Road Network Generalization: A Multi Agent System Approach

Cécile Duchêne[1], **Mathieu Barrault**[2], **Kelvin Haire**[3]

[1] IGN/SR/Cogit Laboratory
2 avenue Pasteur
94165 Saint-Mandé Cedex
France,
cecile.duchene@ign.fr

[2] Department of geography
Spatial Data Handling
University of Zurich-Irchel
Winterthurestrasse 190
CH-8057 Zurich
Switzerland,
fax : +41 1 635 68 48
barrault@geo.unizh.ch

[3] Laser-Scan Limited,
Cambridge Science Park,
Milton Road,
Cambridge. CB4 0FY
United Kingdom
kelvinh@lsl.co.uk

Abstract

This paper intends to introduce an original infrastructure based on a multi-agents systems approach for automating the generalisation of the road network , a key issue for an efficient map. This paper takes place in the context of the European research project AGENT (ESPRIT 24939). It highlights the need for a multi-level analysis. It demonstrates how the multi-agent based infrastructure used meets the requirements of road network generalisation and illustrates with results the relevance of this approach.

Keywords: Road Network Generalisation, Automated Generalisation, Digital Cartography, Vector data, GIS, Multi-agents System.

Introduction

Road network plays a key role in a map. It supplies accessibility to thematic spots and spatially structures the drawn area. Its symbolisation reflects such a role and require space the scale change can not supply. Its generalisation is based on displacement, exaggeration and, for high scale change, removals. Previous research provided fruitful algorithms and processes ([Nickerson 88], [Brazile 98], [Thompson & Richardson 99]), but they cannot be used successfully without first identifying the right moment and the right working space to apply them Moreover, variety of situations to manage require adapted algorithms and adequate set up.

This paper presents the research led on road network automated generalisation during the European project AGENT (ESPRIT 24939). The AGENT project approach relies on two principles highlighted by [Ruas 99]. The first one consists in considering the geographical features as local decisional entities which act to generalise themselves according to knowledge and capacities of analysis. The second principle consists in explicitly distinguish several levels of analysis of the geographical features : the individual objects of the database (one building, one road…), but also groups of spatially organised objects (a town, a road network…).

In the first part of this paper we remind the cartographic rules which justify a generalisation of the road network. We then introduce the baselines of the multi-agents infra-structure which has been developed during the AGENT project, and highlight how the necessary multi-level analysis is handled. We then present how this infra-structure has been specified for the generalisation of the road network, with appropriate levels of analysis: the cluster of roads level, the road level and the road section level. Finally we present results on real subsets of data.

1. Road network generalisation: the cartographic point of view

When generalising a road network, both independent and contextual generalisation are to be dealt with. Independent generalisation of roads aims to solve legibility conflicts occurring inside one single road feature. Contextual generalisation aims to solve conflicts between road features (e.g. over-density of roads, overlapping between roads). Relevant cartographic constraints at the road and the road network levels are briefly described hereafter.

1.1. Individual road cartographic constraints

Two main kinds of symbolisation conflicts can occur inside a road, justifying its generalisation: coalescence conflicts, when the symbol of the road overlaps itself, and granularity conflicts, when the line is too detailed for the displayed scale and thus looks noisy. Inside one road, it is possible to find coalesced portions, granular portions and portions with no conflict (Figure 1).

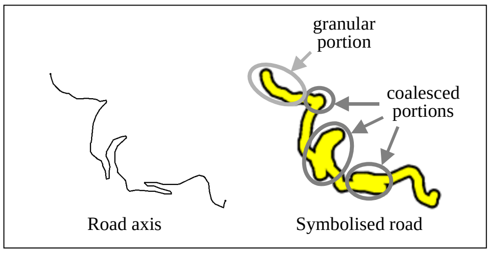


Figure 1 - Legibility conflicts inside a road

The generalisation, while reducing these legibility conflicts, should also respect the users specifications describing the acceptable degradations in terms of shape and positional accuracy. Moreover, the generalisation should never create incoherencies (the road intersects itself).

1.2. Road network cartographic constraints

Increasing road symbols creates overlaps between close road sections (Figure 2) which mislead to wrong topological relations. It also hampers junctions configuration analysis since it conceals the smallest road sections. Angles between roads and event right junction locations are falsified (Figure 3). Minimum legible distances must be provided between road sections and minimal extents to connecting segments ensured by displacing road vertices. Topological characteristics must be preserved. In worst cases, acceptable degradations may lead to remove less important roads. Routing efficiency and global density heterogeneity must also be guaranteed.

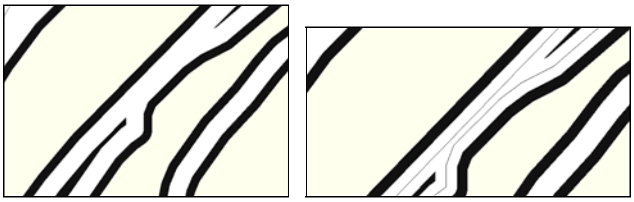


Figure 2 - Road symbol Overlaps: the two white roads seem to be connected while they only have very close sections

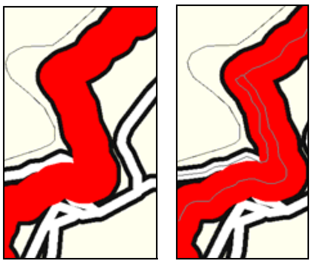


Figure 3 - Road Junction conflict: the left white road seems to end in the bottom curve of the biggest road while it is actually connected to the top curve.

2. Principles of the AGENT project's prototype

2.1. Agents and constraints

In the model designed for the AGENT prototype, the geographical entities have been designed as agents. In Artificial Intelligence, agents are objects that have a goal and a certain autonomy to reach this goal. The geographical agents are described by a set of characters that constrain the generalisation operation, either because they should trigger the generalisation (e.g. the size of a building, when too small), or because they could be damaged by the generalisation (e.g. the global shape, the positional accuracy). The pertinent characters have been identified for each geographical theme. Each geographical agent is guided by a set of "constraints" objects that act as advisers, each of them watching on a particular character of the object and proposing possible plans (i.e. generalisation algorithms) to apply in order to improve the state of this character. The aim of the agent is to satisfy as well as possible all its constraints.

To achieve this aim, the agent has the capacity of choosing a plan amongst those proposed by the constraints (the *a priori* best plan), applying this plan, and *a posteriori* evaluating the improvement. Moreover a "first-depth search" mechanism, which enables to backtrack to any state and try other plans, ensures that the system reaches the best possible solution according to its evaluation criteria. Figure 4 shows the generic life-cycle of an agent when it is activated (see [Regnauld 01] for more details).

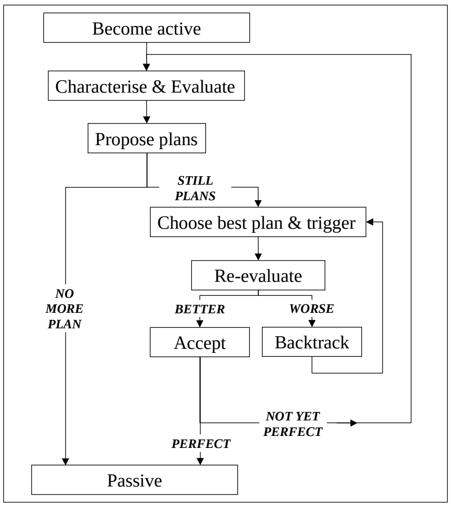


Figure 4 - Generic life-cycle of an agent

This behaviour of the agents supposes to have, in addition to the mechanism itself, a given number of tools on the platform: measures to characterise the objects and generalisation algorithms to transform them. It also implies to have formalised knowledge: which measure is the most appropriate to quantify each character? which algorithm is adapted to which problem? how to compare two states of the agent (what to conclude if a character has been improved and another one damaged)?

2.2. 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 simple objects' level: some operations are performed on groups of spatially organised objects (e.g. a group of buildings aligned along a road), others are performed on parts of an object (e.g. a series of hairpin bends inside a road). In the AGENT prototype, these levels of geographical analysis have been explicitly distinguished. The lowest level of analysis, which contains the single objects, is called the "micro" level. The groups level is called the "meso" level. Thanks to this structure, we can formalise the role of the groups of objects identified as pertinent for generalisation. As the meso level is not present in the geographical databases, we have to identify the pertinent types of meso objects and construct these objects using methods of spatial analysis. Several nested meso levels can be defined, e.g. a town (meso) contains districts (meso) that 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 (it is the case of the town and the districts), or it can be built *dynamically* during the generalisation, when the need occurs (e.g. a meso object "group of aligned buildings" can be created during the generalisation of a district).

2.3. Behaviour of the meso level of analysis

The meso level has several roles for generalisation [Ruas 00]. First it is responsible for the generalisation operations occurring at its level (e.g. elimination of objects inside a group). Secondly it manages the generalisation of its components through three possible behaviours:

- the co-ordination, where it activates the micros trying to optimise the order of activation (since the order of activation greatly influences the management of side-effects),
- the control, where it manages the side-effects of the micro generalisation,
- the legislation, where it gives orders to the micros or changes their constraints or goals, either to help them to solve a conflict they cannot solve themselves, or to solve over-constrained situations.

To be able to manage the generalisation of its components, on top of its normal agent characteristics a meso-agent has specific functionalities like autonomy order computation, micro-agents triggering, side-effects management. These functionalities can be specified on 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, as shown in Figure 5. 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*, which have different methods. The fact that a given type of geographical objects is micro or meso makes the corresponding geographical object class inherit from either *micro-agent* or *meso-agent*.

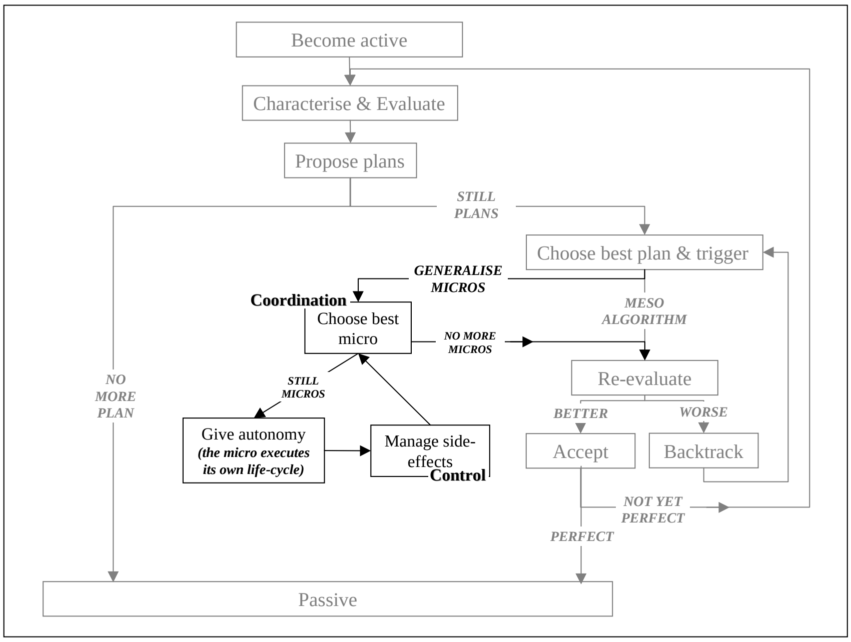


Figure 5 - Life-cycle of a meso-agent

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

– to which micro(s) to give autonomy at a given time ? how to find the “best autonomy order” ?

– how to manage the side-effects ? how to preserve the characters of the micros when propagating side-effects, who is responsible for this propagation ?

– if a micro-agent has been generalised at a time of the process but is damaged by another micro-agent’s side-effects propagation, shall we activate it again? When ? And then, how to ensure the process does not fall into an infinite loop ?

Part 3 of this paper specifies the AGENT model for the road theme and presents the way these questions have been answered for this theme.

3. Specification of the AGENT model for the road theme

Two levels of agents, corresponding to the two relevant levels of analysis identified in Part 1, are considered for the road theme: the road (micro-level) and the road network (meso-level). What we call a road is an edge of the road graph of the database, which is supposed to be planar. A road network is a set of connected roads.

In 3.1 and 3.2 we present how the AGENT model has been specified for each of these two levels (available algorithms, constraints and behaviour). Then in 3.3 we explain how the two level interact, since the *road network* as meso-agent is responsible for the management of its micro *roads* components.

3.1. Constraints and behaviours of the road agent

Two main approaches exist to generalise a road. The first one (e.g. [Fritsch 97]) tries to handle the whole line with one algorithm. The second one splits the line into homogeneous parts with regard to the shape [Plazanet 96] or the coalescence [Mustière 98]. It then separately handles these parts with appropriate algorithms. In other terms, it tries to optimise the working space, handling the conflicts locally where they occur. This local approach has been privileged in the AGENT prototype because it has proved to provide good results and because it fits to the AGENT philosophy: handle 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.

Algorithms available for road generalisation

Eight algorithms dedicated to road generalisation are present on 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 the coalescence, and another one is used to propagate side effects due to local algorithms transformations. These algorithms are illustrated in Figure 6. The detailed descriptions of these algorithms can be found in the following papers: [Plazanet 96] for *Accordion*, [Lecordix, Plazanet & Lagrange, 97] for *Bend Removal*, [Fritsch 97] for *Plaster*, [Mustière 98] for *Maximal Break, Minimal Break* and *Coalescence Based Splitting*.

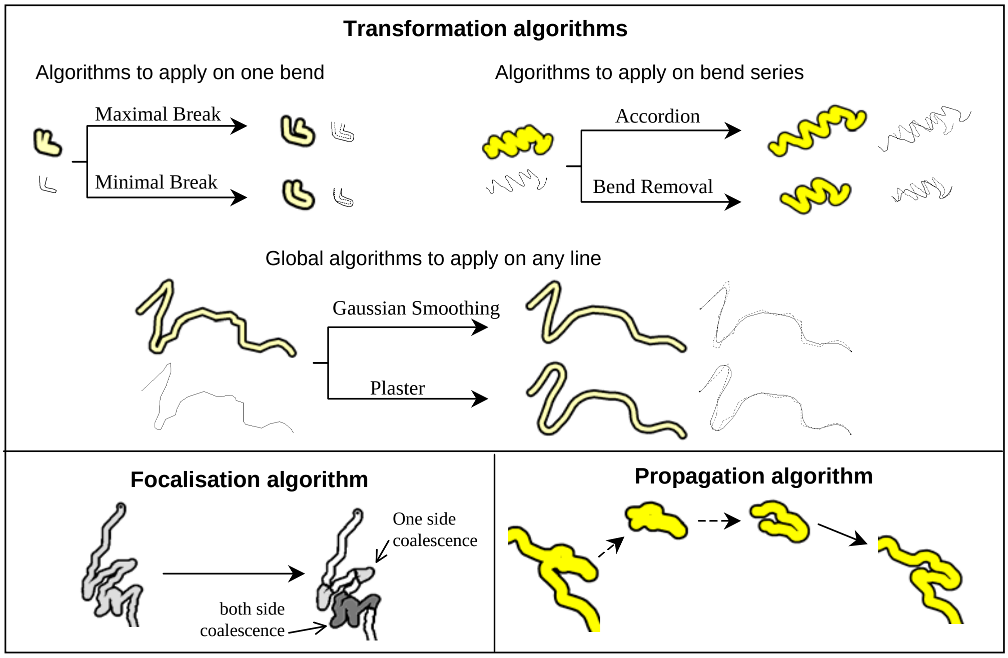


Figure 6 - Road generalisation algorithms available in the AGENT prototype

Constraints designed on the road agents

In the AGENT model, a constraint is in charge of watching on the state of one character of the agent. When a constraint is violated (i.e. the current state of the associated character is not acceptable), it proposes algorithms to the agent so that it improves its state (cf. 2.1).

Three kinds of constraints have been identified for roads:

1. The coalescence constraint triggers the generalisation. According to the side and strength of the detected coalescence it proposes
   - Heterogeneous coalescence: *generalisation by parts* (splitting and supervised generalisation as explained below), *plaster*
   - both-sides coalescence: *accordion, schematisation, plaster*
   - One-side coalescence: *maximal break, minimal break, plaster*
2. Three constraints aim to preserve characters: they do not propose any action but can lead to refuse a transformation during the re-evaluation stage (constraints for positional accuracy, internal topology, absence of "holes" in the symbol).
3. One constraint looking for the available space around the road advises it against the algorithms needing too much space. This constraint is not taken into account for the re-evaluation.

The choice of the constraints and associated proposed plans has been made according to the available algorithms and measures. Let us notice that amongst the defined constraints, no constraint triggers a smoothing operation. This is because the smoothing aims to solve granularity conflicts, and no robust measure of granularity (i.e. detecting the noise inside a line) has been introduced in the system for the time being. In the same way, the constraint looking for "holes in the symbol" does not propose any plan because no algorithm to handle this case is available.

Road splitting mechanism and associated levels of analysis.

In order to enable a road to split itself and generalise its parts, another level of analysis has subsequently been added: the "meso-road" level. When a road detects its heterogeneity and splits itself, it gives birth to a set of temporary agents: the parts of road stemming from the splitting. These parts become the lowest level of analysis. 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, in order to get the meso functionalities necessary to control its parts. This "meso-road" then manages the generalisation of its parts, successively giving autonomy to each of them (Figure 7). Each part which is given autonomy triggers its micro life-cycle to generalise itself, exactly as the initial road did. 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 which 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 needed.

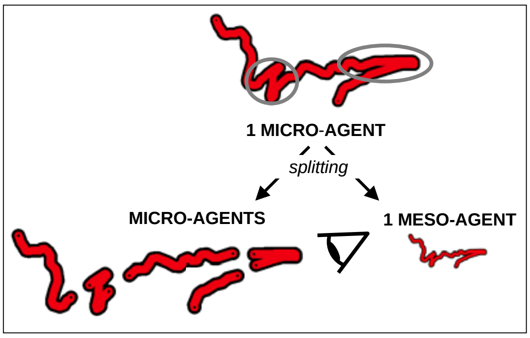


Figure 7 – A micro-agent road “transforming itself” into a meso-agent to generalise its parts

3.2. Constraints and behaviours of the road network agent

Algorithms available for road network generalisation

Most behaviours for the road network generalisation attempt to shift a heavily conflicting vertex of a road polyline.

Five algorithms are used at the road network level, to solve conflicts involving several roads:

– *Displacement* shifts apart a vertex, in order to enables to remove overlap between roads symbols.

– *Propagation* aims to propagate the displaced vertices through the rest of the road so as to preserve each road’s shape characteristics. This propagation is performed in a cushioned way so as neither to translate the whole dataset nor breaking the shape. The cushioning is adapted to each line’s shape (Figure 8). If shape can not fully cushion the shifting, its nodes are then shifted.

– *Diffusion* aims to diffuse node displacements among the connected roads and the roads graph until the initial shifting is fully cushioned (Figure 9), in the case where road shape characteristics constrain the *propagation* not to stop to the ends of the road.

– *Node junction emphasis:* Edge of an hampered connecting road can be stretched or rotated so as to identify its orientation against the other connected roads. The vertex is then shifted, the shifting propagated and diffused. For the very small edges, it can be merely removed if the resulting new edge preserve a close orientation.

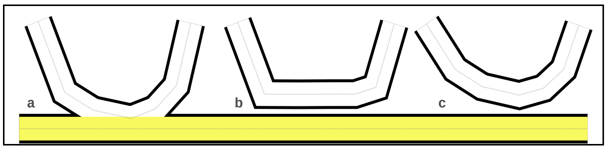


Figure 8 – Propagation: shifting only too close vertices (a) break the shape (b).
Shifts must then be propagated to the rest of the road (c).

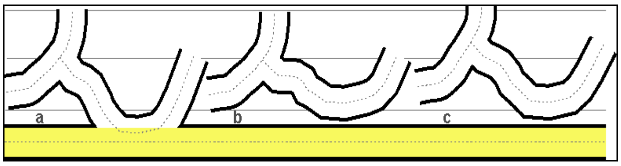


Figure 9 – Diffusion: Propagation cushion initial shifted vertices. But short sections cushion too much (b) and break their shape. The cushioning must instead be shared by the connected roads(c). Initial shifts is then diffused along nodes of the road network.

Constraints designed on the road agents

Four constraints are designed on the road network agent:

1. Topology Constraint: It is in charge of guarantying no roads intersect nor one is self-intersecting. It also preserve connections. This constraint is boolean. It doesn’t propose plans but only reject an applied plan if it violates one of its conditions (and so the agent).

2. Junction Clearance constraint: This constraint is in charge of ensuring each junction configuration's clarity. Proposed plans elongate and rotate extreme vertices of roads in conflict to ensure all orientations to be red, and so the junction's location.
3. Roads Overlap constraint: It checks that no road symbol overlaps another one. Although the previous constraint can be considered as a particular cases of this one, tests quickly revealed that a dedicated constraint was more adapted. It proposes plans which shift one or several vertices, propagate them and diffuse if required. It should be able to propose roads to be removed when possible or to require the micro-agent to modify the road so as to remove conflicts.
4. Micro-roads satisfaction constraint: It checks the satisfaction of the roads components of the road network meso-agent, ensuring the communication between the meso and the micro level. It is detailed in part 3.3.

As for the roads, the choice of the constraints and associated proposed plans has been made according to the available algorithms and measures. Thus no constraint is in charge of the over-density issues, because no pruning algorithm is available in the AGENT prototype at the moment.

How the road network generalisation happens

Road network generalisation is based on local interventions. Constraints detect conflicts and propose one or several plans for each of them. To optimise results convergence, road generalisation is first prioritised and when nodes are displaced, diffusion is triggered. Then, since most of overlapping roads are also connected, their junctions emphasis are first solved when needed. If one shifted vertex does not improve results, the conflicting opposite road is then displaced, or another conflict is managed or a road generalisation is triggered and the current conflict handled later on.

3.3. Interaction mechanisms between road and road network levels

Three kinds of analysis levels are finally 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*). But during the life of the *meso-road*, no interaction with the *road network* occurs. The interactions between a *road* and the *road network* only take place before or after the life of the *meso-road* associated with the *road*. These interactions occur when the *road network* triggers the *road* (then the *meso-road* does not exist yet), and when it re-connect the generalised *road* to the network after it has finished its generalisation (then the *meso-road* has disappeared).

Giving autonomy to a road so that it generalises itself

In order to trigger its roads components, the road network has to know whether these roads are in a satisfying state or not. This is done by means of an additional constraint on the road network: the Micro-roads satisfaction constraint, looking to the satisfaction state of its micro roads. This constraint is considered violated if at least one micro road is not fully satisfied. The proposed plan consists in triggering one road agent amongst the not satisfied ones, and then diffuse the possibly resulting displacement of the extremities of this road. In order to ensure the process does not fall into an infinite loop, only the road agents that have not yet tried to generalise themselves can be triggered.

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

The *diffusion* through the whole network is systematically performed after each road generalisation. The road network 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 the propagation.

During a *diffusion* performed by the road network due to the displacement of a road, the other roads can be more or less damaged. The only feature able to detect this kind of degradation is the damaged road itself: as the possibly damaged constraints are designed at the road level, the road network has no direct access to them. Moreover, because of the road network complexity, many roads are concerned with a diffusion through the network, so that we cannot wait for the re-evaluation at the road network level to decide a diffusion has to be backtracked. For these reasons, the roads must be enabled to refuse a geometry computed by the network, during the diffusion.

This is done by means of a micro "reactive" life-cycle: after computing the new geometries for the roads, the network triggers each of them, proposing them the new geometry. The roads re-evaluate themselves with the proposed geometry, but considering only the most important constraints. As "most important constraints", we consider at least the internal topology constraint, which checks that the road does not intersect itself. But other constraints can be added according to the user needs. During this stage the roads are in a "reactive" state: they can react to the proposal of the network by accepting or rejecting the propose geometry, but they do not act autonomously. For the time being, if a road rejects its "diffused" geometry, the whole diffusion is backtracked, as well as the micro generalisation that was at the origin of the diffusion. It is a heavy process in terms of computation time, this is why the diffusion can only be rejected by irreversible constraints violations, that no algorithm can remedy. A possible improvement of the system would be to enable a road rejecting a proposed geometry to find itself another solution, given the new co-ordinates of its extremities. Another idea would be to enable negotiations between the road and the network, but further research is needed to achieve this.

4. Results

Some results of road independent generalisation are shown in Figure 10. The results have been evaluated by a cartographer of IGN-France, who considers them as very good, except the lack of smoothing and in some few cases a bad preservation of shapes (region surrounded by a circle in Figure 10).

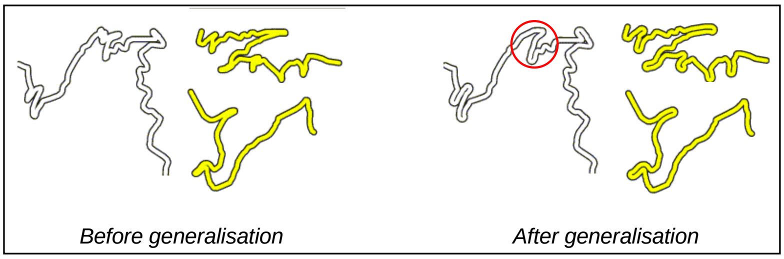


Figure 10 - Some results of individual road generalisation with AGENT

An example showing how the legibility of nodes junctions is improved is given in Figure 11. A full agent process is shown on a sub-set of roads in Figure 12. It took 45 steps for solving overlaps and junctions identifications. In this example no road required micro-generalisation.

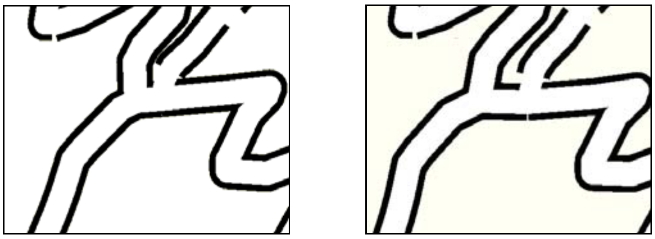
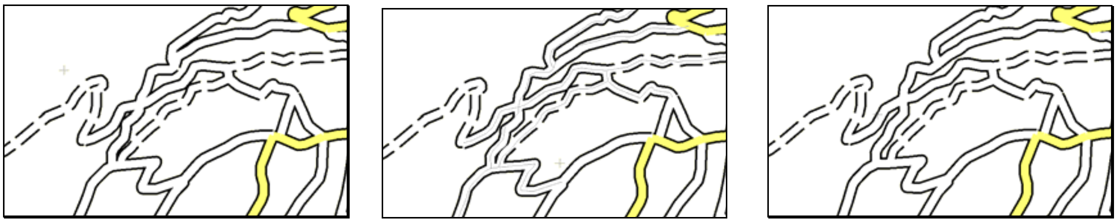


Figure 11 – Junction Configuration highlighted by the Road Network Agent (before, after)



Initial State *After 20 steps* *Final State (45 steps)*

Figure 12 - A cluster of roads generalised by a Road Network Agent. It took 45 steps to clean up junctions and overlaps. No backtracks were required. No road generalisation were required here.

Conclusion and perspectives

This paper introduced an original multi-agents infrastructure which allows to manage the road network generalisation, by providing different levels of analysis. Agents are delegated to the cluster of road level which is in charge of solving contextual conflicts or delegate an agent to a road. This latter one analyses the whole road and may delegate agents to sections when shapes are too complex. Each agent triggers the *a priori* best generalisation algorithm or tries alternative plans if needed (other parameters or another algorithm). Each upper agent analyses results and manage side-effects or rejects the lower agent's decision.

If road generalisation provides very efficient result, the higher level generalisation has to be improved. The introduced local interventions are only one of the possible behaviours that exist. This infra-structure has been designed to allow an easy integration of better existing or future generalisation algorithms. Optimisation techniques like snakes ([Burghardt & Meier 97], [Bader & Barrault 00]) are very promising. But higher levels generate much more complex and various situations to solve which necessitate an accurate spatial analysis, and a fine tuning of the relevant algorithms. Another issue to improve the system capabilities is to introduce more robust and efficient knowledge to take the good generalisation decision at a given time. This better knowledge can be set up by machine learning techniques [Mustière 01]. A further step would be to directly integrate machine learning techniques (and not only resulting knowledge) inside the system, so that it learns on-line. It would be a means to improve the necessary reverse-engineering process of the cartographic generalisation task.

References

BADER M., BARRAULT M., 2000, “Improving Snakes for Linear Features Displacement in Cartographic Generalisation”, 4th International Conference on GeoComputation proceedings (CD-ROM - 10 pages), August 2000.

BRAZILE, F., 1998, "A generalisation Machine Design that Incorporates Quality Assessment", Proceedings of Spatial Data Handling, Vancouver, Canada, pp.349-360.

98 , p. 349-360.BURGHARDT, D. and MEIER, S., 1997, “Cartographic Displacement using the Snakes Concept”, In: Foerstner W. and L. Pluemer (eds): Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, Birkhäuser, Basel, Switzerland.

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.

LAMY, S., RUAS, A., DEMAZEAU, Y., JACKSON, M., MACKANESS, W.A., and WEIBEL, R., 1999, “The Application of Agents in Automated Map Generalisation”, Proceedings of the 19th International Cartographic Conference (Ottawa), 1225-1234.

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., 2001, “Apprentissage supervisé pour la généralisation cartographique”, PhD Thesis, Université Paris VI, France, to be published.

NICKERSON, B.G., 1988, “ Automated Cartographic Generalization for Linear Features”, Cartographica, vol. 25(3), pp.15-66.

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.

THOMPSON, R.C., RICHARSON, D.E., 1999, “The ‘Good Continuity’ Principle of Perceptual Organisation applied to Generalisation of road Networks”, Proceedings of the 19th International Cartographic Conference, Ottawa, Canada, 1215-1225.

REGNAULD, N., 2001, “Constraint based mechanism to achieve automatic generalisation using agent modelling”, Proceedings of 9th Annual Conference on GIS Research in United Kingdom, Glamorgan.

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.