



HAL
open science

Un système multi-agent pour l'orchestration de processus de généralisation cartographique

Guillaume Touya

► **To cite this version:**

Guillaume Touya. Un système multi-agent pour l'orchestration de processus de généralisation cartographique. Journées Françaises des Systèmes Multi-Agents, Oct 2011, Valenciennes, France. pp.73-82. hal-02196601

HAL Id: hal-02196601

<https://hal.science/hal-02196601>

Submitted on 29 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un système multi-agent pour l'orchestration de processus de généralisation cartographique

Guillaume Touya

► **To cite this version:**

Guillaume Touya. Un système multi-agent pour l'orchestration de processus de généralisation cartographique. Journées Françaises des Systèmes Multi-Agents, Oct 2011, Valenciennes, France. pp.73-82. hal-02196601

HAL Id: hal-02196601

<https://hal.archives-ouvertes.fr/hal-02196601>

Submitted on 29 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un système multi-agent pour l'orchestration de processus de généralisation cartographique

G.Touya^a

guillaume.touya[at]ign.fr

^aLaboratoire COGIT Institut Géographique National 73 avenue de Paris 94160 Saint-Mandé – France

Résumé

La généralisation de données géographiques vectorielles est un problème spatialisé complexe qui consiste à simplifier des données géographiques (bâtiments, routes, etc.) pour satisfaire au mieux des contraintes de lisibilité. Il existe beaucoup de processus automatiques (dont certains reposent sur des SMA) spécialisés sur une partie du problème. Nous proposons une approche de résolution par agents, que nous appelons généralisation collaborative, pour réaliser l'orchestration de ces processus automatiques dans le but de généraliser une carte complètement. Notre approche combine des techniques de résolution spatiale, utilisées dans les SMA dédiés à la généralisation cartographique, avec des agents processus, choisis selon leurs capacités, pour traiter une partie de l'espace. Un prototype a été développé et des expérimentations de généralisation collaborative sont présentées.

Mots-clés : résolution de problème spatial, coordination, orchestration, appariement, généralisation cartographique

Abstract

Vector geographic data generalisation is a complex spatialised problem that aims at simplifying geographic data (i.e. a building, a road) in order to satisfy legibility constraints. Many automatic processes exist but are specialised to specific parts of the problem. The proposed agent-based resolution approach, that we call collaborative generalisation, orchestrates the automatic processes in order to generalise a complete map. The approach mixes classical spatial techniques from existing agent-based generalisation with matchmaking techniques to match the automatic process with the capabilities that best suits the characteristics of a part of space. A prototype has been developed and collaborative generalisation

experiments have been carried out.

Keywords: spatial problem resolution, coordination, orchestration, matchmaking, cartographic generalisation

1. Introduction

Ce papier présente une méthode de résolution de problème spatialisé par des agents, dédiée au problème de la généralisation cartographique. Le cadre de ce travail est donc l'automatisation de la généralisation cartographique de données géographiques vectorielles qui consiste en une simplification de l'information à représenter sur une carte lorsque l'échelle de représentation diminue (Fig. 1).

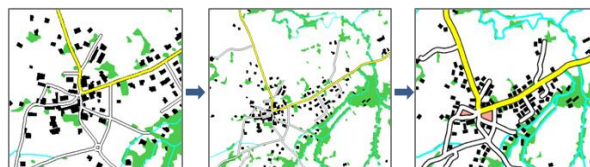


FIG. 1 – Changement d'échelle, puis, généralisation cartographique

La recherche s'intéresse à l'automatisation de la généralisation cartographique depuis de nombreuses années et plusieurs types de processus automatiques ont été développés, certains s'appuyant sur des techniques SMA [1], [16], [6], [12] ou [10] et d'autres non ([19] ou [1] par exemple). Ces processus sont tous spécialisés pour une sous-partie du problème de la généralisation d'une carte complète : ils peuvent être adaptés à un type de paysage tel que la ville ou la montagne, ou être adaptés à un thème de données seulement comme les routes ou l'occupation du sol. Il n'existe pas de processus capable de gérer tous les types de paysages et tous les thèmes de données de manière complète et efficace. La généralisation complète d'une carte comportant des paysages et des thèmes hétérogènes nécessite donc de recourir à

l'utilisation de plusieurs de ces processus automatiques existants bien qu'ils n'aient pas été conçus pour fonctionner ensemble. La problématique de notre travail est de modéliser la collaboration de processus de généralisation pour obtenir une utilisation optimale de ces processus.

La partie suivante présente plus en détail le contexte de ce travail et la problématique (la généralisation de cartes topographiques complètes), en insistant sur les précédents travaux s'appuyant sur des agents géographiques. La partie 3 présente notre approche, le modèle CollaGen, qui permet la collaboration des processus par leur orchestration et leur application sur de nouveaux types d'agents géographiques. Des résultats sont présentés et évalués en partie 4 et, enfin, la partie 5 conclut et propose quelques perspectives de recherche.

2. La généralisation de cartes topographiques complètes

2.1. La généralisation cartographique

Dans une base de données géographique, source à partir de laquelle une carte est construite, à chaque objet de la base est associée une géométrie ponctuelle, linéaire ou surfacique. La généralisation consiste, lors d'une diminution de l'échelle, à transformer ces géométries pour qu'elles restent lisibles tout en conservant les caractéristiques géographiques initiales (forme, position, signification) : par exemple, sur la Fig. 1, la géométrie des bâtiments a été grossie, simplifiée, certains ont été éliminés, d'autres déplacés ; des routes ont été éliminées. L'automatisation de ce processus est fondamentale pour permettre une meilleure visualisation des cartes à la demande sur les géoportails publics ou privés comme GoogleMaps, mais aussi pour la production de cartes en série, une des missions de l'Institut Géographique National.

Pour pouvoir être affichés sur une carte, les objets doivent vérifier certaines contraintes. Ces contraintes (voir les exemples de la Fig. 2) peuvent porter sur un objet, une relation entre objets, un groupe d'objets, une population (limitation de la surface totale occupée par les routes) – nous renvoyons à [22] pour une modélisation détaillée de ces contraintes.

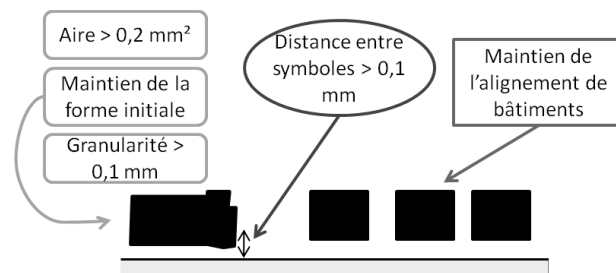


FIG. 2 – Exemples de contraintes de généralisation, sur un objet, entre deux objets et sur un groupe d'objets.

Dans la Fig. 2 et en général, les contraintes sont concurrentes car les simplifications pour permettre la lisibilité modifient les caractéristiques initiales des objets qui doivent aussi être maintenues pour permettre la compréhension de la carte. La généralisation cartographique peut donc être modélisée comme un problème de satisfaction de contraintes (CSP) [3]. Toutefois, les techniques de résolution classiques de CSP ne sont pas applicables à la généralisation cartographique pour plusieurs raisons :

- la géométrie est une variable très complexe,
- les résultats des algorithmes qui transforment les géométries ne sont pas prévisibles,
- les objets peuvent être éliminés ou agrégés ce qui entraîne des disparitions ou des fusions de variables.

D'après [6], il n'est donc pas possible de chercher la solution optimisant la satisfaction des contraintes en faisant varier les variables. L'automatisation de la généralisation requiert donc une méthode ad-hoc pour permettre la satisfaction des contraintes de manière optimale. Nous appelons algorithme de généralisation la transformation (e.g. la simplification, le déplacement) automatique de la géométrie d'un plusieurs objets. Et nous appelons ces méthodes pour enchaîner les algorithmes de généralisation de manière à optimiser la satisfaction des contraintes, des *processus de généralisation*. Des processus de généralisation existants adoptent des approches séquentielles, globales (une solution globale pour l'ensemble des contraintes est recherchée) ou pas à pas locales. La Fig. 3 montre deux résultats de généralisation par des processus globaux, par moindres carrés [19] et par 'poutres élastiques' [1]. Une approche séquentielle

définit à l'avance les algorithmes à utiliser les uns après les autres. Une approche globale cherche soit une position optimale des objets (sans utiliser d'algorithmes comme les moindres carrés), soit une séquence optimale d'algorithmes en fonction de la situation. L'optimalité recherchée étant la satisfaction optimale de l'ensemble des contraintes (par exemple, les moindres carrés minimisent la somme des carrés des résidus de chaque contrainte (écart avec la solution exacte)). Enfin, l'approche pas à pas locale identifie des espaces de travail locaux, et les généralise pas à pas avec des algorithmes, jusqu'à ce qu'on ne puisse plus rien améliorer, l'ordre étant primordial.



FIG. 3 – (a) généralisation par moindres carrés. (b) généralisation par poutres élastiques. Les conflits initiaux sont entourés.

Il convient de noter que la généralisation est un problème qui nécessite souvent de travailler à plusieurs niveaux d'abstraction : par exemple, une ville (niveau meso) est composée de d'îlots de bâtiments (niveau meso), eux-mêmes composés de routes et de bâtiments (niveau micro). Le problème des interactions multi-niveaux a déjà été abordé en simulation multi-agents dans [11] ou [14].

2.2. Des agents géographiques

Les processus de généralisation suivant une approche pas à pas locale sont adaptés à l'utilisation d'un système multi-agents :

- C'est un problème *fortement distribué*, car les transformations sont choisies dans un contexte local.
- C'est un processus *holistique*, car il faut analyser le contexte d'une situation (les relations géographiques, les voisinages,

les structures meso...) pour établir la bonne stratégie de généralisation [18].

- Il s'agit de *satisfaction de contraintes*, mais non résolvable par les techniques de CSP classiques.
- La généralisation cartographique dispose d'un *environnement évolutif*, car les objets géographiques sont transformés petit à petit.
- Il existe des *heuristiques* [17] pour guider efficacement la généralisation de certaines situations (e.g. les villes).

Plusieurs processus de généralisation ont donc été développés, considérant les objets géographiques (bâtiment, route, etc.) comme des agents géographiques dont l'objectif est de bien se généraliser en appliquant sur lui ou ses voisins des transformations. Les travaux suivants ([2], [16], [8], [6], [10] et [12]) proposent des processus, complémentaires les uns des autres, autour de ce concept d'agent géographique. Notamment, AGENT [16] propose une modélisation multi-niveaux d'agents géographiques, l'agent ville étant composé d'agents îlots urbains, eux-mêmes composés (et guidant la généralisation) des agents bâtiments. CartACom [6] propose un modèle de communications entre deux agents de même niveau pour résoudre des contraintes partagées entre les deux agents. La Fig. 4 montre des résultats obtenus avec ces deux derniers processus.

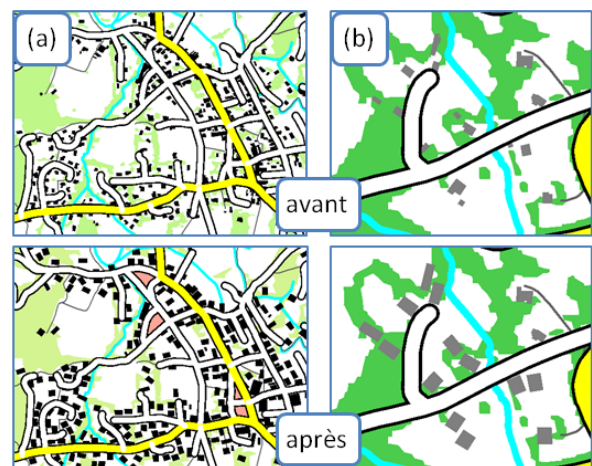


FIG. 4 – (a) généralisation par AGENT. (b) généralisation par CartACom.

2.3. La généralisation collaborative de cartes complètes

Les processus de généralisation existants

permettent de traiter certains paysages ou certains types de données mais aucun n'est capable de généraliser efficacement une carte présentant des paysages hétérogènes et des types de données variés. De plus, les types de paysages et le choix des données représentées varient beaucoup suivant les pays, les cultures et les applications cartographiques (e.g. carte topographique, touristique ou géologique), il nous paraît illusoire de chercher à concevoir un processus universel pour généraliser tout type de carte complète.

L'objectif de notre travail est donc de généraliser une carte complète en mettant en place une collaboration de processus, afin que chacun soit utilisé sur la partie optimale des données à généraliser. Le principe de la généralisation collaborative est de découper automatiquement les données en espaces géographiques qui peuvent être des territoires comme les zones urbaines, rurales ou de montagnes, mais aussi des espaces thématiques comme l'ensemble des routes [22]. Ces espaces ne sont pas nécessairement disjoints et peuvent se croiser (e.g. les routes traversent les zones rurales, qui peuvent empiéter sur les zones de montagne). Ensuite, la collaboration permet de choisir le meilleur processus disponible pour généraliser un espace donné (Fig. 5). Enfin, il faut s'assurer que la généralisation n'a pas créé de problème au bord de l'espace traité (et les corriger s'il y en a).

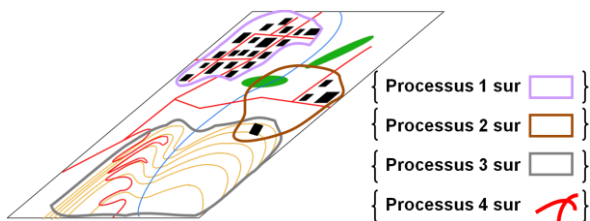


FIG. 5 – Principes de la généralisation collaborative.

Afin de mettre en place une généralisation collaborative automatique, nous proposons le modèle CollaGen (pour *Collaborative Generalisation*) qui s'appuie sur un système multi-agent pour orchestrer l'application de processus (qu'ils soient SMA ou non) sur des espaces géographiques. La partie suivante décrit les agents composant le modèle CollaGen ainsi que leurs interactions dans le système.

3. CollaGen, un SMA mixant agents géographiques et agents logiciels

3.1. Architecture générale de CollaGen

Dans notre système CollaGen, les agents sont à la fois les espaces géographiques et les processus de généralisation. Le système compte également un agent observateur qui analyse les généralisations en cours, ainsi qu'un agent registre et un agent chef d'orchestre qui détermine quel espace doit être généralisé avec quel processus. Dans CollaGen, contrairement aux précédents modèles de généralisation fondés sur des SMA ([16], [6], [10], [12]), les objets géographiques à généraliser (e. g. les bâtiments, les routes etc.) ne sont pas considérés comme des agents mais comme des éléments de l'environnement. En effet, les objets géographiques n'ont pas de comportement proactif du point de vue de CollaGen : ils sont traités globalement par un processus.

Le système multi-agent de CollaGen compte deux mondes qui ne communiquent pas directement : le monde géographique qui contient les *agents espaces* et l'*observateur* ; le monde logiciel dans lequel évoluent les *agents processus* et *correcteurs* d'effet de bord. Deux agents coordinateurs, le *registre* et le *chef d'orchestre* sont chargés de mettre en relation les agents des deux mondes pour réaliser la collaboration (Fig. 6).

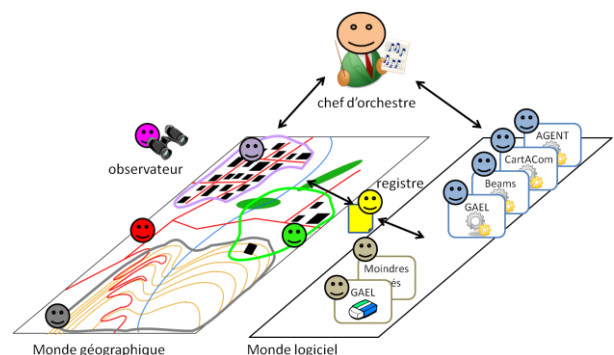


FIG. 6 – Architecture de CollaGen : des agents espaces sont appariés à des agents processus par un agent registre ; les appariements sont orchestrés par l'agent chef d'orchestre ; l'observateur s'assure du bon déroulement des généralisations.

L'environnement géographique représente l'extrait de la base de données géographique

que l'on souhaite généraliser. Les éléments de l'environnement (les bâtiments, les routes, etc.) comme les agents placés dans cet environnement ont des coordonnées pour les situer précisément. Ainsi les agents espaces qui évoluent dans cet environnement géographique sont situés par leur géométrie. Les entités que les agents espaces peuvent percevoir dans cet environnement sont les objets géographiques de la base de donnée (bâtiments, route, groupes de bâtiments etc.) et des 'moniteurs de contrainte' [7]. Ces moniteurs sont des artefacts, liés chacun à une contrainte de généralisation et à un objet soumis à cette contrainte, qui, comme dans [15], permettent aux agents d'interagir avec l'environnement géographique pour localiser et évaluer la satisfaction des contraintes. Les moniteurs permettent de localiser une contrainte dans un espace et d'en évaluer la satisfaction à tout moment du processus.

D'un autre côté, les agents processus ne sont pas situés sur une carte mais évoluent dans un monde purement logiciel : ils sont localisés par un emplacement sur la machine accueillant le système collaboratif. Ce monde logiciel est minimaliste car adapté à la problématique de CollaGen : les agents sont statiques et n'ont pas accès à un réseau. Ils peuvent uniquement communiquer avec le registre et le chef d'orchestre et percevoir des ressources utiles au fonctionnement de CollaGen comme une ontologie et les contraintes formalisées (décrites dans [22], elles permettent aux agents processus de se paramétrer de manière interopérable). Les contraintes formalisées sont les contraintes de généralisation mises dans un format générique pour être exploitées par le système et chacun des moniteurs de contrainte est relié à une contrainte formalisée.

Dans les parties suivantes, nous décrivons le fonctionnement de chaque agent du système CollaGen, permettant de réaliser cette généralisation collaborative.

3.2. Les agents espaces géographiques

Les agents espaces géographiques, comme les agents de [16] ou [6], ont pour objectif de bien se généraliser ce qui revient à une satisfaction optimale des moniteurs de contrainte contenus dans l'étendue spatiale de l'espace. Le chef d'orchestre choisit les types

d'espaces à construire selon les spécifications de la carte et ses connaissances, et les construit automatiquement par des algorithmes d'analyse spatiale [21]. Les espaces s'enrichissent par introspection en créant de nouveaux objets par analyse spatiale, comme des alignements de bâtiments ou des relations entre deux objets. Le processus AGENT a par exemple besoin de l'existence de ces alignements en entrée du processus. Les espaces cherchent aussi s'il y a d'autres agents espaces voisins (en adjacence, en intersection ou simplement proches) avec lesquels des effets de bord pourraient apparaître (e.g. le déplacement d'un bâtiment dans le premier espace viole une contrainte avec un objet de l'autre espace). Ils attendent ensuite que le chef d'orchestre les désigne comme l'espace à généraliser à un moment (Fig. 7). La détermination de l'ordre des généralisations est détaillée en 3.5.

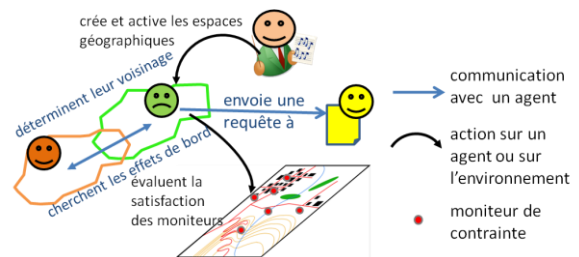


FIG. 7 – Interactions d'un agent espace avec les autres agents.

Une fois désigné, un espace s'évalue à partir des moniteurs et fait une requête au registre s'il n'est pas satisfait. Puis, il attend sa généralisation par un processus. Une fois généralisé, il s'évalue en calculant la satisfaction moyenne de ses moniteurs de contrainte et l'évolution de leur satisfaction. Enfin, l'espace généralisé vérifie avec ses voisins si la généralisation n'a pas généré d'effets de bord. Si c'est le cas, l'agent espace fait appel à un agent processus, correcteur d'effets de bord, pour corriger ces effets de bord.

3.3. Les agents processus de généralisation

Situé dans le monde logiciel, un agent processus a pour objectif de s'exécuter sur des espaces géographiques conformes à ses capacités. Pour atteindre cet objectif, il s'inscrit auprès du registre en indiquant ses capacités formelles : décrit dans [22], le modèle comprend des pré-conditions sous la

forme d'espaces pouvant être passés en entrée (e.g. AGENT fonctionne sur les espaces urbains) et des post-conditions sous la forme de contraintes formalisées gérées *a priori* (e.g. AGENT gère les contraintes de taille et de granularité des bâtiments). Puis, il attend d'être choisi par le registre lors d'une requête d'un espace. Quand il a été choisi pour généraliser un espace, un agent processus détermine ses paramètres en fonction des contraintes, en utilisant une ontologie des concepts de généralisation [22]. En fait, une fonction de traduction est associée à chaque type de processus et l'agent peut l'exécuter pour se généraliser. Par exemple, pour le processus AGENT, la taille minimum des bâtiments est un paramètre spécifique du processus à remplir pour utiliser le processus donc la fonction de traduction cherche s'il y a, parmi les contraintes formalisées, une contrainte portant sur le concept de 'bâtiment' et la caractéristique 'aire'; si c'est le cas, la fonction de traduction récupère la valeur de la contrainte (e.g. 0,2 mm² carte) et l'affecte au paramètre. Une fois paramétré, le processus s'exécute sur l'espace, puis, attend d'être de nouveau sollicité (Fig. 8).

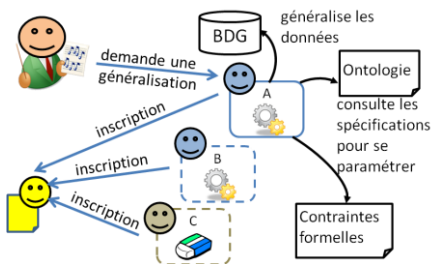


FIG. 8 – Interactions d'un processus avec les autres agents.

3.4. L'agent registre

L'agent registre est un agent intermédiaire ('middle agent') au sens de [20]: il a pour objectif d'apparier un besoin (l'agent espace nécessitant une généralisation) et un service (l'agent processus) (Fig. 9). Nous faisons le choix de modéliser le registre comme un agent intermédiaire, car la norme FIPA-ACL sur les registres [9] ainsi que les travaux sur l'appariement besoin/service [5] suggèrent que le registre doit être un agent proactif plutôt qu'un élément consultable de l'environnement. L'agent registre reçoit des requêtes de la part d'agents espaces. Il analyse alors les moniteurs de contrainte

contenus dans l'espace, calcule sa réponse (à partir des capacités formelles) et recherche les agents processus constituant cette réponse dans l'environnement logiciel. Finalement, il renvoie au chef d'orchestre une liste des processus avec leurs emplacements pour que le chef d'orchestre puisse communiquer avec eux.

L'appariement est fait en cherchant la meilleure adéquation entre les moniteurs de contrainte présents dans l'espace faisant la requête et les capacités formelles des processus disponibles stockées dans le registre [21]: par exemple, si l'espace ne contient que des moniteurs de contrainte de taille minimum des bâtiments et qu'un seul agent processus gère ce type de contrainte d'après les capacités formelles, alors c'est ce processus qui est choisi par le registre.

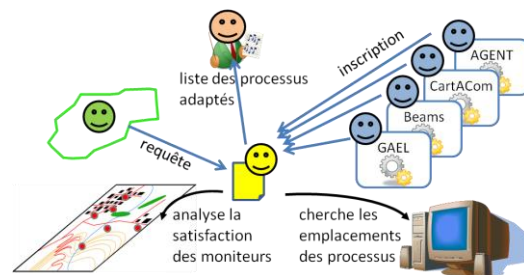


FIG. 9 – Interactions du registre avec les autres agents.

Selon la classification de [5], notre registre peut être un agent intermédiaire de type *courtier* ou un *apparieur*. Un *courtier* met en relation un client qui ne connaît aucune des capacités des fournisseurs mais dont il connaît déjà les besoins. Par contre, un agent *apparieur* ne connaît pas *a priori* les besoins du client mais les capacités des fournisseurs sont connues de tous. Notre agent registre est donc plutôt du type apparieur (matchmaker) que courtier car seul le « client » connaît ses préférences (seul l'espace géographique connaît ses caractéristiques) et tous les intervenants connaissent les capacités des processus via le registre qui rend ses informations publiques (les agents espaces consultent notamment ces capacités pour guider leur enrichissement par introspection). Cette analyse est confirmée par l'utilisation de la taxonomie de [25] qui classe les agents intermédiaires en fonction de six critères (e.g. « Quel type d'information est envoyée dans la requête ? »). Selon cette classification, les

critères de notre agent registre correspondent à une variante d'apparieur.

3.5. L'agent chef d'orchestre

L'agent chef d'orchestre est chargé de la coordination des généralisations : il choisit le prochain agent espace à être généralisé. La coordination est plutôt réalisée ici par orchestration (décision centralisée) que par chorégraphie (décision distribuée) [4]. De plus, les agents espaces et processus ne pouvant communiquer directement, le chef d'orchestre est un intermédiaire de communication entre les agents du système (Fig. 10).

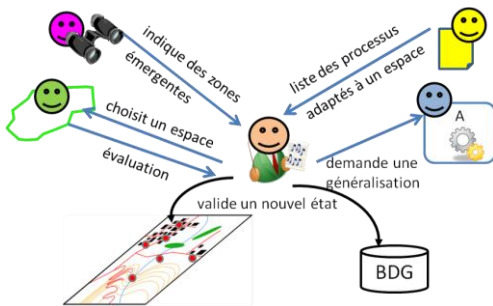


FIG. 10 – Interactions du chef d'orchestre avec les autres agents.

Pour réaliser son orchestration, le chef d'orchestre dispose d'heuristiques, tirées de travaux passés comme [18], exprimées sous forme de règles de la forme :

$Espace_précédent \wedge Processus_précédent \Rightarrow type\ d'espace$

Une règle peut être par exemple « on généralise le réseau routier après les villes ». Le chef d'orchestre est donc doté d'un moteur d'inférence capable d'inférer un type d'espace à généraliser à partir des règles. Ce moteur d'inférence est mixte : par défaut, il réalise un chaînage avant pour trouver la règle dont la prémisse est la plus proche sémantiquement (selon l'ontologie) de la situation courante. Si aucune règle ne correspond, le moteur réalise un chaînage arrière en partant de la règle qui annonce la fin de la généralisation collaborative. Si des types d'espaces sont oubliés par le chaînage arrière, le moteur les insère avant la première règle du chaînage arrière, par ordre décroissant de la quantité de moniteurs de contrainte insatisfaits dans les agents de ce type. Le chef d'orchestre ordonne ensuite les instances du type choisi (e.g. tous les agents espaces de type ville) par ordre décroissant de la quantité de moniteurs

de contrainte insatisfaits.

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

- (R1). début \Rightarrow espaces urbains.
- (R2). espace urbain \Rightarrow réseau routier.
- (R3). réseau routier \Rightarrow espaces ruraux.
- (R4). espace de montagne \Rightarrow on s'arrête.

Dans notre exemple, le fait est le suivant : le système vient de généraliser les graphes de flexibilité par un processus spécialisant le modèle des Beams [1], sachant que dans l'ontologie, le concept graphe de flexibilité est un réseau routier. La règle R3 est donc celle qui est choisie par le moteur d'inférence par chaînage avant. Le prochain espace à généraliser est donc l'espace rural. Ensuite, une fois les espaces ruraux généralisés, le moteur est de nouveau interrogé avec 'espace rural' comme espace précédent mais il ne trouve aucune règle qui correspond. Il cherche donc la règle terminant la généralisation et trouve la règle R4. Et par chaînage arrière, le moteur infère que le prochain type d'espace à généraliser est 'espace de montagne' (en supposant qu'il ne reste plus d'autre type d'espace à généraliser).

La Fig. 10 montre, enfin, que le chef d'orchestre veille au déroulement de la partie pas à pas (par essai/erreur) du système : à partir de l'évaluation globale d'une généralisation, fournie par l'espace concerné, il décide si le nouvel état est validé ou si un autre processus doit être essayé.

3.6. L'agent observateur

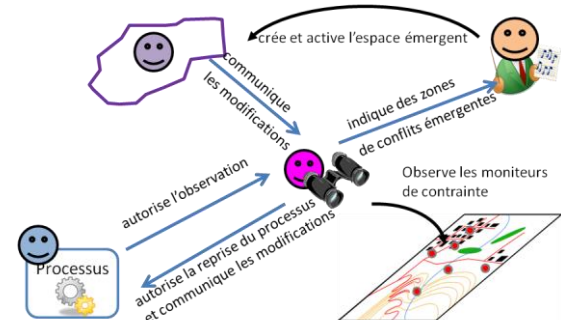


FIG. 11 – Interactions de l'observateur.

L'agent observateur est chargé d'observer le déroulement d'une généralisation et de l'interrompre si un groupe de conflits persistants, non résolus par le processus, émerge (Fig. 11). S'il observe un tel groupe émergent, il charge le chef d'orchestre de créer un nouvel agent espace à partir du groupe, pour le traiter différemment. Le fonctionnement de l'agent observateur a été décrit en détail dans [7] et comparé à un mécanisme similaire dans le processus de généralisation CartACom.

4. Résultats et évaluation

Un prototype de ce modèle CollaGen a été implémenté : le prototype a été doté de 9 processus automatiques et peut généraliser des cartes présentant des paysages et des thèmes différents. La Fig. 12 montre un résultat, pour lequel une quarantaine d'agents espaces ont été construits et les contraintes correspondent aux spécifications de la carte au 1 : 50000 de l'IGN.

Les résultats de généralisation collaborative obtenus par CollaGen (Fig. 12) sont très positifs car les différents thèmes et paysages sont bien généralisés tout en gardant une carte homogène (grâce aux contraintes uniques servant à paramétrer l'ensemble des proces-

sus). S'il reste des imperfections (trop petites pour les distinguer sur la Fig. 12), elles sont dues aux processus eux-mêmes car CollaGen ne peut pas généraliser beaucoup mieux que le meilleur processus disponible pour un type d'espace (le mécanisme d'observation de conflits émergents décrit dans [7] permet des améliorations).

Nous avons par ailleurs comparé sur une zone les résultats obtenus avec CollaGen avec ceux obtenus avec des logiciels du commerce (issus de travaux de recherche), combinant souvent deux ou trois processus de généralisation [21]. Dans cette zone, les problèmes de généralisation sont nombreux et divers (zones urbaine, périurbaine, rurale, montagne, routes en lacets, remontées mécaniques, etc.). CollaGen s'avère meilleur par sa capacité à différencier les situations (urbain, périurbain et rural dans ce cas) et à les traiter par un processus adapté alors que les logiciels comparés utilisent un seul processus, seulement efficace sur un de ces espaces (généralement l'urbain). Par ailleurs, CollaGen permet d'ajouter facilement des processus pour traiter, par exemple, la végétation et les remontées mécaniques, tout en restant interopérable.



FIG. 12 – Extrait d'un résultat de généralisation obtenu avec CollaGen, autour de la ville d'Orthez, avec la collaboration de neuf processus.

Grâce aux tests réalisés, nous avons étudié les choix d'orchestration réalisés par nos agents pour voir s'ils sont bien cohérents avec les choix que nous aurions effectués, en tant qu'expert cartographe, manuellement. Dans le cas où plusieurs processus sont réellement en concurrence, les choix effectués sont les bons, seul un des espaces de la zone d'Orthez (Fig.12) est généralisé par CartACom, alors que nous aurions choisi AGENT. Mais l'état n'étant pas validé par le chef d'orchestre (la généralisation par CartACom ne satisfait pas assez les contraintes), et l'espace est finalement bien généralisé par AGENT.

Par ailleurs, nous avons évalué la convergence de notre système. CollaGen peut se résumer en une série de cycles dans lesquels un agent espace, désigné par le chef d'orchestre, cherche à se généraliser au mieux en faisant appel à un processus par l'intermédiaire du registre. Il faut faire attention à la convergence à l'intérieur d'un cycle (comment un espace arrête d'essayer de se généraliser) et à la convergence des cycles de généralisation collaborative (comment le chef d'orchestre arrête de désigner des espaces pour la généralisation). Deux points peuvent poser problème dans la convergence d'un cycle de généralisation : l'émergence de sous-espaces [7] (plusieurs conflits proches non résolus par un processus peuvent émerger comme un sous-espace à généraliser par un autre processus, par exemple, un groupe dense de bâtiments trop proches dans une zone rurale, que n'arrive pas à traiter un processus dédiés aux zones rurales) et le choix d'un processus tant que l'état global des moniteurs n'est pas suffisant. Concernant l'émergence de sous-espaces, nous avons limité le cycle de vie des espaces émergents pour éviter un phénomène récursif (l'espace émergent n'est généralisé qu'une fois par un processus). De plus, l'observateur n'observe plus les zones dans lesquelles un espace a déjà émergé empêchant l'observation à l'infini de nouveaux espaces émergents (l'espace observé n'est pas infini). Pour le nombre de généralisations, par des processus, d'un même espace, nous n'avons pas placé de garde-fou théorique. Nous avons simplement limité à un le nombre des applications possibles d'un même processus sur un espace (il faut noter que deux fonctions de traduction du paramétrage d'un processus signifient dans CollaGen que deux processus sont

disponibles, ce qui autorise d'essayer une autre fois un processus avec un paramétrage calculé différemment). Ainsi, un espace ne peut pas être généralisé plus de fois qu'il n'existe de processus disponibles adaptés à lui (selon la liste fournie par le registre). Ce nombre de processus adaptés n'étant pas infini, le cycle de généralisation collaborative d'un espace finit par se terminer, même quand l'état n'est pas complètement satisfaisant.

Enfin, concernant la convergence globale du système, pour éviter que l'orchestration aboutisse à la généralisation des mêmes espaces indéfiniment, le moteur d'inférence est doté d'un outil de vérification de la convergence des règles (on évite $A \Rightarrow B \Rightarrow C \Rightarrow A$). Ces dispositifs permettent de garantir la convergence théorique et ne présentent pas d'inconvénient sur la qualité du résultat, le système obtenant malgré tout de bons résultats globaux.

D'autre part, bien que ce travail ne s'inscrive pas dans le cadre de la généralisation en temps réel (le temps d'un zoom arrière par exemple), il convient de mentionner le temps de calcul d'une généralisation par CollaGen. Ici, le temps clé représente une nuit de calculs sur des machines classiques et CollaGen permet des généralisations de cet ordre de durée pour une carte complète, sans que l'implémentation soit optimale.

5. Conclusion et Perspectives

Dans ce papier, nous avons présenté une application des Systèmes Multi-Agents pour la collaboration de processus automatiques de généralisation cartographique. Ce modèle combine les techniques de résolution de problèmes situés des précédents modèles SMA dédiés à la généralisation et des techniques d'appariement entre demande et fournisseur de service, utilisées dans le problème de la composition de services Web. Un prototype a été implémenté et les résultats obtenus sont très intéressants.

Les perspectives de ce travail sont assez nombreuses, la première étant de réaliser une implémentation complète de tout le système robuste aux pannes, inévitables, dans les processus utilisés. Dans un deuxième temps, nous souhaitons étudier l'intégration dans notre modèle de vrais mécanismes de

communication entre agents (seuls des mécanismes ad-hoc simplistes ont été implémentés) pour le rendre plus souple aux évolutions futures (si les agents disposent d'un protocole de communication, il est facile d'ajouter de nouveaux types de communications, pour améliorer la gestion des effets de bord entre deux agents par exemple). Enfin, nous souhaitons améliorer CollaGen en intégrant les principes de deux travaux de recherches : [13] pourrait permettre d'améliorer l'évaluation globale de la généralisation d'un espace et [14] pourrait permettre de mieux modéliser les mécanismes de gestion des effets de bord après une généralisation.

Références

- [1] M. Bader, M. Barrault, R. Weibel, Building displacement over a ductile truss, *International Journal of Geographical Information Science*, **Vol.19**, pp. 915-936, 2005
- [2] C. Baeijs, Y. Demazeau, L. Alvares, Sigma: Application of multi-agent systems to cartographic generalization, *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, Eindhoven, The Netherlands. W. Van de Velde & J.W. Perram, eds, Springer Verlag, LNAI n°1038, pp.163-176
- [3] A. Bellicha, B. Neveu, B. Trousse. et al. (équipe du groupe de recherche CSP'Flex du PRC-IA, 27 auteurs). Autour du problème de satisfaction de contraintes. *Actes des 5èmes journées nationales du PRC-GDR Intelligence Artificielle*, pp. 159-178, 1995.
- [4] Y. Charif, N. Sabouret, Un protocole de coordination d'agents introspectifs pour la chorégraphie dynamique de services, *Revue d'Intelligence Artificielle*, **Vol. 23**, pp. 47-80, 2009
- [5] K. Decker, K. Sycara, M. Williamson, Middle-Agents for the internet, *15th International Joint Conference on Artificial Intelligence*, 1997
- [6] C. Duchêne, C. Cambier, Généralisation cartographique avec des agents qui voient et communiquent, *9èmes Journées Francophones sur les Systèmes Multi-Agents*, Hammamet, Tunisie, pp. 175-188, 2003
- [7] C. Duchêne, G. Touya, Emergence de zones conflits dans deux modèles de généralisation cartographique multi-agents. *18èmes Journées Françaises des Systèmes Multi-Agents*, Cepaduès, Mahdia, 2010, pp. 33-42
- [8] K. Fernandes, M. Occhetto, « Une approche multi-agents hybride pour la conception de systèmes complexes à raisonnement intégré ». XII Congrès Francophone AFRIF-AFIA RFIA'2000, Paris, France, Février 2000.
- [9] FIPA agent management specification. Rapport technique, 2004
- [10] J. Gaffuri, Système multi-agent pour la déformation en généralisation cartographique. *Journées Francophones sur les Systèmes Multi-Agents*, Annecy, 2006
- [11] J. Gil Quijano, G. Hutzler, T. Louail, De la cellule biologique à la cellule urbaine : retour sur trois expériences de modélisation multi-échelles à base d'agents, *17èmes Journées Françaises des Systèmes Multi-Agents*, Cepaduès, 2009
- [12] N. Jabeur, B. Boulekrouche, B. Moulin, Using multiagent systems to improve real-time map generation, *Advances in Artificial Intelligence*, pp. 37-48, 2006
- [13] A. Nongaillard, P. Mathieu, P. Everaere, Bien-être de nash : enjeux et difficultés pour une société d'agents. M. Occhetto, L. Rejeb, eds, *Systèmes Multi-Agents, Défis Sociétaux, JFSMA'10*, Mahdia, 2010, pp.129-138, Cépaduès
- [14] S. Picault, P. Mathieu, Y. Kubera, PADAWAN, un modèle multi-échelles pour la simulation orientée interactions. M. Occhetto, L. Rejeb, eds, *Systèmes Multi-Agents, Défis Sociétaux, JFSMA'10*, Mahdia, 2010, pp.193-202, Cépaduès
- [15] M. Piunti, A. Ricci, O. Boissier, J. F. Hübner, Manipulation implicite d'une organisation multiagent via l'environnement, *17ème Journées Francophones des Systèmes Multi-Agents*, pp. 69-78, 2009
- [16] A. Ruas, Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. Thèse de doctorat, Université de Marne-la-Vallée, France, 1999.
- [17] A. Ruas, W. Mackaness, Strategies for urban map generalization. *Proceedings of 18th International Cartographic Conference*, 1997, pp. 1387-1394.
- [18] A. Ruas, C. Plazanet C., Strategies for Automated Generalization. *Proc. of the 7th International Symposium on Spatial Data Handling*, Delft, Pays-Bas, 1996, pp.6.1-6.17.
- [19] M. Sester, Optimizing Approaches for Generalization and Data Abstraction, *International Journal of Geographic Information Science*, **Vol. 19**, pp. 871-897, 2005
- [20] K. Sycara, S. Widoff, M. Klusch, J. Lu, Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, **Vol. 5**, pp. 173-203, 2002
- [21] G. Touya, C. Duchêne, CollaGen: Collaboration between Automatic Cartographic Generalisation Processes, A. Ruas, ed, *Advances in Cartography and GIScience*, pp. 541-558, 2011, Springer.
- [22] G. Touya, C. Duchêne, A. Ruas, Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes, *6th GIScience Conference, Zurich*, 2010

- [23] UDDI version 3.0.2. Rapport technique, 2004.
- [24] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 2000
- [25] H. C. Wong, K. Sycara, A taxonomy of Middle-Agents for the internet, *4th International Conference on Multi-Agent Systems*, pp. 465-466. IEEE Computer Society, 2000