Combining Three Multi-agent Based Generalisation Models: AGENT, CartACom and GAEL

Cécile Duchêne, Julien Gaffuri

IGN, COGIT Laboratory, 2-4 avenue Pasteur, 94165 Saint-Mandé cedex, France.
email: {cecile.duchene,julien.gaffuri}@ign.fr

Abstract

This paper is concerned with the automated generalisation of vector geographic databases. It studies the possible synergies between three existing, complementary models of generalisation, all based on the multi-agent paradigm. These models are respectively well adapted for the generalisation of urban spaces (AGENT model), rural spaces (CartACom model) and background themes (GAEL model). In these models, the geographic objects are modelled as agents that apply generalisation algorithms to themselves, guided by cartographic constraints to satisfy. The differences between them particularly lie in their constraint modelling and their agent coordination model. Three complementary ways of combining these models are proposed: separate use on separate zones, "interlaced" sequential use on the same zone, and shared use of data internal to the models. The last one is further investigated and a partial re-engineering of the models is proposed.

Keywords: Automated generalisation, Multi-agent-systems, Generalisation models, Models combination.

1. Introduction

In this paper, we deal with automated cartographic generalisation of topographic vector databases. Cartographic generalisation aims at decreasing the level of detail of a vector database in order to make it suitable for a given display scale and a given set of symbols, while preserving the main characteristics of the data. It is often referred to as the derivation of a Digital Cartographic Model (DCM) from a Digital Landscape Model (DLM) (Meyer 1986). In the DCM, the objects have to satisfy a set of constraints that represent the specifications of the expected cartographic product (Beard 1991; Weibel and Dutton 1998). A constraint can be related to one object (building minimum size, global shape preservation), several objects (minimum distance, spatial distribution preservation), or a part of object (road coalescence, local shape preservation). Different approaches to automate generalisation handle the constraints expression in different ways. For instance, in approaches based on optimisation techniques (Sester 2000; Højholt 2000; Bader 2001), the constraints are translated into equations on the point coordinates.

The work presented in this paper relies on an approach of generalisation that is step by step, local (Brassel and Weibel 1988; McMaster and Shea 1988), and explicitly constraint driven (Beard 1991). More precisely, our work is concerned with three complementary models based on this approach, which also rely on the multi-agent paradigm. These three models are respectively dedicated to the generalisation of dense, well-structured data (AGENT model), low density, heterogeneous zones (CARTACOM model), and to the management of background themes during generalisation (GAEL model). The purpose of this paper is to investigate the possible synergies between the three models.

The next section of the paper presents in a comparative way the major aspects of the AGENT, CARTACOM and GAEL models. In section 3, three complementary scenarios for a combined use of these models are proposed, and the underlying technical requirements are identified. One of them is further investigated in section 4, where a partial re-engineering of the models is proposed. Finally, section 5 concludes and draws some perspectives for on-going work.

2. Comparative presentation of AGENT, CartACom and GAEL

2.1. The AGENT model

The AGENT generalisation model has first been proposed by Ruas (1998, 2000). It has then been used and enriched during the European AGENT project (Barrault et al 2001).

In the AGENT model, objects of the database to generalise are modelled as agent, i.e. autonomous entities that try to reach a goal thanks to capacities of perception, deliberation, action, and communication (Weiss 1999). Two levels of agents are considered. A *micro* agent is a single geographic object (e.g. road segment, building). A *meso* agent is composed of micro or meso agents that need to be considered together for generalisation (e.g. a group of aligned buildings, a urban block). This results in a pyramidal hierarchical structure where agents of one level are disjoints. Cartographic constraints can be defined for each agent (Figure 1). If a cartographic constraint concerns several agents it is translated into a constraint on the meso agent they are part of, thus a constraint is always internal to an agent.

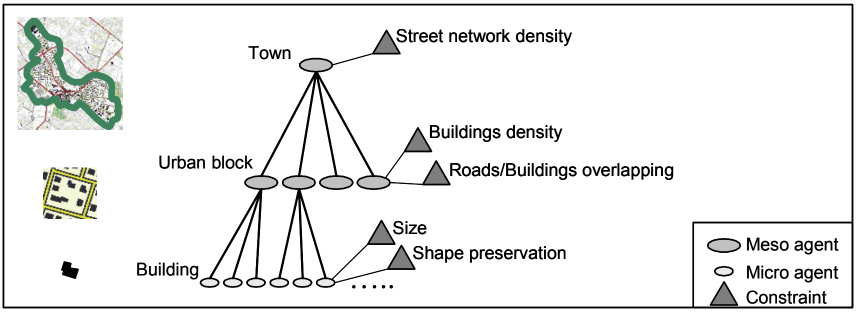

Fig. 1. The AGENT model: agents and constraints

The constraints are modelled as objects. A constraint object can be thought of as an entity, part of the "brain" of an agent, in charge of managing one of its cartographic constraints. In terms of data schema (cf Figure 2a), a generic *Constraint* class is defined, linked to the generic *Agent* class. The attributes defined on the *Constraint* class are as follows:

- *current_value*: result of a measure of the constrained property (e.g. area, for the building size constraint). It is computed by the *compute_current_value* method,
- *goal_value*: what the current value should be,

- *satisfaction*: how satisfied the constraint is, i.e. how close the current value is from the goal value. It is computed by the *compute_satisfaction* method,
- *importance*: how important it is according the specifications that this constraint is satisfied, on an absolute scale shared by all the constraints,
- *priority*: how urgent it is for the agent to try and satisfy this constraint, compared to its other constraints. It is computed by the *compute_priority* method depending on the satisfaction

Two additional methods are defined:

- *compute_proposals*: computes a list a possible plans (generalisation algorithms) that might help to better satisfy the constraint, and
- *re-evaluate*: 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)

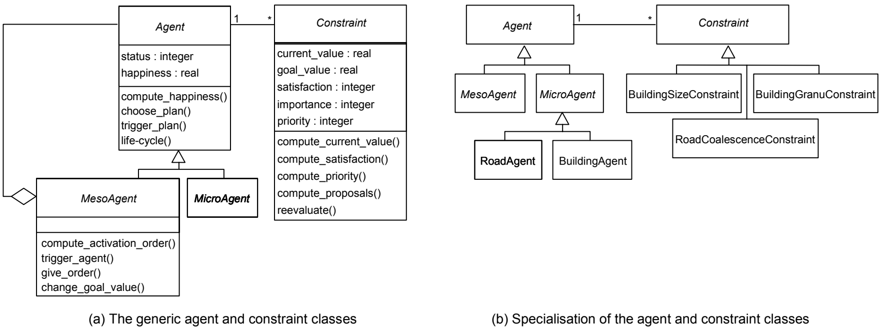

Fig. 2. AGENT static model : data schema

The generic *Constraint* class is specialised into several specific constraints classes, one for every kind of cartographic constraint (cf Figure 2b). One agent is linked to one constraint object of every specific constraint class that is relevant to its geographic nature (e.g. for a building, BuildingSizeConstraint, BuildingShapeConstraint, etc.).

When a geographic agent is activated, it performs a life-cycle where it successively chooses one plan among those proposed by its constraints, tries it, validates its new state according to the constraints re-evaluation, and so on. The interactions between agents are hierarchical: a meso agent triggers its components, gives them orders or changes the goal values of their constraints (Ruas 2000).

The AGENT model has been successfully applied to the generalisation of hierarchically structured data like topographical urban data (Lecordix et al 2007) and categorical land use data (Galanda 2003).

2.2. The CARTACOM model

The CARTACOM model has been proposed by Duchêne (2004) to go beyond the identified limits of the AGENT pyramidal model. It is intended for data where no obvious pyramidal organisation of the space is present, like topographical data of rural areas. In this kind of situation, it is difficult to identify pertinent disjoint groups of objects that should be generalised together, and constraints shared by two objects are difficult to express as an internal constraint of a meso object.

In CARTACOM, only the micro level of agents is considered, and agents have direct transversal interactions between each other. CARTACOM focuses on the management of constraints shared by two micro agents, that we call *relational constraints*. Examples of relational constraints are, for a building and a road, constraints of non overlapping, relative position, relative orientation.

The object representation of the constraints proposed in the AGENT model has been adapted to the relational constraints, which are shared by two agents instead of being internal to a single agent (Fig. 3). Two classes instead of one are used to represent the constraints: *Relation* and *Constraint*. The representation of a relational constraint is split into two parts:

- the first part is relative to the objective description of the state of the constrained relation, which is identical from the point of view of both agents and can thus be shared by them. This description is supported by a *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 separately described for each of them. This part is described by two *Constraint* objects: one for each agent sharing the relational constraint.

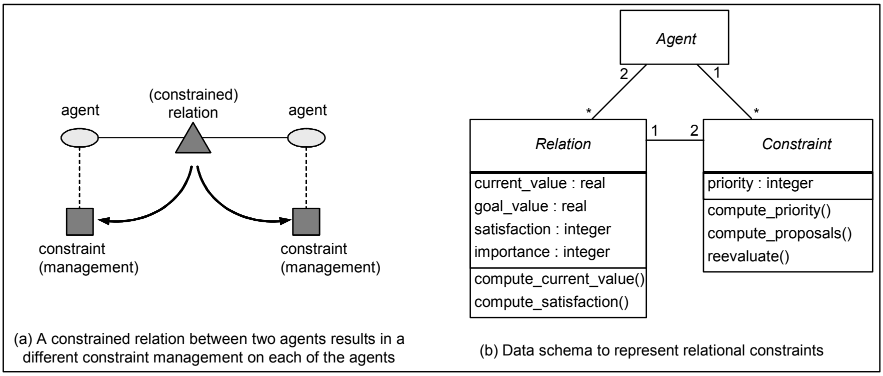

Fig. 3. CARTACOM static model: agents and constraints

In order to improve the state of a relational constraint, in CARTACOM an agent can use two kinds of “plans”: either apply to itself a generalisation algorithm, like in AGENT, or ask the other agent sharing the constraint to apply an algorithm to itself.

When activated, an agent performs a life-cycle similar to the AGENT life-cycle. If AGENT internal constraints have been defined on the agent on top of its CARTACOM relational constraints, the agent can perform its internal generalisation through a call to the AGENT life-cycle, which is then seen as a black box. In the case where the agent asks another agent to perform an action, it ends its life-cycle with a “waiting” status, and resumes action at the same point when it is next activated. The agents are activated in turn by a scheduler. Sending a message to another agent places it on the top of the scheduler’s stack, i.e. the agents trigger each others by sending messages.

The CARTACOM model has been successfully applied to low density, rural zones of topographical data, where the density is such that few contextual elimination is necessary (Duchêne 2004).

2.3. The GAEL model

The GAEL model has been proposed by Gaffuri (2007). Its is intended for the management of the background themes like relief or land use, during an agent generalisation of “foreground” topographic themes by means of the AGENT or CARTACOM model. The background themes differ from the foreground themes in that they are continuous (defined everywhere in the space) instead of being discrete and, from a generalisation point of view, they are more flexible than the foreground themes (thus they can absorb most of the transformations of the foreground themes). Two types of cartographic constraints are considered in the GAEL model: constraints of shape preservation internal to a field theme, and constraints that aim to preserve a relation between a foreground object and a part of a background field (object-field constraint). An example of an object-field constraint is, for a river and the relief, the fact that the river has to remain in its drainage channel.

In the GAEL model, a field theme is decomposed into subparts by means of a constrained Delaunay triangulation, like in (Højholt 2000). The field’s shape preservation constraints are expressed as constraints on subparts of the triangulations called *sub-micro* objects: segments, triangles, points (Figure 4a). The object-field constraints are expressed as relational constraints between a field agent and a micro agent of the AGENT or CARTACOM model (Figure 4b, not represented in the class diagram of

Figure 4a), and translated into constraints on sub-micro objects. The points that compose the triangulation are modelled as agents. The sub-micro objects are thus groups of point agents. Each internal or object-field constraint that concerns a sub-micro object is translated into forces on the point agents that compose it.

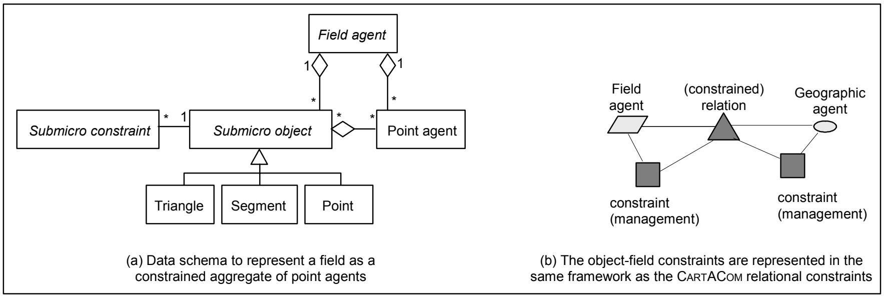

Fig. 4. GAEL static model : sub-micro level, point agents, sub-micro and object-field constraints

When a point agent is activated, it computes and applies to itself a small displacement in the direction that would enable it to reach a balance between the forces resulting from its constraints. Interactions between agents can be hierarchical or transversal. Field agents can trigger their point agents (hierarchical interaction), and point agents can directly trigger their neighbours (transversal interactions). This results in a progressive deformation of the field in answer to the deformations of the foreground themes.

The GAEL model has been successfully applied (Gaffuri 2007) to the preservation of relations between buildings and relief (elevation) and hydrographic network and relief (overland flow).

2.4. Areas of applications of AGENT, CARTACOM and GAEL: schematic summary

Figure 5 summarizes the main characteristics of the AGENT, CARTACOM and GAEL models. AGENT is based on hierarchical interactions between agents that represent single geographic objects or groups of objects. The considered constraints are described as internal to a single agent and managed by this agent. This model is best suited for generalising dense areas where density and non-overlapping constraints are prevalent and strong contextual elimination is required. CARTACOM is based on transversal interactions between agents that represent single geographic objects. The considered constraints are described as shared by two agents and managed by both concerned agents. This model is best suited for generalising low density areas where more subtile relational constraints like relative orientation are manageable. GAEL is based on transversal interactions between agents that represent points of geographic objects connected by a triangulation, and hierarchical interactions between these agents and agents that represent field geographical objects. The considered constraints are described either as shared by a field agent and a micro agent, or as internal to groups of connected point agents, and handled by these point agents. This model is best suited for the management of side-effects of generalisation on the background themes.

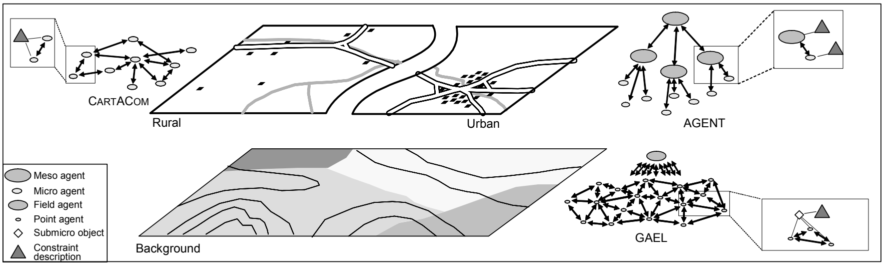

Fig. 5. AGENT, CARTACOM and GAEL model: target areas of application and levels at which constraints are described (GAEL object_field constraints are not represented)

The three models are best suited for different kinds of situations that are all present on any complete topographic map. Thus they will have to be used together in a complete generalisation process. In the next section, scenarios are proposed for the combined use of the three models.

3 Proposed scenarios to combine AGENT, CARTACOM and GAEL

In the subsections 3.1, 3.2 and 3.3, three complementary scenarios for the combined use of the models are studied, in which the synergy takes place at different levels. For each scenario, the underlying technical and research issues are identified.

3.1. Scenario 1: separate use of AGENT, GAEL and CARTACOM on a spatially and/or thematically partitioned dataset

This first scenario concerns the generalisation of a complete topographical dataset. Such a dataset contains both foreground and background themes (everywhere), and both rural and urban zones. In this scenario, we propose to split the space as shown in figure 5, both spatially and thematically, in order to apply each of the three models where it is *a priori* best suited:

- urban foreground partitions are generalised using AGENT,
- rural foreground partitions are generalised using CARTACOM,
- background partitions follow using GAEL.

Let us notice that this scenario does not cover the complete generalisation process but only a part of it. It is intended to be included in a larger generalisation process or Global Master Plan (Ruas and Plazanet 1996) that also includes steps for overall network pruning, road displacement using e.g. the beams model (Bader 2001), and generalisation of background themes (on top of letting them follow the foreground themes). Actually, these additional steps would also be applied on either thematically or spatially split portions of the space.

This scenario first requires adapted methods to partition the data in a pertinent way (here into foreground and background themes, into urban and rural zones). Then, whatever the partitioning, the resulting space portions are not independent because strong constraints exist between objects situated on each side of the borders: continuity of roads and other networks at spatial borders, inter-theme topological relations, etc. This interdependence requires mechanisms for the management of side-effects on the data, i.e. to ensure that no spatial inconsistency is created with other portions of the space when applying one model on portion of the space. It also requires pertinent heuristics for the orchestration of the process: when to apply which model on which partition.

These issues are not new: they have already been discussed by (McMaster and Shea 1988; Brassel and Weibel 1988; Ruas and Plazanet 1996) regarding the design of generalisation process composed of elementary algorithms. We are just a step forward here, since now we consider the combination of several generalisation processes based on different models.

3.2. Scenario 2: "interlaced" sequential use of AGENT, CARTACOM and GAEL on a set of objects

This second scenario concerns the generalisation of a set of objects included in a single partition of the previous scenario i.e. a portion of either urban foreground space, rural foreground space, or background space. In this scenario, we propose to enable the "interlaced sequential use" of the models, i.e. calling successively two or more of the models on the same objects, possibly several times (e.g. AGENT then CARTACOM then AGENT again).

Indeed, experiments performed with the AGENT and CARTACOM models show that the previous scenario is not sufficient. The limit between a rural space that should a priori be generalised by CARTACOM and a urban space that should a priori be generalised by AGENT is not so clear. In some zones of medium density, CARTACOM enables to solve most of the conflicts while tackling also more subtile constraints like relative orientation, but can locally encounter over-constrained situations. In this second scenario, such locally over-constrained situations can be solved by a dynamic call to an AGENT hierarchical resolution. Conversely, not all the constraints shared by two objects in an urban zone can easily be expressed as an internal constraint of a group (meso agent) and solved at the group level. Thus, in scenario 1, some of them are given up, e.g. the constraint of relative orientation. Scenario 2 enables punctual use of CARTACOM inside a urban zone, which could help in solving such subtile relational constraints for which no group treatment is available. Regarding the thematic split between foreground and background, it seems that this distinction is not so well defined either. This is why in this scenario, some objects can be considered as foreground at one time of the process and background at other times. For instance, buildings are foreground when handling there relational constraints with the roads thanks to a CARTACOM activation; but they are rather background when handling the overlapping constraints between roads, as their behaviour at this time should just be to follow the other feature classes in order to prevent topological inconsistencies.

To summarize, in scenario 2 the geographic objects of a dataset are able to play several roles during a generalisation process: an object can be modelled as an AGENT, CARTACOM and GAEL agent at the same time and be successively triggered with an AGENT, CARTACOM or GAEL behaviour (life-cycle). To be more precise, a same object of the micro level can be modelled and triggered both as an AGENT and CARTACOM agent, and the points that compose it modelled and triggered as GAEL agents (as the GAEL model operates at the points level).

To enable this, some mechanisms are required to detect the need to dynamically switch to another model. This means, a mechanism is needed detect that the currently used model is unable to solve the situation, and identify the pertinent set of objects that should temporarily be activated with another model. Then, some consistency preservation mechanisms are required, not from a spatial point of view (this has already been tackled in scenario 1), but regarding the data in which an agent stores its representation of the world. For instance, if a CARTACOM activation is interrupted and an AGENT activation is performed that eliminates some agents, the neighbours of the eliminated agents should be warned when the CARTACOM activation resumes, so that they can update their "mental state". Otherwise, they could enter in an inconsistent state, with pending conversations and relational constraints with agents that do no longer exist.

3.3. Scenario 3: simultaneous use of AGENT and CARTACOM data on one object

This third scenario concerns the generalisation decisions taken by an agent of the micro level (single geographic object) that is both modelled as an AGENT and as a CARTACOM agent as proposed in scenario 2. Only the AGENT and CARTACOM models are considered here since only these models operate at a common level (micro level).

An agent that is both modelled as an AGENT and as a CARTACOM agent has knowledge both of its internal constraints and of relational constraints shared with other agents. But so far, including in scenario 2 above, only the internal constraints are taken into account when it behaves as an AGENT agent, and only the relational constraints are taken into account when it behaves as a CARTACOM agent (during its CARTACOM life-cycle, it can perform internal generalisation thanks to a call to the AGENT life-cycle as explained in 2.2, but the AGENT life-cycle is then seen as a "black box"). In this third scenario, an agent is able to consider both kinds of constraints at the same time when making a generalisation decision, be it in a CARTACOM or in an AGENT activation. This means that, when choosing the next action to try, the agent takes into account both the proposals made by its internal and relational constraints (with the restriction that an agent activated by AGENT does not try an action consisting in asking another agent to do something). And, to validate the action it has just tried, the agent takes into account the satisfaction improvement of both its internal and relational constraints. This scenario is not intended to introduce more relational constraints in urban zones than in scenario 2. It just proposes that, when such constraints have been defined (like the relative orientation constraint), they can be taken into account at the same time as the internal constraints. Provided relational constraints are parsimoniously added, and the relative importances and the relaxation rules of the internal and relational constraints are well defined, this scenario should not result in over-constrained situations anywhere. And it has multiple advantages:

- The aim of an agent activated by CARTACOM (e.g. a rural building) is still to satisfy both internal and relational constraints, but it can satisfy all of them by trying the actions they suggest, while remaining in its CARTACOM life-cycle. This is less computationally heavy than calling the AGENT life-cycle as a "black box".
- The aim of an agent activated by AGENT (e.g. a urban building) is still first to satisfy as well as possible its internal constraints. But, if it has relational constraints defined, they can prevent it from applying an internal algorithm that would decrease their satisfaction too much. For example, algorithms that square the angles of a building, or that transform it into a rectangle, can easily break relations of local parallelism between the building (or one of its walls) and another building or a road. (Steiniger, 2007, p. II-C-13) proposes to prevent this by forbidding the use of these algorithms in the parts of urban space classified as "inner city", because this problem frequently occurs in this kind of area. This scenario 3 enables to avoid this kind of problem in a more adaptive way (only when it really occurs).
- An agent activated by AGENT can also try internal actions specifically in order to improve the satisfaction of one of its relational constraints (like a small rotation in order to achieve parallelism with a neighbouring road). This is far less heavy than having to stop the AGENT activation and start a CARTACOM activation on the whole urban block containing the building.
- If micro-agents activated with AGENT cannot cope with some relational constraints because of "domino effects", another way of solving these constraints can also be that the meso agent above seeks for a global solution by analysing the relational constraints of its components (e.g., in the above case the urban block identifies the buildings that should rotate).

To enable this scenario 3, it is necessary to re-engineer the parts of the AGENT and CARTACOM static models related to constraint representation so that internal (AGENT) and relational (CARTACOM) constraints can both be handled by an agent within the same methods. Hence, the methods of the "Agent" class that use the constraints have to be modified, both in the

AGENT and in the CARTACOM model, in order to take into account the presence of both internal and relational constraints.

4. How to put the proposed scenarios into practice

4.1. Technical requirements underlying scenarios 1, 2 and 3: summary

In sections 3.1, 3.2 and 3.3 we have presented three scenarios where the AGENT, CARTACOM and GAEL models are used with an increasing degree of combination: separate use on separate zones (scenario 1), "interlaced" sequential use on the same zone (scenario 2), shared use of data internal to the models (scenario 3). The three scenarios are complementary and we intend to put all the three into practice in a medium term. The identified underlying issues are summarized hereafter, starting from the most external elements of the models, to the most internal:

1. Define methods to split the map space into relevant partitions, on spatial and/or thematic criteria [scenario 1]
2. Define a strategy to know which model to apply when on which portion of space [scenario 1]
3. Define mechanisms to manage the side-effects at borders, when generalising one partition with one model [scenario 1]
4. Define mechanisms to dynamically identify a set of geographical objects that require a temporary activation of another model than the currently active one [scenario 2]
5. Define mechanisms to preserve the consistency of data internal to one model, when another model is running [scenario 2]
6. Re-engineer the representation and management of constraints in AGENT and CARTACOM so that internal and relational constraints can be handled together [scenario 3]

The current status of the issues (1) to (5) is briefly described in the next section. The issue (6) is tackled more in deep in section 4.3.

4.2. Status of the technical issues underlying scenarios 1 and 2

The issues underlying the scenarios 1 and 2 (issues 1 to 5) in the list above) are part of a research that is currently beginning. However, for some of them we already have some elements of answer. Regarding the space partitioning (issue 1), previous research like (Boffet 2000; Chaudhry 2007) provide specific methods to identify urban or mountainous areas. Regarding the management of side-effects at thematic borders (issue 3), the *Object-field constraints* have been defined in the GAEL model in order to manage, thanks to a GAEL activation, the side-effects induced on the background themes by the AGENT or CARTACOM activations performed on foreground themes. This has already been implemented and tested for the themes building-relief and hydrography-relief during an AGENT activation (Gaffuri 2007). However, the question of when optimally to trigger GAEL during the AGENT activation (issue 2) is not solved yet. Regarding the interlaced use of two models, (Duchêne 2004) tackles the automatic triggering of group operations during a CARTACOM activation (e.g. with an AGENT meso activation). Consistency preservation mechanisms (issue 5) have been implemented and tested with manually triggered group operations. To detect that a group operation is needed (issue 4), a model has been proposed but not implemented at this time.

4.3. Re-engineering of constraint modelling in AGENT and CARTACOM to support scenario 3

In this section, we focus on the re-engineering of the AGENT and CARTACOM constraint modelling in order to enable that an agent modelled both as AGENT and CARTACOM agent can handle its internal and relational constraints at the same time. This means that in any method of an AGENT or CARTACOM agent that handles constraints, the role of constraint can be played either by an internal or a relational constraint. In other words, the agent has to see its internal and relational constraints within the same framework. In AGENT, an internal constraint is modelled as an entity in charge of both the description and the management of the constraint (Figure 6a). In CARTACOM, because the descriptive part is shared by two agents, two different entities are used for the description and the management of a relational constraint (Figure 6b). To integrate the two representations in a common framework, we first propose to modify the relational constraint modelling in CARTACOM: the descriptive part of the constraint (*Relation* object) is “replicated” on the two linked *Constraint* objects so that a CARTACOM agent “sees” the same thing as an AGENT agent (Figure 6c).

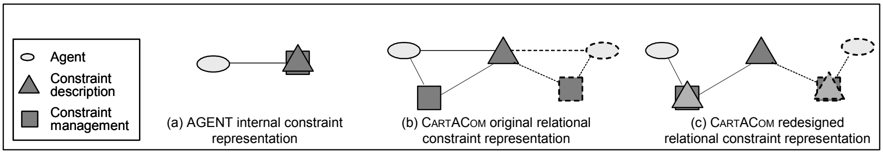

Fig. 6. How to ensure that an agent "sees" its internal and relational constraints in the same framework

More precisely, there is no data replication, but all the information supported by the *Relation* object is made available from the linked *Constraint* objects. The resulting data schema is shown in Figure 7b: getter methods have been added to the CARTACOM *Constraint* class, which get the values carried by the attributes of the CARTACOM *Relation* class. The AGENT data schema is modified accordingly (Figure 7a: the same getter methods are added, but they get the values from the attributes of the AGENT *Constraint* class). Attributes and methods in bold are the ones that the agent can use because they are common to internal (AGENT) and relational (CARTACOM) constraints.

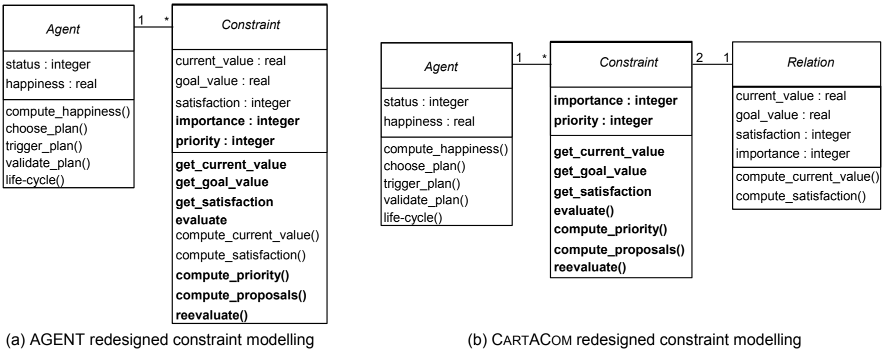

Fig. 7. Formatting the AGENT and CARTACOM constraint representation in the same framework results in splitting the AGENT Constraint class.

Once this "replication" has been performed, we can merge the AGENT and CARTACOM data schema (Figure 8). A generic *Agent* class is specialised into *AGENTAgent* and *CARTACOMAgent*. Similarily, a generic *Constraint* class is specialised into *InternalConstraint* and *RelationalConstraint*. The attributes and methods common to the AGENT and CARTACOM classes are transferred to the generic classes.

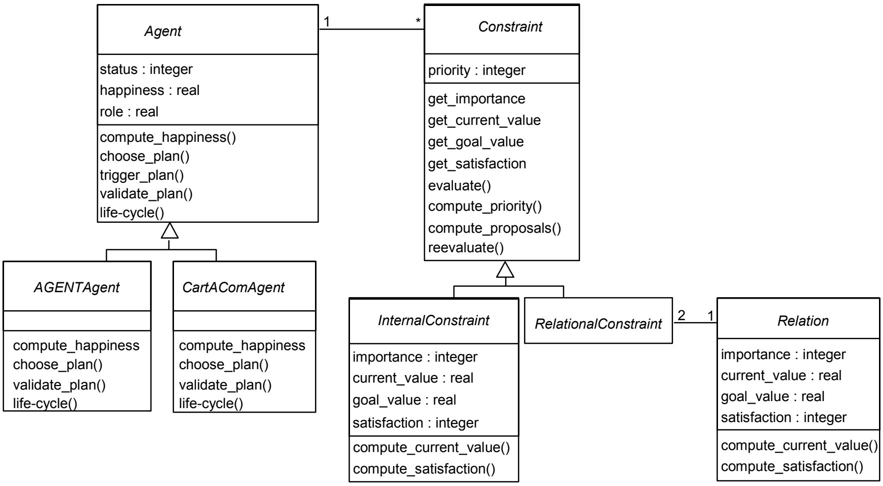

Fig. 8. Factorisation of common aspects of the agents and constraints: re-engineered constraint representation of AGENT and CARTACOM.

As geographic objects can be modelled both as AGENT and CARTACOM agents (i.e. their class can inherit both from *AGENTAgent* and *CARTACOMAgent*), an attribute *role* is added to the generic Agent class. This attribute indicates wether the agent has to be activated as AGENT or CARTACOM agent, i.e. which version of the life-cycle (and the methods it uses) has to be applied to it. Apart from the method that triggers a plan, the other methods used by the life-cycle (and the life-cycle itself) are indeed different for an AGENT and a CARTACOM agent. In other words, these methods, defined at the generic *Agent* level, are specialised in the *AGENTAgent* and *CARTACOMAgent* classes. The *role* of an agent can change during the generalisation process.

The re-engineered data schema presented in figure 8 ensures that an agent modelled both as AGENT and CARTACOM agent can handle its internal and relational constraints at the same time. This was indeed the aim of this re-engineering. But an additional effect of merging the *Agent* and *Constraint* classes, while factorizing the properties and methods common to the models at the most generic level, is to allow an easier maintenance of the system. To go further in this direction, we propose to include the classes of the GAEL data schema (cf. Figure 4, section 2.3) in the merged schema. This is quite straightforward. Regarding constraints modelling, the GAEL model already uses the AGENT modelling for internal constraints associated to point agents and to submicro objects, and the CARTACOM modelling for relational constraints associated with field agents. We just have to add the newly defined *Field-object Relation* class, as a subclass of the CARTACOM *Relation* class. Regarding agents modelling, the GAEL *Field agent* and *Point agent* classes already have the same attributes and methods as the generic *Agent* class of the merged schema. Thus the GAEL agent classes are added as new subclasses of the *Agent* generic class. The final merged schema is shown on Figure 9.

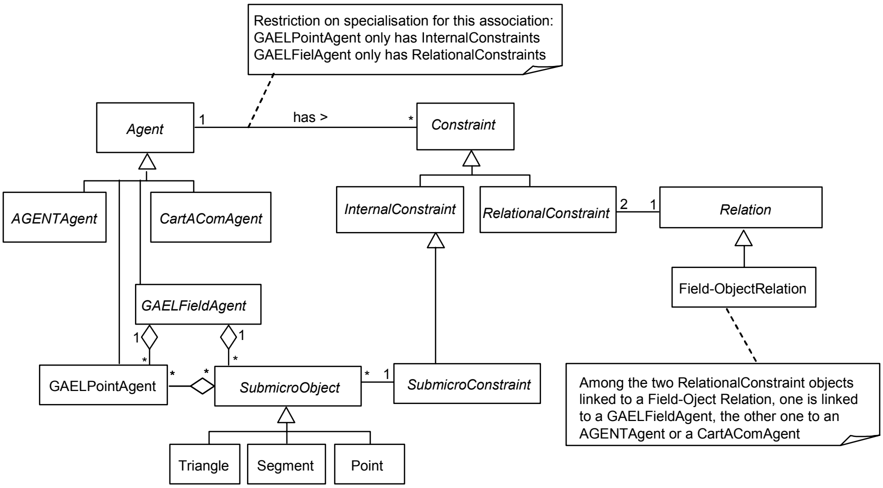

Fig. 9. Introduction of the GAEL classes to the merged schema

5. Discussion

In the two previous sections, we proposed three scenarios to combine the AGENT, CARTACOM and GAEL generalisation models in order to take advantage of each of them. Among these scenarios, at least the first one can also be extended to other generalisation models. We think that such scenarios are needed to go further in the automation of generalisation, without relaxing the cartographic quality too much.

However, complexity or tractability problems are necessarily attached to a system that would implement these three scenarios. This complexity takes place at two levels: firstly, getting familiar to the system and tuning it for a particular use is fastidious; and secondly, the automated generalisation process itself is of high computational complexity (high numbers of agents, constraints and links between both, time consuming algorithms, etc.).

This double complexity calls some clarifications on the target usages of such a system. If we consider the map series making (for map producers, namely NMAs), this double complexity is not a huge problem: building a new map production line is anyway time and resource consuming, and the computational constraints attached to the actual production of one map are not very strong as long as no memory overflow is encountered. Indeed, if the process is highly automated, it can run on a dedicated machine overnight, and will anyway be far quicker than manual generalisation. Now, if we consider on demand mapping, the computational complexity clearly prevents from using such a system for on the fly generalisation. But it could still be used for off-line customised cartography, provided the tuning (parametrization) can be assisted. Research works that can help in this are (Hubert and Ruas 2003), for the translation of user needs into a generalisation system, and (Taillandier 2007), to help the automated revision of the procedural knowledge within the AGENT model.

6. Conclusion and perspectives

In this paper, we have presented a comparative analysis of three agent-based generalisation models dedicated to three different kinds of geographic data and cartographic constraints: AGENT, CARTACOM and GAEL. Three complementary scenarios have been proposed to use them in a combined way, with an increasing degree of combination. For each scenario, the underlying issues have been described. The issue that is the most internal to the models has been tackled and as a result, a partial re-engineering of the models has been proposed.

This re-engineered version will now be implemented in Clarity®, the generalisation platform commercialised by 1Spatial, where AGENT and GAEL are already implemented. CARTACOM, which is for the time being implemented in LAMPS2, will be ported to Clarity on that occasion. The re-engineered model will then be tested on three different topographical data extracts separately:

- an urban zone with classical internal constraints and a few relational constraints defined, that will be generalised with an AGENT activation,
- a rural zone with classical relational constraints and some internal constraints defined, that will be generalised with a CARTACOM activation,
- a mountainous zone with object-field and internal field constraints defined, where the generalisation of foreground themes will be interactively performed and their side-effects managed by GAEL activations.

The issues related to the first two scenarios, numbered (1) to (5) in section 4.1, are being tackled in a parallel research project. This way, we hope to make significant progress in multi-theme generalisation.

Acknowledgement

The authors would like to warmly thank the anonymous reviewer who sent his marked-up manuscript for his constructive remarks and his English language corrections.

References

Bader M (2001) Energy Minimization Methods for Feature Displacement in Map Generalization. Ph.D. thesis, University of Zürich

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. In: Proc. of the 20th International Cartographic Conference, Beijing, China, 2001, vol.3, pp 2110-2116

Beard K (1991) Constraints on rule formation. In: Buttenfield B., McMaster R. (eds) Map Generalization: Making Rules for Knowledge Representation, Longman Scientific and Technical, Harlow, Essex, pp 32-58

Brassel K, Weibel R (1988) A review and conceptual framework of automated map generalization. International Journal of Geographic Information Systems, 1988, 2(3):229-244

Boffet A (2000) Creating urban information for cartographic generalisation. In: Proceedings of the 9th International Symposium on Spatial Data Handling (SDH 2000), Beijing, China, pp 3b4-16

Chaudhry O (2007) Automated scale dependent views of hills and ranges via morphometric analysis. In: Proceedings of the 23rd International Cartographic Conference, Moscow, Russia

Duchêne C (2004) The CARTACOM model: a generalisation model for taking relational constraints into account. 6th ICA Workshop on progress in automated map generalisation, Leicester

Gaffuri J (2007) Field deformation in an agent-based generalisation model: the GAEL model. Proceedings of GI-days 2007 - young researches forum, Münster, Germany, 2007, vol. 30, pp 1-24

Galanda M (2003) Automated Polygon Generalization in a Multi Agent System. Ph.D. thesis, University of Zürich

Højholt P (2000) Solving Space Conflicts in Map Generalization: Using a Finite Element Method. Cartography and Geographic Information Science, 27(1): 65-73

Hubert F, Ruas A (2003) A method based on samples to capture user needs for generalisation. 5th ICA Workshop on progress in automated map generalisation, Paris

Lecordix F, Le Gallic J-M, Gondol L, Braun A (2007) Development of a new generalisation flowline for topographic maps. 10th ICA Workshop on Generalisation and Multiple Representation, Moscow, Russie

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, USA, pp 240-249

Meyer U (1986) Software developments for computer-assisted generalization. In: Proceedings of Auto-Carto, London, 2:247-256

Ruas A, Plazanet C (1996) Strategies for Automated Generalization. Proc. of the 7th International Symposium on Spatial Data Handling, Delft, The Netherlands, pp 6.1-6.17

Ruas A (1998) OO-Constraint modelling to automate urban generalisation process. In: Proceedings of the 8th International Symposium on Spatial Data Handling, pp 225-235

Ruas A (2000) The Roles Of Meso Objects for Generalisation. Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, pp3b50-3b63

Sester M (2000) Generalization Based on Least Squares Adjustment. International Archives of Photogrammetry and Remote Sensing, vol.33

Steiniger S (2007) Enabling Pattern-Aware Automated Map Generalization. Ph.D. thesis, University of Zürich

Taillandier P (2007) Automatic Knowledge Revision of a Generalisation System. 10th ICA Workshop on Generalisation and Multiple Representation, Moscou

Weibel R, Dutton G (1998) Constraint-Based Automated Map Generalization. In: Proceedings of the 8th International Symposium on Spatial Data Handling, pp 214-224

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