

An Agent-Based Framework for Polygonal Subdivision Generalisation

Martin Galanda and Robert Weibel

Department of Geography, University of Zurich, Winterthurerstr. 190, 8057 Zurich, Switzerland, {mgalanda,weibel}@geo.unizh.ch

Abstract

Comprehensive techniques for the generalisation of polygonal subdivisions are still lacking in research as well as in commercial GIS, but required for the derivation of arbitrary scales from a master database. This paper outlines a framework for the automated generalisation of polygon mosaics (polygonal subdivisions) based on a multi agent system by extending previous research carried out by the AGENT consortium. After listing generic properties of agents spatial levels of polygon generalisation agents are identified, namely *map*, *group*, *polygon* and *line*. Each of them is linked to a specific agent type. Both the process of polygon generalisation based on a multi agent system as well as the evolution of an agent during the generalisation process is discussed theoretically. Then, a worked example clarifies and illustrates the concepts and methods presented before. Finally, foreseeable challenges of the framework's implementation based on preliminarily experiments are pointed out.

Keywords: cartographic generalisation, polygonal subdivision, multi agent system

1 Introduction

One of the goals of current research in map generalisation is to enable the automated derivation of arbitrary scales from a master database (Bengtson 2001, Ruas 2001). Although progress in research is reported continuously, some key generalisation tasks are not yet solved satisfactorily. Amongst other aspects, a framework that includes both methods and concepts for the generalisation of categorical data in the vector data model (polygon mosaics or polygonal subdivisions) are missing.

A categorical data set is collectively space exhaustive over the entire map, i.e. all its points must belong to only one category and polygon. Thus, whenever one

polygon object is generalized, that is, its geometry is changed all polygons sharing a geometric primitive with that object¹ are affected, too. In practice, however line-based algorithms are commonly used to generalise polygonal data. Since the specific structure and topology of polygons is ignored, unsatisfying results are not surprising. Existing approaches dedicated to the generalisation of categorical data concern either very specific kinds of data (e.g. Jaakkola 1998, Mueller and Wang 1992) or the isolated implementation of single geometric algorithms (e.g. Bader and Weibel 1997, Galanda 2001). Comprehensive techniques for polygon generalisation have neither been addressed in the research community nor through commercial GIS vendors.

This paper outlines a framework for the automated generalisation of polygonal subdivisions (polygon generalisation) based on a multi agent system (MAS). The research and results presented extend previous work carried out by the AGENT consortium (Lamy et al. 1999, Ruas 1999, Barrault et al. 2001, Regnauld 2001). Section 2 gives a short introduction to MAS and discusses the applications in map generalisation. Section 3 presents the specifications for spatial levels of polygon generalisation. In section 4, an outline of the process of polygon generalisation based on agents is presented, while section 5, illustrates a first worked example. Section 6 discusses the selected implementation issues, and finally, some conclusions are drawn and an outlook on the next steps of the research work is presented in section 7.

2 MAS in Cartographic Generalisation

Borrowing from a definition given by (Luck 1997) an agent for cartographic generalisation denotes a geographic entity capable of controlling its own decision making and generalisation guided by a set of cartographic constraints². Thus, an agent (Barrault et al. 2001, Duchêne et al. 2001, Ruas 1999)

- is linked to a set of constraints;
- possesses a method to determine its so-called happiness, i.e., it evaluates and summarises the satisfaction of all its constraints;
- aims to reach a perfect state (= perfect happiness; all constraints are satisfied);
- proposes plans (cartographic operations) according to the violated constraints in order to improve its happiness;
- triggers plans autonomously starting with the one supposed to be best;
- is able to compare and store different states;

¹ According to the characteristics of a polygon mosaic at least one other polygon is concerned.

² "A constraint is a condition similar to the predicate in a production rule. The distinction is that a constraint is not bound to a particular action."(Beard 1991, p.122) The concept of a constraint-based approach to cartographic generalisation has been emphasised by (Weibel 1996) and (Ruas 1999), among others.

- can return (backtrack) to any previous state;
- may instantiate, trigger and coordinate other agents³.

A MAS designates several independent agents cooperating to solve problems at both a local (related to a single agent) and a global (related to a group of agents) level (Luck 1997, p. 1227). Cartographic generalisation is an holistic and subjective task, that is, each conflict should be solved in its spatial context, and several 'correct' solutions are conceivable. A solution often constitutes a compromise between several local and global constraints. Modelling the generalisation process by a MAS means that "a sub-optimal but acceptable solution can often be reached" (Lamy et al. 1999). Compared to other technologies used in map generalisation like expert systems or neural networks a MAS supports an holistic approach and allows a dynamic adaptation to a changing environment. As a result of these properties, the MAS technology seems to provide an adequate framework for map generalisation (Regnauld 2001).

In the context of the project AGENT funded by the European Commission, a consortium of experts in multi agent systems, experts in automated map generalisation and GIS specialists have succeeded in setting up a MAS for generalisation tasks of topographic mapping (Lamy et al. 1999, Barrault et al. 2001)⁴. The AGENT package (data schema and the generic agent engine) is implemented in a commercial GIS and has already proven useful in map production (Bengtson 2001). This paper extends methods and concepts - developed with respect to the generalisation of road networks and urban settlements during the AGENT project - for use with polygonal subdivisions.

3 Spatial Levels of Polygon Generalisation

The more a cartographic conflict is narrowed in on a spatial context, and the more precisely the situation is characterised, the more specific transformations are applied and the faster an adequate result (i.e., a compromise) is achieved (Ruas 1999, Barrault et al. 2001). Thus, map space is organised in so-called spatial levels of polygon generalisation. So each constraint, cartographic operation etc. is delegated to a specific spatial level⁵, that is, it holds a specific scope. For instance, the constraint of minimal size refers to a single polygon object while the minimal distance between objects is related to a group of polygons. Related to polygon generalisation, 4 different spatial levels, namely

- map,

³ In the following an agent supervising other agents is generally called a *parent agent* while an agent triggered and controlled by a parent agent is termed a *child agent*.

⁴ More information on this project can be found at <http://agent.ign.fr/>.

⁵ The spatial level of constraints can be found in the complete list of constraints to polygon generalisation at <http://www.geo.unizh.ch/~mgalanda/poly-www/constraints.html>. The spatial level of generalisation operators is listed in section 4.2.

- group,
- polygon and
- line,

seem to be reasonable. Agent prototypes for these different levels are available in the framework. The generic properties and behaviours listed in the previous section are independent of an agent's spatial level.

Map Agent. Every polygon mosaic holds only one map agent (cf. Fig. 1). It is responsible for constraints and cartographic operations concerning the whole polygon map (e.g. reclassification) as well as the instantiation of the group agents.

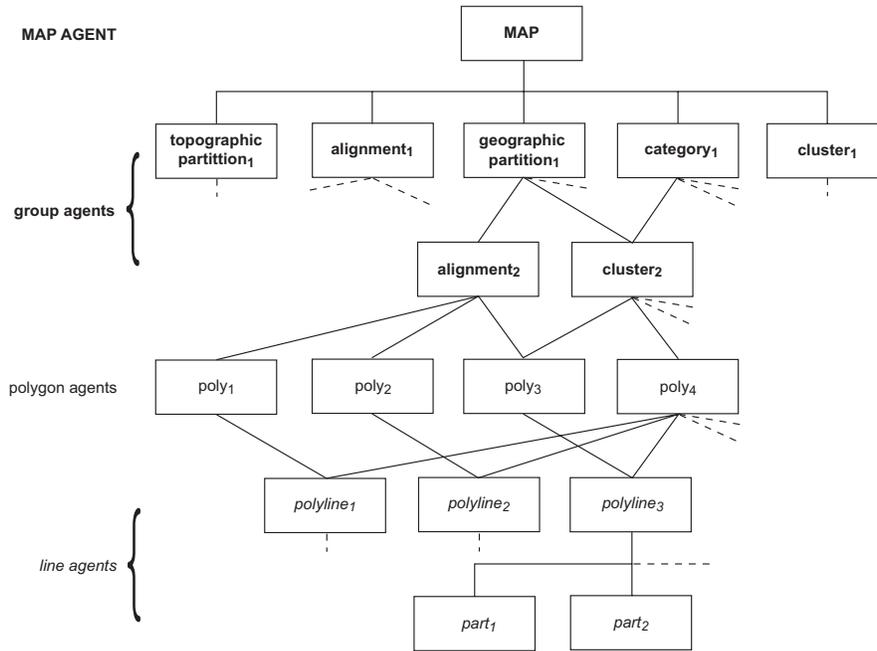


Fig. 1. The organisation of agents at the four different spatial levels of polygon generalisation. See also Fig. 2 below for some graphic examples

Group Agents. Group agents handle contextual generalisation, i.e. conflicts between polygon objects. They are composed of several polygon objects sharing a common geometric or semantic relation. Thus, reasonable group agents may be attached to a cluster, an alignment, a category, a topologic partition (e.g. neighbours of 1st and 2nd order) or a geographic partition (e.g. a collection of polygons bounded by rivers or roads). Group agents can recursively subdivide themselves if needed, that is, a group agent can spawn off other group agents besides the polygon agents. For instance, a geographic partition agent may supervise a cluster agent - compare *geographic partition₁* in Fig. 1.

Polygon Agents. A polygon agent coordinates the generalisation of an area object. Constraints and operations acting on an individual polygon are evaluated and performed without considering the agent's spatial context (e.g. enlargement). However, as already mentioned above the geometric transformation of one polygon induces always at least the modification of one other polygon. So every change must be propagated to those objects sharing a common geometric primitive with the transformed object.

Line Agents. Line agents are delegated to polylines bordering a polygon object and their generalisation (e.g. simplification). Exactly two polygon agents can supervise - assuming a clean topology - a line agent of the first generation⁶. As group agents they are enabled to recursively subdivide themselves in order to

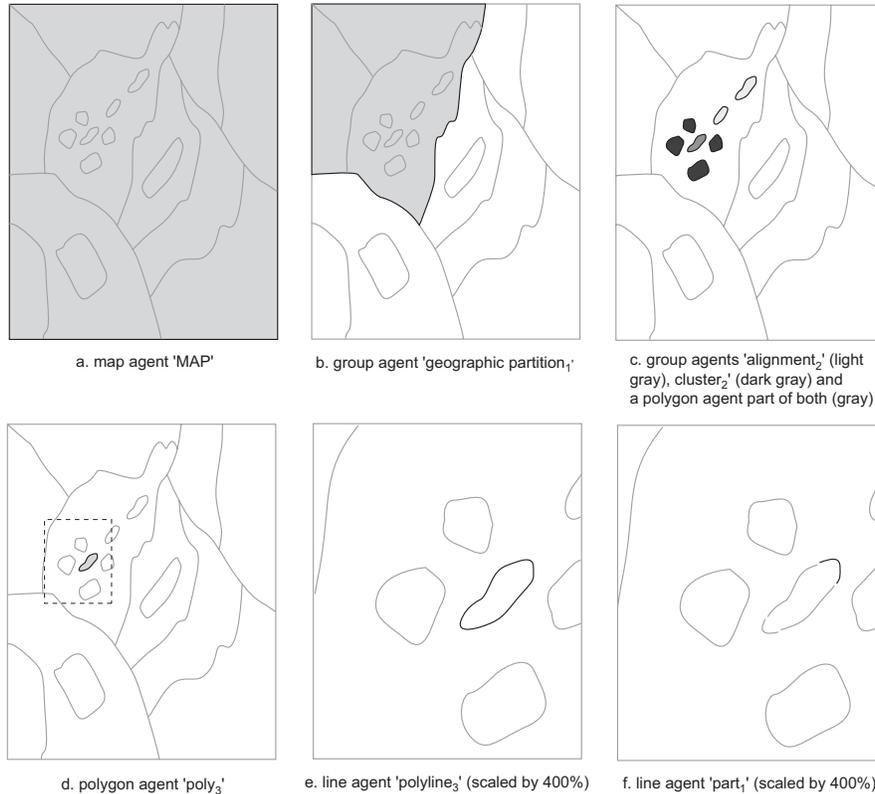


Fig. 2. Some examples of different spatial levels of polygon generalisation in a polygon mosaic according to the organization of agents shown in Fig. 1

⁶ A child agent of the first generation always has parents of a superior agent type - e.g. a line agent of the first generation is exclusively supervised by polygon agents. A child agent of the second generation always has parents of the same agent type - e.g. a group agent of the second generation always has another group agent as a parent.

If necessary parent agents build their child agents at run time. Parent agents are enabled to specify the child agents' constraints according to an analysis of all their child agents or the failure of a previous plan, e.g. a group agent can tell a polygon perform generalisation on homogenous line parts (Duchêne et al. 2001), see *polyline₃* in Fig. 1 and 2(e,f).

agent to not enlarge itself when it knows about a lack of free map space. Different parent agents can supervise one child agent in turn. For instance, *poly₃* in Fig. 1 and Fig. 2c is part of the group agents *alignment₂* and *cluster₂*. Thus, the agents' generalisation must be done in a sequential process.

4 Agent Life Cycle in Polygon Generalisation

The generalisation process for polygonal data presented in the next section reverts to experiences gained and concepts developed by (Ruas 1999) and the AGENT project (Barrault et al 2001). It is organised into three main stages. A pre-processing stage prepares the database for the second stage, the iterative agent-based generalisation. Finally, the user undertakes an evaluation of the automated generalisation results during the third stage.

4.1 Pre-Processing: Data Specification and Analysis

The first step in the pre-processing stage is that all constraints are specified according to the basic conditions of the generalisation task. In addition, supplementary information (auxiliary data) is calculated on the database objects.

4.2 Generalisation: Life Cycles of Agents

At the very beginning of the generalisation process the map agent is activated. It has to care about its own constraints but also to identify and trigger group agents on the fly. Procedural knowledge of polygon generalisation is needed to decide either to try first to satisfy a parent agent's constraint and then work on the child agent's constraint or to first initialise the child agent's generalisation and then the parent's constraints. In practice, a mixture of these approaches may be applied. Again, procedural knowledge can help in choosing which child agent is to be triggered first amongst all the child agents.

The AGENT package uses a constraint-based approach to generalisation (Beard 1991, Weibel 1996, Ruas 1999), that is, constraints initialise and control the entire generalisation process. Constraints to polygon generalisation are the subject of (Weibel 1996), (Peter and Weibel 1999) and (Edwardes and Mackaness 2000). Details of the agent engine to be used in our project (i.e. the one developed for the AGENT project) are described in (Regnauld 2001). In a MAS for map generalisation constraints are linked to agents that aim to satisfy the *offensive*

constraints without violating one of the *defensive constraints*⁷. To every constraint a measure⁸ is linked to evaluate its satisfaction (= to determine the severity of violation). A list of possible plans is attached, too, that propose cartographic operations (geometric and semantic transformations) for improving the agent's happiness – Fig. 3. The final ordered list of plans results from the weighted sum of all plans suggested by any violated constraint of the agent. The current best plan is then triggered.

constraint	The distance between two polygon objects should not be less than the minimal distance (e.g. minimum visual separability distance).
agent	Group
measure	proximity measure (e.g. Hausdorff distance)
plan	<ul style="list-style-type: none"> a. Displacement - algorithm based on 'snakes' (Galanda 2001) b. Displacement - vector displacement c. Exaggeration - algorithm based on 'snakes' (Galanda 2001) d. Aggregation - convex hull e. Typification

Fig. 3. Example of an offensive metric constraint at the group agent level in polygon generalisation, the attached measure and list of possible plans.

The process of improving an agent's happiness starts when its parent agent changes its state to *active*. The following sequence of constraints evaluation [evaluate constr.]⁹, proposing plans [propose plans], triggering the best plan [trigger best plan] and re-evaluation [re-evaluate] is the same generic behaviour of all agents and called an agent's life cycle. A modified 'hill-climbing' algorithm is applied within this life cycle - see Fig. 4 - to find the best compromise to all constraints of an agent (Regnauld 2001). The life cycle ends when either all constraints are satisfied completely (perfect state) or there is no plan left to try. Nevertheless, the database is updated by the best state ever reached in the life cycle [update database]. The agent is set to *passive* again, that is, the control of the generalisation process is returned to the parent agent.

⁷ An *offensive constraint* is an indicator for the need of generalisation, e.g. a minimum size constraint, while a *defensive constraint* controls the preservation of a certain property of an agent such as its topologic relation.

⁸ For details on measures in polygon generalisation refer to (Peter 2001).

⁹ Terms in brackets relate to life cycle steps displayed in Fig. 4.

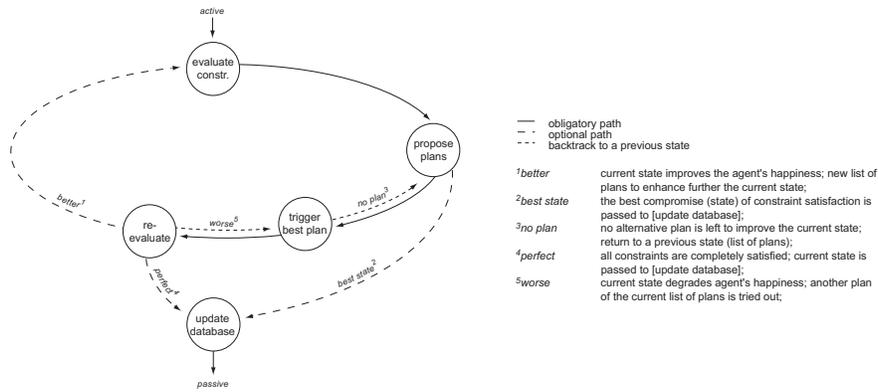


Fig. 4. An agent's life cycle in polygon generalisation with the incorporated 'hill-climbing' algorithm.

One constraint attached to every parent agent assesses the happiness of its child agents. A violation of that constraint requires either plan *a* or *b*:

- a. the instantiation of the child agent's life cycles; the autonomous generalisation of the child agents, that is, the control of the generalisation process is passed to the individual child agents in turn; each of them tries to independently improve its happiness.
- b. the state of the conflicting child agents is set to *reactive* in turn; each of them executes an order (a method and parameters) given by the parent agent¹⁰.

In both cases, the child agent reports modified properties (geometry and semantics) to its parent agent that then continues its life cycle at the re-evaluation step.

The example of a constraint to polygon generalisation shown in Fig. 3 presents a set of possible plans (cartographic operations) to enhance the agent's happiness if required. In polygon generalisation we distinguish the following cartographic operators (spatial agent type given in parentheses):

For semantic transformation

- *Reclassification* (map) changes the category an object belongs to.

For geometric transformation

- *Aggregation* (group) combines a polygon with other polygons of the same or a similar class to a new polygon object.
- *Typification* (group) reduces the complexity of a group of polygon objects by removing, displacing, aggregating and enlarging single objects. The readability is ensured and the significant alignment pattern is preserved.

¹⁰ In general a reactive order can be given to an agent by any other agent.

- *Displacement* (polygon or group) denotes the movement of a whole polygon object - no shape distortion.
- *Collapse* (polygon) reduces the polygon geometry either to a line or to a point geometry.
- *Elimination* (polygon) removes an object from the polygon mosaic; the freed space is assigned to other categories.
- *Enlargement* (polygon) denotes a global increase (decrease) of a polygon's geometry.
- *Exaggeration* (polygon or group) defines a local increase (decrease) of a polygon's geometry.
- *Simplification* (line) reduces the granularity of a linear geometry by removing vertices.
- *Smoothing* (line) improves the visual appearance of a line object.

Algorithms implement the concepts described by cartographic operators. Both a review of existing algorithms and some newly developed algorithms for polygon generalisation based on optimisation techniques can be found in (Galanda 2001).

4.3 Post Processing: Final Evaluation

After the map agent's life cycle is completed the user evaluates the final result with the help of a detailed report (severity of remaining conflicts, happiness per agent or category etc.) output automatically by the system (Ruas 2001). If necessary some interactive (re)generalisation is performed.

5 A Worked Example

In this section, we present a worked example for the agent-based framework for polygon generalisation. The example is intended to both clarify and illustrate the ideas and concepts discussed theoretically above. It demonstrates some excerpts of the agent-based generalisation of one cluster agent (Fig. 6). It consists of five polygons belonging to two categories (light and dark gray) and is embedded in a polygon environment - compare *state0* in Fig. 6. The different states of the cluster agent and the cartographic operations they result from (shown in Fig. 5) as well as the evolution of the constraint satisfaction (diagrams displayed in Fig. 6) summarise how the 'hill-climbing' algorithm finds the best compromise between several competing constraints.

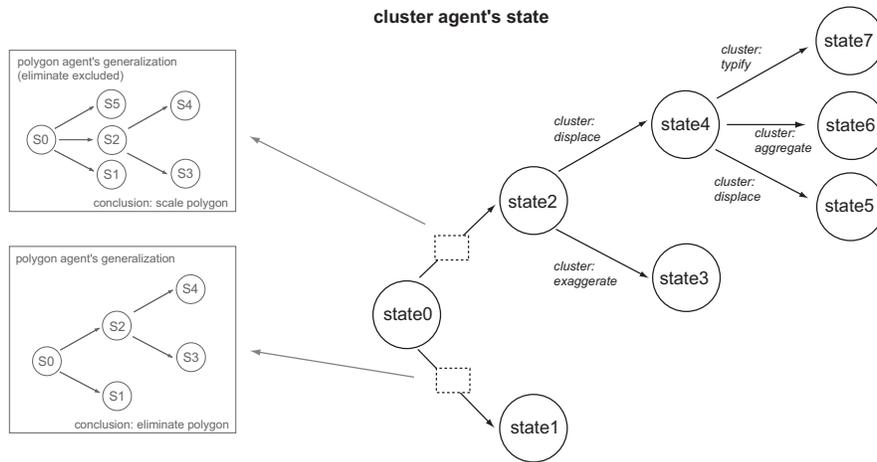


Fig. 5. The different states of the cluster agent (state0 to state7) in the generalisation process and the cartographic operations they result from. The states (s0 to s5) of the triggered polygon agent are indicated in the two small insets to the left.

In the example the following three constraints¹¹ related to cluster agents are considered:

- 'minimal distance' (constraint A)
The distance between two polygon objects should not be less than the minimal distance.
- 'preserve alignment' (constraint B)
The alignment of polygons of different classes (characteristic patterns of alignment) should be preserved.
- 'child agents' happiness' (constraint C)
The constraints of a parent's child agents must reach a defined level of satisfaction.

state0 The *state0* in Fig. 6 shows the agent's state when activated. The life cycle (cf. Fig. 4) starts with an evaluation of the constraints in order to determine if there is a need for generalisation or not. Those constraints directly related to a property of the example agent (constraint 'minimal distance' and 'preserve alignment') are satisfied. However, the third constraint observing the happiness of the supervised child agents is not fulfilled, because the area property of one polygon agent (the polygon in the centre of the cluster) falls below the threshold defined in a minimum area constraint. The polygon agent's generalisation

¹¹ In the given example the satisfaction of the individual constraints is rated on a continuous scale ranging between perfect satisfaction and total violation (see the diagrams in Fig. 6.)

	performed in a separate life cycle (s_0 to s_5 in Fig. 5) results in the elimination of that polygon.
state1	The modified geometry is reported to the cluster agent that subsequently re-evaluates the changed situation ($state1$).
state0	As $state1$ does not conform to the ' <i>preserve alignment</i> ' constraint a backtrack to the previous state ($state0$) is made, the only plan provided for such a conflict. Next, the polygon agents are again activated. A restriction is then set with respect to the conflicting polygon, namely the elimination operator is removed from the list of possible plans.
state2	The cluster agent's $state2$ results from an enlargement of that polygon. Due to the enlargement of this polygon the distance to the other polygons falls below the threshold defined in the ' <i>minimal distance</i> '-constraint. From an agent point of view, an improvement of the overall constraint satisfaction (see Fig. 6) has been achieved, that is, $state2$ is considered to be better than $state0$. Possible plans to solve that conflict are proposed (compare also the list of plans in Fig. 3).
state3	The application of an exaggeration operation ($state3$) decreases the cluster agent's happiness because the ' <i>preserve alignment</i> '-constraint is violated even further as a result of the shape distortion at the polygon level.
state2	Subsequently, a backtrack to $state2$ is performed and the next plan of the list is applied.
state4	The displacement of polygons resulting in $state4$ represents the best compromise so far with respect to the different constraint satisfaction. As this state is considered better than its previous state ($state2$) a new list of plans is proposed and tried out.
state5-7	No further improvement of the agent's happiness occurs; each time a backtrack to $state2$ is performed.
state2	Thus, the life cycle of the cluster agent continues at $state2$ by searching for alternative plans to improve its happiness. In this example it is assumed that at $state2$ no further plans are left (typification and aggregation are not considered)
state0	No plans to be tried out are left at $state0$, too. Consequently the best solution, the one stored in $state4$, is used to update the agent's geometry. The cluster agent is set to <i>passive</i> again.

6 Implementation Issues

Although the MAS-based approach is not new and advances have been made, extensive additional research efforts are required, in particular for subdivision and generalization of polygons. For instance, a comprehensive set of automated, robust and efficient algorithms specifically designed for use with polygonal data

along with measures to determine the satisfaction of all constraints attached to one spatial level of polygon generalization need to be explored.

Applying the existing AGENT package, experiments were undertaken with the landscape model, land use data (VECTOR25 and VECTOR200) of the Suisse Federal Office of Topography and geological data of the Swiss National Park at a scale of 1:25,000. These real data tests identified amongst other things the following major challenges in the implementation of the proposed framework:

- the retrieval of procedural knowledge of polygon generalisation (choice of the 'best plan'; which child agent first? etc.);
- the interplay and interaction of different agents - both of the same and of different spatial levels - acting on the same subset of polygons or the same polygon object;
- the improvement (implementation) of the communication between agents, both of the same and of different spatial levels;
- the determination of reasonable plans and ordered lists of possible plans related to every constraint, that are independent of both the kind of data and the scale range;
- the automated identification of group agents belonging to a map agent or a superior group agent by spatial and/or semantic analysis.

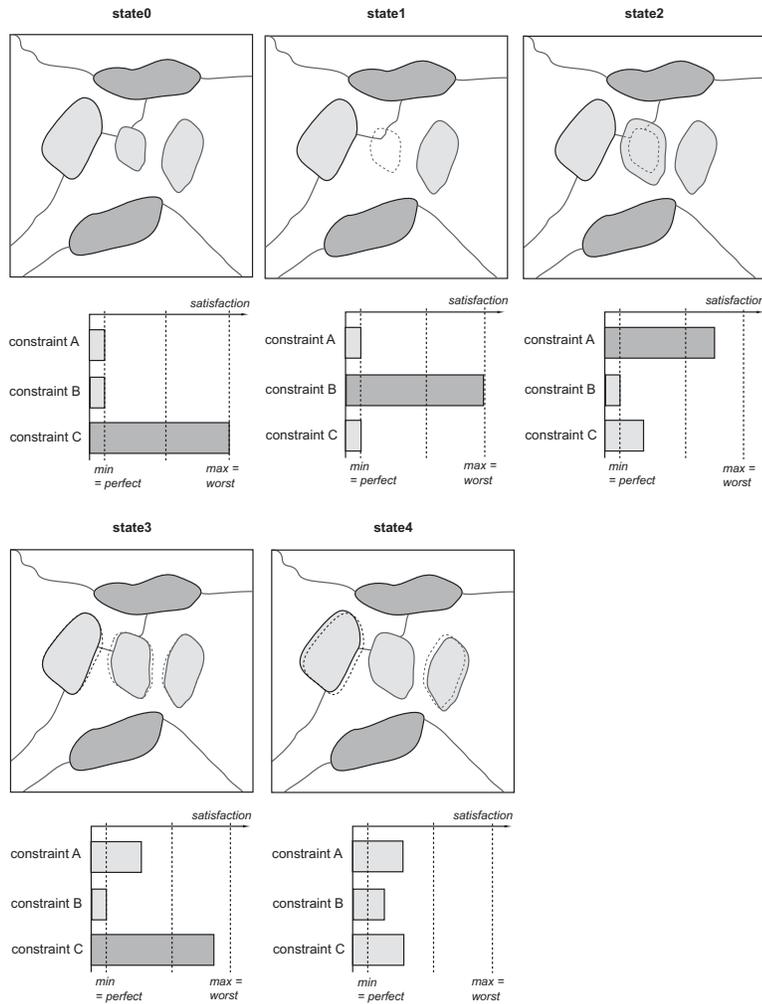


Fig. 6. The different states of the example agent and the corresponding constraint satisfaction

7 Conclusions and Future Work

The main goal of this research project is the evaluation of the suitability of MAS for automated polygon generalization. This paper outlined a framework for the generalization of polygonal subdivisions based on a MAS. Concepts and methods developed by the AGENT consortium (Lamy et al. 1999, Barrault et al. 2001) for the generalisation of topographic maps (road networks and urban settlements) were adapted and extended. The key points of the presented research concerned:

- the definition of the agents' organisation in a polygonal subdivision (sect. 3);
- a first acquisition of constraints related to the different agent types including measures and potential plans - for an example cf. Fig. 3;
- the interplay of different agents and agent types within the so-called agent's life cycle (sect. 4).

While a worked example (sect. 5) clarified the theoretical discussion of the framework and demonstrated the underlying ideas and concepts section 6 pointed out relevant implementation issues.

As indicated in the previous section our work concentrates currently on the extension and fine-tuning of the existing AGENT package (Lamy et al. 1999, Barrault et al. 2001) in order to put the presented framework into practice. First results of real world data experiments will be forthcoming in July 2002.

Acknowledgements

This research is part of project 'GENDEM: Map Generalisation for Thematic and On-Demand Mapping in GIS' supported by the Swiss National Science Foundation under contract 20-52759.97. Many thanks go to Mathieu Barrault for valuable assistance and comments.

References

- Bader M, Weibel R (1997) Detecting and Resolving Size and Proximity Conflicts in the Generalisation of Polygon Maps. In: Proceedings of the 18th International Cartographic Conference. Stockholm, pp 1525–1532
- Barrault M, Regnauld N, Duchêne C, Haire K, Baeijs C, Demazeau Y, Hardy P, Mackaness W, Ruas A, Weibel R (2001) Integrating multi-agent, object-oriented and algorithmic techniques for improved automated map generalization. In: Proceedings of the 20th International Cartographic Conference. Beijing, pp 2110-2116
- Beard M (1991) Constraints on Rule Formation. In: Buttenfield B, McMaster R (eds) Map Generalization: Making Rules for Knowledge Representation. Longman, London, pp 121–135
- Bengtson M (2001) Design and implementing of automatic generalisation in a new production environment for datasets in scale 1:50000 (and 1:100000). In: Proceedings of the 20th International Cartographic Conference. Beijing, pp 1076–1083
- Duchêne C, Barrault M, Haire K (2001) Road Network Generalization: A Multi Agent Approach. In: Proceedings of the 20th International Cartographic Conference. Beijing, pp 2166–2177
- Edwardes A, Mackaness W (2000) Modelling knowledge for automated generalization of categorical maps – a constraint based approach. In: Atkinson P, Martin D (eds) GIS and Geocomputation (Innovations in GIS 7). Taylor&Francis, London, pp 161-173
- Galanda M (2001) Optimization techniques for polygon generalization [online]. In: Proceedings of the 4th Workshop on Progress in Automated Map Generalization.

- Beijing, Available from:
http://www.geo.unizh.ch/ICA/docs/beijing2001/galanda_v0.pdf
- Jaakkola O (1998) Multi-Scale Categorical Databases with Automatic Generalization Transformation Based on Map Algebra. *Cartography and Geographic Information Systems* 25(4):195–207
- Lamy S, Ruas A, Demazeau Y, Jackson M, Mackaness W, Weibel R (1999) The Application of Agents in Automated Map Generalisation. In: *Proceedings of the 19th International Cartographic Conference*. Ottawa, pp 1225–1234
- Luck M (1997) *Foundation of Multi Agent Systems: Issues and Directions*. *Knowledge Engineering Review* 12(3): 307–318
- Mueller J, Wang Z (1992) Area-Patch Generalization: A Competitive Approach. *The Cartographic Journal* 29(2): 137–144
- Peter B (2001) Measures for the Generalization of Polygonal Maps with Categorical Data [online]. In: *Proceedings of the 4th Workshop on Progress in Automated Map Generalization*. Beijing, Available from:
http://www.geo.unizh.ch/ICA/docs/beijing2001/Peter_v2.pdf
- Peter B, Weibel R (1999) Using Vector and Raster-Based Techniques in Categorical Map Generalization. In: *Proceedings of the 19th International Cartographic Conference*. Ottawa, pp 1135–1145
- Regnauld N (2001) Constraint based mechanism to achieve automatic generalization using agent modeling. In: *Proceedings of the GIS Research UK 9th Annual Conference*. Glamorgan, pp 329–332
- Ruas A (1999) *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. Ph.D. thesis, Université de Marne-la-Vallée
- Ruas A (2001) Automating the generalisation of geographical data: the age of maturity?. In: *Proceedings of the 20th International Cartographic Conference*. Beijing, pp 1943–1953
- Weibel R (1996) A Typology of Constraints of Line Simplification. In: *Proceedings of the 7th International Symposium on Spatial Data Handling*. Delft, pp 9A.1–9A.14