. 23, 123-130.
- *Müller J. C., "Generalization of spatial database". Maguire D., Goodchild M. et Rhind D. (eds.), *Geographical Information Systems Principles and Applications*, Vol. 1, Harlow, Logman, p. 457-475.
- Müller J.C., "The removal of spatial conflicts in line generalization". *Cartography and GIS*, Vol. 17(2), 1990, p. 141-149.
- *Müller J.C., Lagrange J. P. et Weibel R., "Editors' Preface". Müller J.C., Lagrange J.P. et Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. ix.

- *Müller J.C., Weibel R., Lagrange J.P. et Salgé F., "Generalization: state of the art and issues". Müller J.C., Lagrange J.P. et Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. 3-18.
- *Newell A. *Unified Theories of Cognition*. Cambridge MA, Harvard University Press, 1990, 549 p.
- *Nickerson B.G. et Freeman H. "Development of a rule-based system for automatic map generalization." In *Proceedings of the 2th International Symposium on Spatial Data Handling*, Seattle, Washington, 1986, p. 537-556.
- *Nickerson B.G., "Automatic cartographic generalization for linear features". *Cartographica* Vol. 25, p. 15-66, 1988.
- *Opheim H., "Fast Reduction of a Digitized Curve". *Geo-Processing*, Vol. 2, p. 33-40, 1982.
- *Perkal J., "An attempt at Objective Generalization". *Michingan Inter-University Community of Mathematical Geographers*, Discussion paper 10, University of Michigan, 1966.
- *Peterson M. P., "Trends in Internet map use, a second look. In touch the past visualize the future". *Proceeding of 19th International Cartographic Conference*, Ottawa, Canada, 1999.
- *Peterson M., "Maps and the Internet: An Introduction". Peterson M. (Ed.), *Maps and the Internet*, London, Elsevier, 2003, p.1-16.
- *Peterson M., "Preface". Peterson M. (Ed.), *Maps and the Internet*, London, Elsevier, 2003, p.ix-x.
- Plazanet C. *Enrichissement des Bases de Données Géographiques : Analyse de la Géométrie des Objets Linéaires pour la Généralisation Cartographique (Application aux Routes)*. Thèse de Doctorat, Université de Marne-la-Vallée, 1996.
- Proulx M.-J., Larrivée S. et Bédard Y., "Représentation multiple et généralisation avec UML et l'outil Perceptory". Ruas A. (Ed.), *Généralisation et Représentation Multiple*, Paris, Hermes science publications, 2002 p. 113-119.
- *Rainsford D. et Mackaness W.A., "Template Matching in Support of Generalisation of Rural Buildings". Richardson D. et Van Oosterom P. (Eds.): *Advances in Spatial Data Handling- 10th International Symposium on Spatial Data Handling*. Springer, Berlin, 2002, p. 137-152.
- *Ratajski L., "Phénomènes des points de généralisation". *International Yearbook of Cartography*, Vol. 7, p. 143-151, 1967.
- *Regnauld N., "Généralisation des bâtiments". Ruas A. (Ed.): *Généralisation et Représentation Multiple*. Paris, Hermes science publications, 2002, p. 257-272.
- *Regnauld N., *Généralisation du Bâti: Structure Spatiale de Type Graphe et Représentation Cartographique*. Thèse de Doctorat, Université de Provence - Aix-Marseille, 1998.

- *Reumann K. et Witkam A.P.M., "Optimizing curve segmentation in computer graphics". *Proceeding of International Computing Symposium*, Amsterdam, 1974, p. 467-472
- Rieger M. et Coulson M., "Consensus or confusion: cartographer's knowledge of generalization". *Cartographica*, 30 (2-3), 1993, p. 69-80.
- *Rigaux P., "La représentation multiple dans les systèmes d'information géographique". *Revue Internationale de Géomatique*, Vol. 4(2), pp.137-164, 1994.
- *Rivest S., Bédard Y. et Marchand P., "Toward better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing (SOLAP)". *Geomatica*, Vol. 55(4), p. 539-555, 2001.
- Ruas A. et Lagrange J.P., "Data & knowledge modelling for generalisation". Müller J.C., Lagrange JP. et Weibel R. (Eds.), *GIS and Generalization*, London, Taylor & Francis, 1995, p. 73-90.
- *Ruas A. *Modèles de Généralisation de Données Géographiques à Base de Contraintes et d'Autonomie*. Thèse de Doctorat, Sciences de l'Information Géographique, Université de Marne-la-Vallée, 1999, 323 p.
- *Ruas A., "A method for building displacement in automated map generalisation". *International Journal of Geographical Information Science*, Vol. 12(8), p. 789-803, 1998.
- *Ruas A., "Les problématiques de l'automatisation de la généralisation". Ruas, A. (Ed.), *Généralisation et Représentation Multiple*, Hermes science publications, 2002, p. 75-90.
- Rusak E., Mazur E. et Castern H.W., "Horston's ordering scheme and the generalisation of river networks". *The Cartographic Journal*, Vol. 27(2), 1990, p. 104 -112.
- *Sabo M.N., Bédard Y. et Bernier E., "Methodology for developing a database of geometric patterns to better support on-the-fly map generalization." In *Proceedings of International Cartographic Conference*, 9-16 July, Coruna, Spain, 2005.
- *Sabo M.N., Bédard Y., Moulin B. et Bernier E., "Integrating geometric patterns with generalization algorithms: Towards Self-Generalizing Objects and On-the-fly map generalization", Submitted to *Cartographica*, 2007.
- *Sabo M.N., Cardenas A., Bédard Y. et Bernier E., "Introduction du concept de patrons géométriques et application aux bâtiments afin de faciliter leur généralisation cartographique à la volée." *Geomatica*, Vol. 59(3), p. 295-311, 2005.
- *Safe Software. Disponible sur <http://www.safe.com/>.
- *Sarjakoski T. et Kilpelainen T., "Holistic Cartographic Generalization by Least Squares Adjustment for Large Data Sets." In *Proceedings of 19th International Cartographic Conference*, Ottawa, Canada, 1999, p. 1091-1098.
- *Schillo M., Fischer K. et Klein C., "The micro-macro link in DAI and sociology", In *Proceedings of the Multiagent-Based Simulation: Second International Workshop on Multiagent-Based Simulation*, Springer-Verlag: Boston MA, USA, 2001, p. 133-148.

- *Scholl M., Voisard A., Peloux J.P., Raynal L. et Rigaux P., *SGBD Géographiques Spécificité*. Paris, International Thomson Publishing, France, 1996, 185 p.
- *Sester M. et Brenne C., "Continuous Generalization for Fast and Smooth Visualization on Small Displays". Fisher P. (Ed.): *Developments in Spatial Data Handling- 11th International Symposium on Spatial Data Handling*. Springer Verlag, 2004, p. 469-480.
- *Sester M., "Generalization Based on Least Squares Adjustment." *International Archives of Photogrammetry and Remote Sensing*, Vol. 33 (B4/3), p. 931-938, 2000.
- *Sester M., "Optimization approaches for generalization and data abstraction", *International Journal of Geographical Information Science*, Vol. 19 (8-9), p. 871-897, 2005.
- *Shea K.S. et McMaster R., "Cartographic generalization in a digital environment: when and how to generalize". *Proceeding of 9th International Symposium on Computer-Assisted Cartography, AUTO-CARTO 9*, Baltimore, Maryland, 1989, p. 56-67
- *Shea K.S., "Design consideration for an artificially intelligent system". Battenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1991, p. 3-20.
- *Spiess E., "The need of generalisation in a GIS environment". Müller J. C., Lagrange J. P. et Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*. Bristol: Taylor & Francis, 1995, p. ix.
- *Stefanovic N., Han J. et Koperski K., "Object-based selective materialization for efficient implementation of spatial data cubes", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12(6), p. 938-958, 2000.
- Thapa K., "Data compression and point's detection using normalized symmetric scattered matrix". In *Proceedings of the AUTO-CARTO*, 1989.
- *Thomson R.C. et Richardson D.E., "The 'Good Continuation' principle of perceptual organization applied to the generalization of road networks". In *Proceedings of the 19th International Cartographic Conference*, Ottawa, 1999, p. 1215-1223.
- *Timpf S. 1998. *Hierarchical Structures in Map Series*. Thèse de Doctorat, Departement of Geoinformation, Technical University of Vienna, Vienna, 1998, 124 p.
- *Timpf S. et A. U. Frank, "A multi-scale data structure for cartographic objects". *Proceeding of 17th International Cartographic Conference*, Barcelone, Spain, 1995.
- *Topfer F. et Pillewizer W., "The principles of selection", *The Cartographic Journal*, Vol. 3(1), p. 10-16, 1966.
- *Van Elzakker C. "Use of Maps on the Web." Kraak M.J. et Brown A. (Eds.), *Web Cartography: Developments and Prospects*, London: Taylor & Francis, 2001, p. 21-36.

- *Van Elzakker C.P.J.M. et Koussoulakou A., "Maps and their use on the Internet". *Proceeding of 18th International Cartographic Conference, ICC'97*, Stockholm, 1997, p. 620-627.
- *Van Oosterom P. et Schenkelaars V. "The development of an interactive multi-scale GIS". *International Journal of Geographical Information Systems*, Vol. 9(5), p. 489-507, 1995.
- *Van Oosterom P., "A storage structure for a multi-scale database: the reactive-tree". *Computer, Environment and Urban Systems*, Vol. 16, p. 239-247, 1992.
- Van Oosterom P., "The GAP-tree, an approach to 'on-the-fly' map generalisation of an area partitioning". Müller J.C., Lagrange J.P. et Weibel R. (Eds.), *GIS and Generalization: Methodology and Practice*, Bristol: Taylor & Francis, 1995, p. 120-132.
- *Vangenot C., "Représentation multi-résolution, concepts pour la description de bases de données avec multi-représentation". *Revue Internationale de Géomatique*, 8(1-2), 1998, p. 121-147.
- *VisValingam M. et Whyatt J.D., "Line generalisation by repeated elimination of points". *Cartographic Journal*, Vol. 30(1), p. 46-51, 1993.
- *Wang Z. et Müller J.C., "Line generalization based on analysis of shape characteristics". *Cartography and GIS*, Vol. 25(1), p. 3-15, 1998.
- Ware J.M., Wilson I.D. et Ware J.A., "A knowledge-based genetic algorithm approach to automating cartographic generalisation". *Knowledge-Based System*, Vol. 16, 2003, p. 295-303.
- *Weibel R. "Amplified intelligence and rule-based systems." Buttenfield B., McMaster R. et Freeman H. (Eds.), *Map Generalization: Making Rules for Knowledge Representation*, Harlow, Essex, England Longman Scientific & Technical, New York NY: Wiley, 1999, p. 172-186.
- Weibel R. et Buttenfield B., "Map design for geographic information systems." In *Proceedings of GIS/LIS*, 30 November 2 December, San Antonio, Texas, 1988, p. 350-359.
- *Weibel R. et Dutton G., "Constraints-based Automated Map Generalization." In *Proceedings of the 8th International Symposium on Spatial Data Handling*, Vancouver, 1998, p. 214-224.
- *Weibel R. et Dutton G., "Generalising spatial data and dealing with multiple representations". Longley P.A., Goodchild M.F., Maguire D.J. et Rhind D.W. (Eds.), *Geographic Information Systems- Principles and Technical Issues*. John Wiley & Sons, 1999, p. 125-155.
- Weibel R. et Jones C.B., "Computation Perspectives on Map generalization". *Geoinformatica*, Vol. 2(4), 1998, p. 307-314.
- *Weibel R., Bernier E, Bédard Y. et Cecconi A., "La généralisation à la volée". Ruas A. (Ed.): *Généralisation et Représentation Multiple*. Paris, Hermes science publications, 2002, p. 319-335.

- *Weibel R., Keller S. et Reichenbacher T., "Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive system and computational intelligence". *Proceeding of COSIT '95: Spatial Information Theory - A Theoretical Basis of GIS, Lecture Notes in Computer Science 988*, Berlin, Springer-Verlag, 1995, p. 139-156
- *Weiss S.M. et Kulikowski C.A., *A Practical Guide to Designing Expert Systems*. Rowman & Littlefield, Totowa, N.J. USA, 1984, 174 p.
- *Wilson I.D., Ware M. et Ware A., "Reducing graphic conflict in scale reduced
- *Wonnacott L. 2000. The speed of business: If your pages are slow, your customers will go. Disponible sur <http://www.infoworld.com/articles/op/xml/00/09/11/000911opsavvy.html>.
- *Young J., Smith S., Akamai and JupiterResearch Identify '4 Seconds' as the New Threshold of Acceptability for Retail Web Page Response Times. Disponible sur http://www.akamai.com/html/about/press/releases/2006/press_110606.html
- *Zacks J. et Tversky B., "Event Structure in Perception and Conception". *Psychological Bulletin*, Vol. 127(1), p. 3-21, 2001.
- *Zona Research Inc. 1999. The Economic Impacts of Unacceptable Web Site Download Speeds. Disponible sur http://www.webperf.net/info/wp_downloadspeed.pdf

ANNEXES

ANNEXE 1: METHODOLOGY FOR DEVELOPING A DATABASE OF GEOMETRIC PATTERNS TO BETTER SUPPORT ON-THE-FLY MAP GENERALIZATION

Mamane Nouri SABO, Yvan BÉDARD, Eveline BERNIER

Department of Geomatics Sciences and Centre for Research in Geomatics
Laval University, Quebec City, Canada G1K 7P4

mamane-nouri.sabo.1@ulaval.ca yvan.bedard@scg.ulaval.ca eveline.bernier@scg.ulaval.ca

Abstract

Cartographic generalization has been a research topic for more than thirty years. In spite of recent advances, a complete automation of this process still faces fundamental problems. Furthermore, recent needs such as web mapping and Spatial On-Line Analytical Processing (SOLAP) require geospatial data that are generated on-the-fly at any scale. This requires processing times and a degree of automation that today's automatic generalization algorithms cannot reach with all data types.

This paper presents the concept of geometric patterns and explains how they can be built and stored in a pattern database for further use in a map generalization process. Geometric patterns are typical shapes that are representative of several occurrences of a cartographic object class. They support simple operations such as rotation, translation or stretch and allow adapting them to the exact geometry of individual objects.

1. INTRODUCTION

The analysis of several geographic phenomena often requires data at various levels of abstraction. Hence, there is a need for the modeling of geographic information at different levels of abstraction [Müller *et al.* 1995]. With the arrival of new geomatics applications such as Webmapping and SOLAP [Rivest *et al.* 2001], the needs for multiscale data has become more pressing. The variety and the increasing number of users of these applications require the systems to be able to generate geospatial data according to the specific user's needs and upon request. But, it is not possible and imaginable to know a priori the needs of the users of these applications. A solution would be to have a large-scale geospatial database from which other smaller scales could be automatically generated. For that purpose, cartographic generalization is necessary. Besides, such applications are characterized by their dynamic and interactive nature and « this dynamic and interactive nature requires almost short response time. Consequently, the scale changes must be made in real time so requiring on-the-fly map generalization » [Weibel *et al.* 2002].

Cartographic generalization has been an important subject of research for more than thirty years. In spite of constant efforts, the generalization remains a semi-automatic process (even manual in certain cases) that leads to slow response times and production costs incompatible to the requirements of these new applications. Solutions based on

multiple representations (RM) thus became an alternative to the on-the-fly map generalization. Typically, in a multiple representation database, the same territory can be associated to several geometric representations, each of them representing the territory at different scales. Explicit links, either between the various scales of the database or between the different representations of the same object can be sometimes create. To create the different scales necessary to build the database, semi-automatic or manual generalization processes are used. Moreover, the redundancy and the possible problems of incoherence between various representations, « the creation of multi-scales data base entails three types of problems: correspondence between various abstractions, correspondence between various instances of an object and the definition of matching process between objects » [Devogele *et al.* 1996].

To minimize the above multiple representation and automatic generalization problems, several approaches aiming at combining cartographic generalization and multiple representation appeared these last years [Cecconi *et al.* 2002; GiMoDig 2004]. [Cecconi *et al.* 2002] recommends to use a database containing a minimum of two scales. After a user request, the most appropriate scale is selected and refined using an on-the-fly generalization process if it is necessary. The [GiMoDig 2004] project proposes a generalization approach similar to the one proposed by [Cecconi *et al.* 2002] and includes an on-the-fly integration process. A philosophy that is common to these two approaches is that the less is the difference between the initial scale and the scale of the map which will be generated, the easier will be the generalization. Unfortunately, to benefit from the advantages of this approach, the difference between the scale of the initial map and the generated map must be small, which is not always the case.

Like the previous projects, the GEMURE project (MUltiple GEneralization and Representations for one-demand map production and delivery) also tackles the problem of on-the-fly map generalization. Within the framework of this project, we've proposed an approach based on the use of geometric patterns. Geometric patterns are objects with a representative shape common to several cartographic objects and that support simple operations such as orientation, displacement or stretch. These operations allow a single pattern to adapt itself to the geometry of several specific objects. So, instead of using algorithms to generalize a cartographic object, we choose a predefined geometric pattern, already stored in a pattern library database, make some adjustments if needed and replace the object by this geometric pattern for smaller scales.

This paper aims at presenting the concept of geometric patterns and explaining how they can be built and stored in a pattern database for further use in the map generalization process. First, a formal description of geometric patterns is formulated. Then, the iterative processes of creating geometric patterns are presented. Throughout the paper, we present various results obtained from concrete experimentations on buildings from the Quebec City area, a city with a mixed architectural style: European and North American. Finally, the mechanism related to the selection of a geometric pattern and its association to a specific cartographic object is introduced.

2. GEOMETRIC PATTERNS

2.1 Some geometric patterns concepts

In the daily life, it is possible to notice a recurrence of several phenomena from a time and/or space point of view. In all application domains, the idea of patterns is closely dependent to this recurrence and similarity. In spite of an abundance of recurring phenomena in our environment, it is only in the 70s that the pattern concept was credited [by the architect Alexander *et al.* 1977]. His widely-cited definition of pattern is:

« A pattern is a solution to a problem in a context where each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice ».

In the 90s, the idea of the patterns was adopted in the computer science field under the name of “concept of design pattern”, popularized by the Gang of four. In their book, Design Patterns: Elements of Reusable Object-Oriented Software, [Gamma *et al.* 1995] propose 23 design patterns, each of them allowing to resolve a recurring problem of object programming. This reuse of code allows increasing programmer’s productivity.

In several domains, such as architecture and computer science, the main interest of the use of patterns is to simplify complex processes. This capacity of the pattern to simplify some processes and the abundance of recurring phenomena and structures in our geographic environment stimulated the interest of some researchers to introduce this pattern concept in geomatics and particularly in map generalization processes. Indeed, when viewing the map, the eye discerns patterns of shape, orientation, connectedness, density and distribution [Mackaness and Edwards 2002].

Besides, in spite of the potential associated to the use of patterns in geomatics, few works are today dedicated to this approach. Nevertheless, we know for many years that the Danish Mapping Agency uses a series of templates to represent farm building in 1: 50 000 scale [Regnauld 2002]. To automate the choice of these templates, [Rainsford and Mackaness 2002] proposes the use of matching algorithms in order to select templates that best characterizes a rural building. The concept of pattern as proposed in this paper innovates in several ways and was first described formally by Cardenas [2004]. The work of the first author to test this concept against buildings is presented for the first time. This has resulted in enrichments to the initial concept, and particularly into the presented methodology.

2.2 Geometric patterns and map generalisation

The existence of recurring geographic phenomena and the hierarchical organization of patterns allow them to be used in a map generalization process. Indeed, certain works evoke a hierarchical organization of patterns; it is the case of [Zacks and Tversky 2001]. For a cartographic object, this hierarchical structure especially appears when we browse through various levels of abstraction (e.g. by changing the scale of visualization). While passing from a large scale to a smaller scale, for the same object or a cluster of objects, we observe an evolution in the shape of several patterns. For example, a single building can have several forms going from a detailed shape to a simple rectangular shape. From a pattern perspective, we can define generalization as a set of transformations (from one pattern to

another) in which certain partonomic qualities are preserved and others are disregarded: as a result of this process we see different patterns according to changes in scale or theme [Mackness and Edwards 2002].

The main idea of the use of patterns in cartographic generalization is to have for each map object, several geometric patterns. Each geometric pattern corresponds to the object shape in a given level of abstraction. When the level of abstraction is modified, an object is replaced by a geometric pattern in order to generalize it, instead of using traditional generalization methods. Given the generic character of the pattern's geometry, a same pattern can be used several times and in various scales to represent various occurrences, every time with appropriate parameters for each occurrence. For example, in 1K scale, for a portion of the test area composed by 365 buildings, 30 % of buildings were associated to a single geometric pattern with a "L" shape form. .

Cardenas [2004] defines the geometric pattern as being a geometric object with typical geometric characteristics representative of a great number of occurrences of a cartographic object or a primitive and that is able to adapt itself to the geometry of these occurrences for various scales. The geometric pattern is composed of primitives. The primitive is the atomic, indivisible element that forms a geometric pattern. The shape of a geometric pattern depends largely on the shape of the various primitives which compose it, the number of primitives and the way these primitives are juxtaposed. There are two types of geometric patterns: simple geometric pattern composed of one or several instances of the same primitive; complex geometric pattern composed of other geometric patterns (simple or/and complex). This definition of geometric patterns is the base of the methodology which will be presented in this paper.

Even if it is possible to replace all the cartographic objects having a similar shape with the same geometric pattern, certain parameters (e.g. the position or the size of every object which will be replaced by the geometric pattern) are necessary in order to adapt the geometric pattern to each specific object. For example, two buildings having the same size and the same shape may be associated to the same geometric pattern but implemented using different positions and different orientations. For these reasons, the following manipulations are necessary to be able to replace a specific cartographic object by a geometric pattern:

- Displacement, to position the geometric pattern;
- Orientation, to orientate the geometric pattern;
- Stretch, to fit the geometric pattern size to the object size.

The simplicity of these operations makes the generalization process based on geometric patterns very powerful. For example, the delay to realize certain generalization operations such as an aggregation and a simplification is about 10 times superior to the time needed for adapting a geometric pattern. The experiment was realized on 2844 buildings situated in Quebec City and 0.4 seconds were enough to adjust all geometric patterns, compared to 3.9 seconds for traditional generalization operations. For this reason, it is possible to use this approach in on-the-fly map generalization processes.

3. METHODOLOGY FOR DEVELOPING THE GEOMETRIC PATTERNS DATABASE

The geometric patterns database creation can be divided in five major steps: the detection of the different patterns on a map, the choice of the primitives that will be used to

construct the geometric patterns, the choice of the most detailed common patterns, the detailed geometric pattern construction, and the construction of derived geometric patterns for each detailed geometric pattern. In this paper, the term pattern will refer to the pattern of shape that can be detected in our environment, or on a map. The term geometric pattern will refer to the constructed geometric pattern that will be stored in the database and used during a generalization process.

3.1 Patterns detection

The aim of this phase is to make a brief inventory of the various patterns associated to a specific object class. This inventory also serves to determine if it is relevant to use geometric patterns for this specific object class. In theory, any object classes can aspire to use geometric patterns. But, the basic concept of the geometric pattern is that the occurrences must present geometric similarities in order to be able to create certain forms that can be easily used by a large number of occurrences. Therefore, it is important right from the start to determine the object classes for which it is relevant to use geometric patterns. This discrimination avoids the creation of a plethoric number of geometric patterns as we create only the essential geometric patterns. A large number of geometric patterns makes the geometric patterns database creation much more laborious and reduce the performance of the system.

The detection of the various patterns is realized by an expert, from the initial map of the selected area. The initial map is the source map from which the smallest scales will be generated using geometric patterns. Using the initial map, the various potential patterns are detected, by visually analyzing the various elements of the object class for which we want to create geometric patterns. A particular attention is accorded to the frequency of appearance of various shapes on the map. The inventory of the different repetitive forms and the estimation of their frequency allow discovering a posteriori the necessary geometric patterns. The geometric patterns that are not representative are eliminated. But the existence of many representative geometric patterns permits to consider this object class as a good candidate for the use of geometric patterns.

There is no magic number to quantify the relevance of the use of geometric patterns for an object class. But it is clear that the best object class for the use of geometric pattern is the one allowing to replace a majority of its elements by a small number of geometric patterns and to cover a large scale range. That is why it is important to make statistics showing the number of times that each pattern appears. After having analyzed these statistical results, we can thus retain only the better patterns, those which present a high frequency of appearance.

Besides, before even this analysis phase, certain object classes can be excluded from the list of the best candidate classes for the use of geometric patterns. Indeed, analyses from the GEMURE project test area showed that the most suited object classes to the use of the geometric patterns are buildings, highway cloverleaves and cul-de-sacs. For example, more than 75 % of the test area buildings were replaced by geometric patterns in 1K scale. But this does not mean that the patterns are not applicable to other object classes.

To facilitate the pattern detection it is sometimes helpful to use maps at various levels of details. Certain objects which seem very different on a large-scale map can present the same pattern on a smaller scale. For example, the majority of buildings are represented by simple rectangles on a small scale map, though they are represented by detailed polygons

on large-scale maps. The use of several levels of abstraction allows detecting patterns appearing only on smaller scales. In the absence of different scales of the same territory, it is more difficult to imagine the various forms than an object can have over several scales. In that case, good cartographic generalization knowledge is necessary.

For certain object classes, it is very difficult to have geometric patterns when we see the complete object geometry. For example, it is very difficult to detect a pattern for entire rivers or entire roads. On the other hand certain parts of these objects (eg. road's cloverleaf) can present good patterns. For these objects, the patterns search must be made not only at the level of whole objects, but also at the level of object geometric primitive (the objects sections). For that, segmentation is sometimes necessary.

Moreover of determining if it is relevant to use the geometric patterns for a given object class, this phase also allows to have a good idea of geometric patterns necessary to represent the selected object class. The analysis of these various patterns will drive us in the geometric patterns construction phase.

3.2 Primitives choice

The basic element for the construction of a geometric pattern is a primitive. That is why, during the previous step (cf. 3.1.), the detected patterns are meticulously analyzed to detect the different primitives that compose them. For that purpose, every pattern must be considered as being a juxtaposition of several primary shapes as rectangles, circles or triangles. For example, the pattern of a building can be created from several instances of a rectangle connected together (figure 1). This object design as an arrangement of different components is now shared by several researchers in the field of shape recognition [Fortin and Rousseau 1992]. According to [Biederman 1987], 36 primary shapes called geon, are enough for representing objects of our environment.

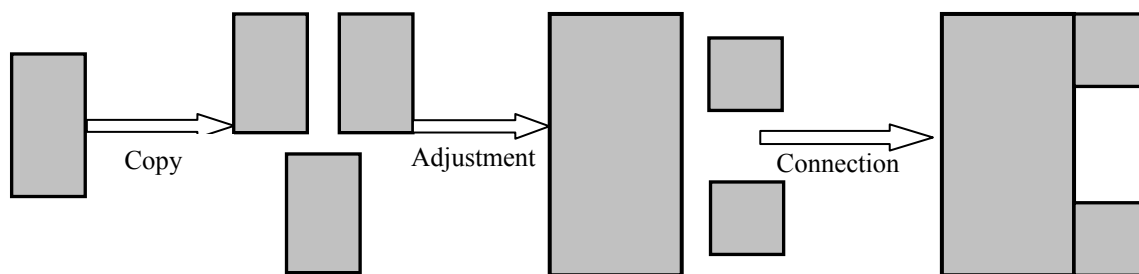


Figure 1 Buildings pattern composed by three instances of a rectangular primitive

The choice of the primitives shape is essential, because the shape of the geometric patterns which will be created and stored in the future database is strongly dependent on this choice. During the choice of the primitive, we must find, the balance between, on one side to have a restricted number of primitive, thus of geometric patterns and, on the other side, to adequately represent objects. Quantify the exact maximum number of primitives necessary to represent a given territory is not possible in our case. But, the 36 geons advanced by [Biederman 1987] seems to be plethoric in our case because geometric patterns are used in a context of cartographic generalization, which implies that they must be simpler than the objects they are meant to replace.

The number of primitives which will be used to create geometric patterns must be reduced to a minimum; otherwise it will lead to the creation of a phenomenal number of geometric patterns. For example, according to [Biederman 1987], by using 36 geons, we can create up to 157 million of objects. On the other hand, the representation potential of even a limited number of primitives is rather impressive. In fact, the various possible combinations between different primitive instances and operations applicable to each primitive instance (eg. change of orientation) allow to generate a large number of patterns. In our case, with six instances of a rectangular primitive, we can create 64 geometric patterns.

The choice of the primitives is closely related to the operations that can be applied on them. These pattern operations (applied on the entire pattern or on its primitives) must not allow to go from one primitive to another. If this case arises, a choice must be made to decide if we keep the primitive or the operation. For example if we have a rectangle and a square as primitive, it is impossible to keep the operation which allows changing the extension of the primitive. Using this operation, we can transform the square in a rectangle and inversely. Thus, the solution will be to take a more generic shape as primitive, for example an orthogonal parallelogram.

3.3 Choice of the most detailed common pattern

For each primitive retained during the previous phase (cf. 3.2), the most detailed common pattern must be defined from the detected pattern in the patterns detection phase (cf. 3.1). The most detailed common pattern is the pattern with the larger number of instances of primitive and from which we can build all derived patterns (formed by the same primitive) by deleting primitive instances. The most detailed common pattern can be seen as a union of the shapes of all patterns of its group. Figure 2 shows several patterns which were formed by the same primitive and the surrounded pattern is the most detailed common pattern of this group. The choice of the most detailed common pattern is a critical phase, because the size and the granularity of the database as well as the quality of generated data depend on this choice. It is about a balance between replacing the maximum number of map objects by fewer geometric patterns and covering a larger scale range with an acceptable degree of geometric accuracy. Indeed, a very detailed geometric pattern (with too many instances of primitive) would imply the creation of a larger number of derived geometric patterns resulting in a substantial increase of the patterns database size.

Some errors may arise during the patterns detection phase. To remedy to this kind of errors, the number of the instances of primitive of a pattern can be increased. For example, if our most detailed common pattern has 4 instances of the same primitive, we can choose a 6 instances pattern to have more flexibility. Raising the number of primitive's instances of the most detailed common pattern allows limiting the number of iterations during the geometric pattern's database creation. On the other hand, a substantial rise of the number of primitive instances can considerably increase the size of the database.

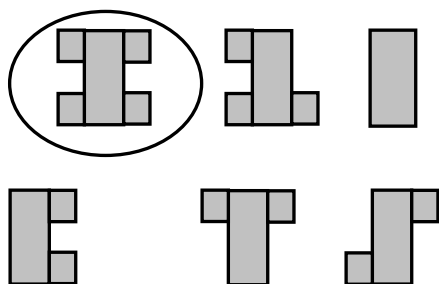


Figure 2 Group of patterns and its most detailed common pattern s creation

attern chosen during the previous phase (cf. 3.3.) is reproduced, using the patterns creation tools. The reproduced pattern is called the initial geometric pattern. Within the framework of the project GEMURE, tools which allow constructing geometric patterns were already created. These tools allow building geometric patterns from several instances of primitive. They also allow to perform several operations on the primitive such as changing the size, the anchoring points, the orientation of the primitive, etc. Geometric patterns are built by connecting several primitive instances. Figure 3.a shows a simple geometric pattern built from three instances of one rectangular primitive. But, an analysis of every most detailed common pattern is necessary to determine other parameters as the number of anchoring points, main and secondary primitives and the way these primitives are connected between them. For example, for the creation of the building geometric patterns, we can determine the number of secondary primitives connected on each main primitive side. Main primitive is the primitive on which secondary primitives are connected. . The creation of a geometric pattern begins with the creation of the main primitive. Afterward, various secondary primitives are connected. During the creation of the initial geometric pattern, an exact reproduction of the most detailed common pattern is not necessary because, the geometric pattern must be generic and manipulations are necessary to adapt it to a concrete map object.

Usually, for an object class that has been identified as relevant to the use of geometric patterns, the majority of its occurrences will be replaced by simple geometric patterns. However, these patterns may not be suited for more complex geometric objects that may be present in the object class. Complex geometric patterns must thus be used. The complex geometric patterns may be created in two ways: 1) by combining some primitives (instances of some primitives), 2) by combining already created geometric patterns. These geometric patterns can be simple and/or complex. Thus, the tools used for the simple geometric patterns creation are necessary. The figure 2.b shows a complex geometric pattern formed by an instance of a circular primitive and four instances of a square primitive.

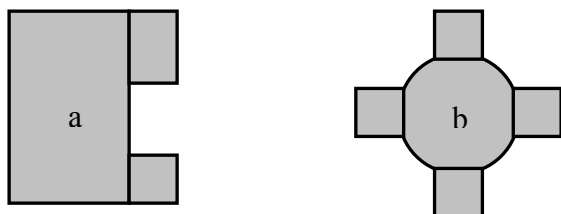


Figure 3 Two geometric patterns: a) simple geometric pattern; b) complexe geometric pattern

3.5 Derived geometric patterns creation and the database creation

Besides operations such as rotation or displacement, which manipulate the entire pattern, they are also operations applicable to a primitive only. These operations allow to simplify the shape of a pattern and to fit it to the geometry of the object to be represented. The most important of these operations are:

- Elimination of a primitive;
- Combination of primitives;
- Change the primitive's size;
- Change the primitive's anchoring points;
- Change the primitive's orientation.

These operations are similar to that used during the generalization of 3D buildings described in [Thiemann 1999].

A derived geometric pattern is obtained by simplifying an initial geometric pattern. In theory, all these operations can be applied repeatedly to each initial geometric pattern, and then the generated derived geometric patterns can be stored in a database. But considering the important number of derived geometric patterns that can be generated with this approach (when applying all primitive operations) and the fact that some operations may be used on-the-fly, it is necessary to limit the number of operations that may be applied on initial geometric patterns when filling the database. Accordingly, only "elimination of a primitive" will be applied. For that purpose, the elimination of primitive is applied to the initial geometric pattern in an iterative way until we obtain the primitive that was used to build the pattern. For example, if we have an initial geometric pattern formed by instances of a rectangular primitive, the operation elimination of a primitive is applied repeatedly until we obtain rectangle. Only one instance of primitive is removed by iteration, and all possible configurations are obtained by permuting the remaining instances. When all derived geometric patterns of the same pattern are generated, the redundant patterns are deleted. Geometric patterns are also considered redundant when they can be obtained by rotating or by scaling an existing geometric pattern.

Each initial geometric pattern and its various derived geometric patterns are stored in a hierarchical tree (figure 4.) whose depth determines the number of primitive instances and consequently the level of detail of the pattern. The hierarchical structures are often used to store geographical data in order to minimize the redundancy and to facilitate the access ([Van Oosterom and Schenkelaars 1995]). Each geometric pattern (initial or derived) is linked to its children. These links forms the tree structure of geometric patterns. This tree structure facilitates the navigation through different levels of abstraction of the same pattern. So, for every map object, we can associate a geometric pattern (if this object can possess one). Also, we can indicate the path in the tree which allows accessing other derived geometric patterns. The choice of the associated geometric pattern and the tree's path depends on the shape of the object. Thus the geometric pattern tree (initial and derived geometric patterns) is a multiscale pattern which can represent different objects in different level of abstraction. So, the generalization using geometric patterns become a simple navigation through different hierarchical levels of the geometric patterns tree.

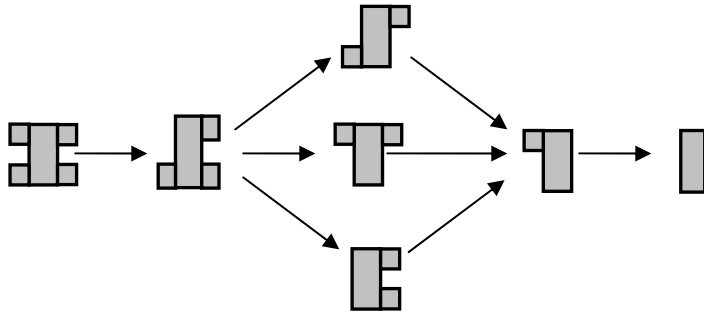


Figure 4 Tree structure of buildings geometric patterns

This research project is continuing towards efficient ways of coding in a unique data string the path or details of each pattern so it even becomes faster. We are also considering the inclusion of additional operations easy to implement in such a string (ex. collapse a primitive by half).

4. CONCLUSION

In this paper, we presented a five-step methodology to create a geometric patterns database. Geometric patterns are generic objects with geometrical basic characteristics used to replace map elements. This methodology allows creating at first one detailed geometric pattern (initial geometric pattern) for each pattern's primitive type. Other derived geometric patterns are generated by iterative simplification of the initial geometric pattern. Results are stored in a geometric patterns database using a tree structure. This methodology simplifies the process of building the geometric patterns database as, for each primitive type, only one geometric pattern is created.

The resulting database can be used in an on-the-fly map generalization process. So, for every map object (if this object can have a corresponding geometric pattern), we can create a link between this object and a geometric pattern. The underlying hierarchical structure allows the access to other derived geometric patterns from this cartographic object. All geometric patterns linked to a cartographic object constitute different shapes of this object, each shape corresponding to a different level of abstraction. Thus, instead of applying complex generalization algorithms when changing the level of abstraction, a corresponding geometric pattern is selected, adjusted and used to replace the object.

The geometric patterns stored in the database can also be used during a multi-scale database acquisition. So, instead of capturing different geometries of the same object, we can only capture the most detailed geometry and associate geometric patterns to provide the object with simplified geometries to be used at different levels of detail. Some researches towards this approach have already been initiated at the Centre for Research in Geomatics at Laval University.

Besides accelerating generalization processes, geometric patterns can also facilitate the data transit through the network. Indeed, in a client/server architecture, all the geometric patterns needed for the generalization of the user's map can be transferred only once to the client side. So, during the generalization processes, instead of transferring the results of the generalization (the generalized object obtained using traditional generalization algorithms or geometric patterns), we transfer only the parameters that will be used to adjust the

geometric pattern. This would considerably decrease the volume of data transfer through the network.

One of the points that appear as a disadvantage of this approach is the amount of work required to create the database and to associate geometric patterns to the different map objects. However, this work is made only once and the results are useful several times and even for various territories. Also, certain stages of this methodology such as the pattern's detection can be automated. Future works will be centered on the automation of certain steps of this methodology and on the integration of traditional generalization methods to support the generalization of the objects that are not suited to the geometric patterns approach.

5. ACKNOWLEDGMENTS

The authors are grateful to the Canadian Network of excellence GEOIDE which made possible these works within the framework of the GEMURE project, as well as the following partners: Research and Development Defense Canada, Natural Resources Canada, Natural Resources, Fauna and Park Quebec, Intergraph.

6. REFERENCES

Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I. and Angel S., (1977). A Pattern Language: Towns, Buildings, Construction. New York, Oxford University Press, 1171 p.

Biederman I., (1987). Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2), pp. 115-147.

Cardenas A., (2004). Utilisation Des Patrons Géométriques Comme Support À La Généralisation Automatique. Faculté de foresterie et géomatique, Université Laval, Québec, 110 p.

Cecconi A., Weibel R. and Barrault M., (2002). Improving automated generalisation for on-demand web mapping by multiscale database. *Proceeding of joint International Symposium on Geospatial Theory, Processing and Applications*, July 8-11, Ottawa, Canada, pp. 1-9.

Devogele T., Trevisan J. and Raynal L., (1996). Processus De Constitution D'une Base De Données Multi-Échelles. *Revue Internationale de Géomatique*, 6(2-3), pp. 249-263.

Fortin C. and Rousseau R., (1992). Psychologie Cognitive : Une Approche de Traitement de L'information. Sillery, Presses de l'université du Québec, 434 p.

Gamma E., Helm R., Johnson R. and Vlissides J., (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Reading, Mass.; Don Mills, Ont., Addison-Wesley, 395 p.

GiMoDig, (2004). Geospatial Info-Mobility Service by Real-Time Data-Integration and Generalisation. <http://gimodig.fgi.fi/>, Accessed 17-12-2004.

Mackaness W.A. and Edwards G., (2002). The Importance of Modelling Pattern and Structure in Automated Map Generalisation. *Join ISPRS/ICA workshop on multi-scale representations of spatial data*, July 7-8, Ottawa, Canada.

Müller J. C., Weibel R., Lagrange J.P. and Salge F., (1995). Generalization: State of the Art and Issues. In Müller J.C., Lagrange J.P. and Weibel R. (eds.): Gis and Generalization: Methodology and Practice, Bristol, Taylor & Francis, pp. 3-18.

Rainsford D. and Mackaness W.A., (2002). Template Matching in Support of Generalisation of Rural Buildings. In Richardson D. and P.V. Oosterom (Eds.): *Advances in Spatial Data Handling 10th International Symposium on Spatial Data Handling*, Springer, Berlin, pp. 137-152.

Regnauld N., (2002). Généralisation des bâtiments. In Ruas A. (Ed.) : Généralisation et Représentation Multiple, Paris, Hermes science publications, pp. 257-272.

Rivest S., Bédard Y. and Marchand P., (2001). Towards better support for spatial decision-making: defining the characteristics of Spatial On-Line Analytical Processing, *Geomatica*, 55(4), pp. 539-555

Thiemann F., (2002). Generalization of 3D Building Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(4), pp. 286-290.

Van Oosterom P. and Schenkelaars V., (1995). The Development of an Interactive Multi-Scale Gis. *International Journal of Geographical Information Systems*, 9(5), pp. 489-507.

Weibel R., Bernier E., Bédard Y. and Cecconi A., (2002). La Généralisation À La Volée. In Ruas A. (Ed.) : Généralisation et Représentation Multiple, Paris, Hermes science publications, pp. 319-335.

Zacks J., Tversky B., (2001). Event Structure in Perception and Conception. *Psychological Bulletin*, 127(1), pp. 3-21.

ANNEXE 2 : Principales distinctions entre notre approche et certaines des approches multi-agents de généralisation

	Notre approche	Jabeur (2006)	AGENT
But	Généralisation à la volée et généralisation automatique traditionnelle	Généralisation à la volée des cartes récréo-touristiques	Généralisation automatique traditionnelle
Les regroupements d'objets sont basés sur	Les contraintes d'intégrité spatiales	Les agrégats contenus dans la base de données à représentation multiple	La structure des villes (délimitée par le réseau routier)
Agents pour représenter des objets individuels de la carte	Plusieurs types d'agents SGO Simples selon les contraintes d'intégrité (ex. alignement) et le type d'objets (ex. bâtiment)	Un seul type (Instance agent)	Un seul type (agent Micro)
Agents pour représenter des groupes d'objets	Plusieurs types d'agents SGO Complexes selon les contraintes d'intégrité et le type d'objets	Un seul type (Container agent)	Un seul type (Agent Méso)
Gestion des principales contraintes de généralisation	Pour chaque contrainte importante, un SGO complexe est créé	?	Toutes les contraintes sont gérées par l'agent Méso (ex. agent îlot)
Le champ de perception des agents	Définie par les liens de proximité	Définie par un buffer créé autour de l'objet	Définie par un rayon de cercle prédéfini
Interactions	Verticales, transversales et obliques	Verticales, transversales	Verticales
Négociations	Oui	Oui	Non
La généralisation est basée sur	Les algorithmes de généralisation, les patrons géométriques et les contraintes d'intégrité	Les algorithmes de généralisation et la représentation multiple traditionnelle	Les algorithmes de généralisation et les contraintes
Niveau d'enrichissement	Élevé	Moyen	Faible
Comportement des agents	Selon le type des agents	Générique	Générique