Object-field relationships modelling in an agent-based generalisation model

Julien Gaffuri, Cécile Duchêne & Anne Ruas
Julien.Gaffuri@ign.fr, Cecile.Duchene@ign.fr, Anne.Ruas@ign.fr
COGIT Laboratory
Institut Géographique National
2/4 avenue Louis Pasteur
F-94165 Saint-Mandé Cedex

Abstract

Many research works in map generalisation are applied to building and network themes. Several generalisation models, such as the agent-based generalisation model of Ruas and Duchêne (2007), on which this paper focuses, have been designed and successfully put to use for these themes. Our purpose is to take into account field themes (relief, land use cover, etc.): many relationships exist between these themes and other objects and should therefore be preserved during the generalisation process. The GAEL model has been set up to enable this.

This paper presents the modelling of relationships between objects (buildings, roads, etc.) and fields, and their integration into the agent-based generalisation process of Ruas and Duchêne (2007) for their preservation. The principle is to allow each geographic agent to asses the state of its relationships with fields, and possibly to apply a specific treatment, such as a field deformation or a displacement of itself, in order to preserve their relationships.

Keywords: agent-based generalisation, object-field relationships, GAEL model.

Introduction

The agent-based generalisation model of Ruas and Duchêne (2007) has been specialised for the generalisation of building and road themes and has given promising results for the automatic generalisation of these themes. The purpose of our work is to go further in the generalisation automation by taking into account *fields themes*. A field provides a way to represent continuously defined variables. It "*allows to assign a value to every location*" (Cova and Goodchild, 2002). On topographic maps, some themes such as the relief and the land use cover are usually represented as fields composing a background, on which the other objects lay. As a consequence, objects share relationships with fields depending on their position on them. For example, a building has a specific elevation value and a hydrographic section flows down on the relief. Some of these relationships are important to preserve during the generalisation process.

In order to allow a preservation of such relationships, we propose a model, called GAEL (for Generalisation based on Agents and ELasticity). The principle of the GAEL model is to model fields as elastic layers interacting with objects above them (cf. Figure 1). In order to preserve the relationships, fields can deform themselves during the generalisation of the objects, and objects can be constrained by fields. In previous works, we presented the principles of the deformation process (Gaffuri, 2007a) and give some examples of application for specific object-field relationships (Gaffuri, 2007a, 2007b). This paper focuses on the object-field relationships modelling and their integration into the agent-based model of Ruas and Duchêne (2007). In a first part, we briefly give the main principles of the agent-based model of Ruas and Duchêne (2007). Then, we present our proposal for the object-field relationships modelling. Finally, we show how these relationships are taken into account by the objects during their generalisation.

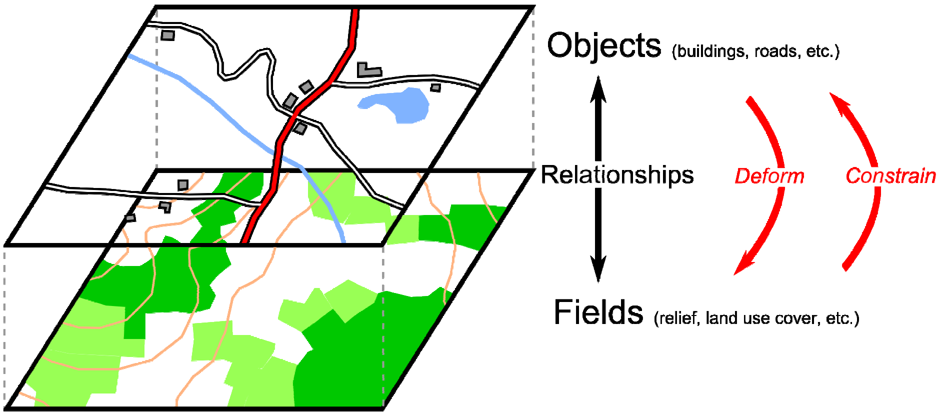


Figure 1: interactions between objects and fields during the generalisation process in the GAEL model

1. Principles of the agent-based generalisation model of Ruas and Duchêne (2007)

Our work aims at enriching the agent-based model of Ruas and Duchêne (2007). This model is based on the works of Ruas (1999), Duchêne (2004) and the AGENT project (Barrault *et al.*, 2001). The three main principles are the following:

1. **Each geographic object is an agent**. An agent is defined by (Weiss, 1999, p.29) as "*a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives*". Each geographic object that needs to be generalised (building, road...) is modelled as a geographic agent: it autonomously applies transformations to itself in order to reach a specific goal.
2. **The goal of each geographic agent is to generalise itself, i.e. to satisfy its cartographic constraints**. The specifications of the target geographic data are translated into constraints carried by the geographic agents (for example, a geographic object that should be big enough to be visible will carry a constraint on its size). The goal of each agent is to satisfy its constraints. An agent is able to achieve this goal by executing a generic life-cycle. During its life-cycle, the agent measures its satisfaction level (which is the result of an aggregation of the satisfaction of all its constraints), and then, if it is not perfectly satisfied, chooses an algorithm to apply to itself in order to improve its satisfaction state. The choice of this algorithm depends on its violated constraints: each constraint proposes a weighted set of algorithms depending on its state of violation. These algorithms a priori allow an improvement of the constraint satisfaction. For example, a constraint on the size of a too small agent will propose to it to try an enlargement algorithm. The geographic agent chooses an algorithm to apply to itself among the proposed algorithms of all its constraints. This choice is performed depending on the weight value of the algorithms and a priority value carried by the constraints proposing the algorithms. Every time an agent applies an algorithm, it checks afterwards if its state has been improved by this algorithm, and

can possibly backtrack and try another proposed algorithm. The agent analyses its states and applies to itself algorithms until it is satisfied, or there is no more algorithm to attempt. This process allows geographic agents to improve their constraints satisfaction autonomously step by step and tend towards a satisfying generalised state.

3. **The use of several levels: the micro and meso levels**. Geographic agents do not generalise themselves independently, but depending on their own context. Indeed, the generalisation process has to take into account relationships between agents: some constraints are relative to groups of agents (for example, the density of an urban block). In order to take into account these constraints, the agent-based model of Ruas and Duchêne (2007) is based on the use of several levels of organisation. In the *micro* level, objects are taken independently. The so-called *meso* level concerns groups of objects. Each meso agent is composed of other agents, meso or micro. For example, a town agent is composed by urban block agents, and a urban block agent is composed of building agents. A meso agent manages the generalisation of its components. Two micro agents also have the capability to communicate in order to satisfy relational constraints they share, such as a proximity constraint between two buildings.

This model has been applied to the generalisation of building and road themes. It allows managing the relationships inside groups and between pairs of objects. The relationship between fields and objects is another kind of relationship, as mentioned in Mustière and Moulin (2002): this relationship concerns how an object lays on a field. We present now our proposal to take into account this new kind of relationship.

2. Modelling of object-field relationships

The GAEL model proposes modelling the object-field relationships in order to take them into account in the agent-based model of Ruas and Duchêne (2007). This modelling is based on the micro agents relationships modelling proposed by Duchêne (2004) in the CartACom model: in this model, both agents involved in a relationship are linked to an explicit object representing the relationship, and each agent is linked to a own specific constraint object that translate how it sees and tries to satisfy the shared relationship. The Figure 2 shows the UML class diagram we propose in the GAEL model for the object-field relationships modelling.

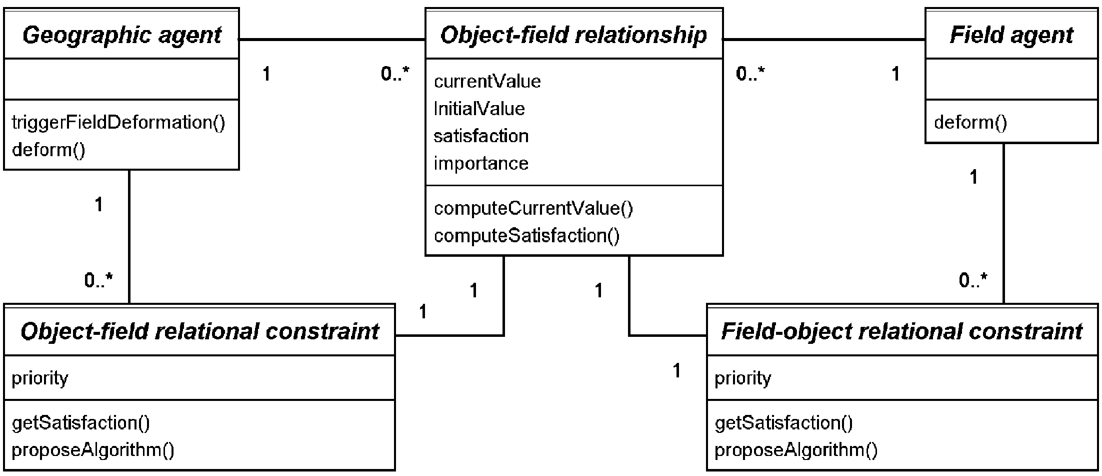


Figure 2: UML class diagram of relationships between objects and fields, and their related constraints.

The object-field relationship modelling involves the following classes:

- a **Field agent** class: fields are *elastic agents* - they satisfy their constraints by deforming themselves using the *deform()* method defined in this class. The deformation aims at finding a balance between some *shape preservation constraints* (which force the field to preserve its shape) and some *deformation constraints* (which force the field to change its shape). These constraints classes are not represented in Figure 2. Further details about the deformation computation are presented in (Gaffuri, 2007a). The principle of this deformation method is to decompose the field into a set of small constrained objects, called *submicro objects* (segments, triangles, etc.). For example, the triangles respectively the segment composing the field can be constrained to preserve their area respectively their length and orientation. The points composing the field are modelled as agents. The goal of each point agent is to reach a balance position between the constraints carried by the submicro objects it belongs to. The autonomous displacement of the points allows making the field elastic. The field triggers its own deformation through its *deform()* method. This method allows the field to manage the activation triggering of its composing points. After an activation of its points, a field agent is able to measure its state and possibly cancel the performed deformation if it has been distorted too much.
- a **Geographic agent** class: objects of this class are geographic agents (like building and road agents) as defined in the model of Ruas and Duchêne (2007). They have a method *triggerFieldDeformation()*, which allow them to activate a specific field agent so that it deforms itself locally under the geographic agent in order to improve the object-field relationships satisfaction. These agents have the capacity to deform themselves too by triggering their *deform()* method.
- an **Object-field relationship** class: an object of this class makes a relationship between an object and a field explicit. It is linked to the *geographic agent* and the *field agent* involved in the relationship. An *object-field relationship* object is characterised by a *current value* and an *initial value*. These values depend on the value of the field under the object respectively in its current and initial states. The *satisfaction* value is a qualitative interpretation of the gap between the current and the initial values of the relationship. It allows a measurement of the relationship preservation. The *importance* value translate the importance of the preservation of the relationship, compared to the other constraints of the geographic agent and the field agent,
- an **Object-field relational constraint** class: such a constraint is linked to both a relationship and the object involved in this object-field relationship. It is characterised by a *priority* value, which translate the emergency for the geographic agent to satisfy this constraint during its generalisation process. The *getSatisfaction()* method returns the satisfaction of the constraint, which is equal to the satisfaction of the object-field relationship linked to the constraint. The *proposeAlgorithm()* method returns a list of possible algorithms to be tried by the geographic agent to improve the satisfaction state of the relationship,
- and a **Field-object relational constraint** class: such a constraint has the same characteristic as the previous one, except it is linked to the field involved in the relationship. The *proposeAlgorithm()* method returns a unique algorithm: a deformation of the field (available through its *deform()* method). This deformation is tuned at the submicro level in order to improve the relationship satisfaction, as described in (Gaffuri, 2007a).

So, like in the CartACom model, the *object-field relationship* is constrained through two kinds of constraints: one *object-field relational constraint* linked to the *geographic agent*, and one *field-object relational constraint* linked to the *field agent* (cf. Figure 3). Both constraints have the same satisfaction value, which is the satisfaction of the relationship. Each of these constraints allows a translation of how the geographic object and the field see the relationship they share. Indeed, the geographic object and the field will not use the same algorithm to try to improve the satisfaction of the relationship.

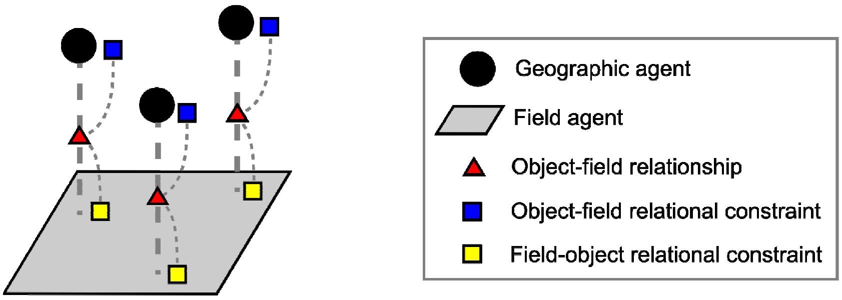


Figure 3 : a field, three geographic agents, and their associated relationships and constraints.

Both kinds of constraints introduced in Figure 2 are designed to be compatible with the model of Ruas and Duchêne (2007). Thanks to that, geographic agents can manage them during their life cycle. We present in the next section how these new constraints are integrated into the generalisation of the geographic agents.

3. Integration of the object-field relationships into the object generalisation process

The principles

The previously presented model gives an explicit representation of the object-field relationships and their constraints. We present now how the introduced classes are managed by the geographic agents during their generalisation process. This integration relies on the three following principles:

- **Each geographic agent takes into account its object-field relational constraints when computing its global satisfaction:** in the model of Ruas and Duchêne (2007), the global satisfaction of a geographic agent is the result of an aggregation of the satisfactions of its constraints (not necessary the average). These constraints concern some characteristics of the agent. The higher the satisfaction values of the agent's constraints, the higher the global satisfaction value of the agent. In order to integrate the object-field relationships into the agent generalisation process, we propose taking into account the object-field relational constraints in the computation of the global satisfaction of the geographic agent. As a consequence, the satisfaction level of each geographic agent is influenced by the satisfaction level of its object-field constraints too.
- **Each geographic agent has the capability to trigger a field deformation:** as mentioned in Figure 2, each geographic agent has a method *triggerFieldDeformation()*. This method allows the geographic agent to activate a field agent so that it deforms itself using its *deform()* method. As shown in (Gaffuri, 2007a), this deformation is computed locally: only a few points around the geographic agent are activated and displaced.
- **Some specific algorithms are proposed by the object-field relational constraints:** in the model of Ruas and Duchêne (2007), each constraint proposes an algorithm set to the agent through its *proposeAlgorithm()* method. These algorithms allow the

geographic agent to improve the relationship satisfaction. Concerning the object-field relational constraints, we suggest making them propose to the geographic agent at least these two algorithms:

- **First, a field deformation:** the geographic agent tries to trigger a field deformation by the mean of its *triggerFieldDeformation()* method. This method triggers an activation of the field agent involved in the relationship. During its activation, the field agent computes the satisfaction of its field-object relational constraint (which is the same as the object-field relational constraint satisfaction) and tries to apply to itself a unique algorithm to improve this satisfaction value: a deformation (all the *proposeAlgorithm()* methods of the field-object relational constraints return only this algorithm). So, when a geographic agent has a violated object-field relational constraint, it tries to improve the level of satisfaction of the constraint not by applying an algorithm to itself, but by deforming the field involved in the relationship.
- **Second, another algorithm applied to the geographic agent:** if a deformation of the agent-field is not possible to improve the relationship satisfaction, then the geographic agent can try to apply to itself another specific algorithm. This algorithm depends on the natures of the geographic agent and the relationship. It could be, for example, an appropriate displacement or deformation of the geographic agent allowing the latter to get a better position on the field.

This way, the satisfaction of the object-field relationship can be improved either by deforming the field, or by changing the object. This proposition enables both the object to deform the field and the field to constrain the state of the object, as represented in Figure 1. Depending on the object-field relationship nature, it could be possible to propose a larger choice of algorithms to both the geographic agent and the field agent: all algorithms allowing a relationship satisfaction improvement can be proposed.

A simple example

We illustrate on a simple example how these principles allow to take into account object-field relationships in the geographic agent generalisation process. We describe the possible interactions occurring between a unique geographic agent and a unique field sharing a unique object-field relationship (cf. Figure 4), although the model allows an instantiation of several of these objects.

At the beginning of the generalisation process (cf. Figure 4 a.), the geographic agent is not in a satisfied state (it has not been generalised yet: let us assume that his satisfaction value is 4), and the satisfaction values of both the relationship and the field (it has not been distorted) are perfect (10).

During its life cycle (cf. Figure 4 b.), the geographic agent changes by applying algorithms to itself in order to improve its satisfaction value. Because of these changes, the satisfaction of the relationship can be affected and possibly decrease (from 10 to, let us assume the value 6). Because the object-field relational constraint of the geographic agent is violated, it proposes to the geographic agent triggering a field agent activation.

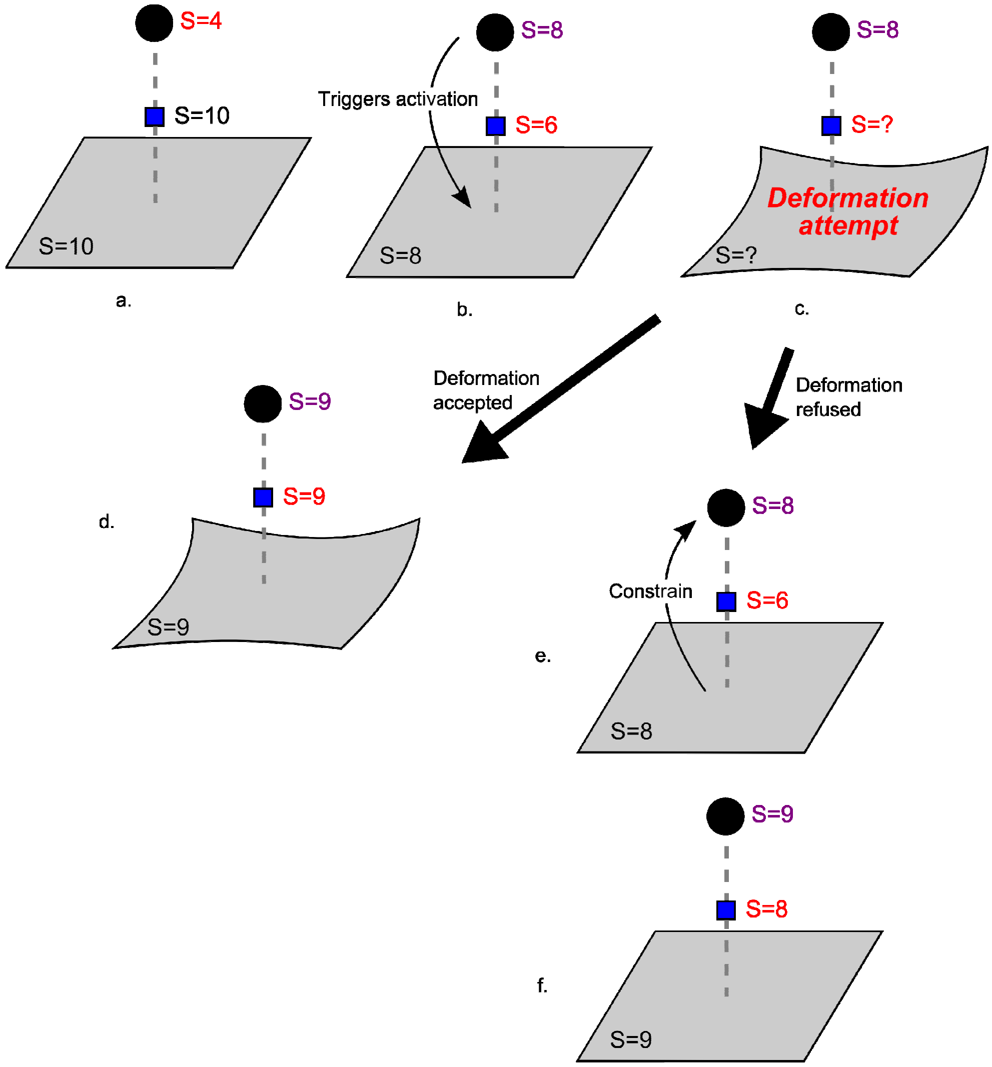


Figure 4 : example of interactions between a geographic agent and a field for the improvement of their relationship satisfaction.

When the field agent activates, it computes its satisfaction value. This satisfaction takes into account the satisfaction of its field-object relational constraint, (which is equal to the relationship constraint one, here 6). So, the field agent is not satisfied and tries a deformation algorithm proposed by its field-object relational constraint to improve its state (cf. Figure 4 c.). After the deformation, the field agent evaluates its satisfaction. Afterward, two cases are possible:

- **The satisfaction value of the field has improved (cf. Figure 4 d.):** in this case, the deformation has successfully improved the state of both the field and the relationship. The field agent finishes its life cycle and the geographic agent continues its own. Because the satisfaction of the relationship has improved by the field agent's activation, the satisfaction value of the geographic agent has increased. The geographic agent can then possibly try other algorithms in order to improve its state. It can possibly improve the satisfaction of the relationship even more (if it is not already perfect) by attempting to apply the other algorithm proposed by the object-field relational constraint.

- **The satisfaction value of the field has not improved:** in this case, the field agent did not achieve to improve its general satisfaction by deformation. It is usually the case when the agent-field deformation is not important enough (the relationship satisfaction has not been improved enough) or too important (in this case, the shape preservation constraints of the field are too much violated). The field agent cancels the deformation, backtracks to its previous state, and finishes its life cycle. The geographic agent continues its life cycle by applying to itself another algorithm proposed by its object-field relational constraint (cf. Figure 4 e.). After, it checks if this algorithm has allowed an improvement of its own satisfaction and keep this state in this case (cf. Figure 4 f.). It backtracks if the algorithm failed to improve its satisfaction.

At the end of the process, four kinds of results are possible (cf. Figure 5):

- **Handling the relationship has resulted in neither a field deformation nor an object change (cf. Figure 5 a.):** this case is possible when the own generalisation of the geographic agent did not affect the satisfaction of the relationship. This case is possible also when neither the field nor the object achieved to improve the satisfaction of the relationship by transforming themselves. As a result, the relationship preservation has been abandoned during the process.
- **Handling the relationship has resulted in a field deformation and no object change (cf. Figure 5 b.):** in this case, the geographic agent's generalisation caused an alteration of the relationship satisfaction. The field tried deforming itself, and achieved to improve the satisfaction of the relationship. After, either the geographic agent did not need to change because the field deformation allowed a perfect satisfaction of the relationship, or the geographic agent tried to change in order to improve the relationship, but failed.
- **Handling the relationship has resulted in no field deformation but an object change (cf. Figure 5 c.):** in this case, the field cancelled its deformation because it did not allow it to improve its satisfaction enough. The object tried changing in order to improve the relationship satisfaction and achieved. In this case, the geographic agent was constrained by the field for the preservation of their common relationship.
- **Handling the relationship has resulted in both a field deformation and an object change (cf. Figure 5 d.):** in this case, both the geographic agent and the field tried to change and achieved to improve the relationship satisfaction thanks to their respective transformations: they shared the work for the relationship preservation.

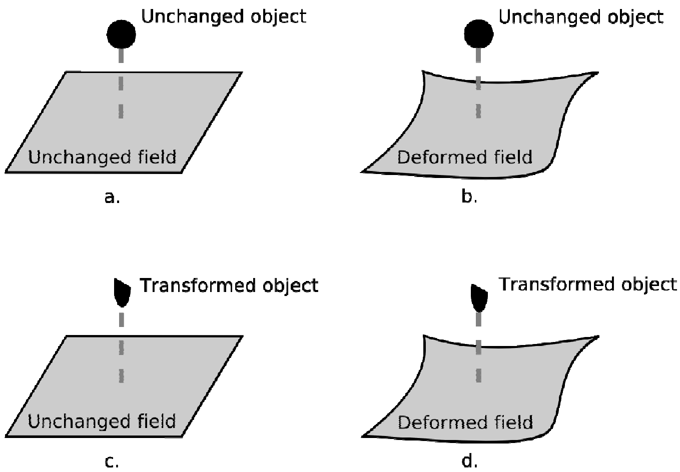
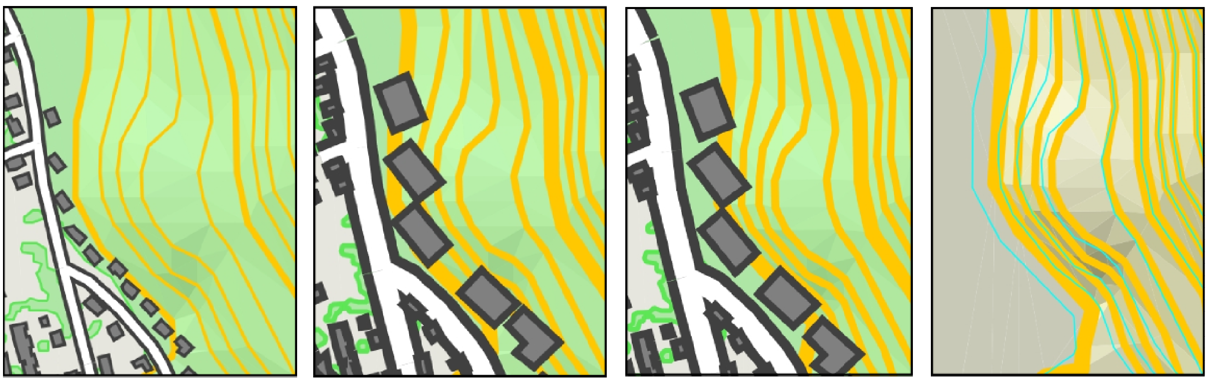


Figure 5: four possible results of the interaction between an object and a field

In the general case, a field can potentially share several object-field relationships with several objects, and an object can share several object-field relationships with one or several fields too. The presented process can be extended to this general case: each geographic agent can try deforming the fields it is in relation with. In some cases, several objects try deforming the field together in the same direction. In other cases, they deform the field toward opposite directions.

Results

The following figures present some results of the implementation of the presented proposal. In Figure 6 (a), some buildings are located at the bottom of a slope. When these buildings are generalised, they seem to have climb the slope because of their displacement (see Figure 6 (b)). By applying our proposal, the relief can deform itself in order to preserve the elevation value of the buildings lying on it (see Figure 6 (c)). The deformation performed by the relief agent is shown on Figure 6 (d). Buildings seem to have "pushed" the mountain. Figure 7 presents another example of building elevation preservation.



(a) Initial state (b) Generalised state without taking into account of the relief (c) Generalised state with taking into account of the relief (d) deformation of the relief

Figure 6: example of buildings elevation preservation

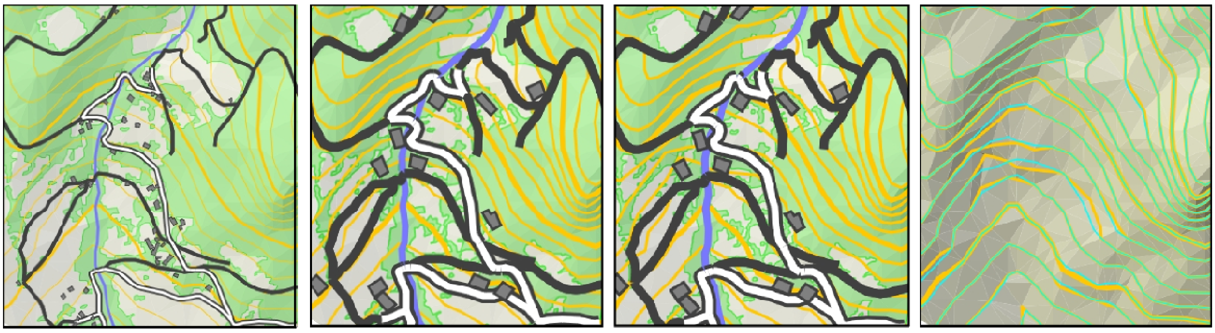


Figure 7: example of buildings elevation preservation

Figure 8 presents another example concerning the hydrographic network. This example involves a road and a river located in a narrow valley (cf. Figure 8 (a)). After generalisation on figure (b), the river has been displaced in order to avoid an overlapping with the road. As a consequence, the river does not flow anymore on the relief. In (Gaffuri 2007b), we present how our proposal allows constraining this outflow relationship involving the hydrographic network and the relief. The result is shown on Figure 8 (c) and Figure 8 (d): the relief has deformed in order to preserve the outflow of the river.

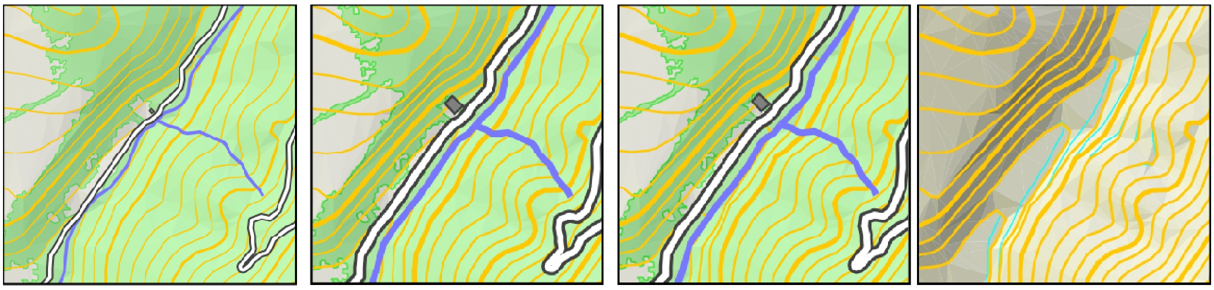


Figure 8: example of hydrographic section deformation constrained by the relief

Conclusion

In this paper, we proposed a model dedicated to the preservation of object-field relationships and proposed a way to take into account these relationships in the agent-based generalisation model of Ruas and Duchêne (2007). Geographic agents are capable of deforming the fields during their generalisation process, and can be constrained by the fields too.

This works improves the agent-based model of Ruas and Duchêne (2007) by allowing the integration of the fields into the generalisation process. Mackaness (2006) reports this quotation, extracted from a discussion in 1965: *"Imagine a gorge with a river and a road and a railway. First we plot the river, then we display the road. The railway is displaced further and finally the contours are moved. This presents a very difficult problem for the machine to solve"*. We think the GAEL model allows to progress towards this goal.

References

Barrault, M.; Regnauld, N.; Duchêne, C.; Haire, K.; Baeijs, C.; Demazeau, Y.; Hardy, P.; Mackaness, W.; Ruas, A. & Weibel, R. Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalization, *20th international conference of cartography,* **2001**, *3*, 2110-2116

Cova, T. J. & Goodchild, M. F., Extending geographical representation to include fields of spatial objects, *International Journal of geographical Information Sciences,* **2002***, 16*, 509-532

Duchêne, C., Généralisation cartographique par agents communicants: le modèle CartACom, *Pierre et Marie Curie Paris VI university, COGIT laboratory,* **2004**

Gaffuri, J., Field deformation in an agent-based generalisation model: the GAEL model, *GI-days-2007 - young researches forum,* **2007a**, 30, 1-24

Gaffuri, J., Outflow preservation of the hydrographic network on the relief in map generalisation, *International Cartographic Conference,* **2007b**

Mackaness, W. A., Automated Cartography in a bush of ghosts, *Cartography and geographic information science,* **2006***, 33*, 245-256

Mustière, S. & Moulin, B., What is spatial context in cartographic generalisation?, *Geospatial theory, processing and applications,* **2002***, 34*, 274-278

Ruas, A., Modèle de généralisation de données géographiques a base de contraintes et d'autonomie, *Marne la Vallée University, COGIT laboratory,* **1999**

Ruas, A. & Duchêne, C., A prototype generalisation system based on the multi-agent system paradigm, chapter 14 of *Generalisation of Geographic information: cartographic modelling and applications,* Mackaness, W.; Ruas, A. & Sarjakoski, L. *(ed.)*, *Elsevier Ltd,* **2007**, 269-284

Weiss, G., Multiagent systems. A modern approach to distributed artificial intelligence, *The MIT Press,* **1999**