The CartACom model : a generalisation model for taking relational constraints into account.

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

Keywords: cartographic generalisation model, relational constraints, communicative agents.

1. Introduction

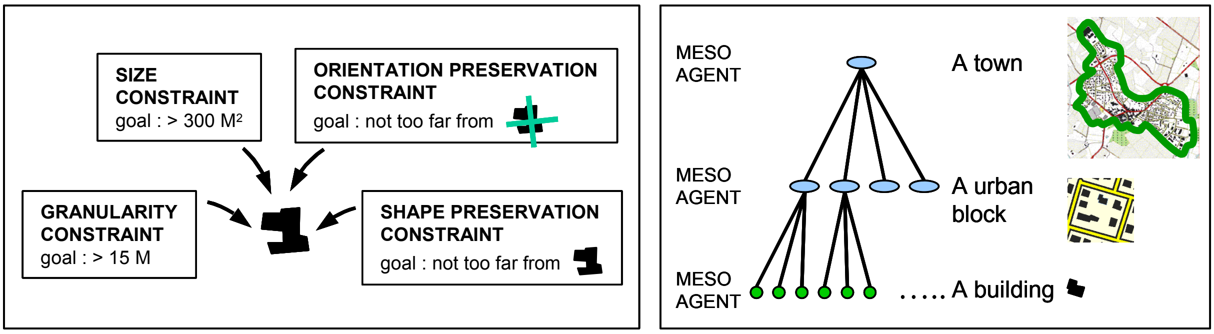
In previous papers [Duchêne 2003a; Duchêne 2003b] we presented a generalisation model relying on communicating agents. The general approach of generalisation it uses is localised, step by step, as proposed by [Brassel and Weibel 88; McMaster and Shea 88; Ruas 99]. This model, now called "CartACom" for "Cartographic generalisation with Communicating Agents", has been improved and made more generic. It still relies on communicating agents, but now it also contains a generic framework to represent cartographic constraints that concern a relation between two agents. It takes into account the operation of elimination, that requires specific management. And it finally provides a hook to handle a geographic object or group of objects by using an external generalisation process, relying on any approach of generalisation. In this paper we present the general architecture of the CartACom model and we focus on the three major improvements mentioned above.

The paper is structured as follows. Part 2 briefly reminds the context and objectives of our work. Part 3 presents the CartACom model and focuses on its most recent improvements. Part 4 shows results obtained by applying the CartACom model to the generalisation of rural topographical data. Finally, part 5 concludes and draws some perspectives of this work.

2. Context and objectives

The context of this work is the generalisation model used in the European project AGENT [Barrault et al. 2001], stemming from [Ruas 1999]. In this model, the geographic objects of the database to generalise are modelled as agents, thus becoming 'geographic agents'. The concept of agent [Weiss 1999] is part of the artificial intelligence domain. An agent can be thought of as an object that has a goal and acts autonomously in order to reach this goal thanks to capacities of perception, deliberation, action, and possibly communication with other agents. In the model used in the AGENT project, each geographic agent is able to identify and assess its internal cartographic constraints, and to apply generalisation algorithms to itself in order to satisfy as well as possible these constraints. Examples of internal

constraints taken into account are, for a building, constraints of size, granularity, squareness, preservation of shape, etc. (Figure 1a).
Now, in generalisation, some constraints can involve more than one geographical object, e.g. the constraint that prevents symbols from overlapping each others, or the constraint that requires aligned buildings to remain aligned. To handle constraints involving several agents, the AGENT model considers several nested levels of agents. The individual agents are called *micro agents*. Any group of agent that needs to be considered as a whole for the purpose of generalisation is explicited as an upper level agent, called *meso agent*. Several nested meso agents can exist (Figure 1b).



(a) A building agent guided by its constraints

(b) Different nested levels of agents

Figure 1. Principles used in the AGENT model

Every constraint involving two or more agents is handled by the meso agent containing these agents. For instance, overlapping conflicts between buildings are handled by the urban block that contains these buildings. During generalisation agents only interact hierarchically, e.g. a meso agent can give an order to one of the agents it contains. No interaction occurs between agents of a same level. This supposes that no agent belongs to two meso agents of the upper level – i.e. meso agents inside a level are disjoint.
Although the AGENT model has proved to give good results, especially for the generalisation of urban areas, the fact that agents cannot have transversal interactions has been identified as a limit. It especially appears in rural area, which are less dense but more heterogeneous in terms of geographic themes and spatial configurations, and where it is difficult to identify disjoint and similarly composed groups of agents to be modelled as meso agents. The objective of our work is thus to set up a new model of coordination of the agents, relying on transversal interactions (Figure 2). The two models are intended to be used together in a medium term.

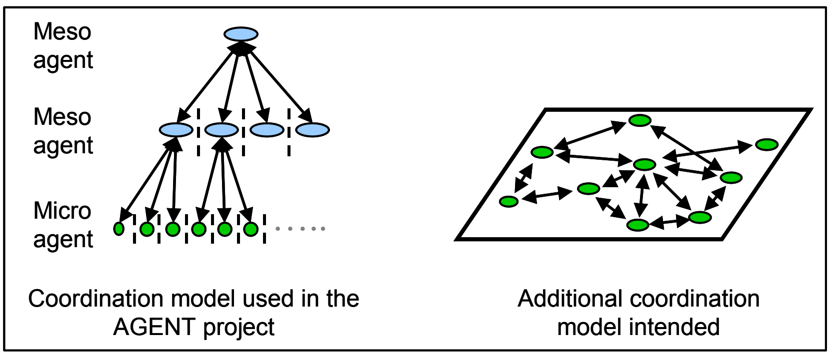


Figure 2. Objective of our work: set up a model where agents can have transversal interactions

The rest of the paper presents the newly set up model, called CartACom, as well as some results obtained by applying this model to the generalisation of topographical data from rural areas.

3. The CartACom model: major aspects

3.1. Principles of the CartACom model

Before describing the CartACom model, we describe hereafter the three main principles it relies on.

Consider relational constraints

The first principle on which the CartACom model relies concerns the kind of cartographic constraints considered. Let us remind that the CartACom model is intended for geographic spaces that are not structured by obvious disjoint groups of objects, but where on the contrary cartographic constraints are more local. Thus, in the CartACom model, we consider the *relational constraints*. What we call a relational constraint is a constraint concerning a relation between two agents. Of course an agent can have relational constraints with several other agents. We identify three kinds of relational constraints (Figure 3): legibility constraints include the constraint that prevents symbols from overlapping, and constraints that prevent a relation from being nearly present (a); constraints of preservation aim to preserve certain relations between objects (b); and constraints of geographic coherence concern particular relations that make sense geographically speaking (c), for instance if a road serves a viewpoint it should be kept unless the viewpoint is removed. CartACom includes a generic model for representing relational constraints, which is presented in 3.3.

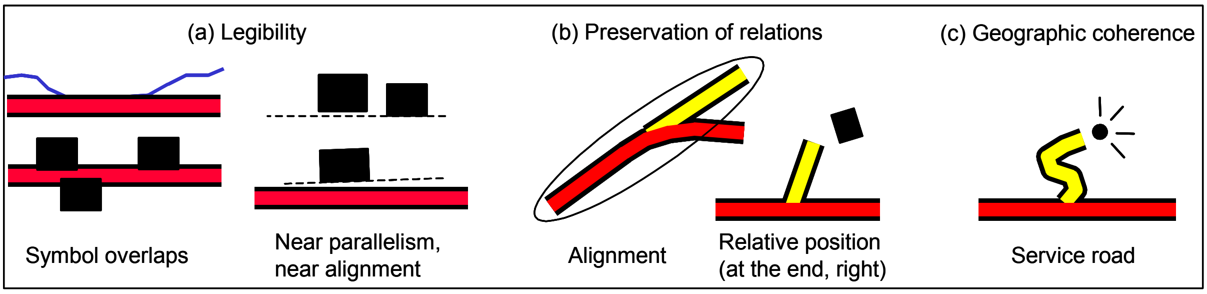


Figure 3. Relational constraints

How the agents interact

The second principle of the CartACom model is relative to the kinds of interactions the agents can have during the generalisation process. In order to identify and assess their relational constraints, the agents are provided with the capacity of perceiving their spatial environment, i.e. the surrounding space and the neighbouring agents (Figure 4a). This kind of interaction – perception – is not sufficient to enable an agent to choose the right generalisation algorithms to apply to itself : an agent often needs to know not only about its own relational constraints, but also on the constraints of its neighbours. That's why our agents are also provided with capacities of communication (Figure 4b).

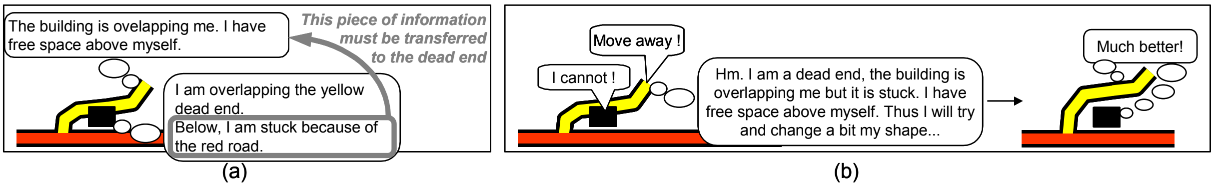


Figure 4. Our agents interact in two ways : by seeing their environment and talking to each others.

The model used by agents for communication, based on the Speech Acts theory [Austin 1962; Searle 1969], has been described in [Duchêne 03a; Duchêne 03b].

Dynamic aspects of the model

The third main principle of CartACom is relative to the dynamic aspects of the system, as well intra-agent (how an agent chooses and chains its actions), as inter-agents (how are agents activated). The intra-agent dynamic, that relies on the notion of task, will be briefly presented in the next section. Concerning the inter-agent point of view, the principle is that agents are activated in turn, and every time an agent is activated, all the transformations it has done are immediately perceivable by its neighbours. This is in order to maintain the system coherent, taking into account the strong dependencies that exist between actions in generalisation.

3.2. General architecture and functioning of an agent

Figure 5 presents the conceptual model of a CartACom agent. Knowledge maintained by the agent about itself and its environment are organised in several modules, figured by grey ellipses. Arrows betweens ellipses represent the interactions between those modules.

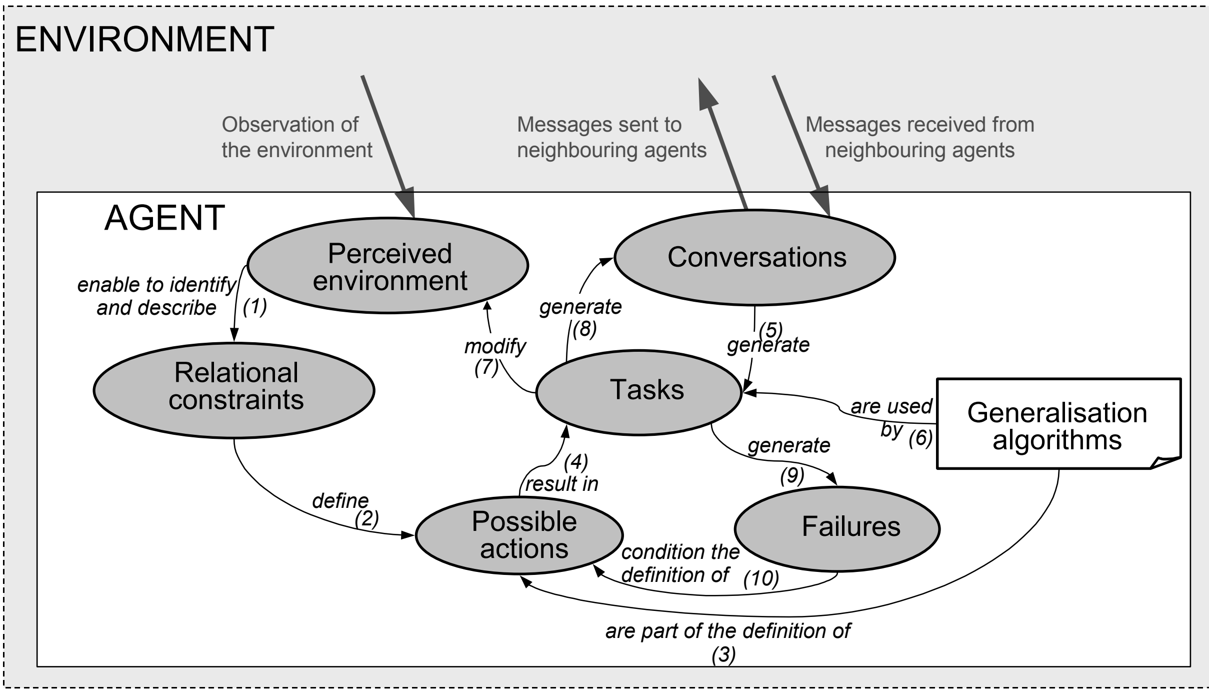


Figure 5. General architecture of an agent in CartACom

By *observing* its environment, the agent builds up a representation of this *perceived environment*. It can then identify the *relational constraints* it shares with its neighbours, and compute their satisfaction (1). The proactive behaviour of the agent makes him define *possible actions* to increase the satisfaction of these constraints, depending of their nature and their state (2). A possible action can consist in applying a *generalisation algorithm* to itself, or in asking the other agent sharing the constraint to apply an algorithm to itself (3). Once the agent has identified its possible actions, it chooses one of them and tries it. This chosen action gives birth to a *task* (4). The notion of *task* is used to represent an action which is in progress or about to begin. It enables the agent to interrupt an action (e.g. because it has to wait for an answer), while keeping a representation of the progress state of this action. Not all the tasks of the agent are directly generated by its constraints: as a communicative agent, the agent also receives messages that are part of *conversations*. Depending on the progress of a conversation, a received message which is part of this conversation can call for the realisation of an action, that the agent then also represents by a task (5). During the execution of a task, the agent may transform itself by applying a generalisation algorithm to itself (6). In that case, as a consequence the environment perceived by the agent is modified as well (7). For instance, if the agent moves or dilates itself, some of its neighbours appear closer to it. During the execution of a task, the agent can also need to begin a conversation with a neighbour (8). Finally, when the execution of a task leads to a *failure*, this failure is stored (9). The storage of failures enables to filter the possible actions (10), avoiding to infinitely try an action that has already failed.

3.3. Generic model used to represent relational constraints

This section presents the generic model set up to represent relational constraints in CartACom. The underlying idea is the following. A relational constraint concerns a relation between two agents. What is constrained is the relation, which means that the relation must have some given properties. This constraint on the relation as a consequence constrains the behaviour of both of the concerned agents (Figure 6a). E.g. for a proximity constraint, the distance between two agents must be greater than a given threshold. This leads each of the agents to try to keep further from the other than this threshold.

In the CartACom model, two generic classes are used to represent the relational constraints between geographic agents: a class called *Relation* (abbreviation of 'constrained relation') and a class called *Constraint* (abbreviation of 'relational constraint'), as shown on Figure 6b.

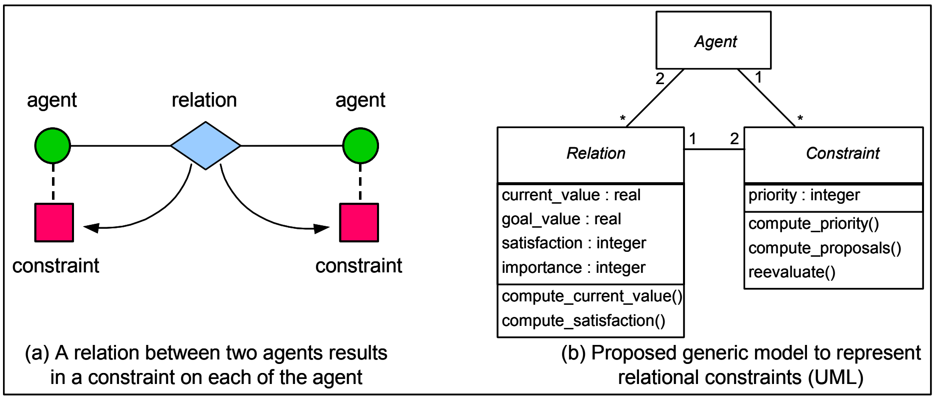


Figure 6. Modelling a relational constraint

Any relational constraint between two agents A and B is represented by three objects:
− a *Relation* object, linked to A and B, that describes the state of the relational constraint,

– and two *Constraint* objects : each of them is linked to the *Relation* object on the one side and to one of the agents on the other side, and represents the way this agent manages the relational constraint.

The attributes and methods of those classes are described below.

The class *Relation* bears the attributes *current_value* (result of a measure describing the relation, e.g. a distance), *goal_value* (what the current value should be), *satisfaction* (how satisfied is the constraint, i.e. how close the current value is from the goal value) and *importance* (how important it is according the specifications that this constraint is satisfied, on an absolute scale shared by all the constraints). It also bears the methods to compute the *current_value* and the *satisfaction*.

The class *Constraint* bears an attribute *priority* that indicates how urgent it is for the agent to try and satisfy this constraint (compared to its other constraints). The priority is computed by the method *compute_priority*, which takes into account the current satisfaction of the constraint. Thus the priority changes dynamically during the process. The class *Constraint* also bears a method *compute_proposals*, that computes a list a possible actions that might help to better satisfy the constraint, and a method *reevaluate*, that after a transformation assesses if the constraint has changed in a right way (if it has been enough improved, or at least if it has not been too much damaged).

The attributes and methods born by the classes *Relation* and *Constraint* in CartACom are very similar to the attributes and methods of the class *Constraint* in the model of [Ruas 1999]. In fact we propose an adaptation of this model in order to manage the fact that the constraints we consider are shared by two agents, instead of being internal to an agent. This adaptation consists in splitting the representation of a relational constraint in two parts :

– the first part is relative to the objective description of the state of the relational constraint, which is identical from the point of view of both agents and can thus be shared by them. This description is born by the *Relation* object linked to both agents,
– the second part is relative to the analysis and management of the constraint, which is different for each agent and should thus be described separately for each of them. This part is described by the to *Constraint* objects.

Figure 7 illustrates the representation of two relational constraints of proximity involving three agents: one road agent R1 and two building agents B1 and B2. The class *Proximity Relation Road-Building* inherits from the class *Relation*. The classes *Proximity constraint Road←Building* and *Proximity constraint Building←Road* inherit from the class *Constraint*. The fact they are different enables to take into account the fact that a building lives its constraint of proximity with a road in a different way as the road lives its constraint of proximity with the building.

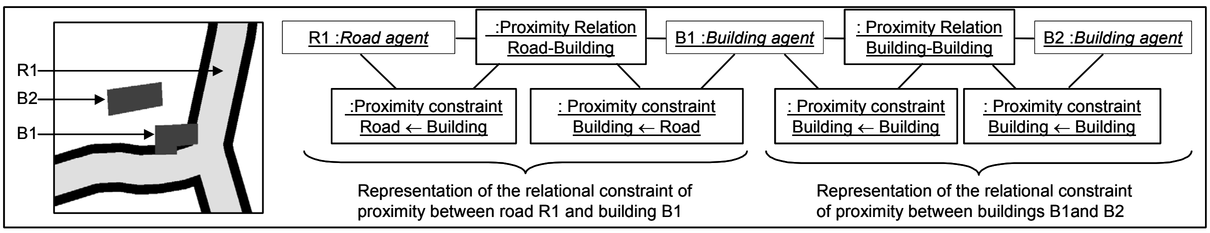


Figure 7. Example of modelled relational constraints: two constraints of proximity, involving three agents. The diagram is a UML object diagram (each box represents an object)

The relational constraints are initialised by the agents themselves in a stage of initialisation, prior to generalisation. The way they are identified and instantiated is described in [Duchêne

2003b]. Then any agent that modifies itself is in charge of updating the *Relation* objects it is linked to as well as its own *Constraint* objects. It then informs the agents sharing the relational constraints of its change, so that they can update their relevant *Constraint* objects the next time they are activated. This way each agent has an up-to-date view of the system.

3.4. Self-elimination of an agent: a specific operation

Elimination is required in generalisation as soon as the density of objects or the change of scale are important. However it must be done cautiously in order to prevent the elimination of 'important' objects, important objects being for instance objects with a particular value of attribute (e.g. the town hall), or simply objects that are exceptions among their surroundings regarding some characteristics. This is why, in the AGENT model, an elimination is always decided at the meso level, that has a global view of a situation. In CartACom, the fact that there is no meso level leads to make agents responsible for the decision of eliminating themselves. The developer who adapts CartACom for a particular generalisation problem is responsible for fixing the conditions under which an agent can decide to eliminate itself. If constraints of geographic coherence (see Figure 3c) are defined, they should be taken into account.

In CartACom, even once an agent has decided to eliminate itself, the elimination is not straightforward because agents do interact with each others. Thus, the agent that decides to eliminate itself is surely sharing relational constraints with other agents, and might even have conversations in progress with some of them at the time where it decides to eliminate itself. If the agent simply eliminates itself without warning the other agents, the system might enter an incoherent state, e.g. with agents waiting for an answer they will never receive. Thus, before eliminating itself, the agent sends a message to all the agents sharing a constraint with it, in order to inform them that it is about to eliminate itself. The agents that receive such a message can then update their mental state accordingly.

3.5. Punctual use of external operations

The last aspect of the CartACom model we would like to discuss here is the possibility to punctually use an external operation on a group of geographical objects (agents) during the execution of the CartACom generalisation process as illustrated in Figure 8. The CartACom model is based on transversal, bilateral interactions. This assumes a strong hypothesis: that a network of bilateral constraints can be solved by using only bilateral interactions. This hypothesis is sometimes false: in over constrained situations, the best solution is often to handle the group of over constrained agents as a whole – assuming an algorithm dedicated to this kind of situation exists.

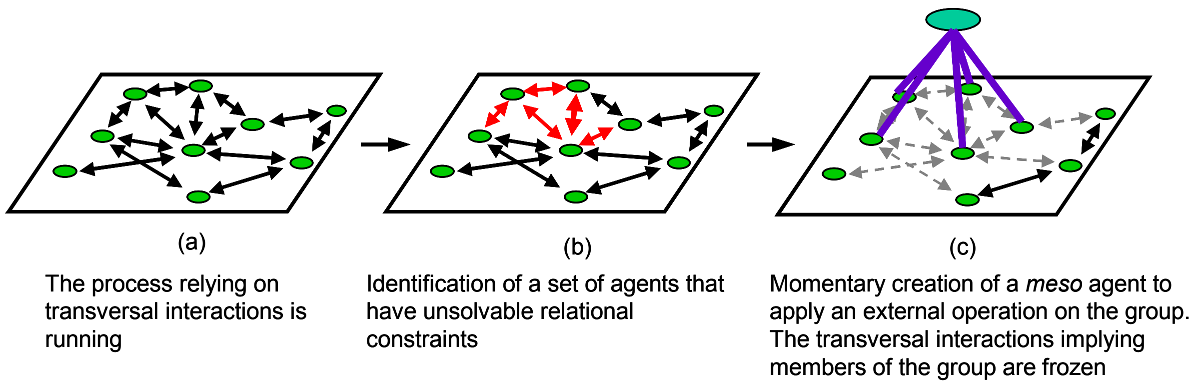


Figure 8. Punctual use of an external operation on a group of agent during the CartACom process

To enable the use of an external operation, two elements are necessary:

(1) Be able to detect a group of agents which could be candidate to an external operation. It means (1a) detecting groups of agents sharing relational constraints that could not be satisfied after all the possible actions have been tried; (1b) theses groups can then be modified by addition/removal of agents or by merging, in order to obtain a group for which a given operation is suitable.

(2) After the execution of this external operation, manage its consequences on the other agents so that the system stays in a coherent state – i.e. the surrounding agents must be aware of the modifications or eliminations that have been made, in order to keep their mental state up to date.

Not all of this process has been implemented for the time being. Part (1a) is generic and is currently being implemented. Part (1b) depending on the available external group operation. Part (2) has been implemented and tested by interrupting the system manually and performing interactive modifications/eliminations.

4. Applying CartACom to generalisation of rural topographical data

4.1. Agents, constraints and operations considered

To test CartACom, we have used topographical data stemming from rural zones of the BD TOPO, the 1m resolution database of the French National Mapping Agency. The reference scale of this database is about 1:10 000-1:15 000. The generalisation is used to produce maps at 1:25 000-1:50 000. The considered agents are roads, buildings, rivers, railways and footpaths. Other themes, like land use, are not generalised but they are modelled as simpler agents, that just follow the other agents to which they are topologically connected. This is important for updating purposes in a global cartographic process.

The considered relational constraints are the following:

- Constraint of proximity (to respect the separability threshold) between roads and buildings, rivers and buildings, railways and buildings, footpaths and buildings, buildings between each others.
- Constraint of preservation of relative positions between roads and buildings, rivers and buildings, railways and buildings, footpaths and buildings.
- Constraint of exaggeration of parallelism between roads and buildings, rivers and buildings, railways and buildings, footpaths and buildings.

Moreover, we consider on the buildings an internal constraint of planimetric accuracy, that prevents buildings to move too far away from their initial position.

The allowed generalisation operations are displacement, rotation and elimination for buildings. The roads, rivers, railways and footpaths don't displace themselves. We also enable roads and buildings to perform internal generalisation thanks to a hook with the prototype stemming from the AGENT project.

4.2. Example of cartographic results

Figure 9 shows results obtained at scale 1:25 000 on a rural zone with locally quite a high density of buildings.

Here the criterion defined for a building to eliminate itself is the following: a relational constraint of proximity exists with another building that is at least 3 times greater that it, and this relational constraint has failed to be solved by a displacement of any of the buildings. Two small buildings have performed self-elimination according this criterion (Figure 9b – regions surrounded by circles).

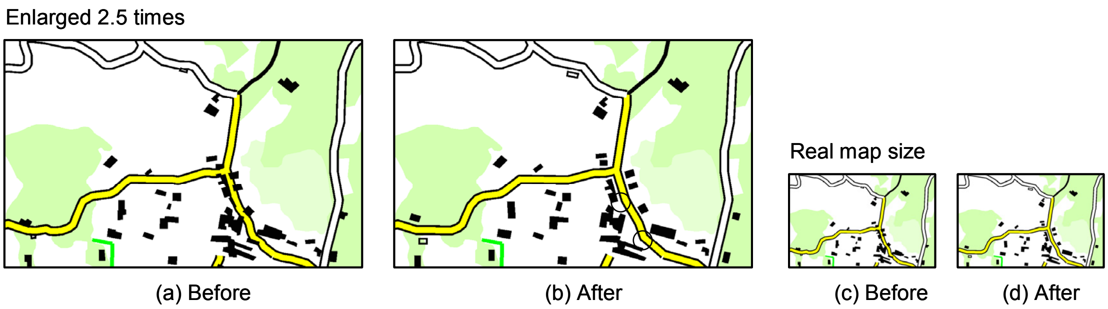


Figure 9. Result obtained at scale 1:25 000 on a rural zone with quite a high density of buildings. The two circles on picture (b) enhance two self-eliminated buildings.

Other results obtained at scales 1:25 000, 1:35 000 and 1:50 000, on the same zone and on two additional zones, are shown on Figure 10. These results can be considered as globally correct from a cartographic point of view. However, the more the zone is dense or the final scale is small (i.e. at the bottom right of the picture), the more it occurs that conflicts of proximity/overlapping remain (detected by the system), and the more conflicts of relative sizes and relative positions between buildings appear (detected by visual assessment). It shows the need of considering constraints of relative sizes and positions between buildings. It also shows the limits of the rough CartACom model relying only on bilateral operations: for most of the over constrained situations appearing here, using external operations as explained in section 3.5, which is not yet possible since the implementation is not complete, would probably help a lot to solve the remaining proximity conflicts.

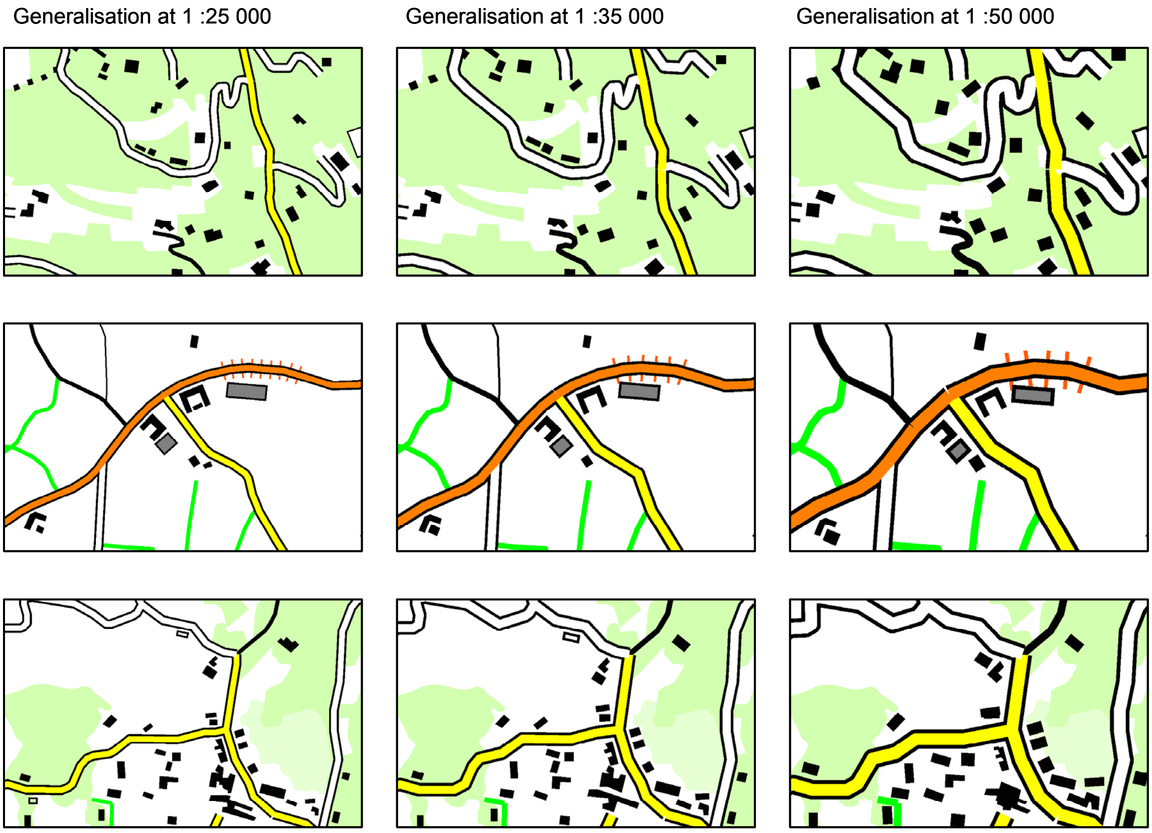


Figure 10. Results obtained on various zones at scales 1:25 000, 1:35 000 and 1:50 000 (display enlarged to 1:10 000).

5. Conclusion and perspectives

In this paper, we have presented the CartACom model of generalisation, based on transversal interaction between agents in order to satisfy relational cartographic constraints. We have emphasized three newly improved aspects of the model: the generic model to represent relational constraints between agents, the operation of self-elimination of an agent, and the possibility to use an external operation on a group of agents (this last one is only partially implemented for the time being). We have presented some cartographic results obtained with CartACom on real data. These results are globally satisfying but they also show the need to take more constraints into account and to complete the implementation of the module dedicated to the use of external operations.

The model proposed by [Ruas 1999] and used in the AGENT project enables to generalise a geographical object (internally), or a group of objects thanks to the meso level. The CartACom model adds the possibility to generalise a geographical object with regard to its spatial environment. The two models are thus complementary. We have already shown two ways of combining them: (1) by using the AGENT model, like a black box, to perform individual generalisation of roads and buildings; (2) by proposing a protocol to use an external operation on a group during the execution of CartACom. This second point is worth for any operation on a group, be it a meso process of the AGENT prototype or any other algorithm, possibly using another approach of generalisation (for instance a global approach like least square [Harrie & Sarjakoski 2002] or finite elements [Bader 2001]).

A next step in our research will consist in combining CartACom more deeply with the AGENT model, in order to make agents able to handle their internal and relational constraints at a time.

Another step, maybe the most important one, would consist in trying to combine CartACom, the AGENT model and other approaches of generalisation in order to generalise a whole set of data, using each approach for the kinds of situations it is adequate. For instance, using CartACom for low density, heterogeneous zones like rural space, using the AGENT model for urban areas, a process based on elastic beams [Bader 2001] for the generalisation of the road network once pruning has been performed, etc. This would enable to better study how several approaches can be used in a synergic way, and hopefully to identify some key elements that are still missing.

References

Austin J.L. 1962. *How to Do Things With Words*. Clarendon Press. Traduction française *Quand dire, c'est faire*, le Seuil, 1970.

Bader M. 2001. Energy Minimization Methods for Feature Displacement in Map Generalization. Dissertation zur Erlangung der Doktorwürde, Mathematisch-naturwissenschaftliche Fakultät, Universität Zürich, 2001.

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.

Duchêne C. 2003a. Coordinative agents for automated generalisation of rural areas. *5th ICA Workshop on progress in automated map generalisation,* Paris, 2003, available on the web site of the ACI commission for generalisation :

http://www.geo.unizh.ch/ICA/docs/paris2003/papers03.html

Duchêne C. 2003b. Automated Map Generalisation using Communicating Agents. In *Proceedings of the 21th International Cartographic Conference*, Durban, 2003, p.160-169.

Harrie L. & Sarjakoski T. 2002. Simultaneous Graphic Generalization of Vector Data Sets. *GeoInformatica*, vol.6, n°3, 09/2002, pp.233-261.

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

Searle J. 1969. *Speechs acts*. Cambridge University Press, 1969.

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