

INTEGRATING MULTI-AGENT, OBJECT-ORIENTED, AND ALGORITHMIC TECHNIQUES FOR IMPROVED AUTOMATED MAP GENERALIZATION

Mathieu BARRAULT (1), Nicolas REGNAULD (2), Cécile DUCHENE (3), Kelvin HAIRE (4),
Christof BAEIJS(5), Yves DEMAZEAU (5) , Paul HARDY (4) , William MACKANESS (2),
Anne RUAS (3) and Robert WEIBEL (1)

- (1) Department of Geography, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland
(2) Department of Geography, University of Edinburgh, Drummond St, Edinburgh EH8 9XP, Scotland UK
(3) Institut Géographique National, Laboratoire COGIT, 2 Avenue Pasteur, B.P. 68, 94160 Saint-Mandé, France
(4) Laser-Scan Ltd., Cambridge Science Park, Milton Rd, Cambridge CB4 4FY, UK
(5) Institut National Polytechnique de Grenoble, LEIBNIZ/IMAG, 46 Avenue Felix Viallet, 38031 Grenoble, France

Abstract

This paper summarizes some of the key architectural elements and reviews some of the results of the AGENT project, a research project funded by the European Commission between 12/1997 and 11/2000 as part of the Esprit Programme. The project involved five partners and successfully developed a prototype system to deal with automated map generalization. The prototype is based on Laser-Scan's object-oriented GIS and map production system LAMPS2, uses a multi-agent approach to model cartographic objects and their treatment in generalisation, and uses various newly developed geometric algorithms to measure and transform the cartographic objects in the generalisation process. More information about the project can be found at <http://agent.ign.fr/>.

Keywords: Automated Generalisation, Digital Cartography, GIS, Multi-Agent Systems.

Introduction

The background, organisation, and the key design elements of the AGENT project have already been reported at the last ICA Conference (Lamy et al. 1999). This paper gives an update after the conclusion of the three-year project and concentrates on architectural considerations of the multi-agent system (MAS) approach used as well as on some of the project results. The project focused on topographic map generalisation and used the French topographic map series as an example. For experimentation, BDCarto® and BDTopo® databases of IGN France were used. A prototype system was implemented on the basis of Laser-Scan's LAMPS2 object-oriented map production system.

In contrast to previous research projects in automated map generalisation, AGENT attempted to address the generalisation process in a comprehensive and holistic manner. While new algorithms for spatial analysis and generalisation were developed during the project, the focus was mainly on building a framework that would allow to choose the right algorithms given a particular problem situation. For that framework, we have found MAS to be a particularly suitable computational paradigm as it has allowed us to match conflict detection (via spatial/geometrical analysis; e.g., detection of overlaps or congestion, object alignments, etc.) with conflict resolution (via generalisation algorithms; e.g., simplification of building outlines, road caricature, etc.) and decompose the problem to different levels of granularity (e.g., the level of an individual building, the level of a city block, the level of a town).

The AGENT Framework

Main principles

The framework is based on a hierarchical multi-agent system and was inspired by the work of Ruas (1999). Each agent aims to improve the situation (a sub-set of conflicting features) with respect to its attached constraints. Both agents and constraints are instances of base classes (cf. Figure 1). Each agent object has referenced constraints objects. But sub-classes of agents may occur, as described below.

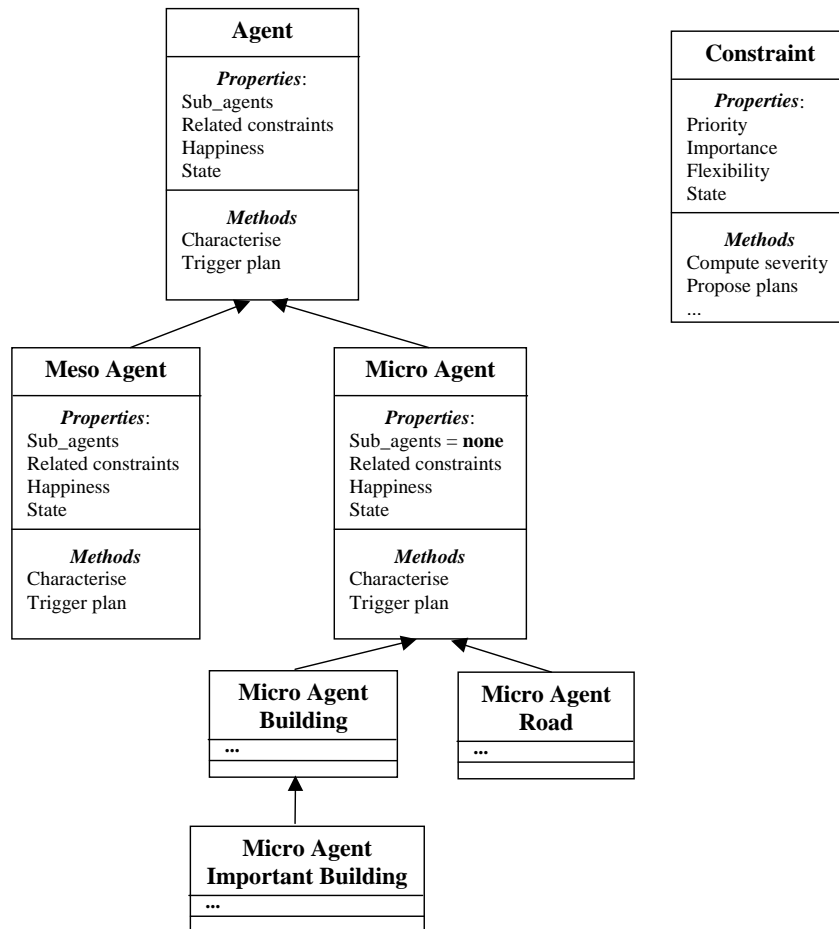


Figure 1 : Main classes – Constraints, agents and examples of sub-agents

Constraints

Each constraint relates to a property of an agent that has to be controlled during generalisation. It can be a graphic legibility rule or an information to emphasise, but it turns out processing constraints may also occur to help the agent to converge faster. Each constraint inherits from the constraint class. It provides a property “goal value” which a set of values the agent attempts to reach by generalising. The method “compute current value” returns a description of the actual state of the situation against this constraint. The method “compute severity” returns a value which characterises the happiness of the constraint, computed against the goal value, the current value, and a possible flexibility

which allows the agent to make compromises between its associated constraints. Properties “importance” and “priority”, respectively, are used to rank the constraint satisfaction of the agent and to help it sorting all the proposed plans of the various constraints. Depending on the severity and the current value of each constraint, a set of sorted proposed plans which improve the constraint happiness is then proposed to the agent.

The Agent

The agent aims to satisfy its constraints. But most generalisations are a compromise between information preservation and legibility rules. All agents inherit from a base class. But as in oriented-objects models, more specialized agents may overload the default methods or get further attributes to benefit from the acquired knowledge and converge faster to a better solution.

Basic Behaviour

The agent characterises its state by consulting all of its constraints. It integrates their severity values and importance to establish its own happiness. The default method to compute the happiness is the sum of the severities, weighted by their importance. When unhappy, the agent collects the plans proposed to remedy a constraint violation. In order to improve the search for the best sequence, the agent sorts them with respect to their own ranking, constraint’s priority and severity.

The default method applied is :

$$100 * \text{constraint_priority} + 10 * \text{constraint_severity} + \text{algorithm_order}$$

where the three variables are defined within the range [0,10].

The agent then triggers the first plan and re-evaluates its own happiness. Depending on the success of the plan, as illustrated in Figure 2, it may backtrack and trigger the next plan, stop if it reaches a perfect state, or start a new cycle if the happiness has improved.

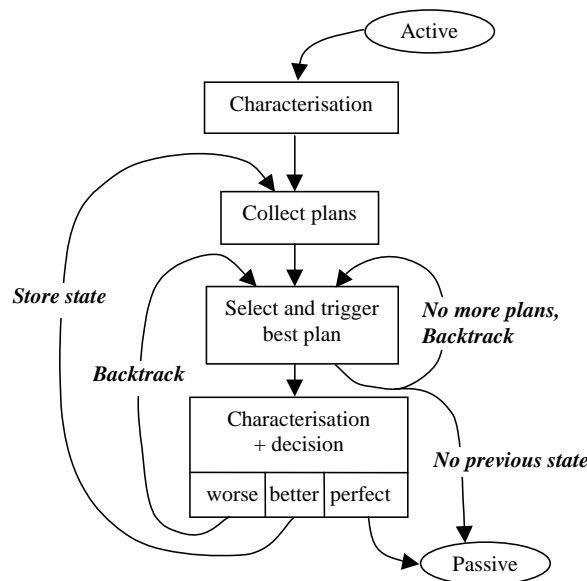


Figure 2: The Agent Cycle

Improved Depth First Search

Such a cycle corresponds to a depth first strategy among a tree of plans which either stops when a perfect state is found or proceeds through the entire tree. In this latter case, the agent will then return the best state it found during the search.

First results highlighted the need for a smarter decision strategy of the agent. The initial requirement of a monotonous improvement of all constraints punished many more efficient sequences which would cause a temporary deterioration of constraints. The agent decision should accept constraint deteriorations as long as the constraint which submitted the applied plan is improved.

To avoid getting caught in a loop, another condition is added to the first one, the constraint proposing the plan must be improved. The new state must provide a set of severities that no previous state can compete with: If there exists a previous state where each severity is equal or better than the state presently evaluated, the agent considers the situation as not improved.

Events List

In certain situations, obvious major priorities within the generalisation process occur. This knowledge is included through the “importance” property which can be used to weight the importance of the various constraints and algorithms. An events list has been attached to the agent. Each item stores a set of constraints. The agent will go through each item, trigger its cycle against the corresponding list of constraints, and return the best improved situation. It then steps to the next list of constraints with this new situation.

Hierarchical MAS

The multi-agent system was built upon two principles.

Divide and Conquer

Many spatial relationships between geographical features, reflected in generalisation knowledge, allow to divide the conflicts and to solve them independently. For instance, city blocks bounded by streets can be dealt with separately. An agent gets attached a processing constraint whose plan spatially divides up the set of features and delegates a sub-agent to each sub-set. The correlations between sub-sets are controlled through the overwritten properties values and methods of the constraints attached to the sub-agents. They are also controlled through the other constraints of the upper agent.

Delegate an Expert

The better known the situation (i.e., an accurate spatial analysis through adequate measures and known generalisation requirements with reliable fine-tuned algorithms was carried out), the smaller the tree and thus the faster the convergence and the better the solution.

The initial configuration of our system was built on a hierarchy consisting of two lower levels (meso and micro level) and an upper level (macro level) used to monitor the lower levels.

The *micro level* is dedicated to *independent generalisation*: Each feature is dealing with its inner conflicts, without consideration of spatial context. A micro-agent is attached to each map feature (e.g., building, road between two intersections).

The *meso level* was then handling the *contextual generalisation*, i.e. conflicts between features. A meso-agent was delegated to each main theme (i.e. a city, a rural road network, etc.). However, having only very few meso-agents would result in overwhelming complexity. Hence, the meso-agents can recursively sub-divide themselves if needed. A typical case is the heterogeneity of conflicts requiring different treatment in different parts of the meso-agent. For instance, a road first leading through a plane then over a mountain chain would consist of a part with low sinuosity and a highly sinuous part with hairpin bends, both requiring different treatment. In such a case, the meso-agent splits itself up and delegates sub-agents with specific requirements, first regarding the situation at the global level (initial meso-agent), then focusing on more local characteristics and conflict solutions.

The hierarchical multi-agent system is then built on conflicts handling knowledge and possible spatial divisions. Each agent, advised through its constraints, may either trigger a lower agent or modify the lower agent's set up. It also has the option to manage conflicts itself and then give orders to the lower agents.

The highest level, the *macro-level* of the MAS, owns only one macro-agent which only transmits user requirements by setting up the constraints of the top-level meso agents before triggering their life-cycles.

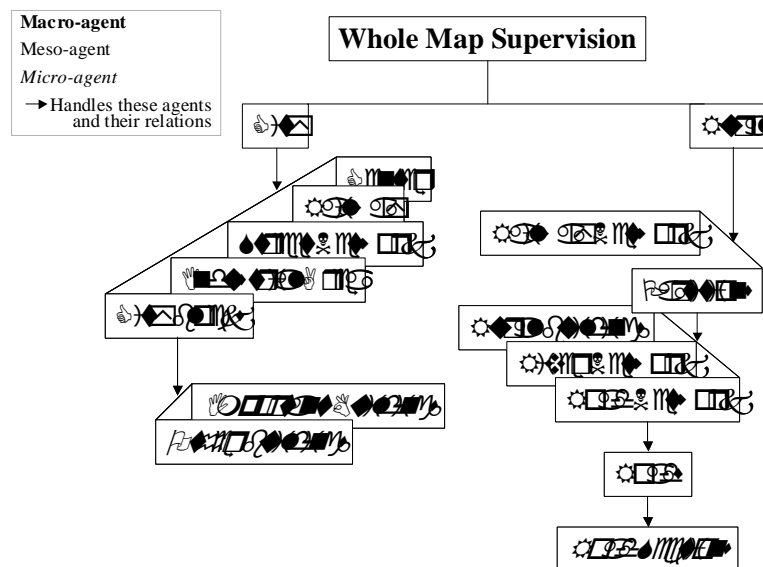


Figure 3 : A possible instance of hierarchical agents

Results

Building Micro-agent

A micro-agent is delegated to each building. Cartographic rules require that the building has minimum size, granularity (size of edges), and inner distances. In most cases, angles must be squared. Finally, the shape must be preserved, monitored here by a concavity measure. Five constraints are thus attached to the agents in this case. They are set up against the theme of the building class and the requirements of the managing meso-agent (city-block).



Figure 4 : Building generalisation by a micro-agent (5 constraints: Size constraint, Granularity constraint, Squareness constraint, Width constraint, Concavity constraint)

*11 states were attempted – The fifth one is the best solution
(all constraints are happy but the concavity)*

Road Micro and Meso-agent

The road micro-agent incarnates the trade-off between efficient generalisation algorithms and the agent management. Each road is associated to an agent. Granularity must be ensured and coalescence avoided. If the shape is simple enough, existing and simple algorithms like smoothing and simplification can solve cartographic conflicts. Otherwise, no algorithms exist to fully solve conflicts along mountains roads, for instance. A constraint is dedicated to detect the complexity of the shape. The micro-agent is turned into a meso-agent which splits up the road into more homogenous sections and delegate a micro-agent to each of them. After having triggered them, it reconnects each section. If the final shape is too much modified, it requires another generalisation from the micro-agent. Each micro-agent's constraints are set up depending on the homogeneity of the corresponding road section. A road micro-agent has thus a shape complexity constraint, a coalescence constraint and a granularity constraint. It also supports shape preservation constraints (positional accuracy, inner topology and monitoring of loops). Figure 5 illustrates some of the steps of the road agent (see Duchêne et al. 2001 for further details). A road agent is triggered by the road network meso-agent.

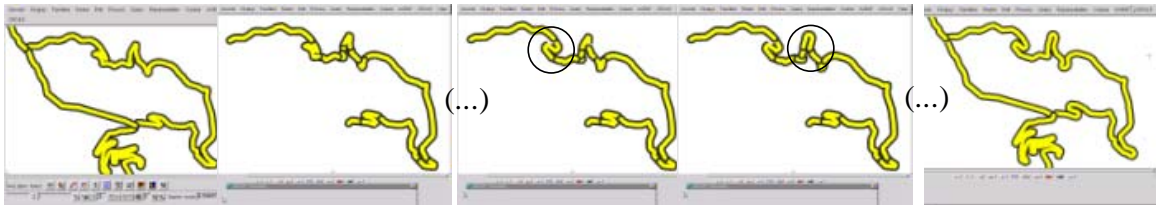


Figure 5 : The road “micro-meso” agent. The initial road is split into more homogeneous parts so as to be better generalised. Each coalescence is solved with the adequate algorithm. The meso-road reconnects all pieces of generalised road sections. The whole process requires 14 steps shown are steps 1, 2, 10, 11, 14).

Road Network Meso-agent

The road network must avoid its road symbols to overlap with one another or to hamper the readability of junction configurations. Its generalisation may remove roads (for large scale reductions), or displace them. But the generalisation must preserve road shapes, graph topology coherence, density through the graph and the accessibility of nodes (e.g. settlements) in the network. The road network has thus a density constraint, an overlap constraint, a junction clearance constraint, but also a graph-topology constraint and a road generalisation constraint. In order to speed up, it also has a portioning constraint when the pruned road network dataset is still very large, which divides space into partitions and delegates a sub-road network meso-agent to generate it. Each road network meso-agent is in charge of triggering the road micro-agent (see Duchêne et al. 2001 for further details).

City-block Meso-agent

The city block has to deal with overlaps that occur between buildings and surrounding street symbols, and between buildings. The meso-agent is in charge of displacing, eliminating, typifying or aggregating buildings. Constraints handled are (road-building proximity, building-building proximity, building density, and building generalisation constraint). They are triggered by the city meso-agent.

City

The city meso-agent is in charge of decreasing the density of objects so as to get enough room to generalise the remaining objects. But it must ensure to preserve its characteristics: density heterogeneity, neighbourhood types (centre, suburbs, industrial areas), preserving important buildings, etc.

Constraints attached to the city are: Street network density, area heterogeneity detection and city-block generalisations. The city districts are characterised (Edwardes and Regnaud 2001). The street network is then pruned against this characterisation (Edwardes and Mackaness 2001). Then meso-agents are delegated to city-blocks and set up with respect to the kind of district to handle (in the example of Figure 6, the centre of Trets is filled up, while the other city-blocks are thinned and their buildings generalised).

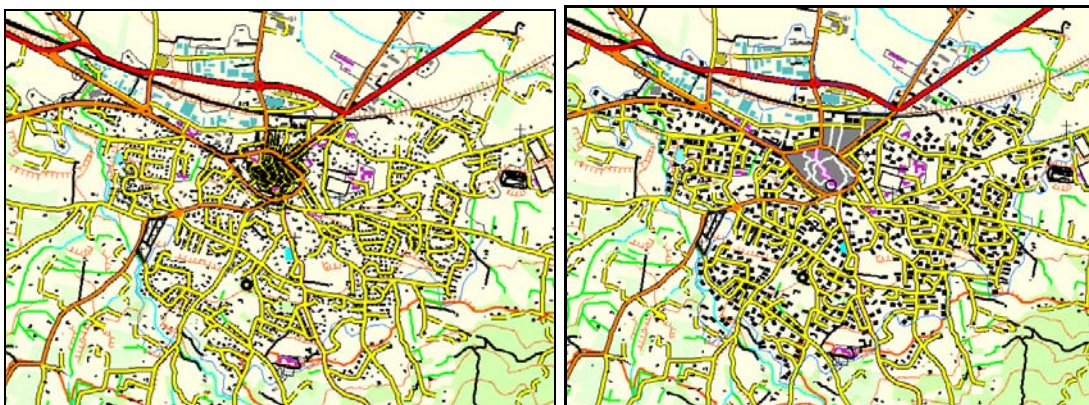


Figure 6 : Trets city before and after the AGENT generalisation (DTtopo[®] dataset, IGN).

Conclusions

The AGENT project has produced a prototype system that allowed to experiment with the MAS approach in map generalisation. The results are encouraging. The MAS approach does offer an interesting framework to model the characteristics of cartographic objects and their transformation through generalisation procedures. The approach taken also allows multi-level analysis through the provision of micro and meso-agents. In the short-term perspective, the AGENT prototype is currently being transformed into a software product by Laser-Scan. This product will be used in operational map and DB production by the national mapping agencies of France and Denmark (IGN and KMS, respectively) and hopefully by other current prospects. From the more long-term and research-oriented point of view, the most important quality of the presented framework is its extensibility. New algorithms and constraints can be added and the prototype can generally be used as a workbench system for research on generalisation algorithms, measures, and their impacts. The system could also be further enhanced by improving the negotiation between agents or incorporating knowledge from external sources such as from cartographic experts or machine learning (Mustière 2001).

References

- Duchêne, C., Barrault M., Haire K. (2001): Road Network Generalisation: A Multi Agent System Approach, *Proceeding of ICC 2001*, Beijing, China.
- Edwardes, A.J. & Mackaness W.A. (2000): Intelligent Generalisation of Urban Road Networks, *Proceedings of GISRUK 2000 Conference*, University of York, UK, pp. 81-85.
- Edwardes, A. J. & Regnault N. (2000): Preserving the Pattern of Density in Urban Network Simplification, *GIScience 2000*, Savannah (GA), USA, pp. 104-105.
- Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W.A. and Weibel, R. (1999) The Application of Agents in Automated Map Generalisation. *19th Int. Cartographic Conf.*, Ottawa, pp. 160-169.
- Mustière, S. (2001): *Apprentissage supervisé pour la généralisation cartographique*. PhD Thesis, Université Pierre et Marie Curie (Paris VI).
- Ruas, A. (1999): *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. PhD Thesis, Université de Marne-la-Vallée.