Individual Road Generalisation in the 1997-2000 AGENT European project

Cécile Duchêne

IGN Laboratoire COGIT & Université Paris-Est,

73 avenue de Paris, 94165 Saint-Mandé Cedex, France

firstname.lastname@ign.fr

Preamble – August 2014

This paper has been prepared, and internally accepted in 2003, for a special issue of the CEUS Journal that eventually never appeared for organisational reasons, dedicated to the European AGENT research project (Project ESPRIT/LTR/24939, from December 1997 to December 2000). It describes the model developed for cartographic generalisation of sinuous mountain roads during the AGENT project. This paper is an extended version of an abstract presented at GISRUK 2001 under the title "Road generalisation using agents". It also extends the paper by [Duchêne et al. 2001]. Compared to this paper, the paper by [Duchêne et al. 2001] additionally describes the mechanisms developed in AGENT for road network generalisation and the connection between road network and individual road generalisation, but it gives less detail on the generalisation process of an individual road.

After the end of the AGENT project, this work has been industrialised and used in production at IGN-France, as part of the production process of the 1:100k topographic map – see [Lecordix et al. 2005] for an overview of the industrialised process, including results (as map extracts) and pointers to associated publications.

We publish this (old!) paper as a technical report, in order to keep a written and accessible record of the model so that some of its principles might be reused for further needs.

Abstract

This paper explores the application of an agent based methodology to the problem of automated road generalisation. It focuses on the generalisation of one single road feature in the case of winding mountain roads, and its implications on the surrounding road network. The aim is to ensure legibility of the road symbol under scale changes, i.e. mainly to avoid coalescence. The solution proposed here is based on the multi-agent framework developed as part of the European AGENT project. It uses a previous approach in automated road generalisation that consists of decomposing the road into parts in order to isolate the coalesced portions, and then handle those portions individually with tailored algorithms. The paper presents the decomposition process in terms of agent modelling. A worked example of the process is presented. A discussion of the results includes comparison with other possible approaches to automated road generalisation.

Introduction

In broad terms, the aim of the AGENT project [Lamy et al. 1999; Barrault et al. 2001] was to build a prototype system for automated cartographic generalisation, on a GIS platform (LAMPS2), using multi-agent techniques. The design is based on two main principles [Ruas 1998; Ruas 1999]:

1. To view geographical map features as local decision entities (agents), that act to generalise themselves according to cartographic and procedural knowledge and their capacity to analyse their state at any given time during the map compilation process,
2. To explicitly distinguish several levels of geographical analysis: the individual objects of the database (one building, one road, etc.), but also groups of spatially organised objects (a town, a road network, etc.).

This paper focuses on a component of the AGENT prototype, concerned with road generalisation. The specific issues were: What knowledge is required for road generalisation? How can we model the interactions between objects from several nested geographical levels of analysis (a bend, a road, a road network)? This paper does not address the problem of road network generalisation (concerned with issues of proximity and overlapping between roads), which is discussed by [Duchêne et al. 2001]. In this paper the focus is on the generalisation of a road such that it has the optimal cartographic representation. In particular, we are interested in the interactions between a road and its components (bends, series of bends, straight parts) during the generalisation process. The generalisation of a road feature can have implications on the neighbouring roads, requiring us to examine the idea of propagating generalisation into the surrounding road network.

The first part of this paper gives an overview of the AGENT project's prototype (generic model and engine). In the second part we present the model with respect to the generalisation of road features. The third part deals with the problem of working with objects at several nested levels of analysis, and describes some solutions. The fourth part presents a worked example on a real subset of data and the fifth part discusses the results obtained in comparison to other approaches for road generalisation. Finally we conclude with some evaluation work, pointing to some perspectives for further research.

1. Overview of the AGENT project's prototype

1.1. Agents and constraints

In the model designed for the AGENT prototype, the geographical entities have been modelled as agents [Ferber 1999; Weiss 1999]. In Artificial Intelligence, agents are objects that have a goal and a degree of autonomy in order to reach that goal [Wooldridge & Jennings 1995]. The geographical agents are described by a set of characteristics (size, shape, etc.) that constrain the generalisation process. Some characteristics govern the requirement for generalisation (for example when the size of a building is too small, the building must be generalised and made bigger). Other characteristics act as 'limiters' to the degree of generalisation (for example that the overall shape of a feature must not be too distorted from its original form). Therefore an important part of the project was to identify the pertinent characteristics for each geographical theme.

In terms of model, each geographical agent is guided by a set of "constraint" objects that act as "advisers". Each constraint object (also called simply "constraint") observes changes to a particular characteristic of the agent, and has a *goal* for the value of that characteristic, according to some map specifications. For example, for a map at 1:50 000, the size of a building should be bigger than 300m². When the current state of the characteristic does not fit the map specifications, the constraint is considered violated. The constraint object then proposes possible plans (i.e. generalisation algorithms) to apply in order to improve the state of the characteristic, i.e. make its current value tend towards the goal value. For example, a constraint governing the size of a building agent might propose a dilation algorithm if at any moment the size is too small compared to the *goal value*. Therefore the aim of the agent is to satisfy as best is possible, all of its constraints (Fig 1).

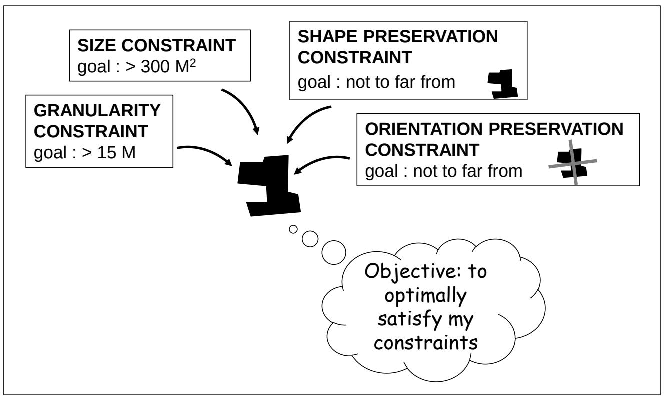

Figure 1. An agent (here a building) surrounded with its constraints.

In order to satisfy all its constraints, the agent has the capacity to
- characterise its state and evaluate it according to its constraints,
- choose a plan amongst those proposed by the constraints (the *a priori* best plan),

– apply this plan (i.e. trigger the algorithm),
– evaluate the improvement of its state (i.e. re-evaluating its state)

Fig 2 shows the generic life-cycle of an agent when activated, as designed by [Ruas 1999]. Furthermore a mechanism has been added, described in more detail by [Regnauld 2001], which enables the agent to backtrack to any state and try other plans, ensuring that the system reaches the best possible solution according to its evaluation criteria. The process stops when either all the possible plans have been tried, or a perfect state has been reach, in other words all the constraints are satisfied. Note that the perfect state for an agent is to have all its constraints satisfied, since its knowledge about itself is comprehensively described through its constraints.

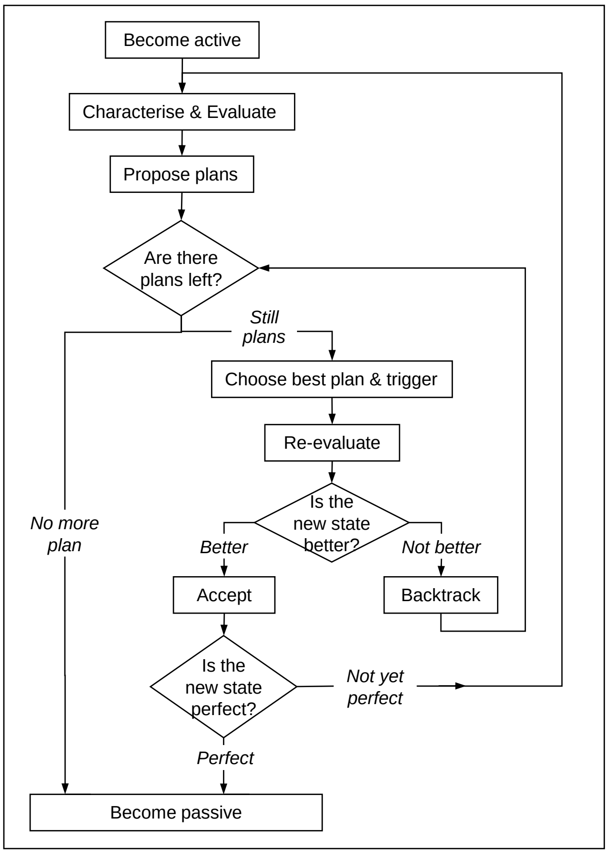

Figure 2. Generic life-cycle of an agent (after [Ruas 1999])

This behaviour of the agents presupposes that, in addition to the mechanism itself, there are a given number of tools available within the system: measures to characterise the objects and generalisation algorithms to transform them. It also assumes that there is formalised knowledge relating to the following questions: which measure is the most appropriate to quantify each characteristic? Which algorithm is best suited to which problem? How to compare any given pair of states of the agent (what to conclude if a characteristic has been improved and another one made worse)?

1.2. Several levels of analysis

Generally, only single geographical objects are represented in databases: one road, one building, one lake, etc. However, for the characterisation of the geographical space as well as for its generalisation, operations are not only performed at this single object's level. Some operations are performed on groups of spatially organised objects (e.g. a group of buildings sharing the same alignment along a section of road). Other operations are performed on a part of an object (e.g. a series of connected hairpin bends within a section of road). In the AGENT prototype, as proposed by [Ruas 1999], these levels of geographical analysis have been explicitly distinguished. The lowest level of analysis, which contains single objects, is called the "micro" level. The group level is called the "meso" level. As the meso level is not present in the geographical database, we have to identify the pertinent types of meso objects and construct these objects using methods of spatial analysis. Several nested meso levels may be defined, e.g. a town (meso) contains districts (meso), which in turn contain buildings (micro).

The meso level can be created in different ways. On the one hand, it can be built either *bottom-up*, i.e. by grouping objects (e.g. close buildings are grouped together to create a town), or *top-down*, i.e. by splitting a whole into parts (e.g. the districts are obtained by partitioning the town). On the other hand, the meso level can either be built *a priori*, in a stage of data enrichment prior to the generalisation process, or it can be built *dynamically* during the generalisation, as and when the need arises. For instance, towns and districts are created *a priori* whereas groups of aligned buildings are created dynamically during the generalisation of a district.

1.3. Behaviour at the meso level of analysis

The meso level performs several roles in generalisation [Ruas 2000]. First, a meso agent is responsible for the generalisation operations occurring at its level (e.g. elimination of objects inside a group, typification of a group). Secondly, the meso agent manages the generalisation of its components through three possible behaviours:

- *co-ordination*: the meso agent activates the micro agents in turn, trying to optimise their order of activation. The order of activation of the micro agents is referred to as the "*order of autonomy*" in the remainder of this paper. This order is very important in order to minimise the knock on effects related to the generalisation of one micro agent (described further for roads in section 3),
- *control*: after triggering each micro agent, the meso agent manages the knock-on effects produced by the micro agent's generalisation,
- *legislation*: the meso agent gives orders to the micro agents or changes their constraints or goals, either to help them to solve a conflict they cannot solve themselves, or to relax over-constrained situations.

These behaviours can be specified for each kind of meso object. It makes the life-cycle of a meso-agent a bit more complex than the generic life-cycle of an agent (Fig 3). Fig 3 shows the parts of the life-cycle that are the same as the life-cycle of a micro-agent with elements greyed out, in order to highlight the parts specific to a meso agent.

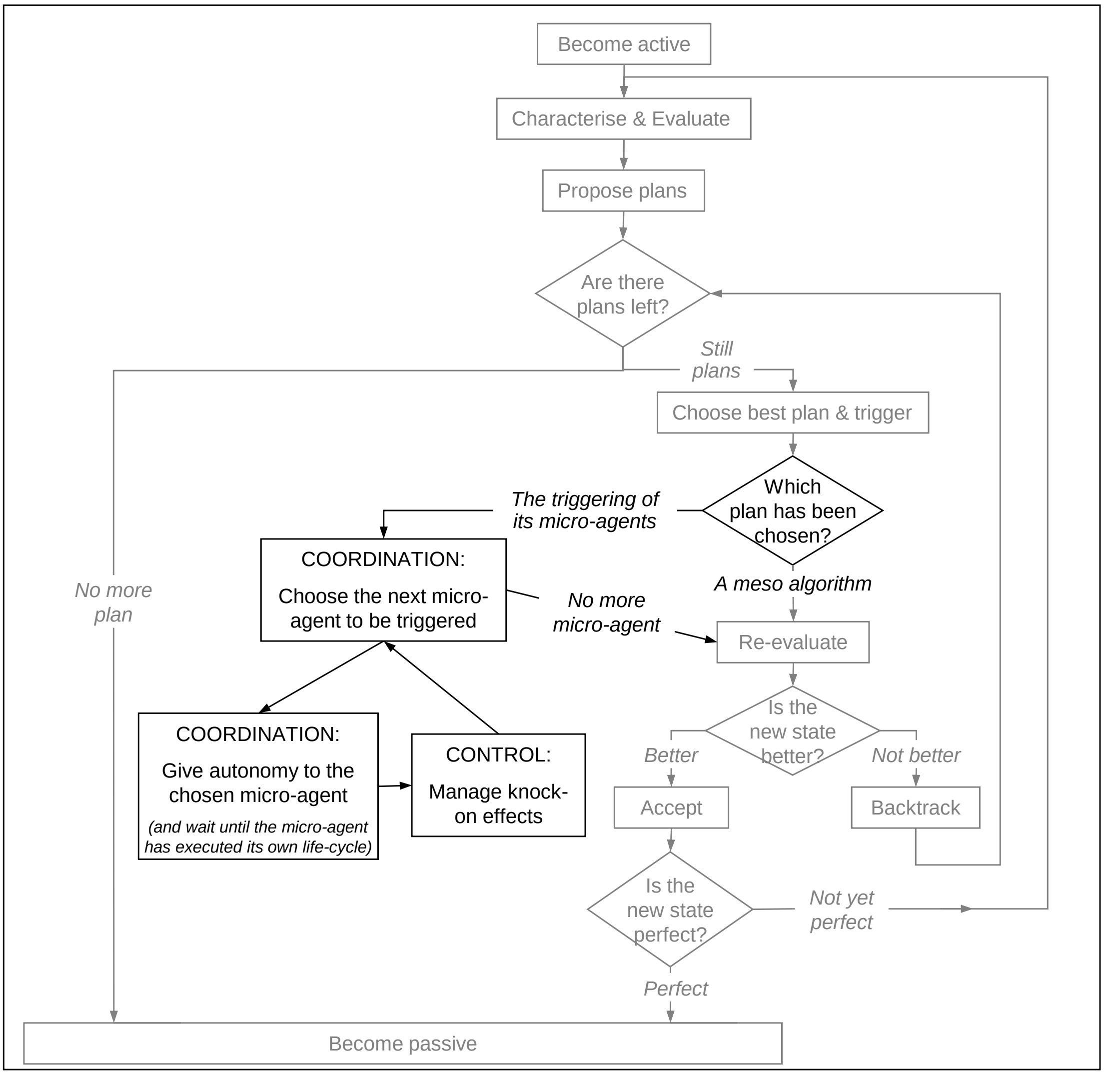

Figure 3. Life-cycle of a meso-agent

Since they have different functionalities, in terms of implementation the two kinds of agents (micro and meso), are translated into two object classes of the AGENT model: *micro-agent* and *meso-agent*. Each of these classes has different methods. The fact that a given type of geographical object is micro or meso governs whether the corresponding geographical object class inherits from either the class *micro-agent* or *meso-agent*.

The underlying questions that must be answered when designing the meso-agent functionality are related to communication between a meso and a micro level:

– To which micro agent(s) should the meso agent give autonomy at any given time? How is the "best autonomy order" determined?

– How are the knock on effects managed? How will the characteristics of the micros be preserved when managing the knock on effects? Which agent is responsible for managing the knock on effects?

– Following the generalisation of a micro-agent, if its shape is subsequently altered as a consequence of another micro-agent's actions, should that initial micro-agent be re-activated? If so, when? How do we prevent the possibility of this becoming an infinite loop?

In the third section of this paper we consider answer to these questions with respect to handling road themes. But before this, we need to consider in more detail how the agent methodology can be applied to road features.

2. Specification of the AGENT model for roads

The focus of this paper is on road independent generalisation; that is the generalisation of a single road feature with the goal of ensuring its internal legibility, without considering the symbolisation conflicts the road may have with its neighbours (an issue that is handled in [Duchêne et al. 2001]). However we still consider the surroundings of the road network, such that the road remains connected to the network after generalisation.

2.1. A localised approach to road generalisation

Two main kinds of symbolisation conflict can occur within a road, that require its generalisation: coalescence conflicts, when the symbol of the road overlaps with itself, and granularity conflicts, when the line is too detailed for the displayed scale and is thus visually 'noisy'. Inside one road, it is possible to find coalesced portions, granular portions and portions with no conflict (Fig 4).

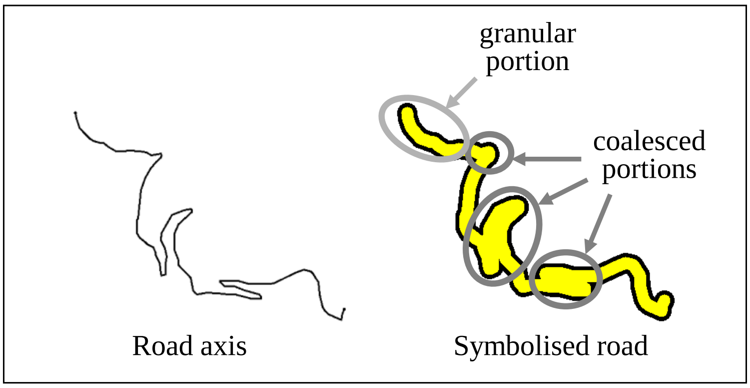

Figure 4. Legibility conflicts inside a road

Two main approaches exist to the generalisation of a section of road. The first one tries to handle the whole line via one single method (e.g. [Fritsch 1997]). A second approach works by first splitting the line into homogeneous parts with regard to their shape [Plazanet 1996] or coalescence [Mustière 1998]. It then separately handles these parts with appropriate algorithms. In other words, it tries to optimise the working space, handling the conflicts locally, as and where they occur. This local approach was preferred in the AGENT prototype because it has proved to provide good results and because it fits the AGENT philosophy: namely handling the problems where they occur, after having identified and characterised them. However, global algorithms that handle a whole line are also present in the system and can alternatively be used by the roads to generalise themselves.

Using the local approach for road generalisation means that the roads must have the capacity of splitting themselves into parts and then generalising these parts. Section 2.2 explains how this is translated in terms of levels of analysis.

2.2. Road splitting mechanism and levels of analysis.

At first, two levels of agents are considered for the road theme: the road (micro level) and the road network (meso level). We define a road from a graph theoretic perspective [Mackaness and Beard 1993; Hartsfield and Ringel 1990]: a road is an edge of the planar road graph (Fig 5). A road network is a set of connected roads. Notice that this definition of a road stands independently of the way the road graph has been split into edges.

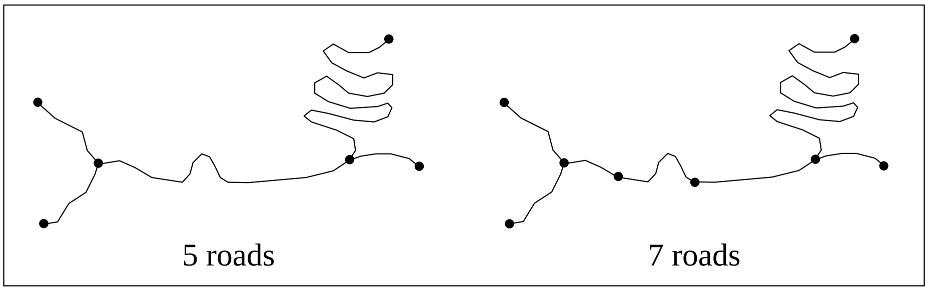

Figure 5. Delimitation of a road inside the road graph

In order for a road to split itself and generalise its parts, a third level of analysis has been added for the road theme: the "*meso-road*" level. When a road assesses its heterogeneity (as a basis for splitting), it splits itself and gives birth to a set of temporary agents: the parts of road stemming from the splitting process. These parts become the lowest level of analysis, i.e. the new micro level. They are created as "road" agents, i.e. in the same analysis level as the initial road. The initial road is then seen as a group (composed of its parts): it temporarily "transforms itself" into a meso agent. This *meso-road* agent then manages the generalisation of its parts (Fig 6). Once all the parts have been generalised, they are merged back and the road transforms itself back into a micro-agent. The *meso-road* is an example of a meso-agent that is created *top-down* and *dynamically*. Furthermore, this splitting process is *recursive*: a road stemming from a splitting can split itself again during its own life-cycle if required.

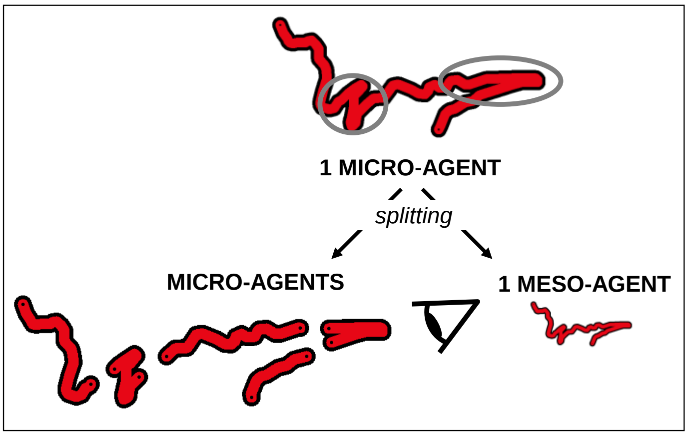

Figure 6. A micro-agent road "transforming itself" into a meso-agent to generalise its parts

Resulting Object Oriented model

The road micro-agent which has transformed itself into a meso-agent has new functionalities. In terms of an object oriented implementation, this road object should thus change class. To support this, a *meso-road* class has been added to the schema, which is a meso-level class (i.e. one that inherits from the *meso-agent* class). In reality, when a *road* transforms itself into a meso-agent, it does not really change class but is duplicated in the *meso-road* class. Thus, one object of the *meso-road* class contains the "meso" aspect of one road that wants to generalise itself by parts. For that, this *meso-road* class is linked to the *road* class by a one-to-one relation, which links the object of the *road* class to its "alter-ego" in the *meso-road* class. The objects of the *meso-road* class have no geometry, since the geometry is carried by the *road* class. Fig 7 shows the resulting class model, using standard UML (Unified Object Modelling) [Booch, Rumbaugh & Jacobson 1999]. Two associations appear between the classes *road* and *meso-road*:

– A one-to-one association that is instantiated between a *road* and its alter-ego *meso-road*,

– An aggregation that occurs between a *meso-road* agent and its *road* subparts.

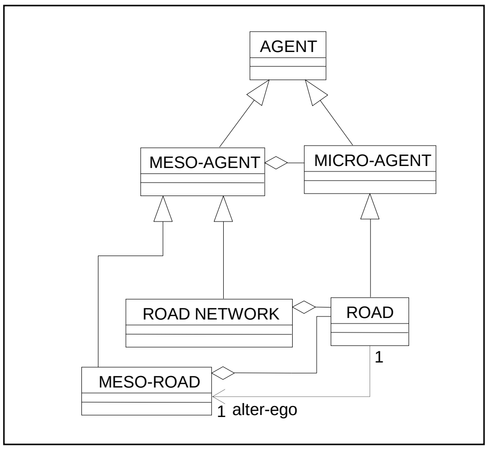

Figure 7. The object oriented model for the road theme in the AGENT prototype

In terms of dynamics, when a *road* decides to generalise itself by parts it creates its alter-ego *meso-road* agent in the *meso-road* class, in a one-to-one relation with itself. Then it stops its micro-agent life-cycle (the life-cycle is put on standby) and triggers its alter-ego *meso-road* agent, which runs its (meso) life-cycle. The first task of the *meso-road* in its life-cycle will then be to perform the splitting in order to create the parts (in the *road* class) and register them as its micro components, in order to manage their generalisation. Once all the parts have been generalised, their geometries are merged and the resulting geometry becomes the new geometry of the initial *road* micro-agent, which resumes its life-cycle. The *meso-road*, as well as the *road* parts, which were temporary agents, are deleted. The entire life-cycle of a *meso-road* is detailed in 3.1.

To summarise, three kinds of analysis levels are distinguished for roads: the *road* (micro), the *road network* (meso, composed of *roads*), and finally the *meso-road* (meso, composed of *roads* stemming from the decomposition of one *road*). During the life of the meso-road, no interaction with the network occurs. The re-connection of the generalised road to the network occurs after the micro-road has finished its generalisation, i.e. after the meso-road has disappeared (cf. 3.2).

2.3. Algorithms for road generalisation

Eight algorithms dedicated to road generalisation have been implemented within the AGENT prototype. Six basic algorithms are used to transform (smooth or caricature) either a part of a line or the whole line, one algorithm is used to split the line according to its coalescence, and another one is used to propagate knock on effects that arise due to local transformations. Constraints of space prevent the detailed description of these algorithms; our focus is on how these algorithms can be combined. However an illustration of each algorithm is provided in Fig 8, which is sufficient for the focus of this paper. The effect of each algorithm is briefly described as follow (a detailed description of each algorithm can be found in the suggested references):

– *Maximal Break* [Mustière 1998] enlarges one bend whilst retaining its shape as much as possible.

– *Minimal Break* [Mustière 1998] minimally enlarges one bend in order to make it legible. The resulting bend takes less space than if it had been enlarged with *Maximal break*, but its shape is less well preserved.

– *Accordion* [Plazanet 1996] stretches a bend series to make each bend distinct from the others.

– *Bend Removal* [Lecordix et al. 1997] removes two consecutive bends from a bend series, in order to provide space for other bends.

– The *Gaussian smoothing* smoothes all the bends of a given line.

– *Plaster* [Fritsch 1997] enlarges sharp bends and smoothes slight bends for any line (the name Plaster partially reflects the way the algorithm operates – akin to straightening something by putting it in a plaster cast).

– *Coalescence Based Splitting* [Mustière 1998] splits a given line into parts with either no coalescence, coalescence on one side, or coalescence on both sides. The splitting is based on the Hausdorff distance from the border of the symbol to the axis of the road: on coalesced parts, the border of the symbol is "far" from the axis of the road.

– *Propagation* algorithm propagates the displacement of some parts of the line to the whole line. The strength of the displacement gradually decreases along the line.

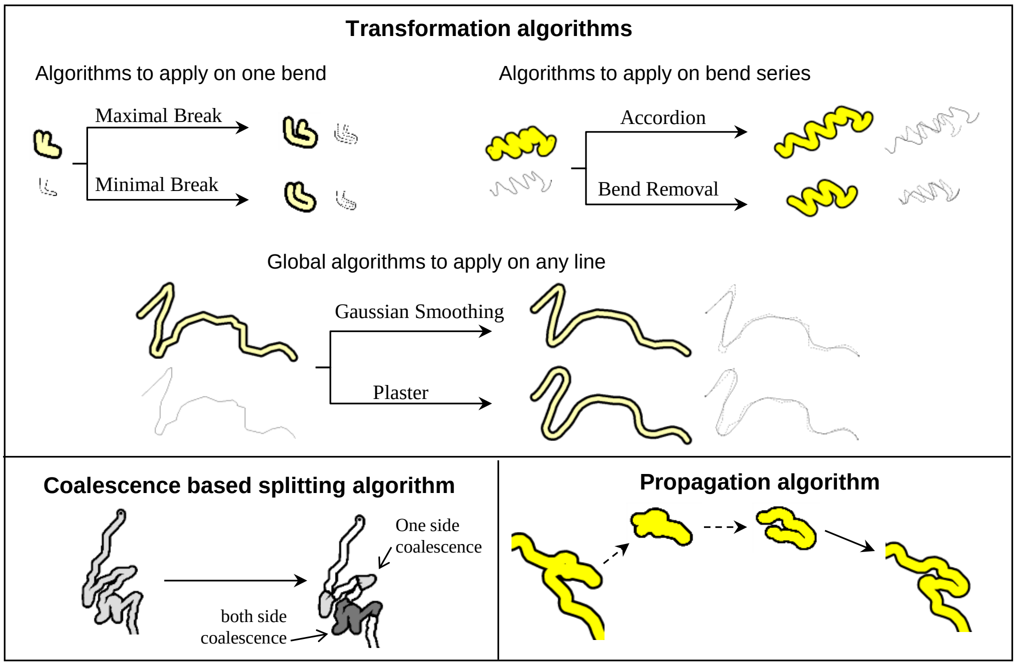

Figure 8. Road generalisation algorithms available in the AGENT prototype (figure from [Mustière 2001])

2.4. Constraints on road agents

In the AGENT model, a constraint object (also called "constraint") exists for each characteristic of the agent. When a constraint is violated, i.e. the current state of the associated characteristic is not acceptable, it proposes plans (i.e. algorithms) to the agent so that it can improve its state (cf. 1.1).

Three kinds of constraints have been identified for roads:

1. **The coalescence constraint triggers generalisation**. Two indicators are computed for this constraint:

–*coalescence side* (see coalescence based splitting introduced in previous section)

–*coalescence strength*: computed as: 1 – *road symbol area* / *road symbol expected area*,

where

road symbol area = area of a buffer of offset 0.5*road symbol width

and therefore *road symbol expected area* = road length*road symbol width + area of a circle of diameter 0.5*road symbol width (actually two half-circles, at each extremity)

With this definition, the coalescence strength value varies between 0 (no coalescence) and, theoretically, 1 (which is never reached in practice).

The coalescence strength is used to compute a degree of satisfaction (integer value from 1 to 5) for the constraint. This will influence whether an action is proposed (an action is proposed when satisfaction < 5), and whether the result of an action is considered correct during re-evaluation stage (the satisfaction must have increased at least by 1). Threshold values have been fixed by supervised machine learning for our case study (generalisation at 1:250k from a 10m resolution database) on a set of representative roads, different from the roads on which we evaluated the process to avoid over fitting. These values need to be adjusted depending on the case study as the computed coalescence strength depends on the sinuosity of roads, the symbol widths and the difference between initial and final scale. For our case study the obtained values were the following:

coalescence strength < 0.02 (or *coalescence side* assessed as “none”) => satisfaction = 5

0.02 ≤ *coalescence strength* < 0.11 => satisfaction = 4

0.11 ≤ *coalescence strength* < 0.17 => satisfaction = 3

0.17 ≤ *coalescence strength* < 0.33 => satisfaction = 2

coalescence strength ≥ 0.33 => satisfaction = 1

When the coalescence constraint is not perfectly satisfied, it proposes the following actions according to the detected coalescence side:

– Heterogeneous coalescence: *generalisation by parts* (splitting and supervised generalisation as explained above), *plaster.*

– Both-sides coalescence: *accordion*, *schematisation*, *plaster*.

– One-sided coalescence: *maximal break*, *minimal break, plaster.*

This is based on the assumption that a part of the road that is coalesced on one side (cf. Fig 8) is likely to be an isolated bend, whereas a part of road that is coalesced on both sides is likely to be a series of hairpin bends. The coalescence strength is computed as a ratio between

2. **Three constraints aim to preserve various characteristics**: they do not propose any action but can result in preventing a transformation during the re-evaluation stage (constraints for positional accuracy, internal topology, absence of “holes” in the symbol as shown by Fig 9).
3. **One constraint advises the road against triggering certain algorithms**: if insufficient space exists around the object, then this constraint does not allow the use of algorithms needing such space (*accordion* and *maximal break*).

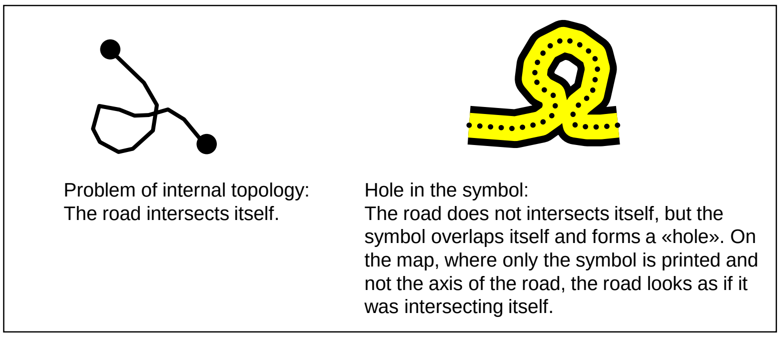

Figure 9. Problems of internal topology and "hole in the symbol" for a road

One could imagine other constraints being added. The only pre-requisite to introducing a constraint is to have a reliable measure describing the related characteristic, and to be able to translate that measure into a violation degree based on some user specifications. In the same way, other plans could be added provided the relevant algorithms are present in the system.

The choice of the constraints and associated proposed plans have been made according to the available algorithms and measures. It is interesting to note that amongst the defined constraints, no constraint triggers a smoothing operation. This is because smoothing aims to solve granularity conflicts, and no robust measure of granularity (i.e. detecting the noise inside a line) has yet been introduced into the system. Similarly, the constraint governing “holes in the symbol” does not propose any plan because no algorithm to handle this case is yet available.

3. Roads specific mechanisms for interaction between agent levels

An important concern when generalising is to minimise the knock on effects associated with modifying the form of an object (in this case roads). When a micro road is locally moved away from an overlapping road or enlarged to solve a coalescence conflict, its extremities may become disconnected from the rest of the network (Fig 10b). To preserve the connectivity, the displacement has to be propagated through the network, but decaying in a way that the change is 'absorbed' among the local space, whilst still trying to conserve the shape (Fig 10c). The challenge is to avoid creating new conflicts when propagating such displacements. In this section we describe how this problem is managed.

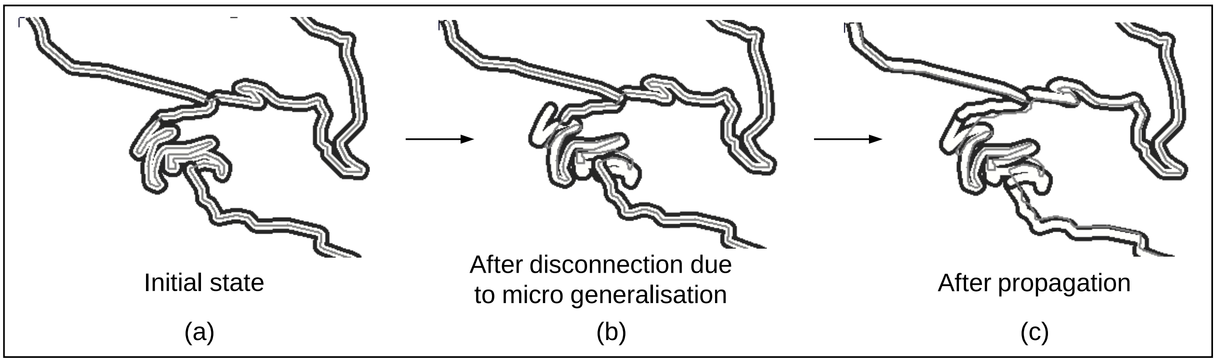

Figure 10. Propagation of a displacement through the network (the thin grey line represents the initial state, the symbolised line the current state)

3.1. Specific behaviours of the "meso-road" agent

<u>Activation and life-cycle</u>

The *meso-road* agent is activated just after its creation by the micro-*road* agent. This is a "depth first" way of activating the objects of different nested levels (an alternative approach, "breadth first", would be to handle first all the parts at a given level before handling any of their subparts). As soon as it is activated, the *meso-road* agent decomposes itself. In effect, it performs the splitting of the initial micro-*road* and creates the micro *road* agents from the resulting parts. In the same way, at the end of its life-cycle the *meso-road* "recomposes itself", merging back all its parts and returning the resulting geometry to the initial micro *road*. These two stages (decomposition and recomposition) are specific to *dynamically*, *top-down* created meso-agents. They must be added to the meso life-cycle for this kind of meso-agents (as shown in Fig 3).

Between these two stages, the life-cycle of the *meso-road* is more simple than the generic life-cycle of a meso-agent: no constraints have been associated with the *meso-road* class, its only role is to manage the generalisation of its parts. Thus, immediately after decomposition, the *meso-road* enters the cycle "Choose the best micro – give autonomy – manage side-effects", i.e. it sequentially activates its micro parts. The stages "characterise and evaluate", "propose plans", "choose best plan and trigger" and "re-evaluate", are not required. Since there are no constraints defined at the meso level, no re-evaluation is needed at this level. However, the final merged geometry will be re-evaluated by the initial micro-*road* before being accepted. The resulting life-cycle of a *meso-road* agent is shown in Fig 11.

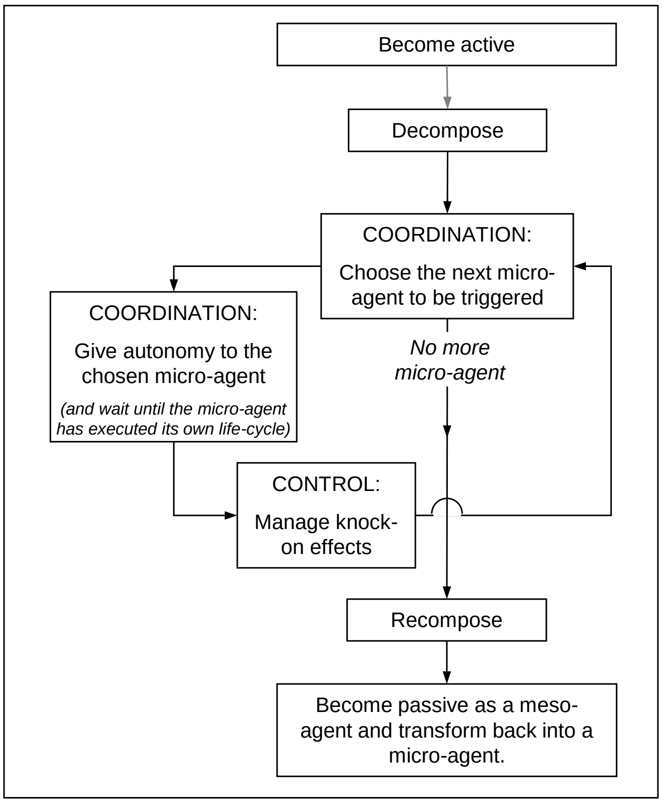

Figure 11. Life-cycle of a meso-road agent

Sequence and Autonomy: which micro-road to handle next ?

At every cycle the next best micro-*road* candidate to be triggered is dynamically computed by the *meso-road* agent. The possible candidates are the parts of the *meso-road* that meet the two following conditions:

- they are not yet in a perfect state, and
- they have not just been handled without reaching a perfect state.

To achieve this, every *road* agent is marked with a boolean flag "unable to reach a better state", initialised to *false,* which is turned to *true* at the end of its life-cycle if the final state reached is not perfect. So, the possible candidates are the parts which are not in a perfect state, and which have their boolean flag set to *false*. Amongst those possible candidates, the best one to generalise first is the one that is deemed to be the most coalesced, i.e. the one with the highest violation degree of its coalescence constraint. This rule of thumb is followed because the more a road is coalesced, the more difficult it is to generalise without creating knock on effects. Generalising a very coalesced road last might easily damage the optimal state already reached by its neighbours. The basic philosophy is to deal with the worst case first.

Propagation of knock on effects inside a meso-road

At every cycle, the side-effects management stage consists of reconnecting the rest of the network to the newly generalised micro-road, using the *propagation* algorithm mentioned in 2.3. Once the micro-road life-cycle is finished, the *meso-road* agent checks if the extremities of the micro *road* agent have been disconnected from its neighbours. If this is the case, it computes a new geometry for all of its micros, using the propagation algorithm, in order to reconnect them while cushioning the deformation.

The propagation algorithm used here works recursively. From one point of the road line that is displaced, it computes a new position for the next point on the road line, given the displacement of the previous point and the current position of the point. The next point is displaced in the same direction as the previous point, but the displacement is a bit smaller compared with the previous point (there is a kind of "return spring" in its current position). Similarly a new position is computed for the following point, and so on until the displacement has been totally cushioned. Fig 10 illustrates the problem of propagating the displacement, and also shows example output of this algorithm.

Triggering a micro several times

It is sometimes the case that a micro-*road* that has first been successfully generalised is later deformed by the propagation of a neighbour. In this case, this micro-road can be triggered again: the meso-road detects that the micro has been deformed during the propagation and changes its boolean flag “unable to reach a best state” to *false*, once again making it a possible candidate for generalisation. However, to avoid falling into a state of deadlock, the system only allows this to occur once.

3.2. Re-connecting a generalised road to the rest of the network

As illustrated in Fig 10, sometimes a displacement inside a "part of road" is not yet cushioned at the end of the "parent road", (the road that has given birth to this part by splitting). In this case, at the end of its own life-cycle this "parent" road is not connected to the surrounding network any more. It must therefore be re-connected by the meso-agent that has triggered it, - the road network meso-agent. How to reconnect a part of road to the other parts of the same road, and how to reconnect a road to a road network, are different issues. The main difference between the meso-road and the road network is that a meso-road is a linear graph (non cyclic, where the micro-roads have only one neighbour at each extremity), whereas inside a road network nodes can be of any valence, and include cycles. Thus the propagation of a road displacement to the whole road network is more complex.

The propagation through the whole network is systematically performed after each road generalisation. *The road network* agent is in charge of checking if the newly generalised road has been disconnected from its neighbours, and then computing a new geometry for the other roads to ensure any propagation.

The road network meso-agent computes a new geometry for each of the roads concerned with the propagation. The road network agent then passes the geometry it has computed to each road. To do this the road network meso-agent triggers a particular life-cycle on each micro-*road*, called "reactive life-cycle", with the proposed geometry as an argument. During the reactive life-cycle, the road re-evaluates its state with the proposed geometry, but considers only its most important constraints. In the current system only one constraint is considered: the internal topology constraint, which checks that the road does not intersect itself. But other constraints could be added according to user needs. During this stage the road is in a “reactive” state: it can react to the proposal of the network by accepting or rejecting the proposed geometry, but it does not act autonomously. Currently, if a road rejects its “propagated” geometry, the whole propagation is backtracked, as well as the micro generalisation at the commencement of the propagation. The process is computationally intensive. A possible improvement to the system would be to enable a road that had rejected a proposed geometry to find for itself another solution. Another idea would be to enable negotiations between the road and the network agents, but further research is needed to achieve this.

4. Worked example

Fig 12 presents results showing the evolution of one road during its generalisation by the AGENT process. The road is shown in its initial state inside the road-network (1), and then in isolated form for the sake of illustration.

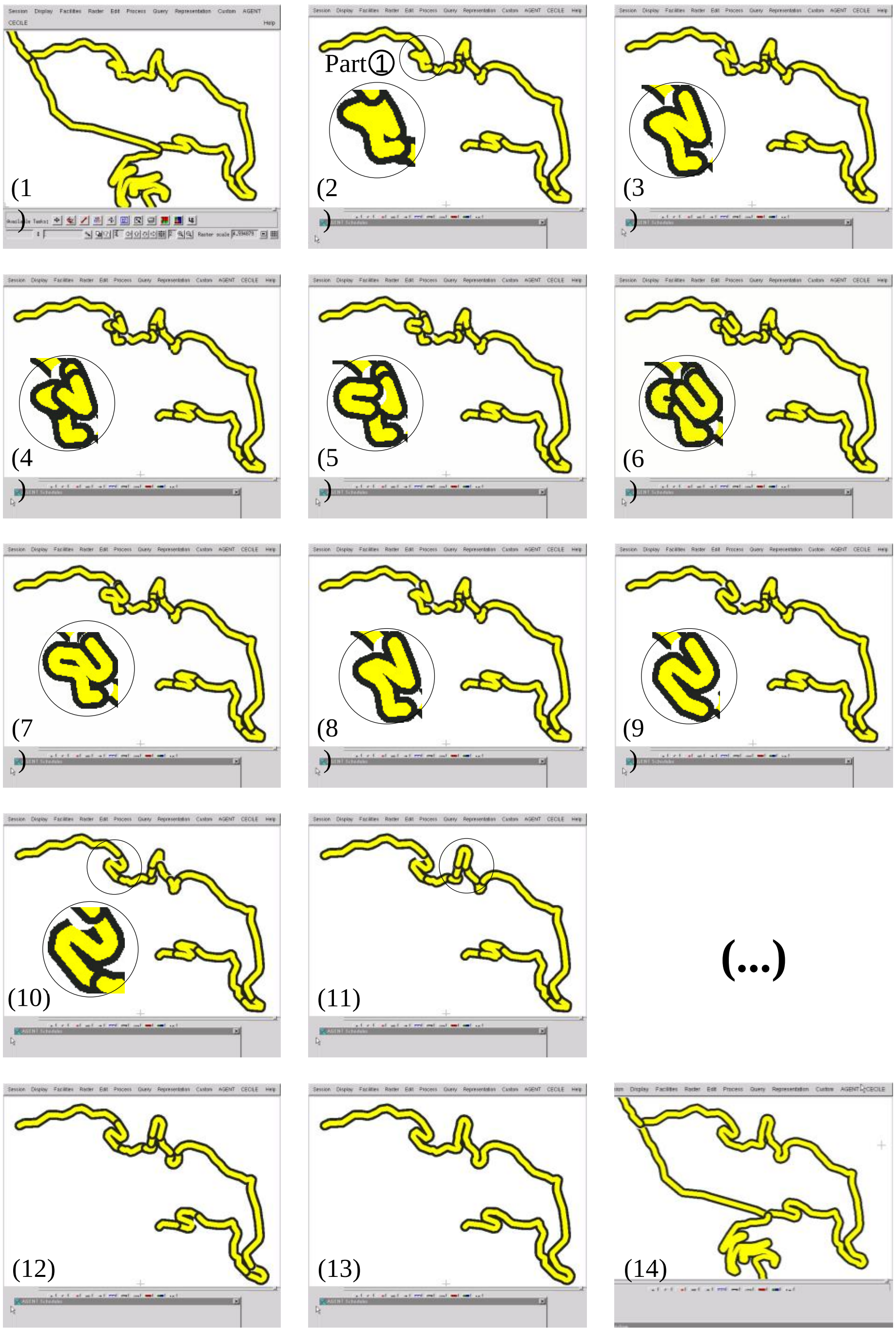

Figure 12. Generalisation of one road by the AGENT process

During its characterisation, the coalescence constraint detects its violation. As the coalescence is measured as heterogeneous, it proposes to the road a *generalisation by parts*. The road then becomes meso, splits itself (2) and computes the autonomy order of its parts. The first part to be handled (let us call it part ①) is the series of two bends in its north-west section (this part ① has been enlarged in the first ten images in order to illustrate the changes). Part ① is triggered as a micro-agent, and it runs its life-cycle. Its coalescence constraint detects its violation and, as it is a both-sides coalescence, it proposes to trigger *accordion*, which is successfully applied (3). The result of the re-evaluation is "better, but not perfect": the summits of the bends are still a bit coalesced, so that now, the coalescence has become heterogeneous (coalesced around the summits, not coalesced at the base of the bends). The coalescence constraint of part ① thus proposes to go on with *generalisation by parts*. Part ① then splits itself again (4), successfully applies *minimal break* on the two bends that were coalesced on one side (5-6), and then merges its parts back (7). But the re-connection creates a hole in the symbol (7), which is detected at the re-evaluation of part ①. As there is no algorithm available to solve this kind of conflict, the constraint looking for holes in the symbol has nothing to propose and it rejects the treatment: part ① thus backtracks to state (3) (frame 8 is actually the same as frame 3). Then the second proposal of the coalescence constraint at state (3), which was *plaster*, is tried (9). The re-evaluation considers that part ① has reached a perfect state. Thus part ① warns the meso-road that it has finished, and the meso-road propagates the displacement of its extremities to the other parts (10). Then the meso-road computes the next best candidate for generalisation amongst its parts: it is the big bend close to part ①, which is thus triggered. This bend runs its life-cycle. Its coalescence constraint proposes {*maximal break* or *minimal break* or *plaster*}, but the constraint that assesses whether there is sufficient space around the bend (described in paragraph 2.4 / 3) finds there is little space available and thus advises against *maximal break*. Thus, *minimal break* is applied, which results in a perfect state for the bend (11). The meso-road then re-connects it and the generalisation carries on. The following stages are not detailed in Fig 12. Once all the parts have been successfully generalised (12), the meso-road merges them back together (13), stops its life-cycle, and disappears, leaving its micro alter-ego (the initial road) to re-evaluate the result. The result is considered perfect, so that the road stops its micro life-cycle and alerts the network that is has finished. The network then propagates the (slight) displacement of its extremities to the other roads (14).

This example shows that the generalisation process of a road in the AGENT prototype is quite long, even on roads of medium complexity. A backtrack could have been avoided at stage (7) if an algorithm to remove the hole in a symbol had existed, in which case the shape of part ① would have been better preserved. However the result is still considered to be a good one.

Fig 13 shows the result once the other roads have also been generalised. Notice that no smoothing has been applied, since no constraint is able to detect a need for a smoothing.

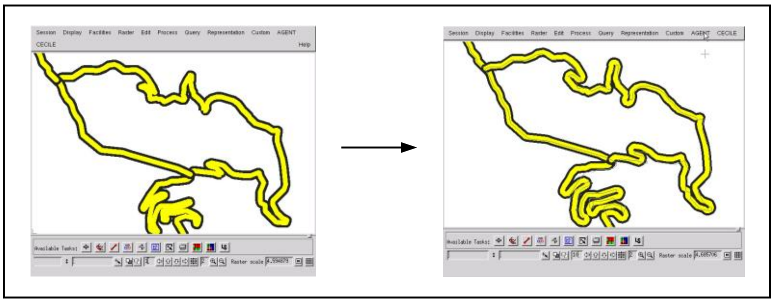

Figure 13. The set of roads before and after generalisation

5. Discussion

5.1. Some other results

Fig 14 shows results obtained with the same process on a set of winding mountain roads extracted from BD CARTO ®, the French 10 m resolution database, for a generalisation at 1:250k. The results obtained with this process are good: the legibility of the road symbols is ensured, which is the crux of road generalisation. The shapes of the roads are well maintained except in a few cases (region surrounded by a circle in Fig 14). The weakest point of the results is the lack of smoothing.

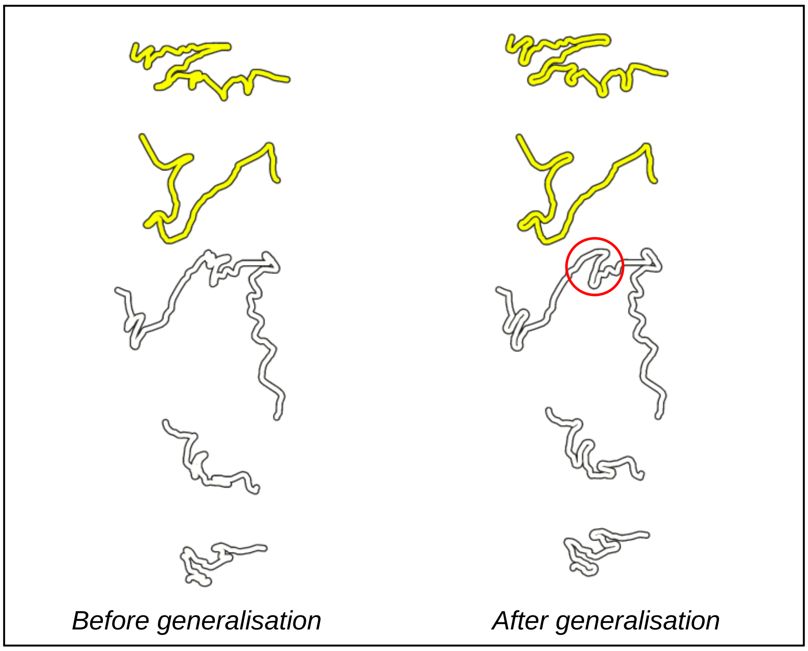

Figure 14. Some results of individual road generalisation with AGENT

5.2. Comparison with other road generalisation approaches

It is interesting to briefly compare this approach with three other approaches.

A typical approach consists of systematically applying an algorithm to the entire extent of the data. For road generalisation, algorithms like [Douglas and Peucker 1973] and its many variants are part of this approach. They handle all the points of a road line in the same way. This kind of approach works for very small changes of scale, where only a very slight filtering is needed. For bigger changes of scale, the main drawback of this approach is that no analysis of the features is performed, so that the main characteristics of these features are lost.

A second generalisation approach proceeds by minimisation techniques. All the constraints applying to a feature or a set of features are mathematically modelled. It results in an over constrained system of equations, that is solved using minimisation methods such as least squares [Harrie 1999] or finite elements [Højholt 2000]. [Bader 2001; Bader and Barrault 2001] applied this technique to road features, with a additional focus on the modelling of the road geometry (using the metaphor of metallic beams). This approach works very well for moderate scale changes, or as a final process once there is enough space for all the objects (or all the details of an object). The reason for that is that this approach only performs continuous changes, such as displacement or deformations. It does not perform discrete operations such as object removal or, for road generalisation, bend removal.

Finally, a third approach considers generalisation as a step by step process, where the state of the data changes progressively by application of localised operations in order to solve conflicts. This approach was proposed by [Brassel and Weibel 1988; McMaster and Shea 1988]. The model of [Ruas 1999], as well as the AGENT model are based on this approach. This approach relies on a characterisation of the geographical space before generalisation, in order to perform it where it is needed, as it is needed. It enables a better adaptation to the data and allows discrete operations like removals or typification. However it does require specific measures of spatial analysis, generalisation algorithms are needed for each kind of geographical feature and it is necessary to formalise all the knowledge used in chaining the operations.

The third approach, which is the approach used in the work presented here, is more appropriate other than for cases of very small changes in scale.. It produces a higher quality solution since conflicts are specifically identified and handled by dedicated algorithms.

Our approach is complementary to that of [Bader 2001], which uses minimisation techniques. [Bader 2001] approach handles both the internal generalisation of the roads and the conflicts of proximity or overlapping between two roads. We can only compare the internal generalisation aspect of this approach with the approach presented here. For moderate scale changes, or when there is enough space to preserve all the details of the road geometry, the results are comparable to ours.. In terms of robustness, the process presented here looks for "the best possible solution that respects a minimal set of constraints". It thus can reject globally a solution because it would locally create a problem such as a "hole in the symbol", even if everywhere else inside the road the legibility has been improved. On the contrary, the process of [Bader 2001] seeks the "best possible solution", so that it always find a solution, even if some local problems remain. [Bader 2001] process is therefore more robust.

For significant changes of scale, the process presented in this paper has the advantage of including two generalisation algorithms dedicated to the caricature of portions of road for cases where insufficient space exists: "bend removal", that typifies a series of hairpin bends, and "minimum break", that minimally enlarges an isolated bend. These two algorithms mimic operations usually performed manually by cartographers when space is limited. Such caricature operations do not exist within the solution proposed by [Bader 2001]. The process presented here is therefore better suited for significant changes of scale.

Finally, we can compare the process presented in this paper with another road generalisation process that uses the same approach and, moreover, the same generalisation algorithms, but that combines these algorithms differently. In this process, the algorithms are triggered by a rule based engine, where the rules have been set up using supervised machine learning. The methodology used to set up those rules are described in [Mustière, Zucker & Saitta 2000], and its application to road generalisation in [Mustière 2001]. [Mustière and Duchêne 2001] provides a detailed comparison of this process and the AGENT process presented in this paper. It was found that in the AGENT process, the mechanism for choosing the best algorithm is not very robust and consequently lots of tries are performed before finding the best solution. This arises because it is not possible to know beforehand when the best possible state has been reached. So, unless it reaches a perfect state, the generalisation process only stops when all the possible combinations of algorithms have been tried. This is in contrast to [Mustière 2001] solution, whereby the learnt rules choose algorithms that are likely to lead to a quick solution, and where some rules indicate when to stop without checking that the state reached is perfect. This process therefore converges more quickly to an acceptable solution, even if sometimes a better solution could have been found by the AGENT process. But contrary to our current model where each constraint proposes its own algorithms, in the learnt rules all the characteristics of a road are taken into account in determining which algorithm is chosen next. [Mustière 2001] solution treats the constraints as being interdependent, whilst AGENT considers constraints to be independent (for ease of modelling and maintenance). Because of this difference of modelling, the two philosophies are difficult to integrate.

Conclusion and perspectives

We have presented a new approach for the cartographic generalisation of road features, based on the use of multi-agent techniques. This approach uses concepts based on previous research: constraint based generalisation, autonomy and levels of analysis [Ruas 1999], and road generalisation using a localised approach lying on coalescence-based splitting [Mustière 1998]. The results show this to be a very promising approach both for road generalisation and for generalisation in general. Furthermore, the system is very flexible, insofar as it supports the addition of new algorithms, and design constraints.

Some areas of further refinement were identified in the evaluation phase. First, the system would benefit from a measure to detect granularity conflicts (in order to know when to trigger a smoothing operation), as well as a measure of shape comparison and an algorithm to handle the cases of a “hole in the road symbol”. Such an algorithm would prevent the system from rejecting a globally good solution where a local conflict of "hole in the symbol" had appeared.

Secondly, the process by which the "a priori best algorithm" is chosen is not optimal and consequently many tries and backtracks are performed before reaching the best possible solution, especially when no perfect solution exist (i.e. a solution where all the constraints are satisfied). A possible solution is to relax the constraints associated with coalescence. Another solution would be to incorporate some more robust rules governing the choice of algorithms (using the philosophy of [Mustière 2001]) but this is not straightforward.

Concerning the contextual aspect of the work presented here, i.e. the propagation of a road transformation to the road network, as a rule of thumb, the approach is to identify and rank road sections according to their ‘severity’. The idea being that the worst cases are in need of radical change, others less so. Whilst this philosophy produces workable results, it could be improved by considering additional factors such as the immediate spatial surroundings, and the anticipated plans for other road sections in that ordering process.

More generally, in the field of contextual generalisation, improving the decision making by a collective effort among a group of agents is a crucial issue. This process can take place at two stages: before triggering a transformation, using co-ordination and planning techniques, and dynamically during the transformations, by introducing negotiation mechanisms. We are currently beginning research on these two themes.

Add-on – August 2014

Regarding the perspectives identified at the end of this research:

- The main issues addressed during the industrialisation stage [Lecordix et al. 2005] were: intensive testing and debugging of algorithms on a large amount of data; adjusting the threshold values for coalescence satisfaction to the considered case study (final scale = 1:100k instead of 1:250k); limiting the number of times a road can be recursively split to 2; as no satisfying measure of granularity could be set up, adding a mechanism ensuring that any portion of road that has no coalescence or no more coalescence, is smoothed.
- Further research have been led on contextual generalisation and collective decision making between agents (roads or others). The main addressed issues where: handling constraints shared by two agents by means of dialog (CartACom model) [Duchêne 2003; Ruas and Duchêne 2007; Duchêne et al. 2012]; mixing agent approaches and strength-based mechanical approaches to handle fields background objects like the relief and their consistency with vector foreground objects (GAEL model) [Gaffuri 2007; Gaffuri et al. 2008]; orchestrating a complete generalisation process while ensuring a synergy between several agent- or not agent-based processes on different zones of a dataset and/or at different stages of generalisation (CollaGen model) [Duchêne and Gaffuri 2008; Touya et al. 2010; Touya and Duchêne 2011]; enabling different kinds of interactions between agents within a unique agent framework [Duchêne and Gaffuri 2008; Maudet et al. 2013; Maudet et al. 2014].

Acknowledgements

I would like to warmly thank William Mackaness for its thoroughly reading of this paper and his many suggestions, and for having helped me to respect the basics of English grammar.

References

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

Bader, M., & Barrault, M. (2001). Cartographic Displacement in Generalization: Introducing Elastic Beams. *4th ICA Workshop on generalisation*, Hong-Kong, available on the web site of the ICA Map Generalisation Commission, http://generalisation.icaci.org/images/files/workshop/workshop2001/bader_barraultv1.pdf (accessed August 2014).

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, Chine, 2001, pp. 2110-2116.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Language User Guide*. Addison-Wesley, Reading, MA, 1999.

Douglas, D.H., & Peucker, T.K. (1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or it's Caricature. *The Canadian Cartographer journal*, 10(2), pp.112-122.

Duchêne C. (2003). Automated map generalisation using communicating agents. Proceedings of the 21st International Cartographic Conference, Durban, South Africa, 2003, p.160-169.

Duchêne C., Barrault M., Haire K. (2001). Road Network Generalization: A Multi-Agent System Approach. In Proceedings of the 20th International Cartographic Conference, vol.3, Beijing, 2001, pp. 2166-2177.

Duchêne C. & Gaffuri J. (2008). Combining three multi-agent based generalisation models : AGENT, CARTACOM and GAEL. In Proccedings of the 13th International Symposium on Spatial Data Handling, Montpellier, France, 2008, p.277-296.

Duchêne C., Ruas A., Cambier C. (2012). The CARTACOM model: transforming cartographic features into communicating agents for cartographic generalisation. International Journal of Geographical Information Science, Volume 26, Issue 9, September 2012, pp. 1533-1562.

Ferber, J. (1999). *Multi-agent Systems. An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.

Fritsch, E. (1997). Représentations de la Géométrie et des Contraintes Cartographiques pour la Généralisation du Linéaire Routier, PhD Thesis, université de Marne-la-Vallée, France.

Gaffuri J. 2007. Field deformation in an agent-based generalisation model: the GAEL model, In Proceedings of GI-days 2007 - young researches forum, IFGI prints vol.30, p.1-24.

Harrie, L. (1999). The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization. *Cartography and Geographic Information Science*, vol.26, n°1, 1999, pp.55-69.

Hartsfield, N. & Ringel, G. (1990). *Pearls in Graph Theory – A comprehensive Introduction*. Academic Press Inc., Boston, 1990.

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

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W.A., & Weibel, R., (1999). The Application of Agents in Automated Map Generalisation, *Proceedings of the 19th International Cartographic Conference*, Ottawa, 1225-1234.

Lecordix F., Jahard Y., Lemarié C., Hautboin E. (2005). The end of Carto 2001 Project: Top100 based on BDCarto® database. Proceedings of the 9th ICA Workshop on Generalisation and Multiple Representation, La Coruña, Spain, 2005. http://generalisation.icaci.org/images/files/workshop/workshop2005/Lecordix_Jahard_et_al_.pdf (accessed August 2014).

Lecordix F., Plazanet C., Lagrange J.P. (1997). A Platform for Research in Generalization: Application to Caricature. GeoInformatica International Journal, vol.1, n°2, 1997, pp.161-182.

Mackaness, W.A., & Beard, M.K. (1993). Use of Graph Theory to Support Map Generalisation. *Cartography and Geographic Information Systems*, vol.20, n°4, 1993, pp.210-221.

Maudet A., Touya G., Duchêne C., Picault S. (2013). Improving multi-level interactions modelling in a multi-agent generalisation model: first experiments. Proceedings of the 16th ICA Workshop on Generalisation and Multiple Representation, jointly organised with the ICA Commission on Map production and Geo-Business, Dresden, August 23-24, http://generalisation.icaci.org/images/files/workshop/workshop2013/genemappro2013_submission_2.pdf

Maudet, A., TouyaG., Duchêne C., Picault S. (2014). Representation of Interactions in a Multi-Level Multi-Agent Model for Cartography Constraint Solving. Advances in Practical Applications of Heterogeneous Multi-Agent Systems: Proceedings of the 12th International Conference PAAMS 2014, Lecture Notes in Computer Science vol. 8473, Springer, p.183-194.

Mustière, S. (1998). GALBE : Adaptive generalisation. The need for an Adaptive Process for Automated Generalisation, an Example on Roads, *Proc. of 1st GIS'PlaNet conference*, Lisbon (Portugal), CDROM, 6 p.

Mustière, S., Zucker, J.-D., & Saitta, L. (2000). An Abstraction-Based Machine Learning Approach to Cartographic Generalisation. *Proc. of 9th International Symposium on Spatial Data Handling*, Pékin, sec.1a, pp.50-63.

Mustière, S. (2001). Apprentissage supervisé pour la généralisation cartographique, PhD Thesis, Université Paris VI, France.

Mustière, S., & Duchêne, C. (2001). Comparison of different approaches to combine road generalisation algorithms: GALBE, AGENT and CartoLearn. *4th ICA Workshop on generalisation*, Hong-Kong, available on the web site of the ICA Map Generalisation Commission, http://generalisation.icaci.org/images/files/workshop/workshop2001/mustiere_duchene_v1.pdf (accessed August 2014).

Plazanet, C. (1996). Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application au routes). PhD Thesis, Université de Marne-la-Vallée, France.

Regnauld N. (2001). Constraint based mechanism to achieve automatic generalisation using agent modelling. *Proc. of 9th Annual Conference on GIS Research in United Kingdom*, Glamorgan, 2001.

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

Ruas, A. (1999). Modèle de généralisation de données géographiques à base de contraintes et d'autonomie, PhD Thesis, Université de Marne-la-Vallée, France.

Ruas, A. (2000). The role of meso objects for generalisation, *Proceedings of the International symposium on Spatial Data Handling*, Beijing, sec. 3b, p.50-61.

Ruas A. & Duchêne C. (2007). A Prototype of Generalisation based on the Multi-Agent System Paradigm. In: Mackaness, W., Ruas, A., Sarjakoski, T. (eds): The Generalisation of Geographic Information : Models and Applications, Elsevier, 2007.

Touya G., Duchêne C. (2011). CollaGen: Collaboration between automatic cartographic generalisation processes. In A. Ruas (Ed.), Advances in Cartography and GIScience, Lecture Notes in Geoinformation and Cartography, vol.1, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 541-558.

Touya G., C. Duchêne, A. Ruas (2010). Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes. In S. I. Fabrikant, T. Reichenbacher, M. van Kreveld, C. Schlieder (Eds) Geographic Information Science, LNCS 6292, Springer, pp. 264-278.

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

Wooldridge, M., & Jennings, N. (1998). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, vol.10, n°2, pp.115-152, 1995.