Coordinative agents for automated generalisation of rural areas

Cécile Duchêne

Institut Géographique National
COGIT Laboratory
2-4 av. Pasteur
94165 Saint-Mandé CEDEX
FRANCE
Tel : (33) 1 43 98 85 45 / Fax : (33) 1 43 98 81 71
cecile.duchene@ign.fr

Abstract
We work on automating the generalisation of topographic databases, in order to produce topographic maps. We use an agent-oriented approach: the geographic features (roads, rivers, buildings etc.) are modelled as autonomous agents, as it has previously been done within the European AGENT project [Barrault et al. 01]. To handle rural areas, our approach then consists in letting these agents interact so that each of them either finds its new place and geometric representation or eliminates itself, so that the whole fits to the generalisation specifications. For that, our agents are provided with capacities of perception of their spatial environment, as well as capacities of dialoguing with the surrounding agents. This approach has been implemented and tested on real subsets of data. In this paper we describe the system. Some encouraging results are also presented and discussed.

Keywords: Rural areas generalisation, topographic data, multi-agent, multi-agent coordination, communicative agents, representation of the spatial environment.

1. Introduction

In this paper, we present a system dedicated to automated cartographic generalisation of topographic data, based on a multi-agent approach.
The general approach of generalisation we use is localised, step by step, as proposed by [Brassel and Weibel 88; McMaster and Shea 88, Ruas 99]. More precisely, our work takes place in the context of the European AGENT project [Barrault et al. 2001], that ended in December 2000. Our research aims to pass beyond some limitations of the generalisation prototype provided by the AGENT project, by proposing an additional method of resolution.
The paper is structured as follows. Part 2 briefly presents the approach of the AGENT project, on which our work relies, and the objectives of our work. Part 3 describes our approach and the designed system. Part 4 presents and evaluates the results provided by the system in its current state. Finally, part 5 draws some perspectives for ongoing work.

2. Context and objectives

2.1. The AGENT approach and its limits

The concepts of agent and multi-agent system [Weiss 99] belong to the domain of Artificial Intelligence. An agent can be thought of as an 'intelligent' object that has a goal and acts autonomously to reach this goal. In the approach of the AGENT project [Barrault et al. 2001], each geographical object of the database (road, building, etc.) is designed as an agent. The goal of each agent is to satisfy a set of constraints that model the generalisation requirements. Some constraints involve one agent: a building must be big enough, a road must not be coalesced, etc. Other constraints involve several agents: symbols must not overlap, relative positions must be preserved, etc. To handle these constraints, the approach of the AGENT project uses a multi-level, pyramidal model of agents. Single agents (like buildings) are called *micro agents*. On top of these micro agents, *meso agents* are designed, that correspond to groups of spatially organised features: a set of aligned buildings, a building block (set of buildings surrounded by roads), a town, etc (Figure 1). *Micro agents* perform individual generalisation without taking their surroundings into account: they apply generalisation algorithms to themselves, like dilation or simplification, in order to satisfy their internal constraints. The constraints that involve several agents are designed and handled at the level of *meso agents*, e.g. a building block agent is in charge of handling the overlap conflicts between its roads and buildings.

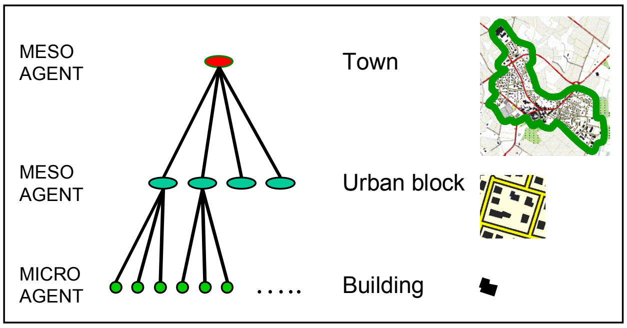

Figure 1. The pyramidal structure of agents used by the European AGENT project (after [AGENT Consortium 98])

This approach has proven to provide good results for the cartographic generalisation of urban areas. The pyramidal organisation of agents fits well for urban spaces, because the urban space is naturally structured in a pyramidal manner, with buildings, urban blocks and towns forming nested levels. The success of the approach is also due to the fact that previous research in automated cartographic generalisation had provided relevant methods for the generalisation of a urban block [Ruas 98].

But this pyramidal organisation has limits. First, it supposes to foresee all the kinds of meso agents, i.e. all kinds of groups of geographical features that can need to be handled together because of shared constraints. Then it supposes to design for each kind of meso-agent the relevant measures to assess its internal constraints, and the relevant transformation algorithms to solve its internal conflicts. Finally, in the AGENT model one agent is not supposed to be part of more that one meso-agent, i.e. the meso-agents are disjoint. Thus this model is only suitable for spaces that have a natural pyramidal organisation, i.e. where it is possible to distinguish disjoint groups of objects so that no constraint is shared by objects belonging to different groups. This is a problem within rural areas, where no meaningful disjoint groups are really distinguishable.

2.2. Our objectives

The aim of our work is to go beyond the limits of pyramidal organisations listed above, by proposing another organisation of agents to handle the relational constraints. *Relational constraints* are constraints that apply to a relation between two geographic objects, e.g. the constraint that prevents symbols overlaps. Relationships between geographic objects are inherent to a geographical database (e.g. proximity, relative position). Because of generalisation we consider these relationships as constrained, therefore the expression "relational constraints".

In our approach, we *a priori* consider only one level of agents. To handle a relational constraint between two geographic agents, we intend to use bilateral interactions between these agents. Relational constraints are thus handled directly by the cartographic agents they involve, like the constraints that apply to one single cartographic agent. This makes the resolution more adaptive, but also a bit more complex to manage. This, because relational constraints differ from the constraints applying to a single agent in two ways. First, to solve a conflict stemming from a relational constraint, there is a choice to apply a transformation algorithm to any of the two geographical agents involved, or even to both of them. Secondly, when an action of a geographical agent creates or increases a relational conflict, it does not only degrade its own state, but also the state of the other agent involved by the conflict. Thus, by 'transitivity', because of relational constraints, modifying one agent can require modifications across a large extent of the database.

We do not aim to replace the pyramidal approach of the AGENT project presented in previous paragraph, but to add another approach for rural areas, or more generally low density areas, where the pyramidal approach does not apply. To begin with, we set up a new system, totally distinct from the one stemming from the AGENT project. In the medium term, the two approaches should be combined.

To set up and test our approach, we work on the generalisation of rural areas of the geographical database 'BD Topo®', the database of the French National Mapping Agency that has approximately the same content and precision as a topographic map at scale 1:15 000. The generalisation requirements are to produce cartographic databases for medium scales (1:25 000 to 1:50 000).

3. Proposed approach

3.1. Principles: our agents need to see and to communicate

As in the approach of the AGENT project (2.1), our agents need capacities of introspection in order to assess and satisfy their internal cartographic constraints. This has already been treated by the AGENT approach, thus we concentrate on the new capacities needed by our agents in order to tackle their relational constraints.

A relational constraint of an agent always involves the agent itself, and another agent that is within its neighbourhood. This is why, to assess their relational constraints, our agents need capacities of perception of their environment, as well as an explicit representation of this environment. The modelled representation of the environment is presented in paragraph 3.2.
Moreover, to decide how to act in order to solve a relational conflict, an agent needs to gather information about itself (its state, what it can do or not), but also about the other agent involved in the conflict. In some cases, it is not able to compute all the information itself and needs to get it from the other agent. This is why our agents need to communicate with their neighbours. Figure 2 illustrates this. The road agent which is a dead-end (yellow) can compute its symbol overlap with the building agent (black). The building agent can compute its proximity with the red road. In order to undertake the right action, the dead-end agent needs to know that the building is stuck. Otherwise, it should wait for the building to move away, because it is a cartographic rule that, *a priori*, moving a building has less consequences than moving a road. The information can be transferred from the building to the dead end thanks to a dialog such as "Move away! No, I cannot". This is why our agents need to communicate. Our agents need to have conversations, i.e. "task-oriented, shared sequences of messages that they observe, in order to accomplish specific tasks" [Labrou 2001].

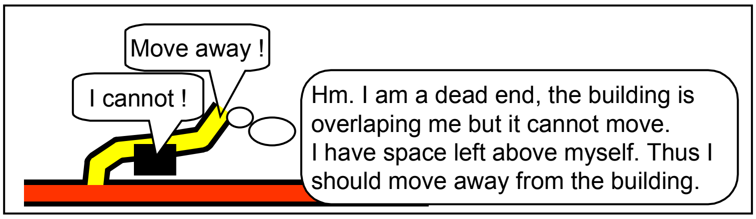

Figure 2. To take the right decisions, our agents need to communicate

The communication system must enable each agent to begin an action, interrupt this action to send a message to another agent, interpret a received message, and go on with its action according to the received messages. For that, we use pre-recorded 'dialog scenarios', designed for binary communication, that indicate how to act and how to answer when receiving a given message. Paragraph 3.3 describes this communication system.

3.2. Representation of the spatial environment

Our modelling of the spatial environment of an agent relies on four main principles:

1. ***Different kinds of geographic agents.*** We distinguish 4 categories of agents with regard to the description of their spatial environment (Figure 3). Note that this classification is neither exhaustive nor definitive. *Small compacts* are small cartographic objects like buildings or point symbols that do not absorb the displacements (they are moved as a whole, even if besides they can need to transform their shape for internal generalisation purpose). *Network linears* are linear objects that belong to a network (roads, railways, streams etc.). *Independent linears* are linear objects that are not part of a network (e.g. hedge, embankment). Contrary to small compacts, they do absorb displacements by modifying their shape. Finally, *Partition areas* are area objects forming a partition of the space (e.g. the landuse).

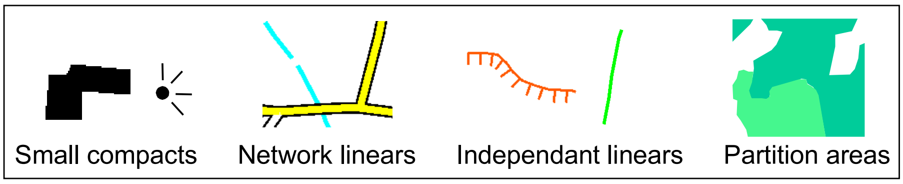

Figure 3. Different kinds of geographic agents

2. ***Consider the topology.*** To facilitate the management of topological relationships between features, we explicitly create the topological faces stemming from a division of the space by the Netwok linear objects. The created objects are called *Network faces*.

3. ***Identify the neighbours, then the relational constraints.*** Our agents need to perceive all the agents with which they share a relational constraint. For the sake of simplicity, we decide that each agent is able to perceive the part of space inside a buffer of offset *d* around itself, the distance *d* being chosen big enough to contain all the agents potentially sharing a relational constraint (*d* depends on the database and the generalisation specifications). We call this buffer the *Environment Zone* of the agent (Figure 4). Each agent computes its Environment Zone during the initialisation stage, before the generalisation itself begins. Then it

detects its "neighbours" (the agents encroaching on this zone), and identifies its relational constraints with each neighbour. A given relational constraint is search for with a given neighbour depending on the nature of both agents and their spatial configuration. For instance, a non-overlap constraint is systematically created for a *Small compact*, with a *Small compact* neighbour that belongs to the same *Network face*. A constraint of "parallelism between road and building" is searched for a building, only with the closest road neighbour (if there is any road in its Environment Zone), and only if the distance to this road is inferior to a given threshold.

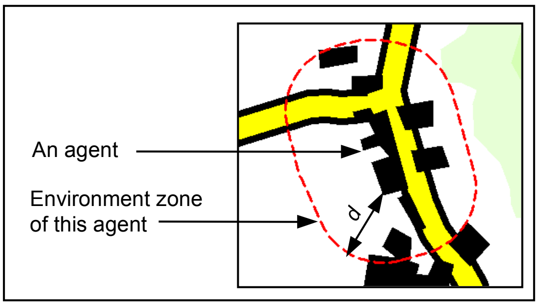

Figure 4. Environement Zone of a building agent.

4. ***Relational constraints do constrain the space.*** Up to now, we have concentrated our work on the representation of the relational constraints for *Small compacts*. To represent their relational constraints, we introduce the concept of *Constrained zone*. The idea is that most of the relational constraints of an agent do constrain the space around it: there are some zones of the space, where the agent cannot be (partially or totally) without violating some constraints. We call these zones *Constrained zones*. For instance, Figure 5 shows the constrained zones of a building relatively to two relational constraints with a dead-end road. If, during its generalisation, the building agent encroaches on the horizontally hatched area, it violates the proximity constraint with the dead end. If it encroaches on the vertically hatched area, it violates the constraint of preservation of the relative position with the dead end (as it is supposed to remain at the end of the dead end). *Constrained zones* are interesting because they enable to replace a theoretic spatial relation like "be at the end of the dead end", with a topologic relation, easier to manage, like "be totally outside a given area (the constrained zone)".

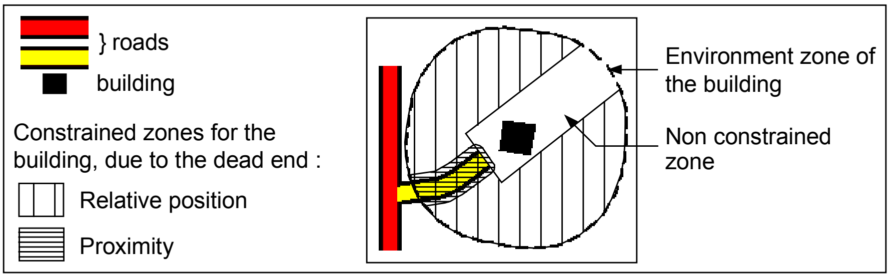

Figure 5. Constrained zones.

Computing its *constrained zones* is part of the initialisation of the agent. Then the agent updates its constrained zones each time it is activated.

The aim of computing the constrained zones of an agent is not to fully forbid the agent from encroaching on them. It is rather a means of computing more easily, for the present position or for a potential future position of the agent, which relational constraint it respects and which ones it does not. When handling its relational conflicts, an agent uses its constrained zones:

- to choose a new position that minimises the constraint violations (i.e. a position that avoids its constrained zones as much as possible),
- to immediately know which relational constraints its new position is still violating, and thus to send a message to the involved agents asking them to solve the problem.

3.3. Communication system

It has been told above that our agents need to communicate by sending messages to each others, so as to satisfy as well as possible their relational constraints. In our system, we only consider conversations involving two agents (dialogs). But an agent can have several conversations with several agents at a time.

To make the communication between agents possible within our system, we have to define the structure of the messages, as well as protocols for chaining the messages into conversations. For that, we base ourselves on works stemming from the multi-agent domain: [Ferber 95; Barbuceanu & Fox 95].

These two works rely on the Speech Acts Theory. This theory and how it can be used for designing agent communication languages is explained in [Labrou 01]. The idea is that an utterance, e.g. "I ask you to move", comprises of an argument (here the fact of moving), and an illocutory force relative to this argument (here asking for an action). The illocutory force is expressed in the utterance by a verb called the performative (here Ask). A small number of performatives exist (ask to = request to, ask if, asserting, deny, refuse, accept, etc.). From these performatives, it is possible to define conversation protocols. A conversation is defined by a sequence of performatives, whatever the argument. For instance, a request for an action can be followed by a refusal or an acceptation, whatever the action.

We store in the system pre-established scenarios of conversation, containing the sequence of exchanged performatives. When receiving a message, an agent thus refers to these scenarios in order to know what it can answer. Moreover, to enable the agent to choose the good answer among the possible ones according to its situation, we store for each message of a conversation:

- the name of the action the agent receiving the message has to perform
- the rule that enables to choose the next message to send according to the result of the action.

For instance, an agent receiving a request for an action has to perform an analysis of its capacity to do the action. If the analysis results in the conclusion that the agent is able to do the action, the agent sends an acceptation message and effectively performs the action. Otherwise, it sends a refusal message.

4. Results

4.1. Implemented system

The system described above has been implemented on the GIS LAMPS2 from Laser-Scan Ltd., which is object oriented. We generalise rural areas of the geographical database "BDTopo®" to produce a topographic map at scales 1:25 000 to 1:50 000, as explained in section 2.2 (the final scale is a parameter of the system).

For the time being, we only work with roads and buildings objects to test the proposed approach. In its present state, the system aims to solve the conflicts of symbol overlap between roads and buildings, and between buildings. The roads are considered fixed, only the buildings are mobile.

Three kinds of relational constraints are considered:

- non-overlap constraint between (1) symbols of roads and buildings, (2) buildings and buildings.
- proximity constraint: symbols of roads and buildings, and symbols of buildings between themselves must respect a given separability threshold (parameter of the system, 0.1mm on the map in our case).
- topology constraint: a building cannot "jump over a road".

Moreover, we consider an internal constraint of positional accuracy for the buildings, that prevents a building from moving too far away from its original position.

Only one dialog scenario is used: the "request action" scenario briefly presented in the previous section (request followed by an acceptation or a refusal).

The standard behaviour of road agents consists in pushing the buildings away from themselves, by asking them to move away. In the same way, the standard behaviour of building agents consists in pushing the other buildings away from themselves. The priority is given to the overlap conflicts between roads and buildings, i.e. buildings prefer moving away from roads, even if it means creating an overlap conflict with another building. On the other hand, a building never accepts to solve an overlap conflict with one road (or a building) if it means creating a conflict with another road.

Moreover, we add to the building agents a method stemming from the system of the AGENT project, that is also implemented on LAMPS2, so that they perform individual generalisation. Thus, in addition to their relational constraints, building agents take into account internal constraints of legibility: minimum size, level of detail of their contour, etc. They handle these constraints by simplifying, squaring and possibly dilating themselves.

4.2. Cartographic results

Figures 6, 7 and 8 show generalisation results obtained at scales 1:25 000, 1:35 000 and 1:50 000 respectively. Let us remind that the original data can be displayed without generalisation at 1:15 000. On each figure, the zones are displayed both enlarged and at real printing scale. Three zones of increasing complexity (increasing density of buildings) are considered. The three zones have been generalised at 1:25 000 and 1:35 000. At scale 1:50 000, only the zones (a) and (b) have been generalised: the zone (c) is too dense to be generalised by the system in its actual state.

At scale 1:25 000, all overlap conflict are solved. However, two slight degradations of the relative positions of buildings can be observed in the zone of highest density (c) (which is normal since no constraint has been designed for that yet).

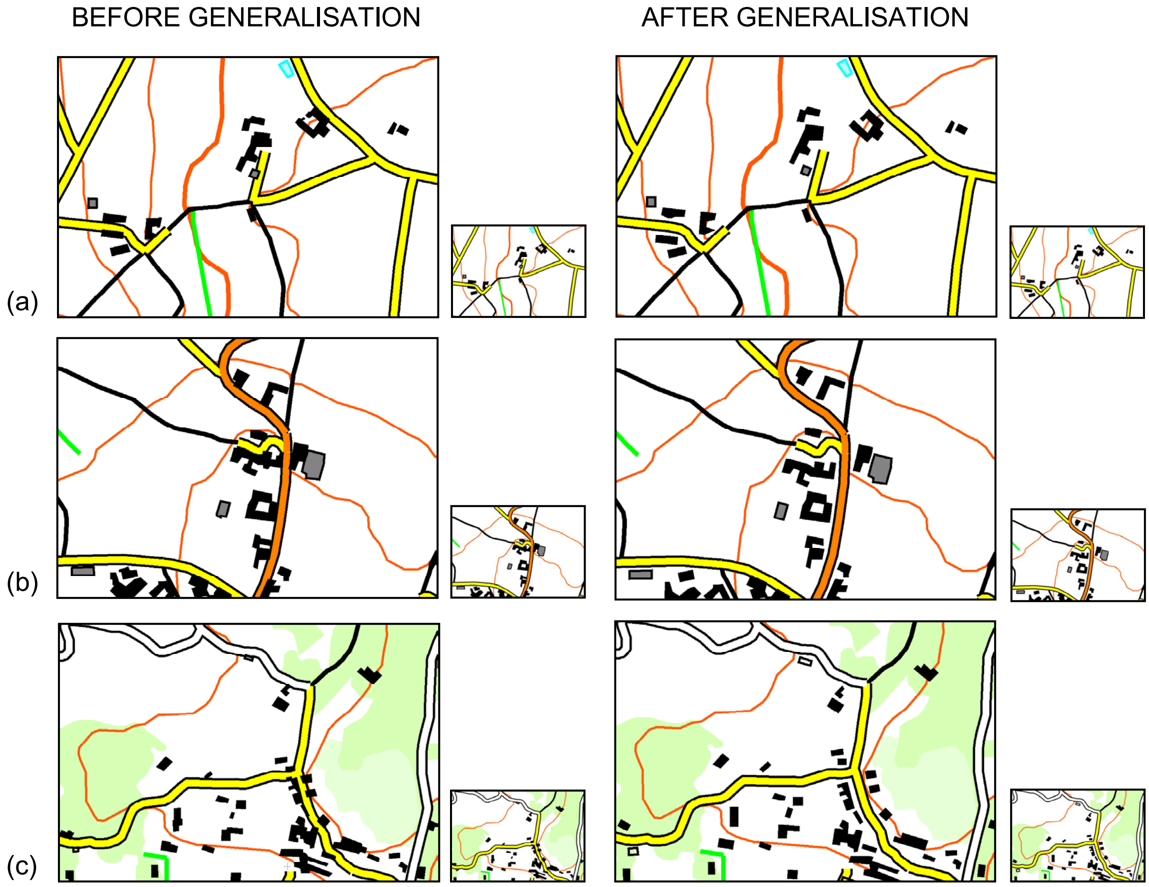

Figure 6. Cartographic results at scale 1:25 000

At smaller scales, there is obviously less space for each building, thus the situations are more constrained and the system has to find other solutions. At 1:35 000, the results on zones (a) and (b), of low density, are still correct. However, the limits of the system in its present state appear on the zone (c), of higher density: not all the overlap conflicts can be solved and, moreover, some relative positions are lost. This shows the need to consider additional constraints, such as constraints to maintain the relative positions, and the need to introduce new generalisation operations like objects removals.

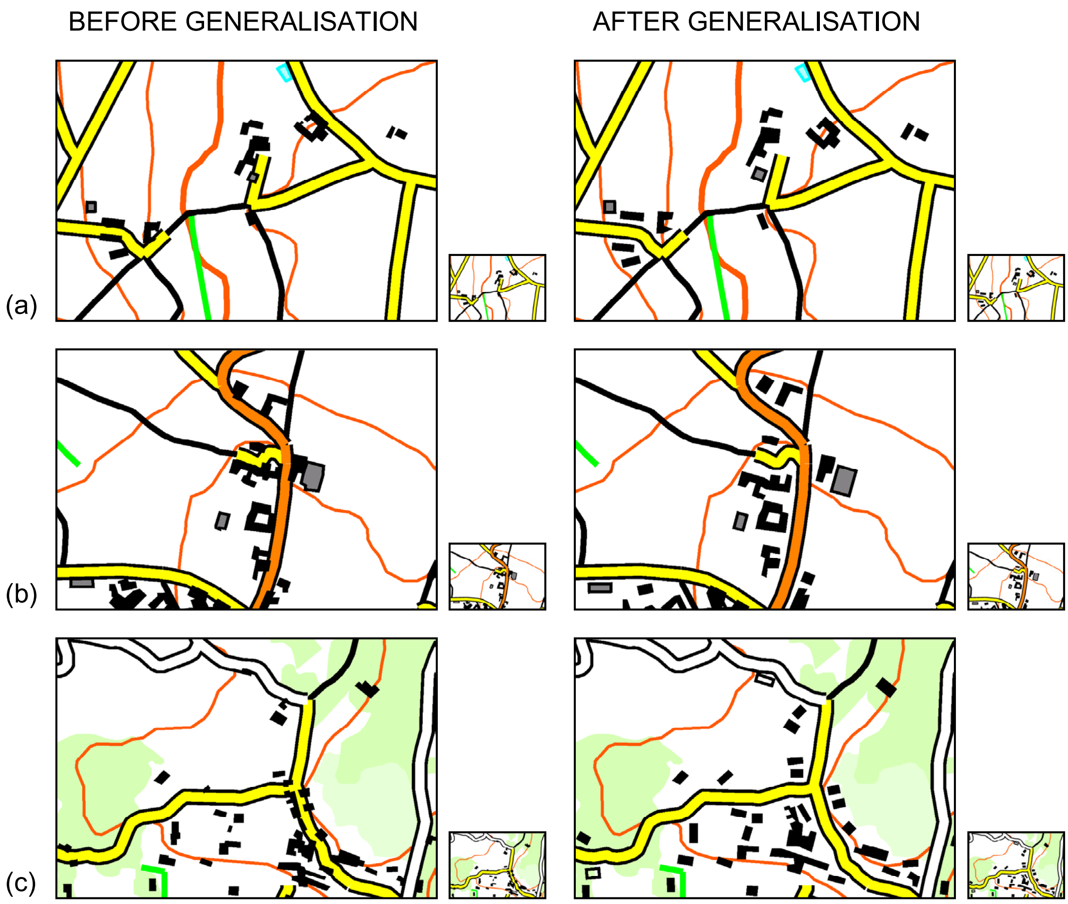

Figure 7. Cartographic results at scale 1:35 000

Finally, at scale 1:50 000 even on the less dense zones some degradations of the relative positions appear, although the overlap conflicts are solved and the results are still acceptable. There is also a problem with individual generalisation of one building on zone (b) but this is out of the scope of our work.

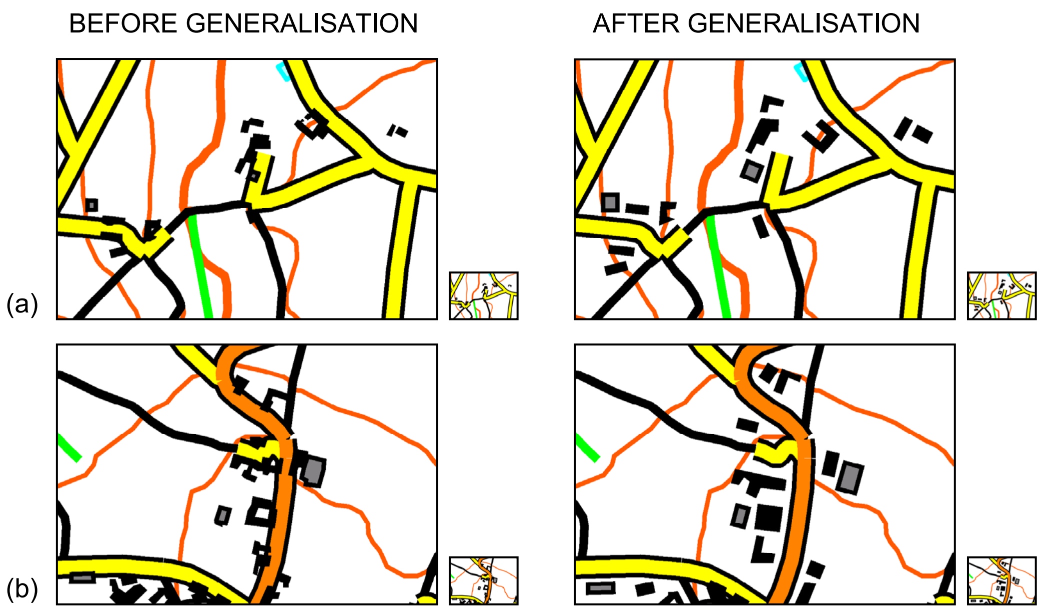

Figure 8. Cartographic results at scale 1:50 000

5. Conclusion and perspectives

The purpose of our work is to set up a system for automated cartographic generalisation, relying on bilateral interactions between 'geographical agents'. The implemented system shows the feasibility of the approach and its pertinence: the system works, and provides encouraging cartographic results, while only a few cartographic constraints have been taken into account. In zones of low density, the results are already cartographically correct. Moreover, it can be noticed that the system is adaptive, since it can perform generalisation for different display scales.

In terms of innovations, the representation of an agent's environment by means of Constrained zones is of great help to choose a position that satisfies several relational constraints at a time. Moreover, it is very encouraging to see that our system and the system stemming from the AGENT project can work together: it proves that the two systems are not incompatible. We can thus take advantage of results stemming from previous research.

In order to improve the cartographic quality of the results, we need to take more cartographic constraints into account: preservation of relative positions, exaggeration of relative orientations (parallelism, perpendicularity), etc. Some of these new constraints, namely those concerned with relative orientations, cannot be represented by means of *Constrained zones*. We are thus working on a new representation of the constraints, partially based, but not only, on the concept of *Constrained zones*.

Still in order to improve the quality of the results, we need to introduce new generalisation operations, already studied but not yet introduced in our system, like roads displacement or building removal. However, removing objects is a delicate operation, because the "most significant objects" must be kept. Thus, allowing buildings to eliminate themselves (commit "suicide") must be done very cautiously, e.g. by reserving it for very small buildings of regular nature (not the town hall!).

A next step in proving the interest of our approach is to introduce other geographic themes than roads and buildings. We plan to add objects from the hydrographic and landuse themes, that are the most present together with roads and buildings.

Finally, the approach of our system, using only bilateral interactions, is not adapted to all cases. It relies on a strong hypothesis that is not always true: that a network of bilateral conflicts within a set of objects can be solved by bilateral interactions. In some over-constrained situations, this approach is not optimal and it is needed to use a resolution by groups. This is why a future step of our work will be to combine more deeply our system to the one stemming from the AGENT European project, in order to use a pyramidal organisation of agents when needed. For that, we need to detect over-constrained situations. We plan to do it by recording and analysing the dialogs between agents.

References

AGENT Consortium 1998. Intermediate presentation to the European Commission, 1998.

Barrault M., Regnauld N., Duchêne C., Haire K., Baeijs C., Demazeau Y., Hardy P., Mackaness W., Ruas A., Weibel R. 2001. Integrating Multi-agent, Object-oriented, And Algorithmic Techniques For Improved Automated Map Generalization. *Proc. of the 20th International Cartographic Conference*, vol.3, Beijing, China, 2001, 2110-2116.

Brassel K. & Weibel R. 1988. A review and conceptual framework of automated map generalization. *International Journal of Geographic Information Systems*, 1988, vol.2, n°3, 229-244.

Labrou, Y 2001. Standardizing Agent Communication. *In Multi-Agent Systems and Applications*, 9th ECCAI Advanced Course, 3rd European Agent Systems Summer School (EASSS'01), Prague, 2001, Selected Tutorial Papers. M. Luck, V. Marik, O. Stepankova, R. Trappl, eds. LNAI n°2086, 75-97.

McMaster R. & Shea K. 1988. Cartographic Generalization in a Digital Environment: a Framework for implementation in a GIS. *Proceedings of GIS/LIS'88*, San Antonio, Texas, États-Unis, 1988, 240-249.

Ruas A. 1998. Method for buiding displacement in automated map generalisation. *International Journal of Geographical Information Science*, vol.12, n°8, 789-803.

Ruas A. 1999. *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie.* Thèse de doctorat, université de Marne-la-Vallée, 1999.

Weiss G. 1999. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press.