Automated Polygon Generalization in a Multi Agent System

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)
vorgelegt der
Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich
von

Martin Galanda
aus Österreich

Promotionskomitee
Prof. Dr. Robert Weibel (Vorsitz)
Dr. Mathieu Barrault
Prof. Dr. Lorenz Hurni
Dr. Anne Ruas

Zürich 2003

Die vorliegende Arbeit wurde von der Mathematisch-naturwissenschaftlichen Fakultät der Universität Zürich im Sommersemester 2003 als Dissertation angenommen.

Promotionskomitee:
Prof. Dr. Robert Weibel (Vorsitz), Dr. Mathieu Barrault, Prof. Dr. Lorenz Hurni, Dr. Anne Ruas

Abstract

Polygonal subdivisions, i.e. the representation of categorical data in the vector model, are a common data type in GIS applications, in thematic maps, in topographic maps and in digital landscape models. Their cartographic generalization is termed polygon generalization. Cartographic generalization or generalization is one of the basic principles of cartography, namely the legible and comprehensible visualization of geographic data at a certain scale and for a given purpose. For the generalization of polygonal subdivisions, a multitude of methods, such as generalization algorithms, measures and generalization constraints, were in the past proposed in isolation from each other. However, what is missing is a comprehensive framework for the orchestration of these tools, that is, their integration into an automated and comprehensive generalization process. Hence, this thesis studies the automation of polygon generalization by means of a multi-agent system (MAS). In doing so, the work extends previous research carried out by the AGENT consortium (i.e. by a consortium of Institut Géographique National France, University of Edinburgh, University of Zurich, Institut National Polytechnique de Grenoble and Laser-Scan Ltd.).

In this thesis, a discussion of possible approaches to automated generalization leads to the proposal of a framework for comprehensive, automated and agent-based polygon generalization. MAS technology together with the concepts developed by the AGENT consortium offer distinct benefits for the automation of map generalization. These benefits include the capabilities 1) to compromise between different generalization constraints associated with a cartographic object, 2) to coordinate the generalization of objects at different spatial levels and 3) to model holistic decision making in map generalization. After listing the generic properties of agents spatial levels of polygon generalization are identified, namely map, group, polygon and line. Each of them is linked to a specific agent type. Both the process of polygon generalization based on a multi agent system as well as the evolution of an agent during the generalization process are discussed theoretically. Next, a worked example clarifies and illustrates the concepts and methods embedded in the proposed framework. Prior to the implementation of the framework generalization algorithms that make use of energy minimization techniques and generalization constraints for polygon generalization are studied. Algorithms are essential to the conflict resolution while constraints control the agent-based generalization process.

The application of an energy minimizing technique, snakes, is investigated for resolving size conflicts (i.e. a polygon too small with respect to the target scale) and proximity conflicts (i.e. polygons are too close to each other) in polygonal subdivisions. The usage of a single snakes-based algorithm is proposed, which can be controlled in such a way that it achieves the displacement, enlargement and exaggeration of polygons or an arbitrary combination of these operations. Thus, size and proximity conflicts within a group of polygons can be solved simultaneously, that is, a holistic solution of such conflicts is accomplished. Moreover, the proposed algorithm enables the direct integration of the update of the neighbors of a modified polygon into the transformation process. The main drawbacks identified are the difficult setup and fine-tuning of snakes parameters and the computational resources required by the algorithm. However, the experiments emphasize that the algorithm constitutes an improvement in comparison to existing algorithms for resolving size and proximity conflicts as well as to sequential approaches of propagation. The computational speed of our test implementation could be vastly improved by more efficient matrix operations that make use of sparse matrix techniques as well as by partitioning schemes.

Generalization constraints denote design specifications to which map generalization should adhere. Besides an inventory of constraints required for automated polygon generalization, this thesis provides methods for the evaluation of the satisfaction of individual constraints, plans (i.e. algorithms and parameters) for improving the satisfaction of constraints and importance and priority values of constraints that enable to automatically compromise between the satisfaction of different constraints. In doing so, constraints have the potential to detect conflicts (i.e. the violation of constraints), to control the resolution of conflicts (i.e. the selection of plans) and to validate accomplished generalization solutions through the evolution of constraints' satisfaction.

The synthesis of the research work is formed by the implementation of a framework for the agent-based, automated generalization of polygonal subdivisions that relies on the developed algorithm and the proposed set of constraints. The prototype system extends the preliminary AGENT engine of the GIS LAMPS2 of Laser-Scan Ltd. The potential of the outlined framework and the prototype, respectively, is studied by conducting experiments with real world data. In practice, the fully automated generalization of an extract of the land cover layer of VECTOR25, i.e. the large scale digital landscape model of Switzerland, to a target scale of 1:50,000 and 1:100,000 is demonstrated. The qualitative and quantitative evaluation of the results of automated generalization illustrate the capability of the agent-based framework and prototype. That is, the results were considered to be promising and appropriate representations of the land cover layer at the target scales. Identified weaknesses refer to algorithms for the shape simplification of polygons, the consideration of methods of semantic and spatial analysis and the lack of negotiation mechanisms of agents.

The implemented prototype is the first system for automated and comprehensive polygon generalization ever reported in generalization research. Thus, it seems legitimate to state that this research work accomplished progress in generalization research. The relevance of the presented research and its results attributes to the fact that the generalization of polygonal data attracts increased interest since this type of data is an integral part of topographic and thematic maps as well as digital landscape models. Hence, it is essential to most maps that are generated in cartographic systems or GIS.

Zusammenfassung

Polygonmosaike finden vielfach Anwendung in der GIS-Analyse, in topographischen und thematischen Karten sowie in digitalen Landschaftsmodellen. Sie beschreiben räumliche, kategoriale Daten im zweidimensionalen Vektorraum. Die kartographische Generalisierung (kurz: Generalisierung) beschäftigt sich mit der Ableitung räumlicher Daten für eine graphisch und inhaltlich angepasste Darstellung unter Berücksichtigung von Massstab und Kartenzweck.

Die Generalisierung von Polygonmosaiken wird unter dem Begriff der Polygongeneralisierung zusammengefasst. In der Polygongeneralisierung wurden bereits verschiedene Konzepte und Methoden entwickelt, wie beispielsweise Generalisierungsalgorithmen, Bedingungen für Polygongeneralisierung (engl. constraints) und Masse. Eine Strategie bzw. ein Konzept für eine umfassende, automatische Polygongeneralisierung wurde jedoch bislang weder in der Forschung noch in der Kartenproduktion thematisiert. Daher untersucht diese Dissertation die Eignung von multiplen Agentensystemen (MAS) für die automatische Polgongeneralisierung. Sie stützt sich dabei auf frühere Forschung des AGENT Konsortiums, welches unter der Leitung vom Institut Géographique National Frankreich die Universitäten von Zürich und Edinburgh, das Institut National Polytechnique de Grenoble und die GIS Firma Laser-Scan Ltd. umfasste.

Ausgehend von einer Diskussion möglicher Strategien für die Automatisierung der kartographischen Generalisierung erarbeitet die Dissertation ein Konzept für die automatische und umfassende Polygongeneralisierung. Dieses stellt eine Erweiterung der vom AGENT Konsortium – für die Generalisierung von topographischen Karten – entwickelten Strategie dar. Das Konzept unterstützt das dynamische Finden eines Kompromisses zwischen verschiedenen Generalisierungsbedingungen, die Koordination der Generalisierung von mehreren Objekten und die Modellierung einer globalen Generalisierungsstrategie. Die Arbeit definiert zunächst die Anforderungen an einen Agenten für die Generalisierung sowie unterschiedliche Abstraktionsebenen eines Polygonmosaiks, nämlich die gesamte Karte, eine Gruppe von Polygonen, ein einzelnes Polygon und die Begrenzungslinie eines Polygons. Jeder dieser Ebenen ist ein eigener Agententyp zugeordnet. In einem nächsten Schritt wird ein Konzept für die Polygongeneralisierung vorgestellt und an Hand eines synthetischen Beispiels verdeutlicht und illustriert. Die Implementierung des Konzeptes erfordert geeignete Generalisierungsalgorithmen für die Konfliktlösung und formalisierte Bedingungen zur Steuerung des Generalisierungsprozesses.

Ein Teilbereich der Arbeit untersucht die Lösung von Grössen- und Distanzkonflikten (d.h. für den Zielmassstab zu kleine Polygone oder zu nahe beieinander liegende Polygone) mit Hilfe eines Verfahrens der Energieminimierung, den sogenannten Snakes. Diese Methode dient als Grundlage für die Entwicklung eines Algorithmus, der die Umsetzung von lokalen und globalen Vergrösserungen von Polygonen, Verdrängung von Polygonen oder eine beliebige Kombination dieser Operationen realisiert. So kann der Algorithmus in einem Prozess verschiedene der genannten Konflikte bereinigen und gleichzeitig das Polygonmosaik an die geänderten Polygongeometrien anzupassen. Damit besitzt dieser vorgestellte Algorithmus entscheidende Vorteile gegenüber den bisher verwendeten Verfahren zur Lösung von Grössen- und Distanzkonflikten. Nachteile des Algorithmus liegen in seiner diffizilen Parametrisierung und der hohen Rechenzeit, die allerdings durch geeignete bessere Implementierungsverfahren entscheidend beschleunigt werden könnte.

Bedingungen (constraints) für die Generalisierung beschreiben Anforderungen – im Sinne von Designspezifikationen – an eine generalisierte Karte. Die vorliegende Arbeit liefert neben einem Inventar an Bedingungen für die Polygongeneralisierung, auch Methoden zur Bestimmung der Erfüllung der einzelnen Bedingungen und mögliche Generalisierungsoperationen zur Verbesserung der Erfüllung von Bedingungen. Des Weiteren wird die relative Bedeutung der einzelnen Generalisierungsbedingungen als Grundlage für die Kompromissfindung zwischen verschiedenen Bedingungen erörtert. Die derart definierten Generalisierungsbedingungen können in einem automatisierten Generalisierungsprozess für das Erkennen von Konflikten, die Steuerung der Konfliktlösung und die Validierung von berechneten Lösungen eingesetzt werden.

Das vorgeschlagene Generalisierungskonzept, der entwickelte Algorithmus, weitere Algorithmen der Polygongeneralisierung und die besprochenen Bedingungen für die Generalisierung sind in einem Prototyp implementiert worden. Dieser unterstützt die vollautomatische Generalisierung von Polygonmosaiken und ist im GIS LAMPS2 von Laser-Scan Ltd. verfügbar. Experimente werden mit realen Daten durchgeführt. Der Prototyp wird verwendet, um aus einem Ausschnitt der Landnutzungsebene (Primärflächen) von VECTOR25, dem grossmassstäbigen digitalen Landschaftsmodell der Schweiz mit einem Erfassungsmassstab von 1:25'000, Polygonmosaike der Landnutzung in den Massstäben 1:50'000 und 1:100'000 abzuleiten. Eine detaillierte qualitative und quantitative Analyse der erzielten Ergebnisse unterstreicht das Potential des vorgestellten Konzeptes für die Generalisierung von Polygonmosaiken. Schwächen bestehen in den für die Formvereinfachung von Polygonen verwendeten Algorithmen, dem Einbezug von Methoden der räumlichen und semantischen Analyse und der noch fehlenden Kommunikation zwischen Agenten.

Indem der implementierte Prototyp wohl das erste bekannte System für die automatische und umfassende Generalisierung von Polygonmosaiken ist versucht die Dissertation einen Beitrag zur Forschung in der automatischen Generalisierung zu liefern. Eine gesteigerte Nachfrage nach Methoden der Polygongeneraliserung, welche eine Grundlage für die automatische Ableitung von topographischen und thematischen Karten aus digitalen Landschaftsmodellen darstellen, unterstreicht zusätzlich die Bedeutung der Dissertation.

Acknowledgments

Robert Weibel who encouraged my research work and offered me much appreciated freedom in research. Despite his busy diary he always managed to discuss emerging research questions and read drafts.

Mathieu Barrault for helping me getting started with the PhD, his endless enthusiasm on automated cartography and reviewing this thesis. If you are convinced of having a good concept then go and ask Mathieu and for sure he will find an example of how you must rethink your concept.

Lorenz Hurni and Anne Ruas for reviewing this thesis and acting as members of the PhD committee.

The AGENT consortium, whose research strongly influenced me and who built the AGENT engine that serves as a basis of this thesis.

Dirk Burghardt who was always ready to assist me in doing my PhD, argue about my research and play table tennis.

Mats Bader who motivated me to apply snakes to polygon generalization and taught me his approach step by step.

Alessandro Cecconi for hours of interesting discussions about map generalization in front of the white board in our office.

Alistair Edwardes and Ross Purves who edited the English of this thesis.

Laser-Scan Ltd. – especially Paul Hardy – for supporting me through a research licence for the GIS LAMPS2 as well as technical support.

The Swiss Federal Office of Topography and the Swiss National Park for providing test data.

All the colleagues of the GIS and GIA division for providing always a friendly and fun working atmosphere in the department.

My parents Brigitte and Josef Bailer, my ‘best’ sister Cornelia Galanda and my ‘best’ brother Hannes Galanda who support me in every situation.

The ‘Raxmapa’-Gang Andi Neumann, Lilli and Peter Sykora, Alex and André Winter, whom I can count on at any time.

The Viennese cartographers related to ‘1993’, the YAKs and the ‘WG’-family for fun and sporty adventures.

The ‘Long Joggers’ for running hundreds of kilometers up and downhill, in and around Zurich. I will miss you!

And most of all my wife Beate for being as she is, her never ending understanding and making over 60 night train trips over weekends from Vienna to Zurich.

This research work is part of the project ‘GENDEM: Map Generalization for Thematic and On-Demand Mapping in GIS’ supported by the Swiss National Science Foundation under contract 20-59249.99.

Contents

Chapter 1

Introduction

1.1 Motivation

The technological revolution of the last 40 years through the emergence of personal computers, the development of Geographic Information Systems and the formation of the Internet and wireless Internet has initiated a paradigm change in cartography. Morrison (1999) argues that this emergence has led to a redefinition of cartography. The traditional analogue (paper) map had to serve different purposes, namely the storage and inventory of spatial data as well as its cartographic representation. Today these tasks are divided. While geographical data are stored and maintained in databases, maps are exclusively used for the visualization of spatial data. Since Geographic Information Systems (GIS) implement this concept map production relies to a significant degree on such systems (Böhme and Illert 2002). In other words, the distinction of maps has changed from a multi-purpose product to (temporary) visualizations of numerical data, that is, digital landscape models and databases, respectively.

Goodchild (1999) characterizes maps as being flat, at fixed scale, static, generic and slow. However, none of these attributes are characteristics that meet geographical data. Digital technology may allow not only the production of on demand, faster and cheaper maps but also overcome some of the drawbacks of paper maps. For instance, digital maps need no longer be fixed in scale, static or exhaustive. Hence, both the new potential of maps and the new technologies initiated the following trends in cartography:

- **Master database.** The vision of national mapping agencies (short: NMA) for map production is to derive automatically any product at arbitrary scale from a single detailed master database that covers the highest needed resolution (scale). Such a database could be the easiest and most flexible way to store and maintain geographical data if methods of automated map generalization are available that allow one to derive on demand arbitrary scales and products from such a database. Since such methods are still lacking geographical data are often stored in so-called multi scale databases that contain diverse data sets at different scales (Kilpeläinen and Sarjakoski 1995). Map products are then derived from these data sets. In doing so, the bottleneck constitutes the update of the different data sets according to changes in the real world.

- **Increased use of thematic maps.** Commercial GIS software packages commonly used offer functionality to easily create maps from spatial data. Furthermore access to geographical data has become easier in the past since they often are available in digital formats and sometimes over the Internet. Thus, the generation of maps is no longer the preserve of only cartographers. In other words, everybody can make a map that meets exactly their needs. This democratization of cartography (Morrison 1994) has led to increased production and use of thematic maps.

- **Web mapping.** In recent years the Internet has become a major new medium for cartography. Maps in the Internet help to locate points of interests such as restaurants, hotels or specific addresses, to visualize natural hazards, to plan trips etc. For instance, the map service provider MapQuest.com[1] reported 20,000,000 map requests on a daily basis in the year 2001 (Peterson 2001). The success of web maps can be attributed mainly to the fact that they are easy to access and adaptable to the user's needs with respect to the content (Reichenbacher 2002) and the scale (Cecconi and Galanda 2002). In other words, they are interactive and dynamic.

- **Mobile cartography.** Mobile cartography deals with the dynamic and interactive visualization of spatial data on mobile devices such as PDAs or Smartphones (Reichenbacher 2001). It combines the advantages of web mapping, i.e. interactivity, up-to-dateness and adaptability, and a paper map, i.e. its mobility. The definite goal of mobile cartography is to provide the user with up-to-date, spatial information with respect to their location whenever the need arises.

These trends reveal not only emerging research challenges in the field of cartography but also point out the necessity to strengthen research effort related to map generalization and motivate this thesis.

Map generalization is a basic task in cartography, allowing the comprehensible visualization of geographic data at a reduced scale. The goal of generalization is to stress the salient objects while omitting less important ones with respect to the scale and the purpose of a map. In doing so, the readability should be ensured as well as the geographical meaning of the map and the objects, respectively, be preserved as faithfully as possible (McMaster and Shea 1992, Weibel 1995b). Map generalization is an over-constrained and ill-structured problem, that is, there exists no best solution for a certain conflict *per se* (Weibel and Jones 1998). Additionally, map generalization has to compromise between different competing needs (i.e. generalization constraints) and spatially correlated objects[2] (Weibel and Dutton 1998, Ruas 1999). In other words, the satisfaction of one generalization constraint usually implies the violation of another one and the modification of an individual map object often has side-effects on other map objects. For instance, the enlargement of an object may lead to an overlap with another object. In both manual and interactive generalization the holistic reasoning and compromise is delegated to the cartographer who does the generalization according to their knowledge, experience etc. (Weibel 1991). In practice, two cartographers will rarely provide identical solutions to the same generalization task.

Since both manual and interactive generalization are time-consuming and costly cartography strives for automation of the generalization process. Although progress in generalization research is reported continuously[3] some key generalization tasks have not been solved satisfactorily so far (Müller et al. 1995b). The reasons can be attributed to the the difficulty to model, on the one hand, the capacity to compromise between competing generalization constraints and objects and, on the other hand, holistic decision making in generalization (Müller and Wang 1992, Richardson and Mackaness 1998). Thus, automated map generalization remains one of the most challenging research topics in cartography with the following key items:

- In *(NMA) map production* both the automated derivation of arbitrary scales from a master database and the automated update of multi scale databases demand methods of cartographic generalization (Kilpeläinen 1997, Ruas 2001b, Harrie 2001).

[1] The service is available at http://www.mapquest.com/ (accessed 03/08/2003).

[2] While generalization constraints are competing in order to achieve a compromise between their compliance (map) objects are competing for map space in order to obtain a 'geometry' that allows them to be well readable.

[3] Compare, for instance, the working papers presented at the biannual Workshops on Progress in Automated Map Generalization, which are organized by the Commission on Map Generalization of the International Cartographic Association (ICA). Working papers are available at the commission's webpage http://www.geo.unizh.ch/ICA/ (accessed 15/11/2002).

- Methods of automated map generalization are a prerequisite for efficient *adaptive mapping* with respect to scale and an user's position (Cecconi and Galanda 2002, Reichenbacher 2002).
- Automated generalization of transmitted and displayed data could help to ensure short loading times and high cartographic quality in *web and mobile cartography* in spite of restrictions imposed by technology (Buttenfield 1999).
- Maps are made in an ever increasing number by people with little cartographic knowledge. Thus, cartographic expertise should be included in map making software, among other things, through methods of automated map generalization in order to facilitate mapping and enable a higher quality of generated maps.

So far, generalization research has focused mainly on the development of methods for the (automated) generalization of data (e.g. buildings, road networks etc.) relevant to topographic maps since the most important drivers of map generalization were NMAs (Weibel 1995a). Proposed methods dealt with the solution of independent problems such as line simplification or label placement and the orchestration of generalization operations in order to establish automated generalization processes (Müller et al. 1995b). Significantly less attention was paid to the generalization of thematic maps (Bader and Weibel 1997). One of the most frequent data types in thematic maps and GIS, respectively, are categorical coverages. Categorical coverages show the variation of a single variable by a finite number of discrete categories, such as geological units or types of land use (Goodchild et al. 1992, Jaakkola 1997). In this thesis the term coverage is used as "a metaphor for phenomena found on or near a portion of the Earths surface" (Open GIS Consortium 1995, p. 1). Thus, 'coverage' does not imply the coverage's representation, that is, either in the raster or vector model.

Methods designed specifically for the generalization of categorical coverages are rare in commercial GIS and cartographic systems. Hence, users are often condemned to use simple tools such as line simplification algorithms for the generalization of polygonal subdivisions (= categorical data represented by vectors) or grow and shrink algorithms for the generalization of categorical data represented by rasters (Galanda and Weibel 2002b). Unsatisfactory results are not surprising as the special topological and semantic structure of categorical data is ignored (cf. section 2.3). In research, better algorithms and concepts were developed and reported, for instance, by De Lucia and Black (1987), Jones et al. (1995), Weibel (1996), Bader (1997) and Ware and Jones (1998). But, they are all implemented in different prototype systems, which tend to accomplish the solution of (very) specific individual problems rather than the generalization of a whole categorical coverage. Hence, what is missing in categorical generalization is a comprehensive system (i.e. a framework) that combines these different algorithms and other concepts of map generalization such as generalization constraints (Beard 1991, Weibel 1996, Ruas 1999) or levels of analysis (Ruas 1999, Barrault et al. 2001) into a single generalization process.

1.2 Objectives

This thesis investigates the automation of polygon generalization by means of a multi-agent system (MAS). It continues research conducted during the AGENT project[4] by extending methods and concepts – developed for the generalization of road networks and urban settlements – for use with polygonal subdivisions. MAS technology meets well the needs of automated map generalization by providing a basis for the modelling of decision making (i.e. the orchestration of generalization tools such as measures, constraints and algorithms), that is, the compromising between different competing constraints and map objects. In other words, MAS technology enables a holistic view of map generalization and a generalization process that can dynamically adapt to different and continuously changing situations. These characteristics are essential to successful automation of

[4]More information on the AGENT project is provided in section 3.4 and at http://agent.ign.fr/ (accessed 11/15/2002).

map generalization since map objects are spatially correlated and the solution of one conflict creates often new conflicts.

During the AGENT project a consortium of experts in multi-agent systems, experts in automated map generalization and the GIS vendor Laser-Scan succeeded in setting up a MAS for generalization tasks of topographic mapping (Lamy et al. 1999, Barrault et al. 2001). The AGENT package has already proved useful in map production (Bengtson 2001, Duchêne and Regnauld 2002, Lemarié 2003).

This work focuses on polygon generalization, that is the generalization of categorical data represented by vectors, so-called polygonal subdivisions. This data type is amongst others used in digital landscape models (e.g. VECTOR25 and VECTOR200 in Switzerland, BD TOPO® and BD CARTO® in France) and is thus the origin of many topographic and thematic maps.

With respect to automation of polygon generalization the main research questions of this thesis are:

- How can the process of comprehensive, automated polygon generalization be modelled?
- Can MAS technology meet the requirements of comprehensive, automated polygon generalization?
- How can the generalization constraints for automated polygon generalization be formalized and integrated into an agent-based generalization approach?
- Are energy minimizing techniques suited to resolve conflicts in automated polygon generalization?

This research intends to contribute to both cartography and Geographic Information Science by providing improved concepts and tools for the generalization of polygonal subdivisions. The implemented framework will allow at a global level the evaluation of MAS technology for polygon generalization and, at a local level, the identification of problems not considered so far in the automation of polygon generalization. With respect to map production it is expected that the results will initiate the implementation of additional, automated generalization functionality in map making software. Hence, (NMA) map production could become cheaper through a higher degree of automation and more flexible according to offered scales and products. Additionally, it is hoped that a wider range of scales of polygonal subdivisions may help to facilitate spatial analysis of such data. This situation may be attributed to the fact that the representation of a polygonal subdivision at various scales allows better identification of such characteristics that are bound to a specific level of abstraction (i.e. scale). Furthermore, a deeper insight into the process of automated polygon generalization and a platform for future research on this topic are expected to be developed.

1.3 Structure of the thesis

The thesis is subdivided into 8 chapters. An introductory part (chapters 2 and 3) is followed by the main part (chapters 4 to 7). The last chapter presents some conclusions of the PhD project and an outlook for future research. A short preview on the contents of the chapters is provided below:

- **Chapter 2.** Chapter 2 introduces digital generalization, the specific data structure of categorical data and highlights specific aspects that have to be considered when generalizing such data. Next, it provides an overview of previous research according to automated generalization of categorical data either in the raster or vector data model.

- **Chapter 3.** This chapter discusses different approaches to map generalization, i.e. possible ways to orchestrate different concepts of map generalization (including generalization algorithms, measures, constraints etc.) to a consistent generalization process. Interactive generalization, workflows of generalization operations, generalization in expert systems, generalization in MAS, optimization techniques and iterative improvement enabled by simulated annealing are presented as possible options. The examination of potential approaches to automated generalization is the basis for the development of a framework for automated polygon generalization.

- **Chapter 4.** This chapter outlines a framework for the automated generalization of polygonal subdivisions within a MAS, i.e. the AGENT prototype. Starting from the basic concepts and functionality of the AGENT prototype this chapter specifies and discusses the framework used throughout this project. Finally, a worked example clarifies the ideas and concepts behind the proposed framework. The successful implementation of such a framework relies to a great extent on the availability of suitable generalization algorithms and constraints for polygon generalization.

- **Chapter 5.** Chapter 5 presents the application of snakes, an energy minimizing optimization technique, to polygon generalization. A corresponding algorithm is outlined and implemented. It focuses on the solution of conflicts caused by the violation of metric constraints, that is, size and proximity conflicts. The potential of using energy minimizing techniques for the simultaneous solution of metric conflicts in map generalization is shown in several experiments.

- **Chapter 6.** The specification of generalization constraints for polygon generalization establishes a basis for the implementation of the proposed framework. Hence, this chapter proposes not only a basic set of such constraints but also provides methods for an evaluation of the satisfaction of individual constraints. It also discusses so-called plans, i.e. algorithms and parameters, that are capable of improving the satisfaction of a constraint with respect to a certain object. An investigation of the interrelations between constraints and importance values that might help to mitigate between different constraints complete this chapter.

- **Chapter 7.** This chapter presents the synthesis of the previous chapters, that is, the implementation of the framework outlined in chapter 4 that relies on algorithms developed in chapter 5 and the constraints proposed in chapter 6. The generalization of a test area by means of the implemented prototype is demonstrated. The evaluation of the results introduces a discussion of both the potential of an agent-based approach to polygon generalization and the quality of the suggested framework. Some final remarks on the implementation in LAMPS2 lead to the conclusions on the project as a whole.

- **Chapter 8.** This chapter concludes the thesis by pinpointing the main achievements and identifying the need for additional research. An outlook on future research challenges with respect to the automated generalization of polygonal subdivisions is given.

Chapter 2

Polygonal subdivisions and their generalization

2.1 Generalization in digital cartography

Generalization in cartography (also known as map generalization, or cartographic generalization) is defined as "the selection and simplified representation of detail appropriate to the scale and/or purpose of the map" (ICA 1973, p. 173). Map generalization[1] strives for a reduction of the complexity of a map in order to achieve a generalized map that is of high graphical quality even at a smaller target scale. Hence, map generalization should emphasize the essential while suppressing the unimportant as well as maintain logical and unambiguous relations between map objects (Weibel 1997). Detailed discussions on the basic needs and issues of map generalization are provided by McMaster and Shea (1992), Müller et al. (1995a), Weibel and Dutton (1999), SSC (2002) and Ruas (2002).

With a focus on digital map generalization McMaster and Shea (1992) defined generalization as the derivation of cartographic data from a data source (data set at a larger scale) through the application of spatial (geometric) and semantic transformations. Thus, the generalization process "goes from a detailed description of a geographic object, considering each part of the object, to a more abstract description of the object, retaining only properties of the object relevant to the map user's needs" (Mustière et al. 2000, p. 56). In digital cartography three types of generalization are commonly distinguished (Grünreich 1985, Müller et al. 1995b, Weibel 1997):

- *Object generalization* takes place whenever a digital representation (e.g. map, database) of the real world is generated. Since it is impossible to capture the real world with infinite detail an abstraction (generalization) occurs.
- *Model generalization* performs a controlled reduction of data (Morgenstern and Schürer 1999), i.e. it deals with the derivation of data sets of reduced accuracy and/or resolution. Model generalization is specific to digital generalization from a more detailed data set.
- *Cartographic generalization* denotes the generalization of spatial data for purpose of visualization. Hence, this generalization considers the symbolization of data and must encompass the resulting conflicts such as symbol overlaps, congested objects etc. (Weibel 1997).

Map generalization is an over-constrained and ill-structured problem, that is, no unique and perfect solution exists *per se*. Generalization should maintain a data set as faithfully as possible while ensuring its readability despite a smaller display scale (and hence smaller available map space). It is evident that a compromise between these two basic objectives has to be found. Even on the level of an individual object and a single conflict several appropriate solutions are usually possible. For instance, an object that is too small to be well perceived can be enlarged,

[1] In this thesis the term generalization is used as a synonym for map generalization.

eliminated or aggregated with neighboring objects in order to achieve a satisfactory solution. However, each of these operations may lead again to the violation of other constraints, for instance, the enlargement and elimination operations change size ratios and the elimination and aggregation operations influence the (characteristic) distribution of objects. Hence, the solution of this local conflict again results in a compromise between different constraints. In manual generalization the cartographer who generalizes a data set compromises between constraints based on mapping guidelines, the consideration of the specific conflict, and their experience. Of course, this process is highly subjective and thus no two cartographers are likely to come up with the same solution. In automated generalization the difficulty is to translate this capacity of compromising between constraints into an automated procedure and to model decision making in generalization.

The state of the art provided below focuses exclusively on categorical data and their generalization. A general overview of past and recent developments in map generalization is presented by, for example, Buttenfield and McMaster (1991), Müller et al. (1995a), Weibel (1995), Richardson and Mackaness (1998) and Weibel and Jones (1998).

2.2 Definitions

Categorical coverage

A categorical coverage represents the variation of a single variable by a finite number of discrete, nominal categories (Goodchild et al. 1992, Jaakkola 1997). The mapped area is covered by mutually exclusive and space exhaustive objects[2] (Beard 1988, Mark and Csillag 1989), that is, every location in the map must only belong to a single class. A categorical coverage consists of two connected components: (1) a spatial component and (2) a thematic component. "Using Sinton's terminology (Sinton 1978), such data sets hold time fixed, control for theme, and measure the location, by searching for the largest areas with uniform properties" (Frank et al. 1997, p. 217).

Categorical data is a frequent data type in GIS applications (e.g. classified terrain parameters such as slope or aspects), in thematic maps (e.g. portraying geological units, land use or habitats of animals) and in topographic maps (e.g. types of land use such as forest or glacier).

Categorical generalization

Categorical generalization denotes the generalization of a categorical coverage. This term does not imply the data model, i.e. raster or vector, that is used to store the generalized categorical data.

Polygonal subdivision

A categorical coverage represented by vectors is termed a polygonal subdivision[3] (or polygon mosaic, or polygon map). Figure 2.1, for example, shows an extract of the layer 'primary surfaces' of VECTOR25. VECTOR25 is the large-scale landscape model of Switzerland, its content and geometry is based on the Swiss National Map 1:25,000[4].

Polygon generalization

The term polygon generalization defines the generalization of polygon subdivisions, that is, the generalization is performed in the vector data model.

[2] An object in the generalization of categorical data denotes a region (= a set of 4- or 8- connected cells of the same category) in the raster data model and one polygon in the vector data model.

[3] The terminology for describing the components of a polygonal subdivision is explained in appendix A.

[4] Detailed information on this data set can be found at http://www.swisstopo.ch/en/digital/INDEX.htm (accessed 11/15/2002).

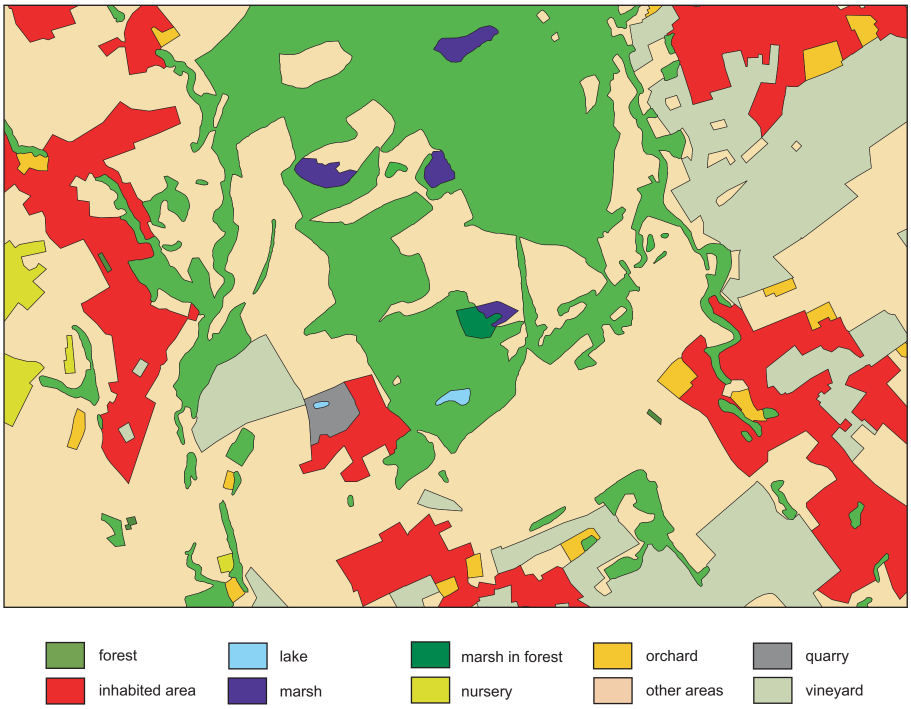


Figure 2.1: A polygonal subdivision that shows the 'primary surfaces' layer of VECTOR25, the digital landscape model of the Swiss Federal Office of Topography – figure at scale of 1:25,000. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

2.3 Generalization of categorical data

The generalization of categorical coverages implies transformations of categorical data sets on both the semantic and the spatial level. On the semantic level, categories are aggregated to higher-level categories, for example the categories 'deciduous forest' and 'coniferous forest' are combined to a category 'forest'. On the spatial level, object geometries are adapted to the perceptual limits imposed by the new (smaller) scale. So, an object is, for instance, enlarged or eliminated if its area is too small to be well perceivable.

As an alternative to modifying the spatial and semantic structure of a polygonal subdivision a generalization effect can be achieved by changing the geometric primitive used to represent an object. For instance, a reduction of dimensionality between an object (e.g. polygon) and its representation (e.g. point symbol) always indicates generalization (Muehrcke 1986, McMaster and Shea 1992). Generalization by changing representation is only mentioned for the sake of completeness. Since the symbolization process is usually seen as an independent process in map making that takes place after the generalization process (Cromley 1992) this type of graphical generalization is not taken into account in this work.

The specific topological and semantic structure of categorical coverages imposes some basic requirements for their generalization:

- Every significant step in the generalization process must represent a consistent (topologically correct) state of a categorical coverage. Likewise, thematic constraints must be respected in order to guarantee the thematic validity of a categorical coverage, for instance, by preventing illogical neighbors, for instance a lake in the sea.

- The mere omission of 'less important' objects, one of the main processes of map generalization, is not feasible because it would invalidate the exhaustive subdivision of space. Generalization has to be accomplished by a combination of cartographic operators, which assign the area of an 'omitted' object to a new category or to some of the neighboring objects, respectively.

- The spatial transformation of one object always requires the modification of at least one other object. Hence, every change of an object's geometry or category must be propagated at least to all adjacent objects.

- Categorical generalization has to compromise between ensuring the legibility of the generalized categorical coverage and preserving the original coverage's characteristics (e.g. the spatial distribution of categories, the shape of an object).

Furthermore, automated generalization can also benefit from the specific data structure since possible solutions are constrained and the neighborhood of generalized objects is explicitly defined. Of course, the argument is only valid if categorical data are generalized independently, that is, without consideration of any additional data.

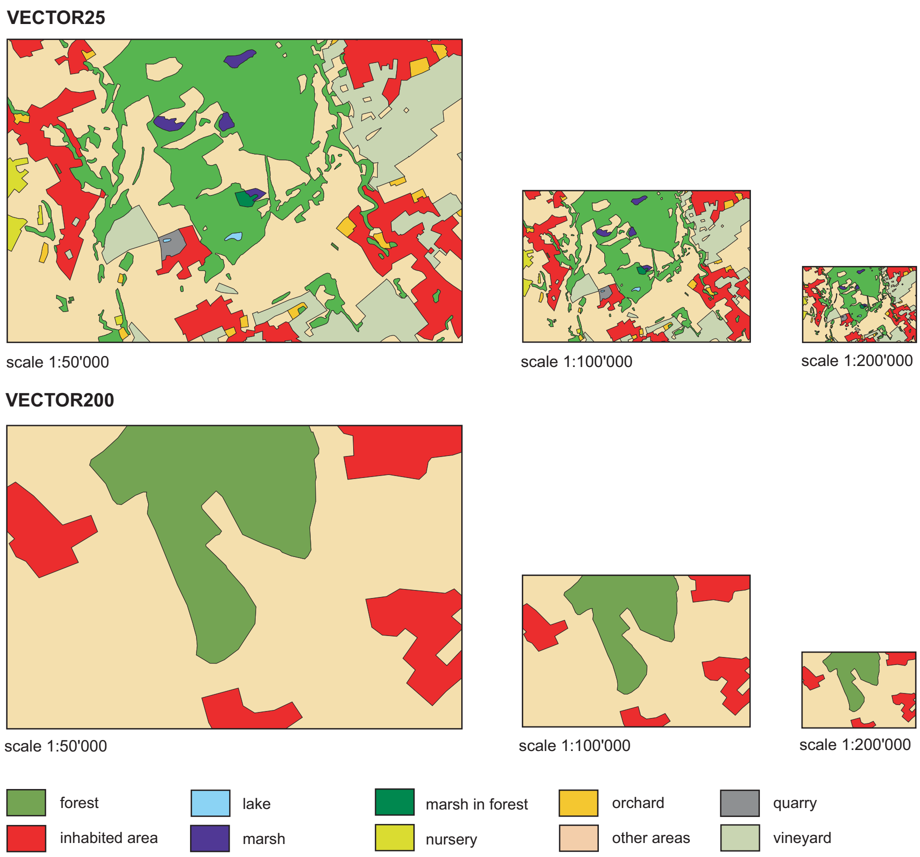


Figure 2.2: The need for categorical generalization demonstrated by a comparison of the 'primary surfaces' layers of VECTOR25 and VECTOR200 at different scales, namely 1:50,000, 1:100,000 and 1:200,0000. The same clipping of VECTOR25 is shown in Figure 2.1 at a scale of 1:25,000. Data: VECTOR25/VECTOR200, reproduced by permission of swisstopo (BA035273).

Although VECTOR25 and VECTOR200[5] are independently derived, a comparison of their 'primary surfaces' layers at different scales (Figure 2.2) may help to better understand the need for map generalization in general and the main aspects of categorical generalization in particular. The primary trigger of generalization is the reduction of available map space for portraying real world phenomena in accordance with a decrease in scale. For instance, a scale change from 1:50,000 to 1:100,000 (factor 1/2) reduces map space to a 1/4 of the previous map space, from 1:50,000 to 1:200,000 to 1/16 and so on. If a categorical coverage is shown at a smaller scale than its original scale without generalization, such as VECTOR25 in Figure 2.2, its legibility is reduced, since information is too dense and too small to be properly perceivable by the map user. In order to achieve graphical clarity and ease perception in spite of a smaller scale representation, categorical generalization needs to reduce the complexity of the categorical coverage. So-called generalization operators describe the spatial and semantic transformations activated to obtain such a reduction.

Generalization operators. Generalization operators (or generalization operations, generalization procedures, depending on the author) designate transformations in the generalization process on a conceptual level. They decompose the overall process into sub-processes (McMaster and Shea 1992, Weibel 1997, Weibel and Dutton 1999). So far, cartographers do not agree to a common base set of required operators for map generalization; the same operations are even named and described differently (Rieger and Coulson 1993).

Figure 2.3 provides a textual and graphical overview of the operators – used in this work – for categorical generalization. Besides the distinction of operators for semantic and spatial transformations, generalization operators are further subdivided according to their scope. This scope relates to those objects of the categorical coverage which are potentially modified by an operator, that is, the whole map, a group of objects, an individual object or its boundary.

Generalization algorithms. "A generalization algorithm is a formal mathematical construct that solves a generalization problem by changing an object's geometry or attribute" (Bader et al. 1999, p. D.1). Algorithms represent concrete implementations of generalization operators. The specific structure of categorical coverages and the desired automation of the generalization process determine some requirements on algorithms for categorical generalization. A suitable algorithm should:

- not require any interactive input;
- support a global generalization approach, i.e. several conflicts can be solved simultaneously;
- ensure consistency of the data set according to the definition of a categorical coverage, i.e. every location in the map must belong to only one category;
- preserve characteristics of the categorical coverage (e.g. distribution of categories, alignments of objects, or specific object shapes etc.); and
- be able to handle different types of categorical data and scale ranges.

Since the objectives of some of the above specifications are in conflict with one another algorithms have to reach a compromise between these different requirements.

Handling side-effects. Due to the space exhaustive character of categorical coverages the transformation of an individual object always requires the subsequent adaptation of all adjacent objects, i.e. at least one other object. In other words, all modifications of a generalized object must be propagated to the coverage. These required adaptations are called side-effects. The handling of side-effects was investigated with respect to the generalization of roads in road networks (Nickerson 1988, Bader 2001) and disjoint polygons such as buildings (Burghardt 2000, Højholt 2000). Side effects can be treated as an independent subtask of the generalization process, that is, changes are propagated to the neighborhood once the transformation (e.g. displacement) is completed. This

[5]Besides VECTOR25 the Swiss federal office of topography offers a small-scale digital landscape model based on the Swiss National Map 1:200,000, namely VECTOR200. For more detailed information about this data refer again to http://www.swisstopo.ch/en/digital/INDEX.htm (accessed 11/15/2002).

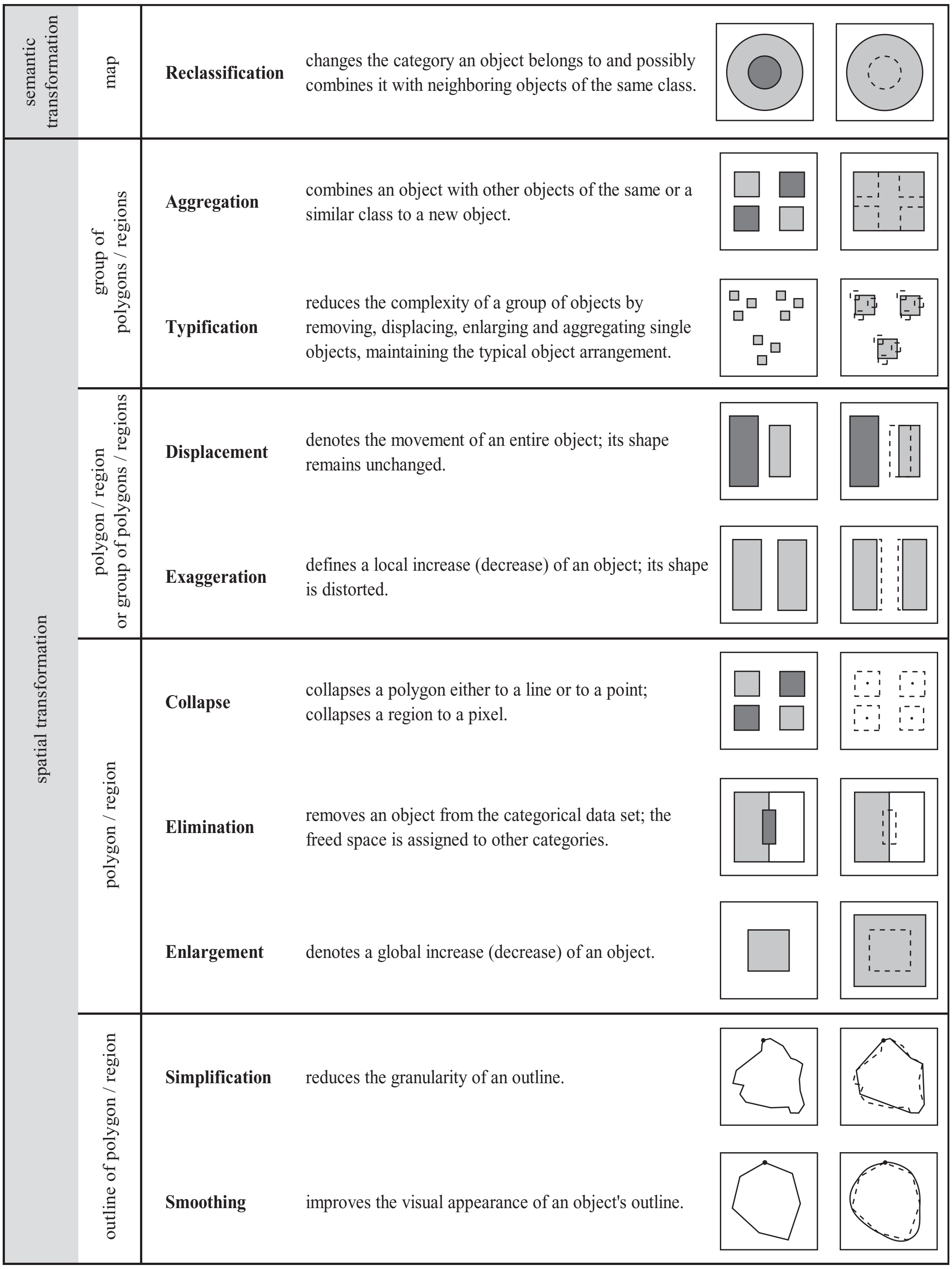


Figure 2.3: Operators used in categorical generalization.

approach risks the formation of new conflicts after the propagation and, hence, tedious loops of transformations and propagations (Bader and Barrault 2000). Contrarily, Burghardt (2000), Bader (2001) and Harrie (1999), among others, made use of techniques that allow the treatment of transformation and propagation in a global sense, that is, both processes are performed concurrently and provide a common solution.

Semantics. When generalizing categorical data semantics play a major role (Le Men 1996). They enable semantic transformation and control the decision making for geometric operations. However, the spatial and the semantic components of a categorical coverage are intimately linked, and any treatment of one in isolation form the other will have a risk of misrepresenting the phenomenon (Mark and Csillag 1989). Thus, knowledge about the primary classification scheme of an expert on the map's theme (and sometimes even auxiliary data) are indispensable to produce not only a 'good looking' but also a meaningful aggregation of classes and generalization of a data set (Imhof 1972), respectively. In categorical generalization semantics are of overriding importance for a variety of reasons:

- They help in selecting the operator applied to a particular conflict. For instance, an isolated object of a rare category may be emphasized rather than eliminated or a significant group of 'too small' disjoint polygons may be aggregated instead of removed.
- Semantics control the reduction of the number of classes displayed (e.g. the combination of classes to so-called superclasses, for instance, the combination of the classes 'deciduous forest' and 'coniferous forest' into a class 'forest'). Beyond any doubt reclassification is the most important operator in reducing the complexity of a categorical subdivision (Jaakkola 1998), especially when performing prominent scale changes (e.g. from 1:25,000 to 1:100,000 or 1:200,000). Further, the geometric combination of adjacent objects of the same category (resulting from reclassification) implies the solution of potential geometric conflicts such as polygons that are too small or too narrow (Spiess 1990).
- The most likely neighboring object to fill the gap which remains after the elimination of an object is identified with respect to semantics. The compatibility of two categorical objects depends, besides their geometric properties, mainly on their semantic similarity. Yaolin et al. (2002b) proposed a measure of semantic similarity for categorical database generalization. The filling of the gap may alternatively be accomplished by a geometric solution, which distributes the area of the eliminated object to the adjacent objects (Bader 1997, Bader and Weibel 1997).
- Semantics influence the resistance of a polygon boundary to deformation. That is, the modification of so-called soft boundaries[6] (e.g. the outline of a scree area) is preferred to that of boundaries considered as 'hard' (e.g. the outline of a lake).
- The validation of a solution obtained according to criteria of semantic plausibility, for example, a lake in the sea is not acceptable.

Constraints. Beard (1991) suggested using constraints instead of rules in map generalization. She defined a constraint as "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) Alternatively, a constraint can be thought of as a design specification, to which an object, a group of objects or a map should adhere (Weibel and Dutton 1998). The constraint-based approach to map generalization was emphasized by Ruas and Plazanet (1996), Weibel (1996), Weibel and Dutton (1998), Ruas (1999), Lamy et al. (1999) and Barrault et al. (2001). Since the constraints defined on an object often contradict, a generalization solution is always a compromise between different constraints and their satisfaction. For instance, it is impossible to avoid shape distortion of a polygon if the polygon outline is illegible at the target scale due to too many details and thus requires simplification.

Peter and Weibel (1999a), Edwardes and Mackaness (2000) and Weibel (1996) discussed constraints to categorical generalization on a conceptual and design level. Though constraints were successfully used in other domains of map generalization, the implementation of a constraint-based

[6]The classification into soft and hard boundaries refers to the fuzziness of a boundary in the real world.

approach to categorical generalization was not reported in literature so far. Constraints in map generalization are closely linked to measures helping to calculate the satisfaction of a constraint[7].

Measures. "A measure is a method that does not change the state of map objects, but is used to characterize it"(Bader et al. 1999, p. D.1). In map generalization measures are used for several purposes:

- Measures are applied to cartographic conflict detection, that is, they determine the need for generalization. In a constraint-based approach to generalization this step corresponds to the evaluation of all constraints defined on map objects.
- The choice of an appropriate generalization operator with respect to the data's semantic and spatial structure often relies on measures. For instance, isolated polygons of the same category may be aggregated while isolated polygons of different categories may be displaced in case of a proximity conflict.
- Measures help to detect significant structures in maps such as groups of map objects (e.g. clusters, alignments) or spatial characteristics (e.g. in the object shapes, in the distribution of objects), which should be preserved or at least taken into account during the generalization process.
- Parameters of generalization algorithms are derived in consideration of the severity of a conflict by means of measures.
- Measures are used in the evaluation of the quality of a generalization solution in comparison to a previous state in order to decide whether a triggered algorithm solved a conflict. With respect to constraints this step examines the change (improvement or aggravation) of the individual constraint's satisfaction.

Peter (2001) studied measures focusing on categorical coverages and compiled a preliminary inventory of measures needed for automated polygon generalization based on earlier works such as Bader (1997) and FRAGSTATS (1994). Chapter 6 deals explicitly with both constraints and measures for polygon generalization and their integration into a comprehensive framework for automated polygon generalization.

Levels of analysis. Ruas (1999) proposed representing explicitly the relations of objets that reflect much of the geographic world's semantics (Mustière and Moulin 2002) for map generalization. In doing so, map space is organized in so-called levels of analysis (spatial levels of map generalization) namely a micro, meso and macro level of analysis (Ruas 1995, Barrault et al. 2001, Ruas in press):

- The *micro level* is attached to individual geographic objects such as a river, a building, a road etc. It deals with the independent generalization of objects.
- The *meso level* is dedicated to a group of objects, for instance, all buildings of a city block or all the roads of a town, and their contextual generalization. An object on the meso level can consist of objects that belong either to the micro or to the meso level.
- The *macro level* refers to a population of objects, for instance, all the buildings or all the polygons of a data set. The main task of this level of generalization is to guide and control the generalization of its population.

These different levels of analysis establish a hierarchical organization of geographic objects. The organization offers multiple hooks to map generalization, that is, for instance, to coordinate map generalization or to define a sequence and hierarchy of objects in generalization. Hence, in considering spatial levels and their hierarchy map generalization can benefit in such a way that better generalization solutions are achieved more efficiently (Barrault et al. 2001). While this concept is successfully applied to the generalization of topographic maps – see, for instance, Barrault et al.

[7]The satisfaction of a constraint describes to which level a constraint is fulfilled (or conversely, violated). The satisfaction of all constraints attached to an object is summarized by the so-called happiness of an object.

(2001) and Duchêne et al. (2001a) – levels of analysis have not been examined according to categorical generalization.

The following two sections provide an overview of methods used and developed explicitly for categorical generalization in the raster and the vector data models.

2.4 Generalization of categorical data in the raster data model[8]

The raster data model is a form of tessellation, and as such it is (2-D) space-covering. This property would seem to make it ideally suited to develop algorithms for categorical generalization. For the very same reason tessellations (in the form of triangulations) are also used in algorithms in the vector domain. Tessellations provide a representation of space that can be exploited for contextual generalization operations such as typification, aggregation or displacement. In contrast to vector-based tessellations rasters are much simpler to handle as a data structure, due to the regular tessellation of space and due to the fact that adjacency is implicitly represented in the matrix structure. Hence, it is not surprising that some of the earliest algorithms for categorical data generalization are based on raster mode representations (Monmonier 1983). The major problem in raster-based generalization is that the quality of the solutions depends to a large degree on the spatial resolution of raster data (Peter and Weibel 1999a). Jaakkola (1994, p. 14) states that "the pixel size should be well below the minimum mapping unit".

Besides easy implementation, however, the development of raster mode generalization algorithms is also justified by the fact that the regular grid is a frequently used sampling scheme for categorical data, in particular for data that originate from remote sensing sources. Examples include CORINE Land Cover (CECA-CEE-CEEA 1993, Fuller and Brown 1996, Jaakkola 1997) or Swiss Landuse Statistics (SFSO 1972, 1992, Peter 1997). The close relationship to remote sensing imagery is also reflected by the fact that many of the existing raster mode algorithms for categorical generalization are based on image processing operations. Such operations can take the form of simple pixel or kernel-based operations (Serra 1982) or more sophisticated procedures as in the example of Jaakkola (1997, 1998) where several primitive operations are combined into complex procedures, thus building contextual generalization operations such as aggregation (Jaakkola 1998). One interesting property of such procedures is that they can be implemented using standard image processing or Map Algebra (Tomlin 1990) operations available in most commercial GIS software packages.

Mathematical morphology. Using a particular image processing technique, so-called mathematical morphology (MM, see Serra 1982), Li (1994) and Su et al. (1998) developed generalization operations for raster data. The underlying idea was to use a common mathematical concept that was powerful enough to ensure compatibility of algorithms and allow the development of a complete set of algorithms for all generalization operations applicable to categorical data – c.f. Figure 2.3. A series of articles describe the development of algorithms for various generalization operations, all based on MM – see Su et al. (1998) for a list of respective publications. The contribution of the work of Li and Su to categorical generalization is mainly in its attempt to cover the entire range of generalization operations, based on cartographic principles and using a comprehensive methodology. The assumption was that vector data could be much more easily converted to raster, generalized, and converted back rather than trying to work directly with vector data (Li 1994).

Regions of pixels. The assumption that vector data could be converted to raster and back (Su et al. 1998) neglects the introduction of conversion errors as well as the loss of the object nature of vector polygons (Peter and Weibel 1999a). Also, the proposed algorithms are restricted to binary

[8] This section is an extended and revised excerpt from Galanda and Weibel (2002b).

raster data (e.g. a grow-and-shrink algorithm for aggregation Schylberg 1993) and hence of little use for data consisting of multiple categories. So, in order for raster-based algorithms to successfully implement cartographic principles of generalization, they cannot simply exploit the easy tessellation structure of rasters but must also attempt to 'mimic' the object nature of polygons represented in the vector data model. Hence, algorithms based on moving kernels (windows) or moving structuring elements (such as in MM) are insufficient and must be extended into the direction of an approach based on regions (zones) and neighborhood relationships rather than individual pixels. Various attempts are leading in this direction. Jaakkola (1998), in addressing the derivation of generalized land cover data for the CORINE Land Cover database, developed procedures using a combination of so-called focal (i.e. kernel-based) and zonal (i.e. region- based) map algebra operations (Tomlin 1990). While these procedures are fine-tuned for the specific purpose at hand, they do show the potential of using combined, category- and region-based operations. Jaakkola (1997) also shows the analysis of the uncertainty introduced in the generalization process, an important aspect in the derivation of generalized data (rather than maps) which have to satisfy given statistical and analytical requirements.

An earlier, similar approach by Schylberg (1993) first segments explicitly the input raster into connected components representing raster objects. Such objects can then be characterized by shape measures (e.g. area, elongation) and selectively addressed in the generalization process. Schylberg (1993) uses a rule-based approach to control the generalization operations and associated parameters, thus leading to a flexible workflow – c.f. section 3.3. Figure 2.4 shows a sample result from Schylberg (1993).

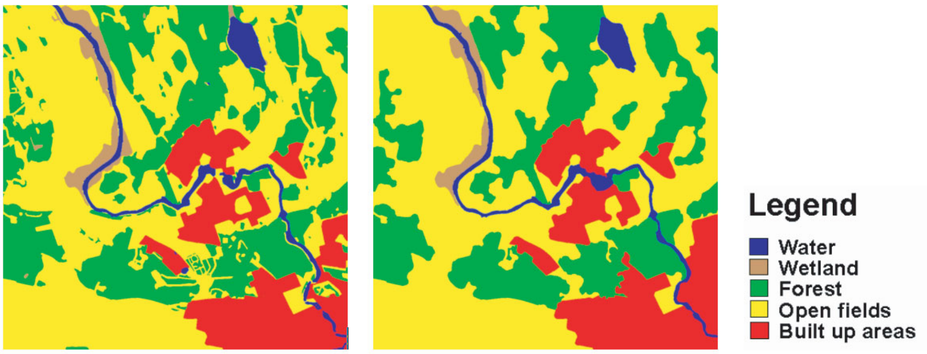


Figure 2.4: Example of raster-based land cover generalization. Original map (left) and the result of combined elimination, aggregation and smoothing (right). (Schylberg 1993, p. 100)

Cost surfaces. Cartographic principles can also be algorithmically addressed using cost surfaces and cost functions, thus representing the degree of resistance to generalization (e.g. certain categories should not be aggregated because they are semantically disparate). Brown et al. (1996) and Peter (1997) presented examples of the use of cost surfaces for modelling the resistance with which a gap between two regions can be bridged in an aggregation operation. For instance, if two forest regions are separated by a water body region, and the water body is given a high priority, it will result in a high cost for the paths between the forest regions, thus preventing the aggregation of the forests – note the categories forest 'F' and water bodies 'W' in Figure 2.5. Alternatively, if the two forest regions are separated by shrubs, which are semantically similar to forests, the resulting paths will have low costs and hence will not provide any resistance to aggregation – note the categories forest 'F' and shrubs 'S' in Figure 2.5. Due to the appropriate quality of the obtained results this procedure has been used in production of land cover maps (Fuller and Brown 1996).

F	F	F	S	W	W
S	S	S	S	W	D
S	S	W	W	W	D
F	F	W	F	F	D
E	F	W	F	F	D
E	E	W	W	D	D

a.

0	0	0	7	40	50
2	1	4	9	32	17
3	1	24	22	21	14
0	0	15	0	0	12
12	0	18	0	0	12
15	11	24	30	18	14

b.

F	F	F	S	W	W
F	F	F	S	W	D
F	F	W	W	W	D
F	F	W	F	F	D
E	F	W	F	F	D
E	E	W	W	D	D

c.

Figure 2.5: Aggregation of the category forest 'F' controlled by costs: **a.** Categorical coverage represented by a regular grid. **b.** Costs for aggregation of category 'F' that were derived from the distance to the least-cost path between candidate regions. **c.** Result of aggregation operation with respect to the category 'F' if the maximum cost that may be bridged is set to 5. (Peter and Weibel 1999b)

Thus, the concept of the cost surface provides a mechanism to not only take into account geometric properties but also semantics. Obviously, it is also simpler to implement cost surfaces in raster mode than in vector mode. Le Men (1996) uses costs in a different way. He proposes a procedure that relies on the minimization of a cost function that measures the difference between the initial and the generalized map. The generalized map is established by searching for a sub-graph of the initial graph of region relations that minimizes the cost function.

2.5 Polygon generalization[9]

Raster-based generalization is reasonable for data collected, stored and analyzed in the raster data model, such as satellite images. Since topographic and thematic map products are usually derived from digital vector data sets this thesis concentrates on the generalization of categorical data represented in the vector data model. The generalization of polygonal subdivisions is termed polygon generalization. Furthermore, it seems that the potential of purely raster-based methods of categorical generalization is rather limited and has been largely exhausted by previous works. Polygon generalization, on the other hand, has so far received less attention.

Algorithms, which are especially designed for the generalization of polygonal subdivisions, are rare both in cartographic software packages and in research. Existing algorithms concern either very specific kinds of data (e.g. Müller and Wang (1992)) or the isolated implementation of spatial algorithms (e.g. De Lucia and Black (1987), Jones et al. (1995), Bader (1997)). A brief overview of existing algorithms for polygon generalization is provided in the following paragraphs.

Delaunay triangulations. The explicit representation of map space by means of a Delaunay Triangulation (DT) has become a common technique for polygon generalization in the past. Amongst others Jones et al. (1995) by using a constrained DT (CDT) and Bader (1997) by using a conforming DT (CfDT) revisiting the concept originally presented in De Lucia and Black (1987). Both modifications of a common DT enforce the edges of the polygons as edges in the triangulation. Thus, the generalization process can benefit in various ways from this auxiliary data structure:

- Both the CDT and the CfDT represent the topological relationship between all map objects explicitly. Their dynamic adaptation during the entire generalization process helps to detect topological errors generated due to generalization (Ware and Jones 1996).

[9]This section is an extended and revised excerpt from Galanda and Weibel (2002b).

- This explicit mapping of proximity relations enables a straightforward detection of conflicting polygons by an analysis of the triangles' geometries (Bader 1997, Ware and Jones 1998).
- The CDT/CfDT data structure offers multiple hooks for the implementation of geometry-based measures and generalization algorithms.

Algorithms based on CDT/CfDT exist for the generalization operators aggregation, collapse, displacement, elimination, enlargement and exaggeration. The first such aggregation procedure for polygonal objects is presented in De Lucia and Black (1987): Disjoint polygons are merged via connecting triangles whose height does not exceed a specified threshold. Thus, island polygons between the merged objects could be created or remain, respectively. An aggregation of objects based on triangles can be achieved alternatively in combination with a previous displacement operation as shown in Jones et al. (1995). The collapse operator is related to the calculation of a polygon's skeleton. Its approximation may result from the set of center points of edges that cross the shape and the center of gravity of those triangles that lie completely inside the polygon (Figure 2.6 middle). Skeleton-based algorithms are described more in detail in De Lucia and Black (1987), Jones et al. (1995) and Bader (1997). An algorithm for the elimination of an object that implies the deletion of the object and the assignment of its area to neighboring polygons is introduced in Bader (1997), Bader and Weibel (1997) and extended in Ai and van Oosterom (2002). Following the construction of the skeleton of the polygon to be eliminated (cf. middle image of figure 2.6), the connections of the incident polygon borders to the nearest point on the skeleton are established. Next, unnecessary skeleton branches are pruned. The remaining parts of the skeleton represent the new border between the neighboring polygons – see the right image of figure 2.6.

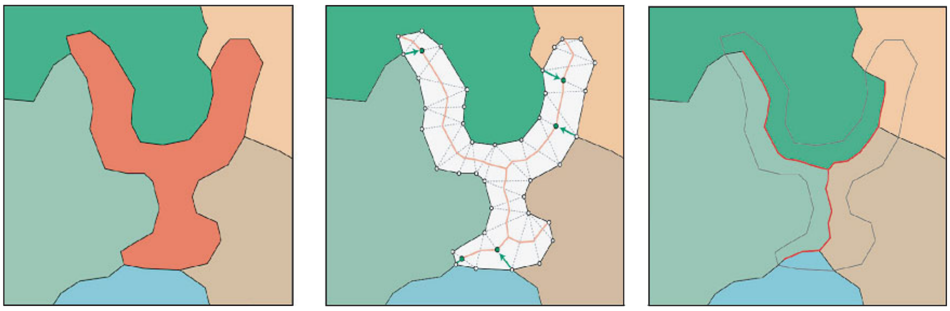


Figure 2.6: Elimination of a polygon based on a CfDT and an approximation of the object's skeleton. The original situation is shown in the left image, the skeleton derived from the object's outline in the middle image and the algorithm's output in the right image. (Bader and Weibel 1997, p. 1529)

Any calculated displacement can be easily applied within the CDT or CfDT as a vector to every vertex of the polygon boundary. Based on that principle Jones et al. (1995) developed an algorithm for an object's enlargement, that is, a displacement vector of the same length derived from incident edges is applied to all points of the polygon outline. Related to that algorithm problems with concave shapes are reported, while convex shapes are handled well. In Bader (1997) a very similar approach is discussed for the exaggeration of polygons, e.g. to widen a narrow section of a polygon. After the triangulation-based geometric transformation of polygon objects a simplification or smoothing of the modified polygon boundaries can help to further improve the visual appearance.

The CDT and CfDT exhibit useful properties for the vector-based generalization of categorical data. They enable robust and efficient algorithms for the solution of local generalization problems. *A priori* the algorithms cannot handle the side-effects of generalization since the creation of new conflicts is not prevented. Triangulations also play a role as auxiliary data structures for polygon generalization with respect to conflict detection, the calculation of measures and the modelling of neighborhood relations (Jones et al. 1995, Ware and Jones 1996, Ruas 1999).

Line simplification. Line simplification is probably the subject most investigated in generalization research. The most commonly used algorithms are the Douglas-Peuker algorithm (Douglas and Peucker 1973, Ramer 1972), the Visvalingam-Whyatt algorithm (Visvalingam and Whyatt 1993) and the bend-simplify algorithm (Wang and Müller 1998). A comprehensive overview of algorithms for line simplification is provided for instance in Bader et al. (1999). Experiments concerning the quality of such algorithms for polygon generalization are presented in Bader (1997) and Saalfeld (1999).

One of the basic requirements of polygon generalization is the topological correctness of the generalized data set, that is, no self-intersections within a polygon outline and no intersections of different objects should occur as a consequence of generalization. One possible concept is to detect and correct topological inconsistencies independently of the generalization process: After the simplification process the conflicting polylines can be either replaced with the original unsimplified polylines (Edwardes et al. 1998) or iteratively enhanced by original vertices (inverse simplification) until topological consistency is reached again (Saalfeld 1999). Another method to avoid topological errors due to simplification is to use more advanced geometric algorithms. An algorithm that prevents any line intersection and additionally ensures points lie on the same side of a line before and after the transformation is reported in de Berg et al. (1998). This method constitutes an enrichment of that algorithm reported in Imai and Iri (1988) and is especially designed for polygonal subdivision simplification.

A simplification of all polylines of a polygon mosaic may produce a less complex appearance and thus a satisfactory solution for small changes in scale. For more important scale reductions, however, the mere application of simplification never succeeds in generating a well-generalized categorical map (Brassel and Weibel 1988). On the contrary, the simplification operation plays a relatively minor role in the overall generalization process compared to other operations.

Other algorithms. Besides these two main groups of algorithms the following methods are relevant to the state of the art in polygon generalization:

- *Elementary geometric algorithms* subsume all the algorithms based on 'simple' vector geometry like: the translation of a polygon or some of its vertices by a vector (Jones et al. 1995, Bader 1997) to achieve the displacement, exaggeration or enlargement of a polygon; scaling algorithms to enlarge a polygon; and aggregation algorithms, which derive the aggregated geometry of a group of polygons by computing an enhanced convex hull (Laser-Scan 1999).

- Müller and Wang (1992) developed a typification algorithm for the generalization of disjoint polygons of the same type. The algorithm is implemented as a sequence of generalization operations, i.e. selection with respect to spatial structure and size of polygons, enlarge polygons, aggregate polygons and subsequently simplify and smooth the polygon outlines. This approach is studied more in detail in section 3.2.

- The *GAP-tree* (Generalized Area Partitioning) denotes a concept for on-the-fly generalization of polygonal subdivisions (van Oosterom 1995). A base data set is enriched by a reactive, hierarchical data structure in such a way that arbitrary scales can be derived from this data set in real time. Map generalization is achieved by dynamic omission of polygons, subsequent filling of the resulting gaps and aggregation of disjoint polygons (van Oosterom 1994, van Putten and van Oosterom 1998, Ai and van Oosterom 2002).

- Molenaar (1996, 1998) and Yaolin et al. (2002a) conducted research on the generalization of *categorical databases*. In this approach categorical objects that require generalization are defined either by the fact that their category is not represented at the target scale or that their size is below a predefined threshold. Conflicts are then solved by aggregation of the categorical objects involved in a conflict. The aggregation operator is guided by an evaluation of semantic similarity among those objects (Yaolin et al. 2002b).

2.6 Conclusions

From an introduction to the field of map generalization and to the characteristics of categorical coverages the requirements for categorical generalization were derived. The state of the art review should have made it clear that categorical generalization attracted less attention in the past compared to other fields in generalization such as line simplification or smoothing. Possible reasons might be, on the one hand, NMAs which drove generalization research over years were mainly interested in the generalization of topographic maps. Since land cover data represented by categorical data is an integrative part of digital landscape models – see Figure 2.2 – a changing research interest of NMAs is beginning to become noticeable. On the other hand, especially the production of thematic maps increased through the widespread use of GIS.

But for all that, the chapter pointed out that previous research achieved a multitude of methods, i.e. algorithms, measures, constraints etc., that were especially developed with respect to categorical generalization and thus they should be considered in continuing research. For combining and enriching these existing methods into a framework for automated polygon generalization a underlying concept of orchestration is required. Hence, the next chapter examines possible approaches to automated map generalization.

Chapter 3

Approaches to process control in map generalization

Research related to polygon generalization focused, so far, on very specific aspects such as algorithms or constraints, i.e. isolated solutions of partial problems are available. Only few integrated approaches that deal with decision making in automated polygon generalization and, thus, combine measures, constraints and algorithms into a comprehensive generalization process are known concerning categorical data, e.g. Schylberg (1993) and Müller and Wang (1992). This chapter discusses possible approaches to orchestration (i.e. process control) in automated (polygon) generalization, namely

- interactive generalization,
- sequential (workflow) generalization,
- generalization in an expert system,
- generalization in a multi-agent system,
- optimization techniques and
- iterative improvement enabled by simulated annealing.

3.1 Interactive generalization

Interactive generalization is characterized by a division of the generalization process into low-level tasks established by the computer and high-level tasks performed or controlled by a human. Typical low-level tasks are the application of a specific algorithm or the evaluation of a specific constraint. Such tasks as the choice of a sequence of generalization operations or the choice of a specific algorithm are high-level tasks and delegated to a human expert. "In other words, the computer implements some tasks (usually execution) which it is good at solving but relies on the user for control and knowledge"(Müller et al. 1995b, p. 8). In doing so, the holistic and subjective nature of generalization is maintained while software substitutes the labor intensive procedures.

With respect to the complexity of map generalization, interactive generalization covers a wide range of possible solutions from tools just replacing the cartographer's pen, to toolboxes of independent algorithms, to an approach termed 'amplified intelligence'[1] proposed by Weibel (1991) - cf. Figure 3.1. The key functionality of an interactive system providing amplified intelligence is: facilities that support the user in decision making (e.g. measures indicating the amount of required generalization, facilities proposing possible algorithms), user-friendly user-interface (e.g. multiple views of the data set showing different generalization states), mechanisms for logging interactions, scripting facilities etc. For a comparison of the most commonly used GIS and cartographic systems for interactive generalization refer to Ruas (2001a).

[1] The user's knowledge is amplified by a range of high-level tools for carrying out generalization operations.

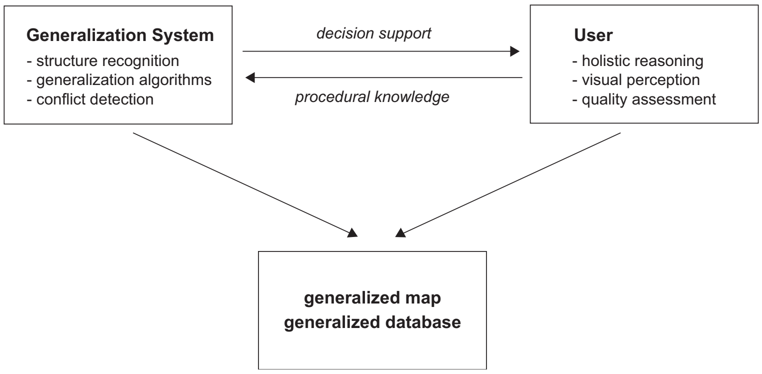


Figure 3.1: The approach of amplified intelligence to interactive generalization. (Weibel 1991)

Besides its holistic character interactive generalization encourages and supports the user in testing several candidate solutions (Müller et al. 1995b) by providing the relevant functionality such as backtracking to any previous state of the data set or logging sequences of applied operations. However, interactive generalization still demands high costs with respect to human resources and offers a low level of automation. Thus, in practical production environments sequences of generalization operations are empirically investigated and tested and are then often combined into one workflow.

3.2 Sequential (workflow) generalization

A sequential (workflow) approach to generalization is defined by a set of manipulations that are applied as 'atomic' operations in a predetermined sequence[2]. "Unfortunately, there is no sequence of operators that is valid for all scale ranges, map purposes and feature classes. [...] For each specific generalization problem, however, the operator/algorithm combination and sequence has to be fine-tuned and calibrated specifically" (Weibel 1996, p. 125,126). In other words, the sequence of operations is context dependent and can not be predefined. Generally, operators with a holistic focus should be triggered prior to operators that affect only few polygons or an individual polygon (Mackaness 1994, Ruas and Plazanet 1996) since it is hoped that operators of the first group reduce also conflicts that refer to a small number of polygons. Mackaness (1994) showed how a varying sequence of a set of operations leads to significantly different results.

Sequences of generalization operators were examined among other things with respect to large scale topographic maps (Lichtner 1979) and line generalization through simplification and smoothing (McMaster 1989). In polygon generalization, Müller and Wang (1992) demonstrated the interaction of measures, constraints and algorithms within a workflow for automated generalization of area patches of same type (e.g. lakes, islands). The proposed workflow pursues the idea of drawing "simple, sketchy pictures which emphasize what should be remembered at the expense of superfluous details" (Müller and Wang 1992, p. 139). It is composed of the following steps[3]:

1. *Preprocess the data*, i.e. rank order patches and determine patches to be expanded or contracted.

2. *Expand or contract* the individual patch with respect to its spatial context.

3. *Eliminate* all patches with an area lower than a threshold derived from the original data.

[2]Triggered as one automated process such a sequence is termed batch generalization. Batch generalization may consist of an individual algorithm or a collection of algorithms, measures, etc. (Müller et al. 1995b).

[3]For a more detailed outline of the workflow refer to Müller and Wang (1992)

4. *Reselect* patches required to maintain characteristic clusters.
5. *Merge* overlapping patches.
6. *Displace* patches too close to each other, the applied displacement is weighted according to the size of the displaced patch.
7. *Verify topological integrity.*
8. *Smooth the patches' outlines* by a combination of low and high pass filters.

The workflow of Müller and Wang (1992) stood the test for the specific task, that is, the generalization of disjoint polygons of the same category. Generally, such a situation is – especially compared to the generalization of an entire data set – considered as rather simple and well defined. However, for a comprehensive solution of cartographic generalization the workflow approach is probably not appropriate due to the missing ability to try out different candidate solutions for one conflict and to adapt to changing situations dynamically. Müller et al. (1995b, p. 10) conclude, "it is clear that the batch and interactive solutions need to be combined with some intelligence if we ever want to attain a performance close to a human expert". So, the next step forward in research was the application of expert systems to map generalization.

3.3 Generalization in an expert system

An expert system is a software that mimics the behavior of human experts in solving problems, that is, in their decision where and when to apply a specific action. The basic components of an expert system are (see also Figure 3.2):

- The *knowledge base* contains formalized knowledge for achieving a solution to a given problem. This knowledge is commonly represented by rules[4] which provide formal descriptions of recommendations, directives, strategies, etc. (Waterman 1986). Rules are expressed by a '**if** condition **then** action' syntax, for example
  1. **if** (object area $\leq$ 1000) **then** (enlarge object)
  2. **if** (distance btw. objects $\leq$ 200) **then** (displace objects)
- The *facts* (facts base, object database) describe the problem to be solved by the expert system, that is, they serve as basis for decision making by the expert system.
- The *inference engine* applies rules from the knowledge base to the facts. In a cycle it matches the facts to the conditions of the rules, selects a rule matching the facts and executes the corresponding action (Torsun 1995). This cycle is repeated until a solution to the problem is reached.

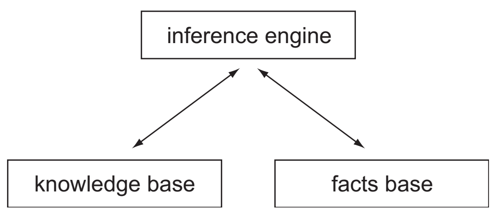


Figure 3.2: The basic components of an expert system.

The success of an expert system is directly linked to the knowledge it contains. Thus, the better the problem is defined (i.e. the more 'textbook knowledge' exists), the better the rules are that

[4]For alternative ways of knowledge representation refer to Waterman (1986) or Torsun (1995).

can be derived and the better the solutions that can be achieved. However, map generalization is highly complex and little or less formal knowledge is available (Fisher and Mackaness 1987, Weibel 1991, Ruas 1995). Armstrong (1991), Ruas and Lagrange (1995) and Weibel et al. (1995) discussed knowledge needed in map generalization, namely

- geometrical knowledge (e.g. size and shape of objects);
- structural knowledge (e.g. hierarchy of categories portrayed by a categorical coverage); and
- procedural knowledge (e.g. for selecting appropriate operators for a given conflict).

Possible sources of cartographic knowledge related to map generalization are cartographic experts, existing map series, textbooks and mapping agencies guidelines (Weibel et al. 1995, Müller et al. 1995b). Weibel et al. (1995) conclude that knowledge acquisition[5] is the major bottleneck in using expert systems in map generalization. In order to overcome this knowledge acquisition bottleneck different approaches were investigated:

- *Interviewing cartographic experts* is accomplished by different methods such as structured interviews, repertory grids, critical incidents or artificial problems (McGraw and Harbison-Briggs 1989, Nickerson 1991, Schylberg 1993, Kilpeläinen 2000).
- *Reverse engineering* attempts to gather knowledge from comparing map objects across scales in map series (Buttenfield et al. 1991, Leitner and Buttenfield 1995, Weibel 1995b, Edwardes and Mackaness 2000).
- *Interactive process tracing* generates rules from interactive generalization carried out by a cartographic expert (Weibel 1991, Weibel et al. 1995, Reichenbacher 1995a,b).
- *Machine learning techniques* aim to derive rules from a set of examples solved by an expert (Plazanet et al. 1998, Mustière et al. 2000). Neural networks as a specific form of machine learning were examined in relation to map generalization by Lagrange et al. (2000), Sester and Brenner (2000) and Werschlein and Weibel (1994).

Despite this effort in knowledge acquisition only few 'experimental' expert systems were developed in generalization research. Generally, their success was restricted to the solution of some well defined partial problems (Herbert and João 1991, Weibel et al. 1995) such as label placement (Cook and Jones 1989, Zoraster 1991, Freeman and Doerschler 1992) or topologically structured linear features (Nickerson 1988). In the generalization of categorical coverages, Schylberg (1993) and Kilpeläinen (2000) – adapting the approach of Schylberg (1993) – derived rules for filling the gap if an object is removed from a categorical coverage. The required knowledge was gathered in both empirical studies with the help of professional cartographers judging prepared examples according to their correctness (Schylberg 1993) or solving test cases (Kilpeläinen 1997, 2000). In doing so, Schylberg (1993) derived rules for the elimination of objects with 2, 3 and 4 neighbors within categorical coverages built up of 5 different classes. Figure 3.3 shows rules represented graphically for filling the area of a removed built-up object in case of two adjacent objects. In addition, Schylberg (1993) proposed some rules related to the aggregation and simplification of objects in categorical coverages.

So far, there is no proof that a rule-based comprehensive generalization system can be built (Müller et al. 1995b). More importantly, it seems that expert systems are not *per se* able to meet the needs of map generalization, namely the dynamic and flexible adaptation to different conflicts. In order to overcome this drawback, generalization research recently started to investigate another technology, namely multi-agent systems.

[5] Knowledge acquisition denotes the process of encoding problem-solving expertise from a knowledge source to a formal structure readable by software (McGraw and Harbison-Briggs 1989). A detailed definition of this term with respect to map generalization is given by McMaster (1995) and Kilpeläinen (2000).

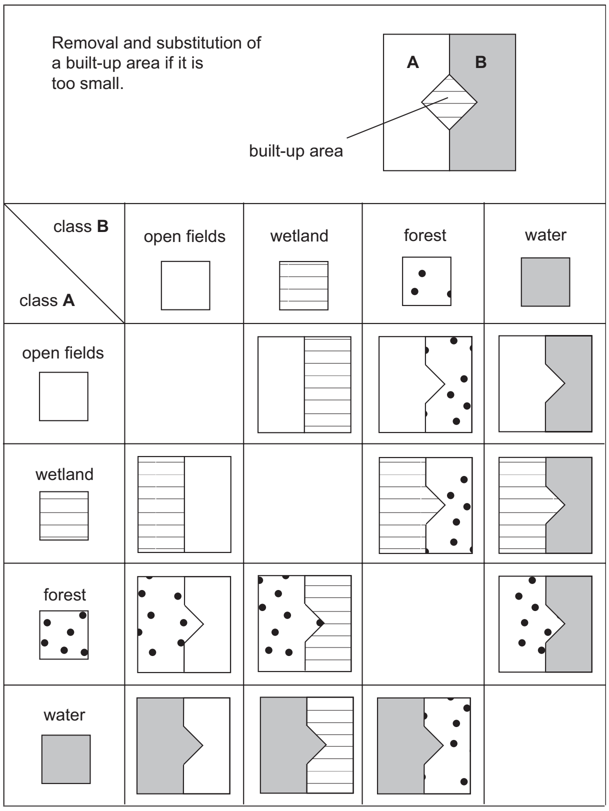


Figure 3.3: Rules for assigning the area of a removed built-up object in case of two adjacent objects. (Schylberg 1993)

3.4 Generalization in a multi-agent system

Definition Agent. “An agent is a computer system that is situated in some environment, and that is capable of autonomous action in order to meet its design objectives” (Wooldridge 1999, p. 29). That is, on the one hand this autonomous agent perceives its environment and on the other hand an agent modifies its environment by its actions – cf. upper part of Figure 3.4. Hence, an agent can dynamically adapt to a changing environment. It is considered as intelligent if it operates flexibly and rationally based on its environment and capabilities (Weiss 1999).

Definition MAS. A multi-agent system (MAS) is a method of distributed artificial intelligence (O’Hare and Jennings 1996, Weiss 1999). A MAS concerns the “cooperative solution of problems by a decentralized group of processes or agents” (Torsun 1995, p. 401). A MAS is defined as a system “in which several interacting and intelligent agents pursue some set of goals or perform some set of tasks” (Weiss 1999, p. 1). Jennings et al. (1998) specify a MAS by the following characteristics:

- Each agent has individual goals, specific capabilities and knowledge. So, agents complement one another in their functionality and ability.
- The system control and the data is decentralized, i.e. problems are decomposed into sub-problems;
- Computation is asynchronous.

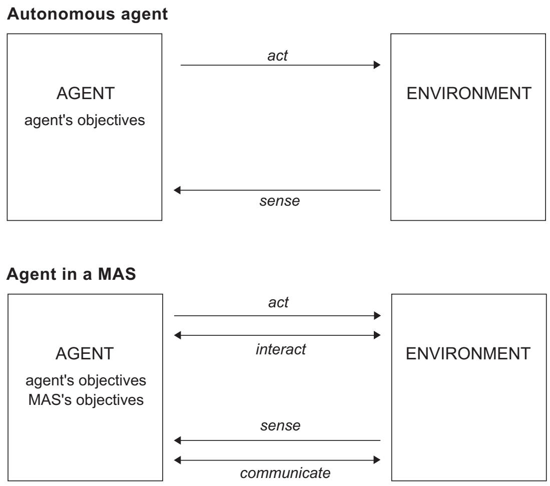


Figure 3.4: The interaction of an autonomous agent and an agent in a MAS, respectively, with their environment. (Wooldridge 1999, Nareyek 2001)

Furthermore, in a MAS an agent is able to communicate and cooperate with other agents (probably in the same environment) to both satisfy its own design objectives related to a subproblem and solve a higher-level problem (objectives of MAS) – cf. lower part of Figure 3.4.

Some examples of areas where MAS technology proved useful are[6]:

- eCommerce where agents trade on behalf of their users, e.g. Chavez and Maes (1996);
- human navigation which is facilitated and modelled by agent-based approaches, e.g. Raubal (2001);
- information retrieval in the Internet such as searching and filtering information, e.g. Bigus and Bigus (2001); and
- modelling in geographic information science, e.g. for growth and densification processes in suburban landscapes, e.g. Loibl and Tötzer (2002).

Concerning map generalization, MAS may offer the needed methodology to orchestrate algorithms, constraints, strategy etc. to a dynamic, flexible and automated generalization system. While Baeijs et al. (1996) studied the convergence of a multi-agent system to external forces resulting from conflicts in map generalization the AGENT project demonstrated strikingly the capabilities of MAS for comprehensive map generalization (Lamy et al. 1999, Barrault et al. 2001, Duchêne and Regnauld 2002).

3.4.1 AGENT project

The AGENT project was funded by the European Commission for the period of December 1997 to November 2000. It investigated the comprehensive solution of map generalization by means of a MAS. This project was a cooperation between a GIS vendor and researchers on map generalization and MAS, namely:

- the COGIT laboratories of IGN France (project leader), the Department of Geography of the University of Zurich and the Department of Geography of the University of Edinburgh with expertise in different areas of cartographic generalization;
- the LEIBNIZ laboratories at the Institut National Polytechnique de Grenoble with expertise in distributed artificial intelligence; and
- the GIS vendor Laser-Scan with expertise in software engineering.

[6]For more examples refer among other things to Weiss (1999)

Basic concepts. *Geographic entities* are designed as agents, that is a building, a road, a group of buildings etc. may become an agent. In consideration of the definition of an agent provided earlier in this chapter every agent (geographic entity) in the AGENT prototype:

- attains a goal, which is formalized by means of constraints;
- owns sensors modelled by measures which evaluate the constraints' satisfaction and characterize both the conflict and the involved objects;
- is able to trigger plans, that improve its attached constraints' satisfaction.

The generalization of an agent is linked to a *set of constraints*, i.e. the need for generalization and the evaluation of a calculated solution is guided by constraints and their satisfaction. As already pointed out in section 2.3, a generalized data set constitutes a compromise between several constraints, i.e. generally a compromise between information preservation and ensuring legibility. Lamy et al. (1999) stated that the modelling of the generalization process by means of a MAS supports the achievement of such a compromise.

As outlined in Ruas (1999, 2000), the data set to be generalized is organized hierarchically in so-called *levels of analysis*. Using such levels enables a better characterization of conflicts, the application of more specific (better suited) algorithms and last but not least more efficient generalization. Thus, every constraint, generalization operation etc. is bound to such a spatial level. The AGENT prototype was built up of two levels (Barrault et al. 2001, Duchêne and Regnauld 2002):

- The *micro level* was dedicated to the individual geographic entity such as a building or a road;
- The *meso level* referred to a group of geographic entities (group of micro objects), e.g. a city block composed of buildings;

The generalization of an agent is performed in the so-called agent's *life cycle*, that is, a sequence of different stages: a characterization of the agent (i.e. an evaluation of its constraints); if generalization is needed (i.e. constraints are not satisfied) plans are collected, selected and triggered to improve the compliance of violated constraints; and again a characterization of the constraints in order to decide whether the agent's satisfaction improved – see also Figure 3.5.

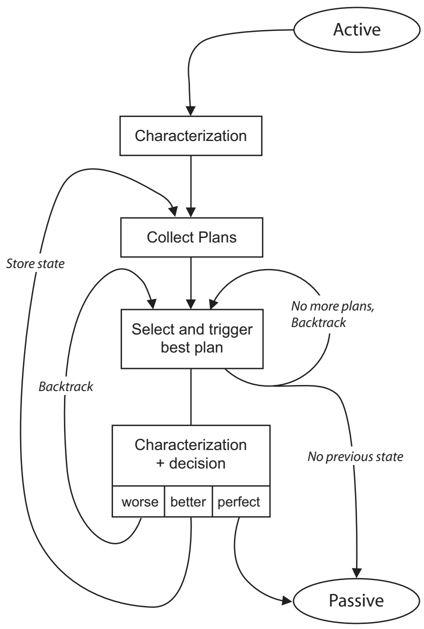


Figure 3.5: Life cycle of an agent. (Barrault et al. 2001)

An agent's life cycle terminates if either a perfect state (all constraints are satisfied entirely) is reached or no plans to improve its states are left[7]. A *modified hill-climbing algorithm* is used to compare different states and to find the best state generated in the iterative generalization process (Regnauld 2001). This state is then used for the agent's update.

Results. A prototype was developed on top of the object-orientated GIS LAMPS2 of Laser-Scan Ltd. It accomplished the automated generalization of urban settlements (roads, city blocks and buildings) and road networks implementing the basic concepts discussed above. Figure 3.6 portrays the results of urban settlement generalization for a scale change from 1:15,000 to 1:50,000. The data stems from IGN BD Topo® with metric resolution. The fully automated process involved the generalization of both the road network and the urban blocks. In an evaluation by professional cartographers from IGN France and Ordnance Survey Great Britain these results were considered as generally good and a valid representation for a scale of 1:50,000 (Duchêne and Regnauld 2002)[8].

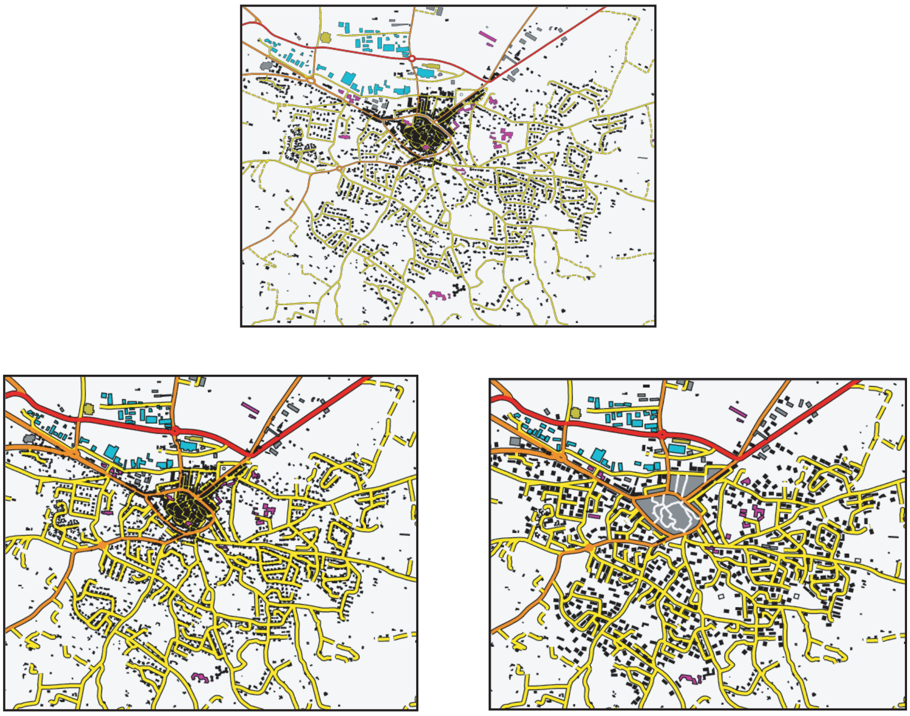


Figure 3.6: Results of automated generalization of urban settlements by means of the AGENT prototype from a scale of 1:15,000 to 1:50,000. Upper figure: original data symbolized for a scale of 1:15,000; Lower left figure: original data symbolized for a scale of 1:50,000; Lower right figure: generalized data symbolized for a scale of 1:50,000; (Data: IGN BD Topo®) (Duchêne and Regnauld 2002, p. 379)

Figure 3.7 represents an example of fully automated road network generalization in the AGENT prototype from a scale of 1:50,000, that is, IGN BD Carto® with 10 meters resolution[9], to a target scale of 1:250,000. Again the results were assessed by a professional cartographer, who emphasized the good readability and the high quality of generalization especially on the level of individual roads (AGENT consortium 2000). Barrault et al. (2001), Duchêne et al. (2001a), Mustière and Duchêne

[7] A more detailed description of an agent's life cycle is provided in Barrault et al. (2001), Regnauld (2001) and Duchêne and Regnauld (2002).

[8] A detailed evaluation of the example shown in figure 3.6 is provided in Duchêne and Regnauld (2002).

[9] For further information about IGN BD Topo® and IGN BD Carto® refer to the web page, http://www.ign.fr/ (accessed 04/24/2002), of the Institut Géographique National France.

(2001) and Duchêne and Regnauld (2002) presented and discussed further results achieved in the AGENT prototype.

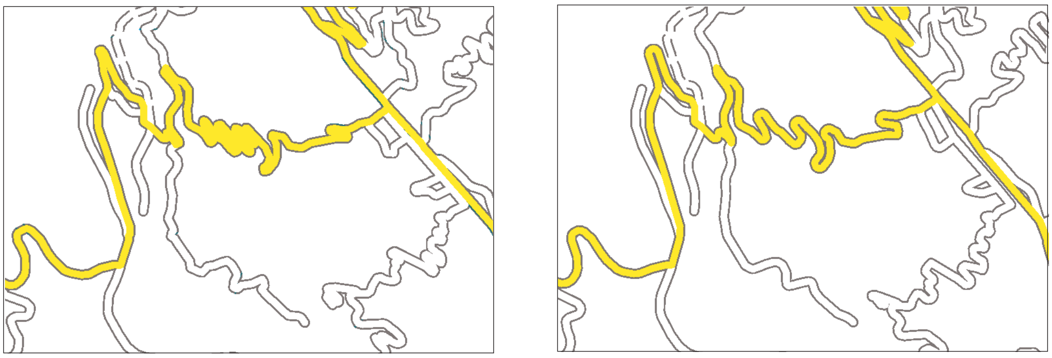


Figure 3.7: Results of fully automated generalization of road networks in the AGENT prototype. Left figure: original data symbolized for a scale of 1:250,000; Right figure: generalized data symbolized for a scale of 1:250,000; (Data: IGN BD Carto®)

The prototype developed during the AGENT project is part of Laser-Scan's commercial GIS LAMPS2. IGN France (Ruas 2001b, Duchêne and Regnauld 2002, Lemarié 2003) and the Danish national mapping agency KMS (Bengtson 2001) reported already on successful application of agent-based generalization in map production. From a research point of view the AGENT prototype was considered as an important step towards a comprehensive, automated solution of map generalization (Duchêne and Regnauld 2002). MAS technology showed to provide an adequate framework for modelling the holistic and highly complex process of map generalization (Regnauld 2001).

Beyond the AGENT project. Recently, Laser-Scan announced the release of a new product for automated map generalization named 'Clarity' which is based entirely on the AGENT prototype (Laser-Scan 2002b). Besides the commercialization of the project's output research on agent-based generalization is carried on by the COGIT laboratories of IGN France and the Department of Geography of the University of Zurich. Research deals generally with aspects of MAS in map generalization and in particular with further improvements of the existing AGENT prototype:

- the introduction of a *macro level* of analysis – as proposed in Ruas (1999) – that shows responsible for a population of objects. It should control their generalization as well as allow statistical analysis and evaluation of the results.
- the improvement of the AGENT engine, i.e. for instance accomplishing the communication (interaction) between agents or introducing *negotiation mechanisms*[10] for agents (Duchêne et al. 2001b, Duchêne and Regnauld 2002).
- the integration of learning techniques in order to improve dynamically the procedural knowledge of generalization and to identify the need for additional constraints (Mustière et al. 2000, Ruas 2001b).
- the application of a MAS/the AGENT prototype to the generalization of *other data types* besides urban settlements and road networks such as river networks or polygonal data. In doing so, the AGENT prototype requires – as outlined in chapter 4 – enhancement according to new agent types, additional constraints, measures and algorithms.

[10]Negotiation mechanisms allow the resolution of conflicts between actions of individual agents in such a way that a joint plan can be developed and executed (Torsun 1995).

3.5 Optimization techniques

Optimization techniques (OT) are well known methods for the solution of over-constrained problems in physics and engineering. Over-defined problems do not exhibit an unique solution. Thus, the solution is considered being the one that best compromises amongst all the observations – such as the regression line of a scatter-plot graph. Map generalization can also be modelled as an optimization problem, where different constraints have to be satisfied simultaneously as faithfully as possible (Sester 2000). To do so, generalization problems and constraints need to be formalized analytically as functional dependencies (Sarjakoski and Kilpeläinen 1999). Then, all the displacements and deformations required in a data set – to meet the imposed generalization constraints – can be applied simultaneously, that is, map objects interact to find an optimal result (the best compromise that satisfies competing constraints). In other words, OT support a holistic approach to generalization, that is, side-effects of generalization can be handled together with the spatial transformations. Thus, the use of OT establishes a significant advantage over sequential methods (cf. section 2.3).

The following optimization methods have been introduced to map generalization, especially related to linear objects:

- spring models (Bobrich 1996),
- least squares adjustment (Sarjakoski and Kilpeläinen 1999, Sester 2000, Harrie and Sarjakoski 2002), and
- energy minimizing methods (i.e. snakes and elastic beams) (Burghardt and Meier 1997, Burghardt 2000, Bader 2001).

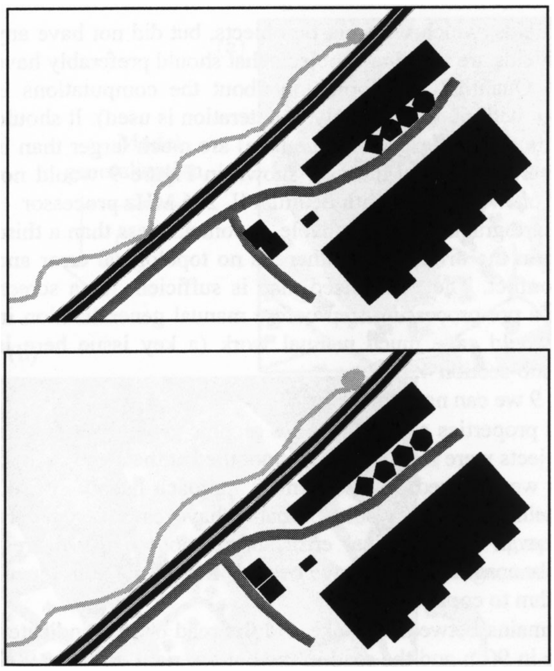


Figure 3.8: Simultaneous graphic generalization: original situation (upper figure) and result of generalization (lower figure). (Harrie and Sarjakoski 2002, p. 253)

The potential of OT for the combination of different generalization operations were studied by Harrie and Sarjakoski (2002) and Burghardt (2002). Harrie and Sarjakoski (2002) proposed a concept for so-called simultaneous graphic generalization based on least-squares adjustment. Their approach combined both the application of generalization operations (i.e. simplification, smoothing, enlargement and displacement) and the maintenance of structural constraints (i.e. curvature, crossings etc.) into a comprehensive generalization process (see Figure 3.8). In doing so, they proved the potential of OT to achieve a compromise between competing generalization operators and constraints. Burghardt (2002) presented an algorithm that establishes the simultaneous smoothing and displacement of linear objects based on, the energy minimizing technique, snakes.

The introduction of OT to map generalization led to a significant advance in map generalization research since OT enable a holistic solution of generalization conflicts. However, optimization techniques can not achieve semantic generalization and comprehensive spatial generalization. It is limited to operations that translate geometries or parts of them (i.e. points) (Sarjakoski and Kilpeläinen 1999) – for instance the generalization operations displacement and enlargement. Hence, essential operations for comprehensive generalization, such as elimination, aggregation or typification etc., can not be implemented by such techniques. A different approach to automated map generalization links OT to other techniques, namely trial position and iterative improvement techniques.

3.6 Iterative improvement enabled by simulated annealing

Ware et al. (2001, 2003) proposed a completely different approach to automated execution of generalization operations in map generalization, that is, the combination of a trial position approach and iterative improvement techniques (Russell and Norvig 2002). The trial position approach indicates for each of the n discrete map objects, k candidate solutions that represent different generalized states of a map object (Ware and Jones 1998). Thus, k^n map configurations are possible and it is assumed that some of these solutions provide a satisfying compromise between the different constraints attached to the map objects. In order to evaluate the quality of a map configuration and compare different solutions overall costs are calculated for every solution. They subsume the remaining conflicts in the solution as well as the generalization effort required to achieve the solution (Ware et al. 2001). Of course, the best solution is the one that minimizes overall costs. Since an exhaustive calculation and evaluation of map configurations is not possible Ware et al. (2001, 2003) showed that simulated annealing[11] allows to find efficiently a near optimum solution. Ware et al. (2001, 2003) demonstrated the potential of this approach for the generalization of buildings belonging to the same city block – see Figure 3.9.

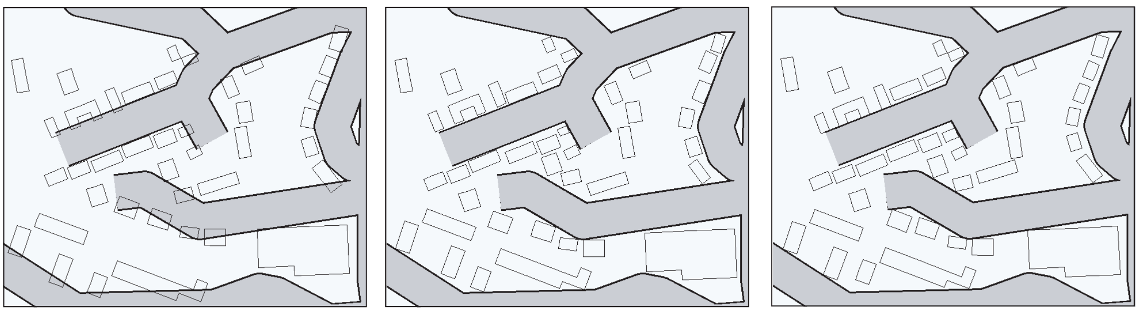


Figure 3.9: The generalization of buildings of a city block: the original situation (left figure) is solved by applying the iterative improvement algorithm to individual buildings (middle figure) and to higher-order objects such as rows or alignments of buildings (right figure). (Ware et al. 2003, p. 5)

In doing so, trial positions considered the operations displacement, elimination, enlargement (i.e. a global increase) and reduction (i.e. a global decrease) of corresponding map objects. In other

[11] Simulated annealing approximates the solution of very large combinatorial optimization problems (Krikpatrick et al. 1983). In other words, it directs the search through the possible solutions.

words, the approach meets the requirements for the generalization of well defined partitions with a limited number of objects (i.e. buildings of a city block). Doubtlessly, the discussed approach is a valuable "part of an automated map generalization solution" (Ware et al. 2003, p. 1) but it is not suited as an approach to model a comprehensive generalization process. The reasons are attributable to, among other things; the computational costs with respect to the number of considered generalization operations and the missing capability to coordinate the generalization of high-order objects (e.g. an alignment or a cluster) with individual objects.

3.7 Conclusions

Interactive generalization, the synthesis of human capability of decision making and computers' capability of running generalization operations, is still the approach most often used in practice. But, a trend to consolidated application of automated solutions is noticeable especially in NMA map production. Automated generalization processes are amongst others applied to production of topographic maps at the French (Ruas 2001b, Duchêne and Regnauld 2002) and Danish (Bengtson 2001) NMA. In fact, workflow approaches – although their drawbacks are well known – and agent-based approaches, i.e. the commercialized AGENT prototype, are operated. Sometimes, they include algorithms that are based on OT or iterative improvement techniques (Lemarié 2003). Automated generalization usually is completed by an interactive post-processing performed by a cartographer who accomplishes a final evaluation and in case interactive (re-)generalization.

In comparing the requirements of map generalization in general and of categorical generalization in particular with the potential of the presented approaches the MAS technology seems to be the most suitable approach to mimic cartographers in decision making (i.e. holistic reasoning and compromising between competing generalization constraints and objects) and to achieve automated generalization, that is, the orchestration of generalization tools such as generalization operators, generalization constraints and measures. Additionally encouraged by the promising results of the AGENT project (Lamy et al. 1999, Barrault et al. 2001) an agent-based framework for automated polygon generalization is presented in the next chapter.

Chapter 4

An agent-based framework for automated polygon generalization[1]

The state of polygon generalization is characterized by the fact that several methods and concepts exist but a framework for their combination into a comprehensive polygon generalization process is missing. Other research, related to the generalization of topographic maps, applied MAS technology successfully and demonstrated the potential of this technology for dynamic decision making (i.e. process control) in map generalization. Hence, MAS technology seems to be today the most powerful and flexible approach to orchestration in map generalization (cf. chapter 3).

This chapter outlines a framework for the automated generalization of polygonal subdivisions based on a multi agent system. In doing so, this chapter extends previous work carried out by Ruas (1999) and the AGENT consortium (Lamy et al. 1999, Barrault et al. 2001). The prototype and results of topographic map generalization established during the AGENT project were already discussed in section 3.4.1.

Besides an introduction to the fundamentals of agent-based generalization (section 4.1), this chapter intends to examine spatial levels of polygonal subdivisions as the basis of different agent types for automated polygon generalization (section 4.2) and to demonstrate theoretically how an agent-based process of polygon generalization operates (section 4.3). A worked example (section 4.4) aims to clarify the basics of agent-based polygon generalization. Both the theoretical discussion and the worked example shall finally help – in consideration of the state of the art of polygon generalization presented in chapter 2 – to identify needs of polygon generalization that should be tackled prior to the implementation of the proposed framework (section 4.5).

4.1 MAS in cartographic generalization

Borrowing from a definition given by Luck (1997) an agent for cartographic generalization denotes a geographic entity (e.g. a building, a river, a road network, a group of polygons etc.) capable of controlling its own decision making and generalization guided by a set of generalization constraints (cf. section 3.4). Thus, an agent (Barrault et al. 2001, Duchêne et al. 2001a, Ruas in press)

- is linked to a set of constraints;
- possesses a method to determine its so-called happiness (satisfaction), that is, it evaluates and summarizes the satisfaction of all its constraints;
- proposes plans (generalization operations specified by generalization algorithms and their parameters) according to the violated constraints in order to improve its happiness;
- triggers plans autonomously starting with the one supposed to be best;
- aims to reach a perfect state (= perfect happiness; all its constraints are satisfied);

[1] This section is an extended and revised version of Galanda and Weibel (2002a).

- is able to compare and store different states;
- can return (backtrack) to any previous state;
- may instantiate, trigger and coordinate other agents[2];

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 (Torsun 1995, Luck 1997). As mentioned before, cartographic generalization is both a holistic and subjective task and an over-constrained problem, 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 generalization process by a MAS means that "a sub-optimal but acceptable solution can often be reached" (Lamy et al. 1999). Compared to other approaches used in map generalization like workflows or expert systems a MAS supports a holistic approach and allows a dynamic adaptation to a changing environment. Due to these properties the MAS technology seems to provide an adequate framework for automated map generalization (Regnauld 2001).

In the context of the AGENT project a consortium of experts in multi agent systems, experts in automated map generalization and GIS specialists (cf. section 3.4.1) have succeeded in setting up a MAS for generalization tasks of topographic mapping (Lamy et al. 1999, Barrault et al. 2001). The AGENT package (data schema and the generic agent engine) is implemented in the commercial GIS LAMPS2 and has already proven useful in map production (Bengtson 2001, Duchêne and Regnauld 2002, Lemarié 2003). This chapter extends methods and concepts – developed with respect to the generalization of road networks and urban settlements during the AGENT project – for use with polygonal subdivisions.

4.2 Spatial levels of polygon generalization

The more a cartographic conflict is narrowed in on a spatial context, and the more precisely the situation is characterized, 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 organized in so-called spatial levels (levels of analysis) of polygon generalization. In other words, the concept of levels of analysis (cf. section 2.3) developed for the generalization of topographic maps is adapted for polygon generalization. So each constraint, generalization operation etc. is delegated to a specific spatial level , 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 generalization, 4 different spatial levels, namely

- map,
- group,
- polygon and
- line,

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

Map Agent. Every polygon mosaic holds only one map agent (Figure 4.1). It is responsible for constraints and generalization operations concerning the whole polygon map (e.g. reclassification) as well as the instantiation of the group agents at run time.

Group Agents. Group agents handle contextual generalization, i.e. conflicts between polygon objects. They are composed of several objects sharing a common geometric, topological or semantic relation. Thus, reasonable group agents may be attached to a cluster, an alignment, characteristic

[2] 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.*

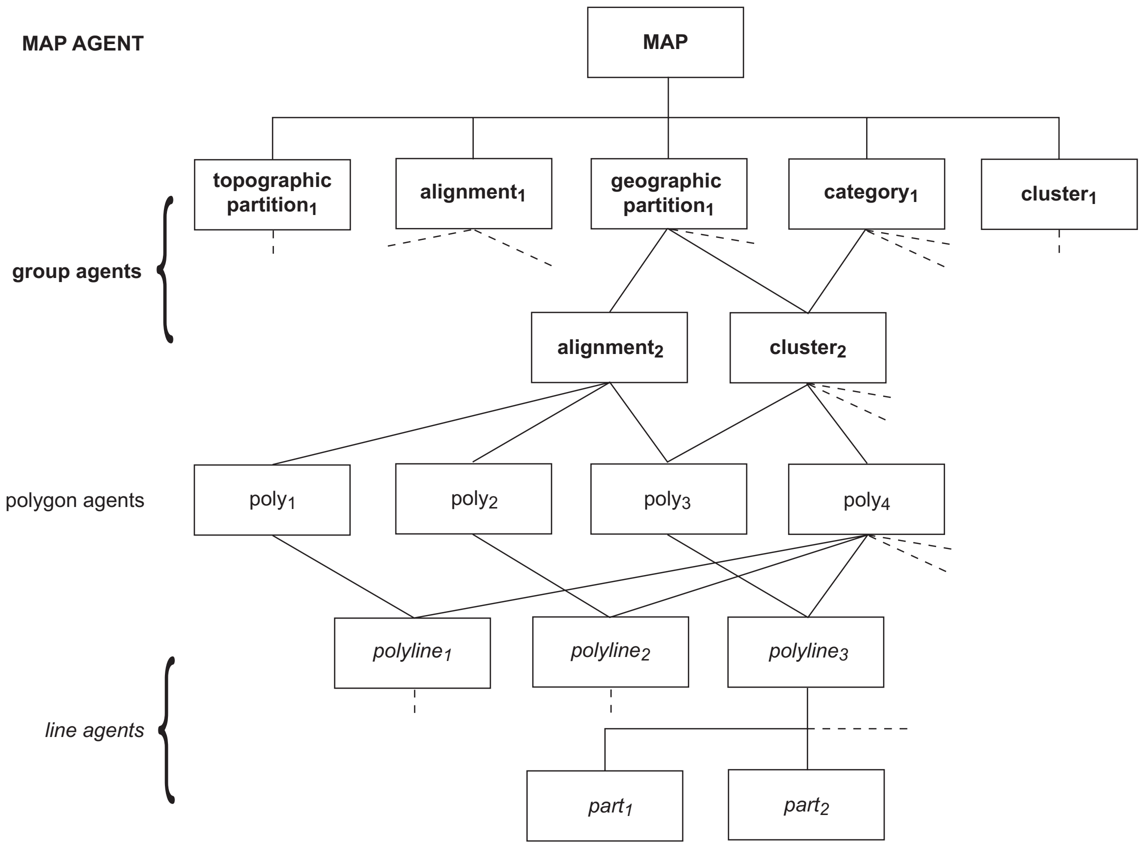


Figure 4.1: The organization of agents at the four different spatial levels of polygon generalization. See also Figure 4.2 below for some graphic examples.

spatial patterns, a category, a topological partition (e.g. first and second order neighbors) or a geographic partition (e.g. a collection of polygons bounded by rivers or roads). A non exhaustive list of group agents includes the following types of group agents:

- A *cluster agent* is, on the one hand, linked to the spatial proximity of polygon agents, that is, it subsumes all polygon agents that are within a certain distance of each other (Figure 4.2c). Otherwise, cluster agents may consist of polygon agents that exhibit a certain level of semantic similarity. Of course, cluster agents may be built upon a combination of spatial and semantic distance.
- A *category agent* subsumes all the polygons belonging to the same category, that is, it relies exclusively on semantics.
- *Topological partition agents* combine polygon agents through topological relations. Examples include a polygon agent and its first order or its second order neighbors. Such groups are, for instance, relevant to the update of a polygonal subdivision with respect to polygon agents generalized independently.
- The identification of *geographic partition agents* relates to additional geographic data of the area portrayed by a polygonal subdivision. In other words, geographic features, such as river or road networks, may be used to partition a polygonal subdivision and make up group agents. Figure 4.2b presents an example of such a geographic partition agent.
- *Pattern agents* refer to polygon agents that exhibit a specific spatial and/or semantic configuration. They usually represent characteristics of a certain geographic area or mapped

theme that should be preserved during generalization. A specialization of a pattern agent is an alignment agent that deals with polygon agents arranged along a linear structure – see Figure 4.2c.

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 partition*$_1$ and *cluster*$_2$ in Figure 4.1. Furthermore, group agents may be built upon arbitrary combinations of the listed types of group agents. For example, a group agent may be made up of all polygons that belong to both a certain topological partition agent and a certain category agent.

Polygon Agents. A polygon agent coordinates the generalization 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 (cf. section 2.3) 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. In other words, side-effects of independent generalization have to be settled.

Line Agents. Line agents are delegated to polylines bordering a polygon object and their generalization (e.g. simplification). Exactly two polygon agents can supervise – assuming a clean topology – a line agent of the first generation[3]. Similar to group agents they are enabled to recursively subdivide themselves in order to perform generalization on homogenous line parts (Duchêne et al. 2001a), see *polyline*$_3$ in Figure 4.1 and 4.2e&f.

The AGENT prototype handles generalization in accordance with the generic levels of analysis (cf. section 2.5) through macro, meso and micro agents (Ruas 1999, Barrault et al. 2001). The proposed agents for polygon generalization relate to these agent types as follows. The map agent refers to the macro agent. But contrary to the macro agent the map agent is enabled to apply generalization operations (e.g. reclassification) to its entire population, i.e. all polygons of a polygonal subdivision. The proposed group agent is equivalent to a meso agent. The polygon agent does not have a direct equivalent in the AGENT prototype since it shares, on the one hand, the commonality of independent generalization with the micro agent and, on the other hand, it supervises child agents like a meso agent. The line agents behave like micro agents except if they recursively subdivide themselves. A micro agent would become a meso agent as it has to control the generalization of child agents and its parts subsequently micro agents. Here, a line agent is also enabled to control other line agents' generalization. Of course, the agent types established in the AGENT project meet more generic requirements while the agents used in this project are exclusively adapted to the spatial levels of polygonal subdivisions.

If necessary parent agents (e.g. group agents) build their child agents (e.g. polygon 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 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, *poly*$_3$ in Figure 4.1 and Figure 4.2c is part of the group agents *alingment*$_2$ and *cluster*$_2$. Thus, the agents' generalization must be done in a sequential process.

[3] A child agent of the first generation has always 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 has always parents of the same agent type – e.g. a group agent of the second generation has always an other group agent as a parent.

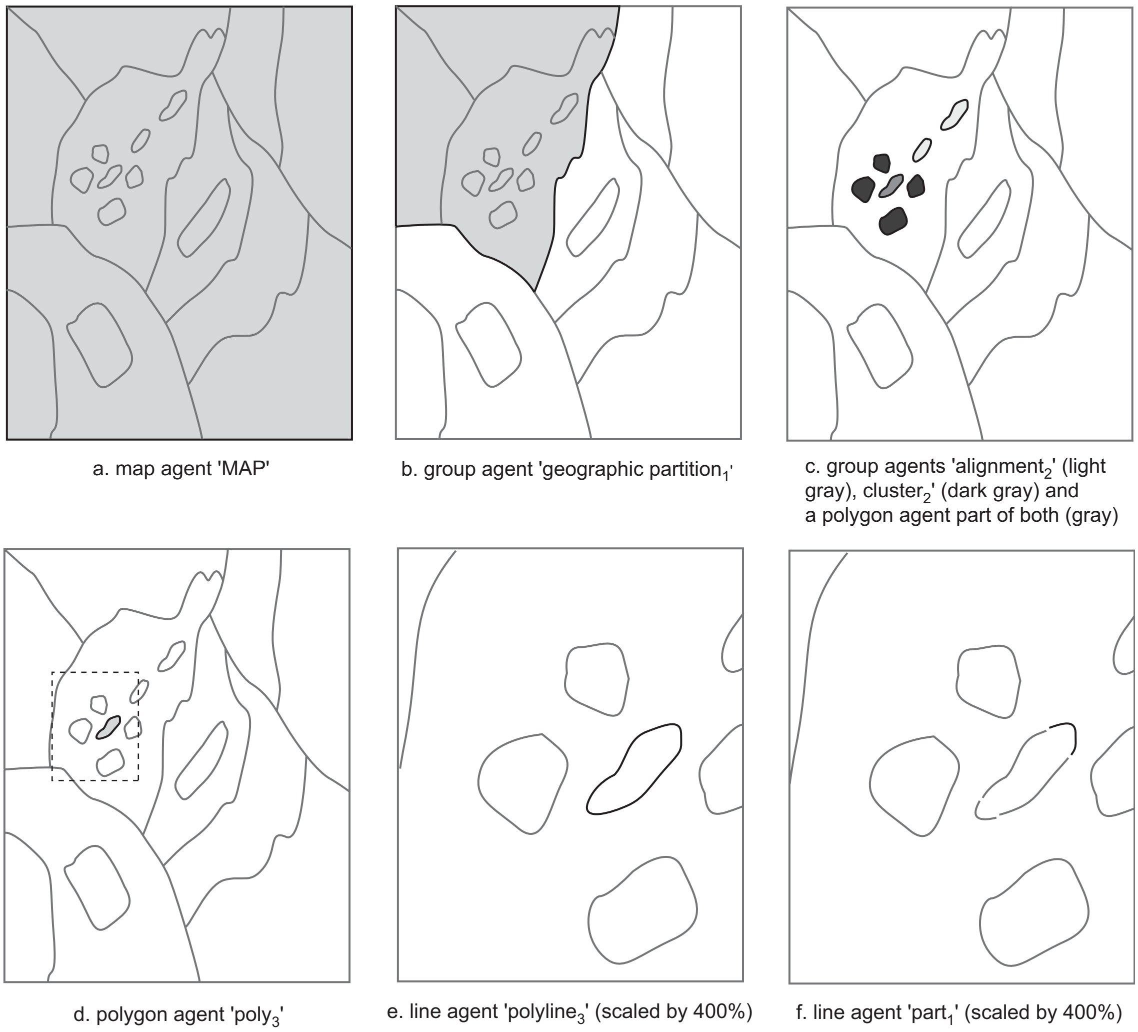


Figure 4.2: Some examples of different spatial levels of polygon generalization in a polygon mosaic according to the organization of agents shown in Figure 4.1.

4.3 Agent Life Cycle in Polygon Generalisation

The generalization 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 organized into three main stages. A pre- processing stage prepares the data base for the second stage, the iterative agent- based generalization. Finally, the user undertakes an evaluation of automated generalization's result during the third stage.

4.3.1 Pre-processing: Data Specification and Analysis.

First of all constraints are specified according to the basic conditions of the generalization task and supplementary information (auxiliary data) is calculated on the database objects.

4.3.2 Generalisation: Life Cycles of Agents.

At the very beginning of the generalization 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 generalization 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 initialize the child agent's generalization and then the parent's constraints. In practice, a mixture of these approaches may be applied. Again, procedural knowledge can help choosing the child agent to be triggered first amongst all the child agents.

The AGENT package uses a constraint-based approach to generalization (Beard 1991, Weibel 1996, Ruas 1999), that is, constraints initialize and control the entire generalization process. 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 press). In a MAS for map generalization constraints are linked to agents that aim to satisfy the offensive constraints without violating one of the defensive constraints[4]. To every constraint a method based on a measure is linked to evaluate its satisfaction (= to determine the severity of violation). A list of possible plans is attached, too, that propose generalization algorithms (geometric and semantic transformations) and their required parameters for improving the agent's happiness. Algorithms are pre-defined while the algorithms' parameters are usually derived at run time from the constraint's severity of violation. Figure 4.3 lists a sample constraint with its corresponding agent type, measure and list of plans.

constraint	The distance between two polygons should not be less than a minimum distance (e.g. minimum visual separability distance).
agent	group
measure	shortest distance between polygons)
plans	a. displacement algorithm 1 b. displacement algorithm 2 c. exaggeration algorithm d. aggregation algorithm e. typification algorithm

Figure 4.3: Example of an offensive metric constraint at the group agent level in polygon generalization, the attached measure and list of possible plans.

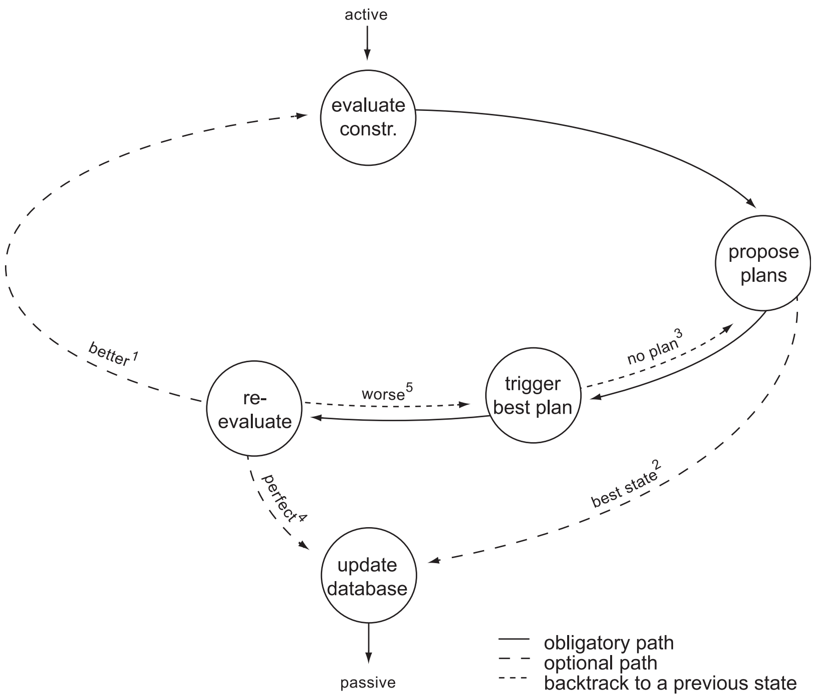
The final ordered list of plans results from the weighted sum of all plans suggested by any violated constraint of the agent (Barrault et al. 2001, Regnauld in press). The current best plan is then triggered. The best plan allows to speed up the iterative generalization process since it is hoped that a perfect solution (state), that is, a complete satisfaction of all constraints, is reached earlier using this heuristic (Regnauld 2001). Note that, although a perfect state implies that all constraints are met it needs not to be the best solution *a priori*. However, a perfect state is considered to be a satisfying (i.e. good enough) compromise between the constraints attached to an agent and an appropriate generalization of an agent, respectively. Whenever a perfect state is reached the iterative generalization of an agent terminates. In doing so, the search for the best possible solution is skipped for the sake of a faster and more efficient generalization process.

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.][5], proposing plans [propose plans], triggering the best plan [trigger best plan] and re-evaluation [re-evaluate] is the same generic behavior of all agents and called an agent's life cycle. A modified 'hill-climbing' algorithm is applied within this life cycle – see Figure 4.4 – to find a satisfying compromise to all

[4]An offensive constraint is an indicator for the need of generalization, e.g. a minimum size constraint, while a defensive constraint controls the preservation of a certain property of an agent such as its shape.

[5]Terms in brackets relate to life cycle steps displayed in Figure 4.4.

constraints of an agent (Regnauld 2001, in press). The life cycle ends when either all constraints are satisfied completely (perfect state) or no plan to be tried is left. In any case the database is updated by the best state ever reached in the life cycle [update database]. In other words, the plan that leads to the perfect or best state is triggered in any case sooner or later, but 'best plan' acts as a heuristic to find the best state more efficiently. The agent is set to passive again, that is, the control of the generalization process is returned to the parent agent.



[1]better	current state improves the agent's happiness; new list of plans to enhance further the current state;
[2]best state	the best compromise (state) of constraint satisfaction is passed to [update database];
[3]no plan	no alternative plan is left to improve the current state; return to a previous state (list of plans);
[4]perfect	all constraints are completely satisfied; current state is passed to [update database];
[5]worse	current state degrades agent's happiness; another plan of the current list of plans is tried out;

Figure 4.4: An agent's life cycle in polygon generalization with the incorporated 'hill-climbing' algorithm.

Due to the hierarchical organization of agents the constraint of parent agents that assesses the satisfaction of its child agents, is the most essential one in the generalization process. A violation of that constraint requires one of the following plans:

- the instantiation of the child agent's life cycles; the autonomous generalization of the child agents, that is, the control of the generalization process is passed to the individual child agents in turn; each of them tries to improve independently its happiness.
- 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 agent[6].

[6]In general a reactive order can be given to an agent by any other agent.

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. If the last group[7] supervised by the map agent has completed its generalization the control of the generalization process returns again to the map agent. The generalization stage ends when the map agent terminates its life cycle.

4.3.3 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 2001b). If necessary some interactive (re)generalization is performed.

4.4 A worked example

In this section, we present a worked example for the agent-based framework for polygon generalization. This example is intended to both clarify and illustrate the ideas and concepts discussed theoretically above. It demonstrates some excerpts of the agent-based generalization of one cluster agent (Figure 4.5). It consists of five polygons belonging to two categories (light and dark gray) and is embedded in a polygon environment – compare state0 in Figure 4.5. The different states of the cluster agent and the generalization operations they result from (shown in Figure 4.6) as well as the evolution of the constraint satisfaction (diagrams displayed in Figure 4.5) summarize how the 'hill-climbing' algorithm finds the best compromise between several competing constraints.

In the example the following three constraints[8] related to cluster agents are considered:

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

The following paragraphs describe the individual states of the cluster agent in the worked example:

state0 The state0 in Figure 4.5 shows the agent's state when activated. The life cycle (Figure 4.4) starts with an evaluation of the constraints in order to determine if there is a need for generalization or not. Those constraints directly related to a property of the example agent (constraint 'Object separation' and 'Relative configuration') 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 center of the cluster) falls below the threshold defined in a minimum area constraint. The polygon agent's generalization performed in a separate life cycle (s0 to s5 in Figure 4.6) results in the elimination of that polygon.

[7] This group agent may control itself the generalization of child agents, i.e. group agents or polygon agents.

[8] 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 Figure 4.5.)

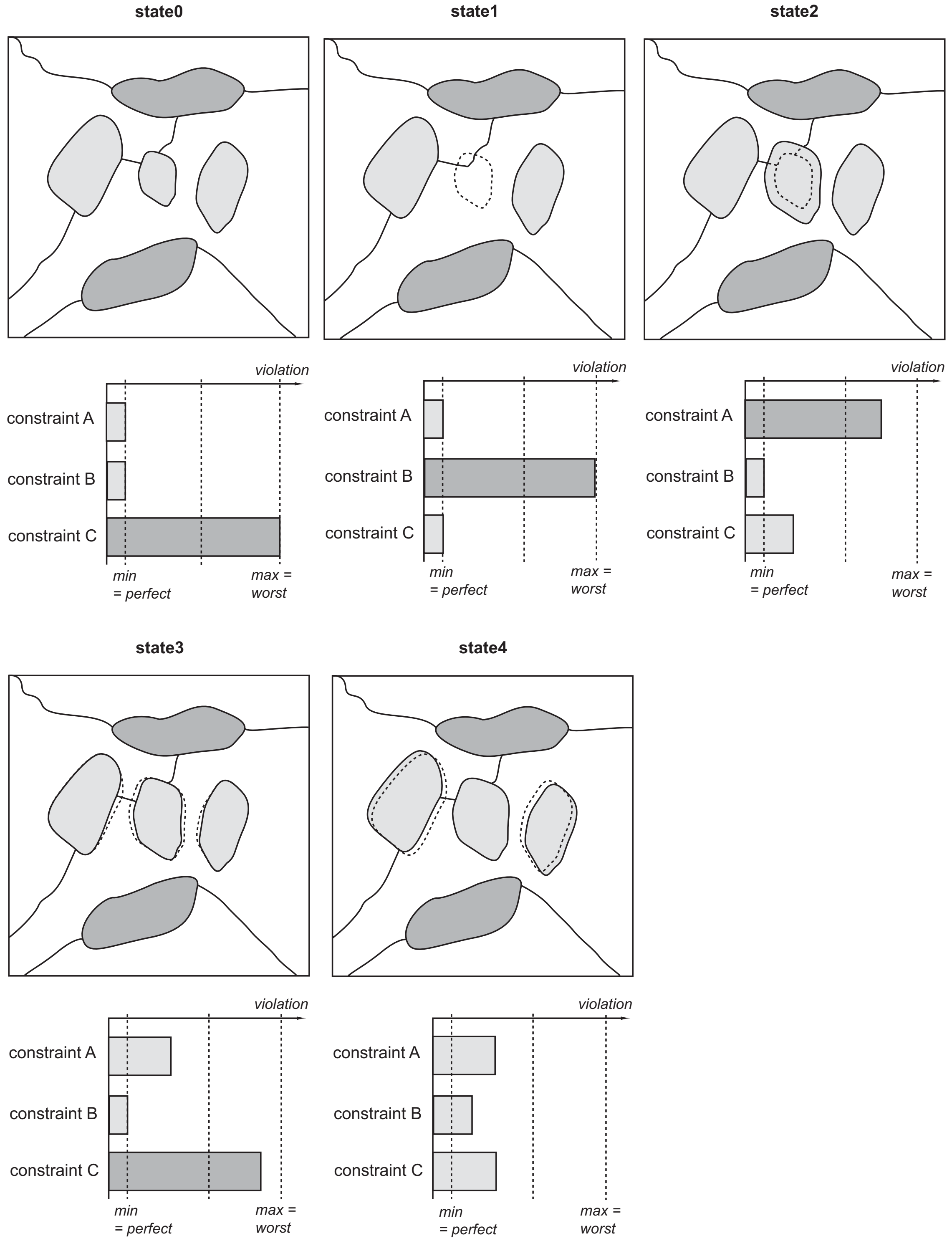


Figure 4.5: The different states of the example agent and the corresponding constraint satisfaction.

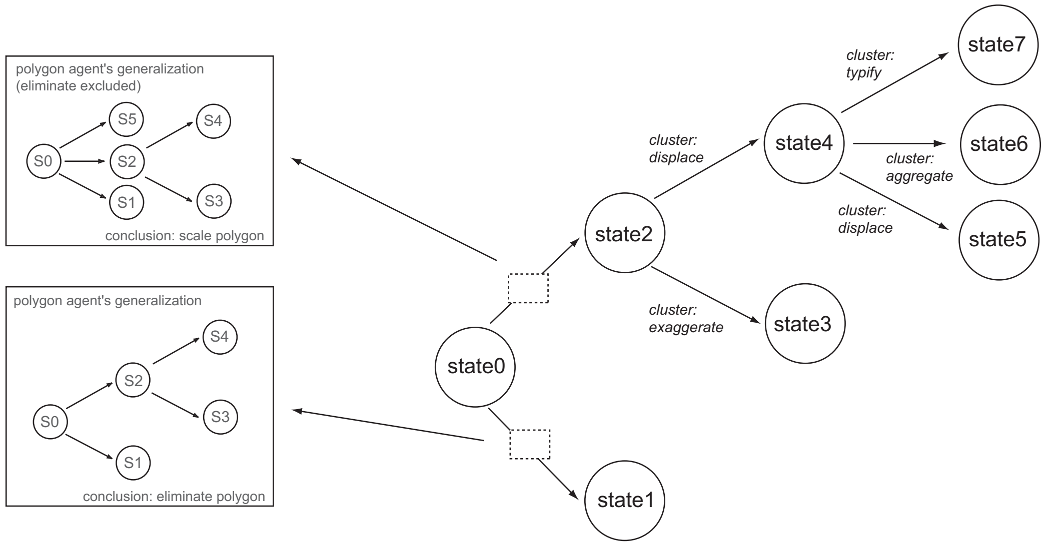


Figure 4.6: The different states of the cluster agent (state0 to state7) in the generalization process and the generalization operations they result from. The states (s0 to s5) of the triggered polygon agent are indicated in the two small insets to the left.

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 'Relative configuration' 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. But a restriction is 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 constraint 'Object separation'. But from an agent point of view an improvement of the overall constraint satisfaction (see Figure 4.5) 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 Figure 3).

state3 The application of an exaggeration operation (state3) decreases the cluster agent's happiness because the 'Child entity's constraints'- constraint is violated even further as a consequence 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 to be 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 passive again.

The worked example demonstrated the basic concepts of agent-based decision making in polygon generalization by stepping through and discussing each step in a group agent's generalization process.

4.5 Conclusions

This chapter described a framework for automated polygon generalization based on a MAS, that is, it outlined a feasible way of how a process for comprehensive polygon generalization may be modelled and implemented, respectively. Such an agent-based approach is not new, experiences gained of the AGENT project are available and the AGENT engine is implemented and customizable in the commercial GIS LAMPS2. Thus, the implementation of the outlined framework seems to be straightforward. However, besides the MAS technology, this framework, and consequently the success of its application to polygon generalization, relies to a good deal on appropriate generalization algorithms for conflict resolution and generalization constraints for polygonal subdivisions for conflict detection and evaluation of generalization solutions. Thus, research prior to the framework's implementation (cf. chapter 7) focuses on these two aspects of polygon generalization.

With respect to algorithms for polygon generalization, optimization techniques are studied (cf. chapter 5) as a basis for the implementation of new generalization algorithms that allow side-effects of generalization to be reduced through a holistic approach as well as the adaptation of a polygonal subdivision to be supported according to an independently generalized agent (e.g. the enlargement of a polygon agent). As highlighted in the worked example, constraints are the core of decision making in the outlined framework. Thus, chapter 6 deals exclusively with constraints and their modelling for automated polygon generalization. It includes the proposal of a preliminary set of constraints for polygon generalization, methods for the evaluation of the constraints' satisfaction, the determination of reasonable plans for every constraint etc.

Chapter 5

Energy minimization techniques in polygon generalization[1]

5.1 Introduction

This chapter concentrates on an algorithm for resolving the conflicts resulting from the violation of metric constraints which exist if a polygonal object is too small (violation of the minimal size constraint), too narrow (minimal width constraint) or too close to another polygon (minimum separability constraint) (Weibel 1996, Peter and Weibel 1999a). A violation of the minimal size constraint is denoted as a size conflict, while a violation of the later constraints characterizes a proximity conflict. Existing approaches usually apply specific algorithms to every type of size or proximity conflict in turn, such as an algorithm for widening narrow parts of a polygon based on a Delaunay triangulation (Bader and Weibel 1997) or a boundary-moving algorithm to enlarge a polygon (Jones et al. 1995) – see also section 2.5. The solution of metric conflicts in a group of polygons through such independent algorithms may lead to tedious loops of different generalization operations due to the side-effects of map generalization (cf. section 2.3).

In order to overcome the drawback of sequenced independent algorithms, Harrie (1999), Sarjakoski and Kilpeläinen (1999), Harrie and Sarjakoski (2002), and Burghardt (2000), among others, made use of optimization techniques that allow the treatment of different generalization conflicts in a comprehensive sense (cf. section 3.5). That is, different generalization operations are performed concurrently and provide a common solution. Their research focused primarily on the generalization of linear objects and the displacement of disjoint (i.e., detached) polygons such as buildings. Following a similar approach, this chapter proposes a single algorithm based on the optimization technique called snakes (Kass et al. 1987), which enables the holistic and simultaneous solution of numerous size and proximity conflicts in a group of polygons. The presented research extends previous work carried out in our group (GIS division of the Department of Geography, University of Zurich), which investigated the application of snakes in road network generalization (Bader 2001). The novelty of the discussed approach lies in the application of the snakes method in polygon generalization.

5.2 The snakes method

Snakes, also called Active Contour Models, were developed in the field of computer vision where they were used for the extraction of contours in raster images (Kass et al. 1987). The term "snakes" refers to the way in which these energy-minimizing splines iteratively approximate and approach the contours in a raster image in a snake-like movement, when visualized. Burghardt

[1] This chapter is adapted from Galanda and Weibel (2003).

and Meier (1997), Burghardt (2000), Bader and Barrault (2000) and Bader (2001) have transferred the method to cartographic generalization, adapting it primarily to the displacement of linear objects. Burghardt (2000) also applied snakes to polygons, but his approach was restricted to the displacement of disjoint (isolated) polygons.

Snakes involve an iterative process whose purpose is to improve and optimize the solution over time, with an energy-minimizing spline controlled by both internal constraint forces (the so-called inner or internal energy E_{int}) and external forces (external energy E_{ext}). Equation (1) shows the total energy of a deformation, while Equation (2) describes the internal energy. In these equations l denotes the line length, $\boldsymbol{d}'$ and $\boldsymbol{d}''$ denote derivatives of $\boldsymbol{d}$ (displacement between the original line and the displaced line) with respect to s (segment length) and α and β are weights associated with these derivates. The weights α and β are also denoted as the shape parameters. For details on the derivation and solution of the equations, refer to Bader (2001).

$$E(\boldsymbol{d}) = \int_l (E_{int} + E_{ext})\, ds \tag{1}$$

$$E_{int} = \alpha \left|\boldsymbol{d}'(s)\right|^2 + \beta \left|\boldsymbol{d}''(s)\right|^2 \tag{2}$$

Internal energy stems from the difference in terms of position and shape between the initial object and the object after displacement or deformation. As the objective is to keep the internal energy minimal, it effectively controls the resistance of the initial object to deformation. The shape parameters α and β allow control of this resistance. Empirical experiments have shown that the results of snakes are only sensitive to a concurrent increase or decrease of both α and β (Bader and Barrault 2000). Their influence on the displacements calculated by snakes is illustrated in Figure 5.1.

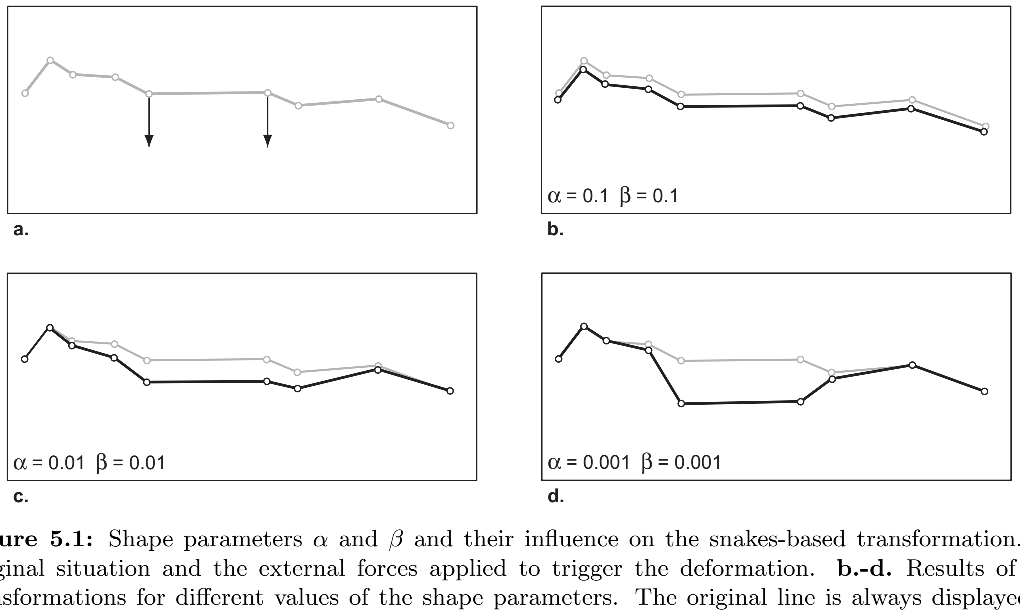


Figure 5.1: Shape parameters α and β and their influence on the snakes-based transformation. **a.** Original situation and the external forces applied to trigger the deformation. **b.-d.** Results of the transformations for different values of the shape parameters. The original line is always displayed in gray.

Figure 5.1a shows the original line and two assumed forces which indicate external energy and are applied to two of the line's vertices in order to trigger snakes-based transformation. The significant differences in these transformation results shown in Figures 5.1b-d relate only to the magnitude of the shape parameters α and β. Note, for instance, the effect of a factor 10 on the shape parameters when comparing the updated line objects (drawn in black) in Figures 5.1b and 5.1c (the original geometry is drawn in gray). We can observe that the higher the shape parameter values are the more likely it is that the entire line is translated (Figure 5.1b) rather than transformed locally (Figure 5.1d). In other words, for lower values of α and β the deformation increases in the vertices close to the external forces, while it becomes minimal in more distant vertices of the line, compared to the situation with larger values of α and β.

External energy is imposed on an object through forces that represent a need of transformation. With respect to map generalization this need is defined through the occurrence of generalization conflicts, such as size or proximity conflicts. For instance, forces may be derived from the ratio of the current distance between objects to the minimal distance that is required to visually separate two objects. The aim of the snakes method is to minimize the sum of internal and external energy. The iterative optimization process is described in matrix terms by Equation (3).

$$(\mathbb{1} + \gamma \boldsymbol{K})\boldsymbol{d}^{(t)} = \boldsymbol{d}^{(t-1)} + \gamma \boldsymbol{f}^{(t-1)} \qquad (3)$$

The displacements $\boldsymbol{d}$ at time step t result from the forces $\boldsymbol{f}$ and displacements $\boldsymbol{d}$ at time $t-1$. $\boldsymbol{K}$ denotes the so-called stiffness matrix, which only depends on the length of the line segments. The optimization procedure employs the calculus of variations (Jost and Li-Jost 1998) and is controlled by the iteration term γ, which determines the inertia of the iterative snakes process. A low γ results in a weaker displacement and deformation in each calculation step. Compare, for instance, the results of varying γ values in the snakes-based transformations represented in the Figures 5.2b-d, which show the transformations' output in black and the original line in gray.

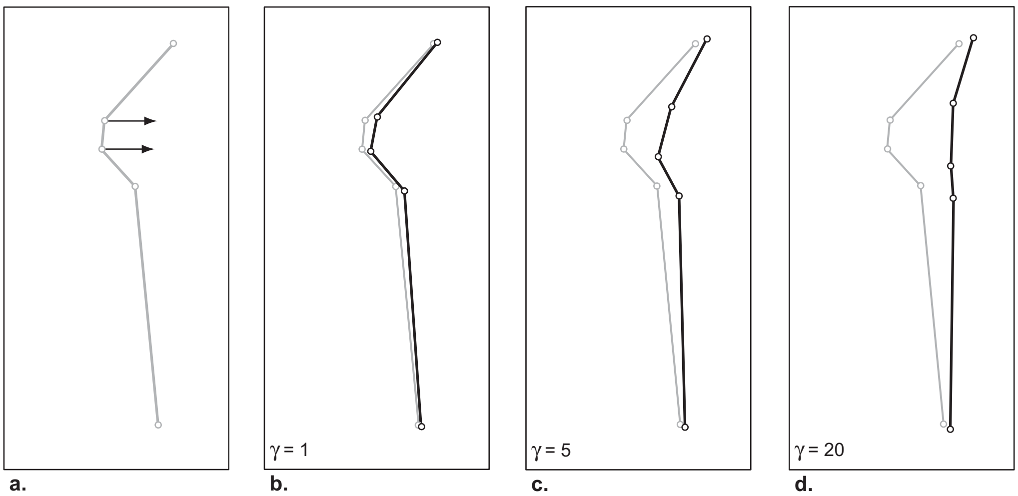


Figure 5.2: The influence of the iteration parameter γ on the results of snakes-based transformations, after the same number of iterations (10) for each experiment. **a.** Original line object and the external forces applied to trigger the transformation. **b.-d.** Results of the transformations achieved by applying increasing magnitudes of γ. The original situation is always displayed in gray.

5.3 Applying snakes in polygon generalization

Previous research (Burghardt and Meier 1997, Burghardt 2000, Bader and Barrault 2000, Bader 2001) illustrated the potential of snakes in line generalization, i.e. the displacement and deformation of line objects. This potential stems from the possibility to model the preservation of the shape (characteristics) of cartographic objects through the internal energy as well as the need for generalization through external energy and forces. The snakes method, which minimizes the sum of the internal and external energy, can thus be used to find a compromise between the preservation of shape and the magnitude of a generalization applied.

Polygon generalization makes use of snakes for those generalization operators which can be implemented as boundary-moving operations. That is, they are applied in a polygonal subdivision by translating some or all vertices of those polylines bounding at least one of the polygons involved. Such generalization operators include:

- The displacement operator describes the translation of an entire polygon in order to solve a proximity conflict between polygons (Figure 5.3a). The shape of a displaced polygon remains unchanged as all its vertices are shifted by the same vector.
- The enlargement of a polygon (Figure 5.3b) involves the constant increase of a polygon in all directions. In polygon generalization this operator usually supports the solution of size conflicts.
- The exaggeration operator defines a local deformation (i.e., an increase or decrease) of a polygon. Because only some vertices of the polygon are shifted (Figure 5.3c), the shape characteristics of the polygon (e.g., the ratio of perimeter to area) are modified. The exaggeration of a polygon enables the solution of proximity conflicts.

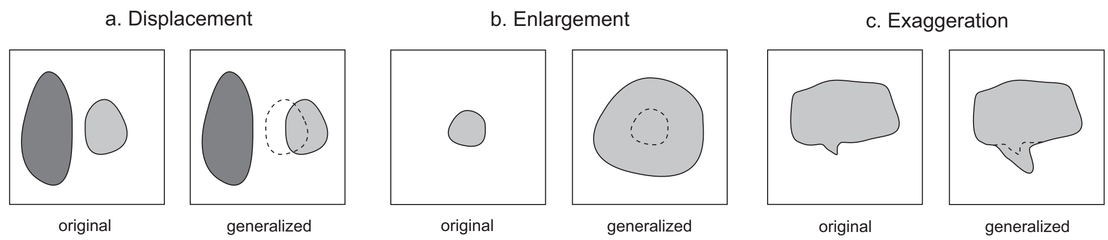


Figure 5.3: Generalization operators in polygon generalization with a potential to be modelled by snakes.

These generalization operators usually involve only a restricted number of polygons (termed the involved polygons here), rather than the whole polygonal subdivision.

5.3.1 Behavior of individual polygons

To accomplish different generalization operators by a single snakes algorithm, polygons have to react in different ways to displacements and deformations calculated by the snakes method (see Equation (3)). In practice, the magnitude of applied is controlled by the so-called weight property at the polygon level, which is defined for every involved polygon and stored as an attribute on the polygons. Values can range between 0.0 and 1.0. 0.0 indicates no change of the polygon's position, that is, the polygon acts as an obstacle for all the other polygons, and 1.0 indicates no resistance to deformation. Intermediate values specify a continuous transition between these two extreme behaviors. In other words, the magnitude of the calculated transformation for the polygon is scaled by its weight attribute. For example, a computed displacement of 100 meters (m) for a polygon with a weight of 0 leads to a displacement of 0 m, while a weight of 0.25 corresponds to a displacement of 25 m.

5.3.2 Computation of forces

External energy and forces derived from size and proximity conflicts act as the primary trigger for snakes. The implementation of different generalization operators by a single snakes-based algorithm requires four different possibilities of force computation, that is, different force models. Depending on the selected generalization operator, a force model describes how forces acting on one polygon are derived from the set of involved polygons to resolve a particular conflict. Forces are considered in the iterative optimization process through the force matrix $\boldsymbol{f}$ (see Equation (3)). In general, forces are computed by virtue of the shortest distance between two polygons. The distance is the shortest vector connecting a point of the first polygon outline with a point on the second polygon outline. In the following, the polygon to be displaced is termed the displaced polygon and the polygon (object) that pushes it away is called the push polygon.

Force model 0 'reactive'. A reactive force model 0 of a polygon object implies that forces are not computed explicitly for this polygon, yet forces determined for a neighboring polygon are allowed to deform it. The polygon shows no resistance to deformation. Example: A polygon whose neighboring polygon is enlarged.

Force model 1 'vertex_line'. According to the force model 1 'vertex_line' the force for one vertex of the displaced polygon is given by the vector representing the shortest path to the push polygon and a minimal distance threshold – see, for instance, the force vector calculated for the vertex vd_1 of the displaced object in Figure 5.4a. If the shortest distance is greater than the required minimal distance the force becomes 0, for example, at the vertex vd_2 of the displaced object in Figure 5.4a. Example: The enlargement of a polygon.

Force model 2 'line_vertex'. The force model 2 'line_vertex' complements model 1 in that it considers the vector representing the shortest path between a vertex on the push polygon and the outline of the displaced polygon – compare the distance between vertex vp_2 and the displaced object in Figure 5.4b. The distance of each vertex of the push polygon is calculated to the nearest vertex of the displaced polygon. If this distance is greater than the shortest distance to the displaced polygon multiplied by 1.5[2], a new force is determined based on the vertex on the push polygon and the nearest position on the displaced polygon – see, for instance, the dashed force vector plotted in the middle of the displaced object in Figure 5.4b. If the nearest position is within the first/last 30 % of the corresponding line segment the force is mapped to the start/end vertex of the segment. If the position lies in between, half the force is projected on both the start and the end vertex (see vertex vd_1 and vd_2 in Figure 5.4b). This force model is always used in combination with other force models.

Force model 3 'combined'. This force model is derived from the output of models 1 and 2. If a vertex of the displaced polygon obtains a force either by force model 1 or 2 the resulting force acting on this vertex is equivalent to the force calculated by the corresponding force model (see, for instance, the vertex vd_2 in Figure 5.4c). If forces calculated by both force models 1 and 2 act on one vertex of the displaced object, the final force results from the mean of these forces. Compare, for instance, the force vectors acting on the vertex vd_1 of the displaced object in Figure 5.4c and the forces calculated for this vertex by the force models 1 (Figure 5.4a) and 2 (Figure 5.4b). Example: The exaggeration of a polygon.

Force model 4 'vertex_line_max'. The force model 4 'vertex_line_max' applies the longest force vector computed by model 1 and is exclusively used for the displacement operator (Figure 5.4d). Example: The displacement of a polygon.

[2] All the thresholds used in the context of this force model are determined through empirical observation and testing.

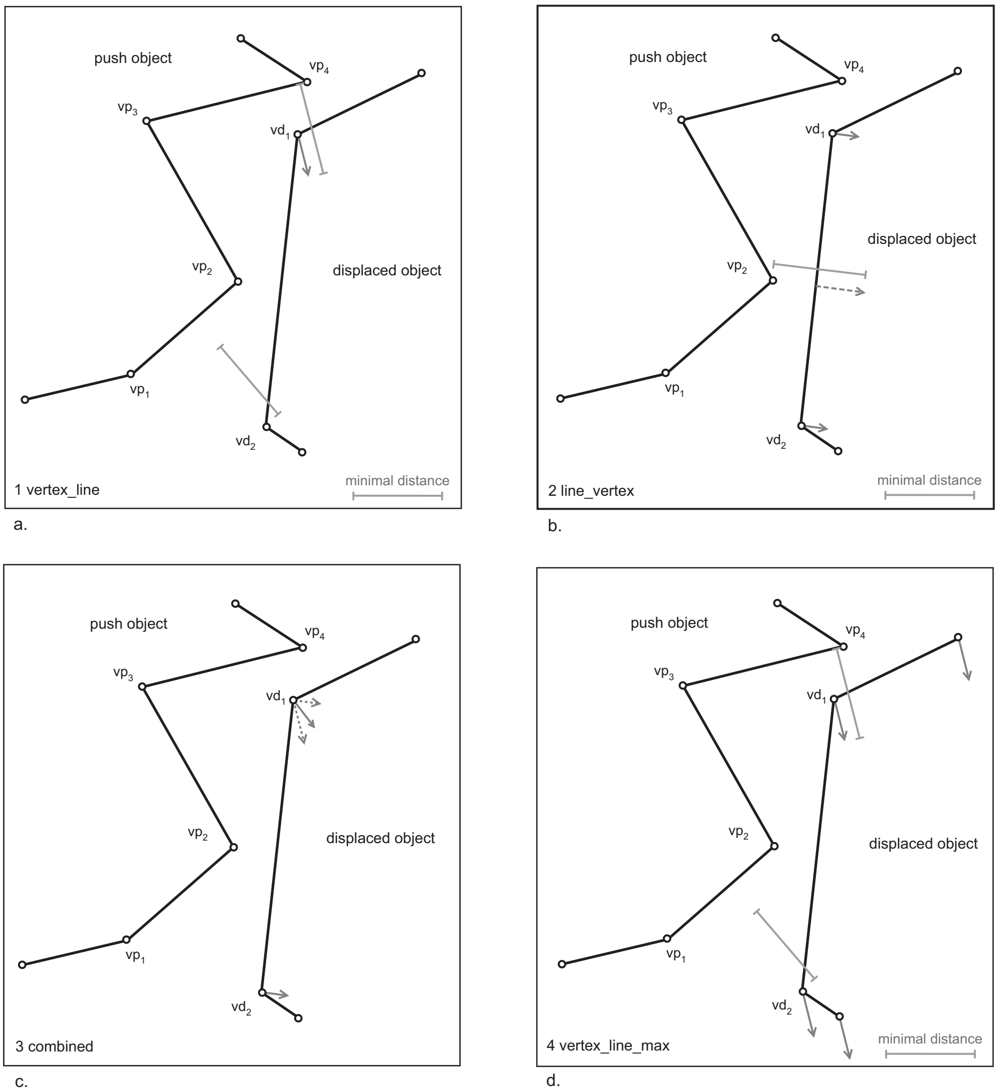


Figure 5.4: The basics of the computation of forces. Force vector are shown in gray. Further explanations are given in the text.

5.3.3 Assigning weights and force models to polygons

Both the weight property and the force model acting on a polygon are stored as attributes of polygons. For the transformation by snakes these attributes are then mapped to the polylines of the involved polygons. If both polygons sharing a polyline are considered in the snakes process, the polyline inherits the higher weight attribute and the force model that exhibits a higher 'priority' value. The priority ranking of force models is reflected in the numbering of the force models and has been derived from empirical observation. Thus, the 'vertex_line_max' model takes precedence over all other force models and the 'combined' model again has higher priority than the 'vertex_line' model.

For a small number of polygons the assignment of the weight and force model attributes to the polygons can be done manually, that is, the cartographer selects the conflicting polygons and specifies these attributes by hand with respect to her knowledge and experience. For the integration of this algorithm into a fully automated generalization process, such as the outlined framework (cf. chapter 4), the weight and force model attributes need to be set automatically in consideration of the following aspects:

- the type of conflict to be resolved (e.g., a size conflict is usually resolved by an enlargement operation);
- the spatial and semantic properties of a polygon (e.g., a polygon representing a feature with rather distinct boundaries such as a lake may be displaced rather than exaggerated); and
- the spatial context of the involved polygons (e.g., the number of polygons possibly affected by a specific transformation).

5.3.4 Generalization operators enabled by snakes

Varying simultaneously the weights of the involved polygons and the force models used enables snakes to resolve size and proximity conflicts in polygonal subdivisions. Figure 5.5 gives some examples (using artificial data) of applying snakes in polygon generalization, where each instance displays an assumed conflict which is resolved by a specific generalization operator (and hence a different setup of weights and force models).

Displacement. The two uppermost graphics of Figure 5.5 deal with the solution of proximity conflicts through the displacement operator. In the upper left figure, which shows the case of disjoint polygons, the displacement of polygon P_2 is specified by the force-model 'vertex_line_max' (4) and a weight of 1.0 assigned to P_2. In this example, P_1 acts only as an obstacle and remains unchanged due to zero weight. The figure at the upper right shows the solution of an assumed proximity conflict between polygons P_1 and P_3 by translating P_3 to the right. The weight attribute of P_3 is 1.0 and the force model used is 'vertex_line_max' (4). The settings on polygon P_2, namely weight = 1.0 and force model = 0, enable its polyline to adapt to the transformation of P_3 in the same snakes process, which displaces P_3 to the right. Here, P_1 serves only as the push polygon. Note that a different setting of the weight and force model attributes of the polygons would lead to the displacement of P_1 or of both P_1 and P_3 (i.e. both polygons would then act in turn as a push and displaced object).

Enlargement. The snakes-based enlargement of a polygon requires the automated creation of a virtual push object, which supports the computation of forces acting from the polygon's interior on its boundary and, thus, the implementation of an enlargement operation by means of a boundary-moving algorithm based on snakes. This virtual push object can be accomplished automatically by inward buffering. The concept of varying weights and forces allows not only the application of a single operator, but also to combine different operators into one snakes-based transformation – cf. the left graphic in the middle row of Figure 5.5. The same snakes-based process which enlarges polygon P_1 exaggerates simultaneously P_2 – in order to avoid an overlap of P_1 and P_2 – despite the enlargement of P_1. The exaggeration of P_2 is defined by its weight value of 1.0 and the force model 'combined' (3) applied on it. Conversely, the force model 'vertex_line_max' (4) applied on P_2 would lead to a displacement of P_2. The solution of a size conflict embedded in a polygonal subdivision is portrayed in the right graphic in the middle row of Figure 5.5. The enlargement is achieved through a weight of 1.0, and forces are calculated for the virtual push object by the vertex_line force model (2).

Exaggeration. The bottom-row of graphics of Figure 5.5 illustrate proximity conflicts that are tackled by a local deformation of polygons (i.e., exaggeration). The conflict at the left is characterized by two disjoint polygons (i.e. P_1 and P_2) that are too close to each other with respect to the required minimal distance between polygons. Alternatively to the displacement of the entire polygon P_2, the conflict is solved here by an exaggeration of P_2. As above, P_1 remains

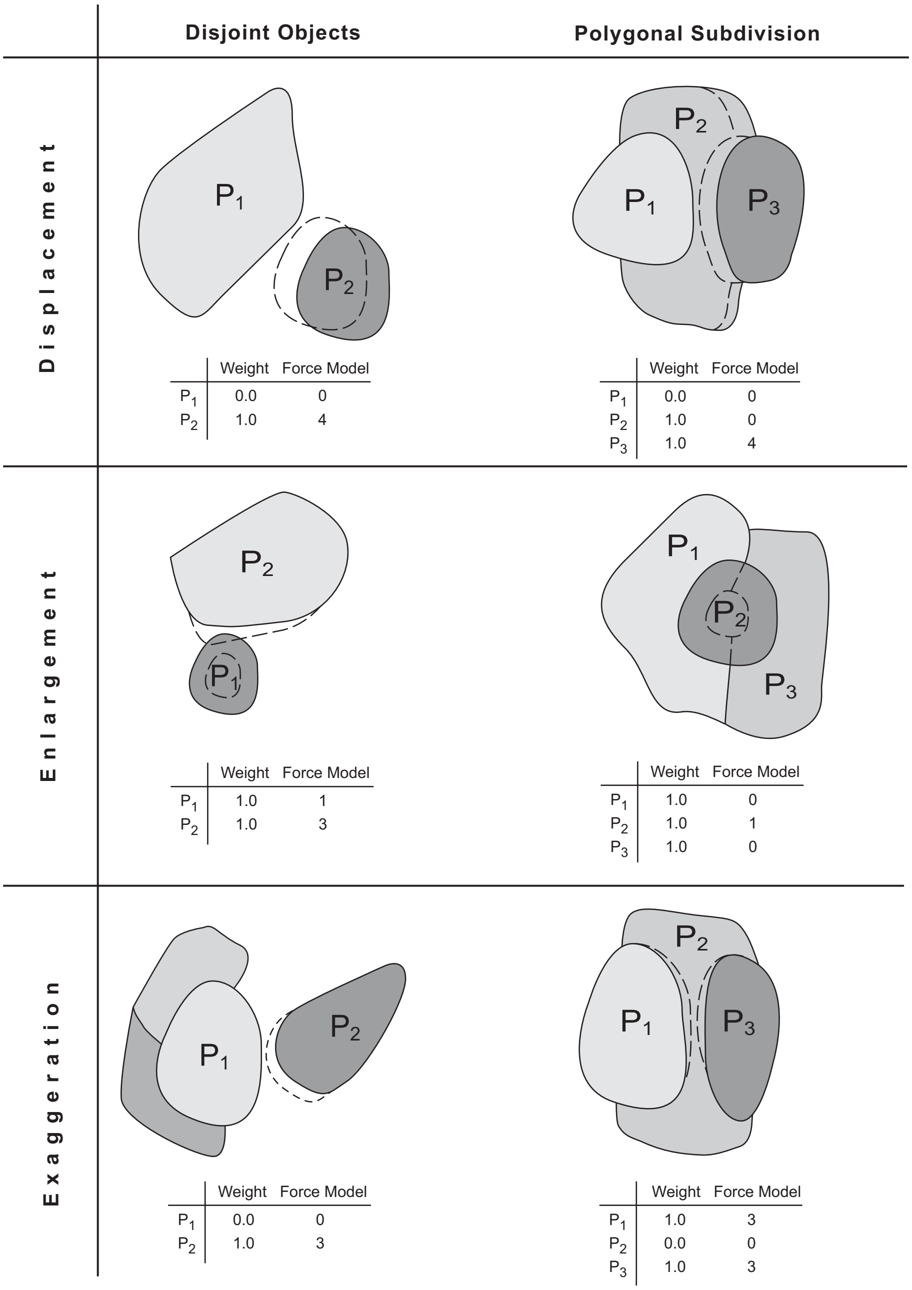


Figure 5.5: Displacement, enlargement and exaggeration accomplished by the snakes method. The solid areas describe the situation after the transformation, while the dashed lines show the original outlines of the polygons.

unchanged due to its weight of 0.0. The exaggeration operator differs from the displacement operator only in the use of a different force model of P_2, namely the combined model (3) instead of the vertex_line_max model (4) as used in the above example for the displacement operator. The right-hand image of the bottom-row of Figure 5.5 illustrates the widening of a narrow passage between the polygons P_1 and P_3 by a local deformation of the two polygons, such that both obtain the same weight of 1.0 and the combined force model (3). Due to the space-exhaustive nature of a polygonal subdivision the same result could be achieved by setting the weight and force model of P_1 and P_3 to 0.0, while assigning to P_2 a weight of 1.0 and the combined force model (3).

5.4 Experiments

While the two previous sections have explained the principles of the snakes method and its application in polygon generalization this section discusses some experiments on the application of snakes in polygon generalization. Following the approach outlined by Bader (2001), the snakes method has been implemented in a development environment, the commercial GIS LAMPS2 by Laser-Scan Ltd. The tests reported here as well as the evaluation of results concerned solely the ability of snakes to resolve size and proximity conflicts in polygon generalization. Other aspects of polygon generalization, such as generalization at the semantic level or processes following a global strategy (e.g., in order to integrate with other generalization operators) were neglected at this point. The integration and orchestration of different operators is the subject of chapter 7.

The aims of the experiments were to

- prove the concept that the snakes-based algorithm can achieve different generalization operators in polygon generalization;
- demonstrate the possible benefits for polygon generalization using a snakes-based algorithm; and
- identify potential drawbacks of the application of snakes in polygon generalization.

The experiments were carried out using the digitized vegetation map of the Swiss National Park (originally compiled at 1:25,000 by H. Zoller in 1992).

5.4.1 Displacement of disjoint polygons

As a basis for the displacement of a polygon, force model 4 was used, i.e., equal forces were applied to all vertices of the displaced polygon (cf. Figure 5.4). Forces sometimes did not result in an equal displacement of all the vertices due to the different lengths of the polyline segments. Consequently, a distortion of the polygons shape might occur. In order to avoid this effect, the maximum translation calculated for one of the polygon vertices was used for all vertices of the polygon. Figure 5.6 illustrates the application of a snakes-based displacement on a group of 4 disjoint polygons. Due to a reduction in map scale by a factor of 3 the minimal allowed distance between polygons, which supports proper legibility of the polygons at the smaller target scale, is increased to 150 m (measured in real world coordinates). The original group of polygons is illustrated in Figure 5.6b at the new map scale and in Figure 5.6a at the original map scale. Because the polygons in Figure 5.6a are buffered with half the minimal separation distance (i.e. 75 m) all overlaps (white fill color) of buffers indicate areas where two polygons are closer than the minimal separation distance, i.e. proximity conflicts occur. The dark gray area P_3 is fixed in its position (weight = 0.0) due to reasons of semantics. For instance, it seems to be reasonable to displace a 'forest polygon' rather than a 'built-up area' polygon if a proximity conflict occurs between polygons of theses categories. The light gray polygons P_1, P_2 and P_4 (weigh t= 1.0) have been rearranged by the algorithm in such a way that the minimal separation distance is ensured for all polygons (Figure 5.6c, d).

The distances between the polygons of the original situation and the situation resulting from the snakes-based displacement represented in Figure 5.6 are shown in Figure 5.7 – values below the goal distance of 150 m are printed in bold typeface. One weakness of snakes as a computational method is the need of computer resources to meet a specified minimal distance completely. Because the snakes process is an energy-minimization technique that compromises between internal and external energy, it is likely that the specified minimal distance is not reached completely.

In other words, the closer a specific distance converges on the target minimal distance the smaller the translations become per iteration of the snakes process. For instance, while the distance between P_2 and P_4 increased from 98.7 m to 147.4 m during the first 50 iterations, the next 50 steps

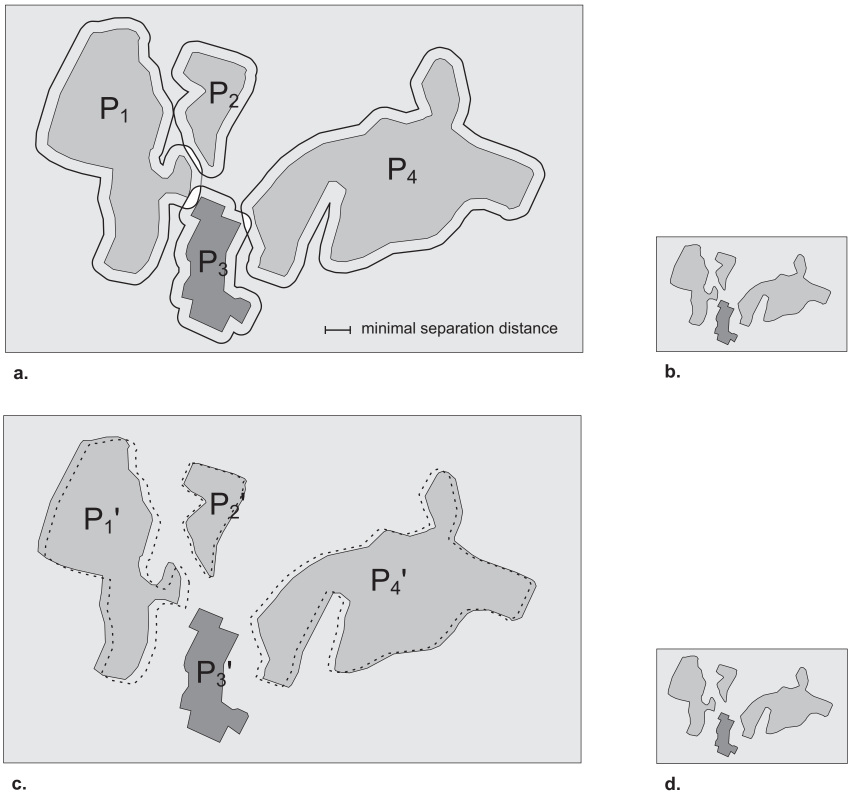


Figure 5.6: Snakes-based displacement to ensure the minimal separation distance between polygons. **a.** Polygons at the original scale with their outline buffered at half of the minimal separation distance of the target scale. **b.** Original polygons at the target scale. **c.** Original polygons (dashed outlines) and generalized polygons (solid contours) at the original scale. **d.** Displaced polygons at the target scale. (data source: © Swiss National Park)

original distances

	P_1	P_2	P_3	P_4
P_1	-	**113.7**	**77.7**	444.1
P_2	-	-	218.6	389.8
P_3	-	-	-	**98.7**

distances after 50 iterations

	P_1	P_2	P_3	P_4
P_1	-	167.7	**148.3**	560.4
P_2	-	-	235.1	422.3
P_3	-	-	-	**147.4**

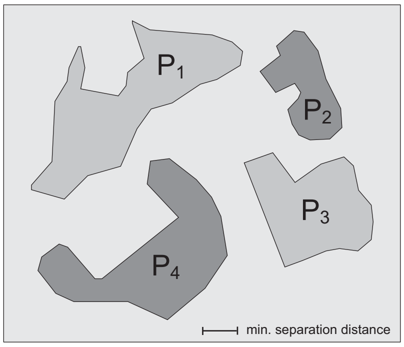
distances after 100 iterations

	P_1	P_2	P_3	P_4
P_1	-	166.9	150.1	562.4
P_2	-	-	233.6	422.9
P_3	-	-	-	**147.6**

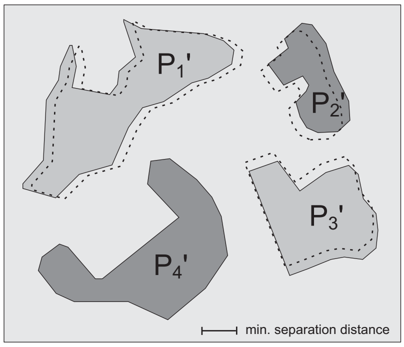
Figure 5.7: Distances between polygons. **a.** Original situation (cf. Figure 5.6). **b.** After 50 iterations of snakes-based displacement (cf. Figure 5.6). **c.** After 100 iterations of snakes-based displacement. Distances below the target minimal distance (150m) are highlighted in bold typeface. Note that although the minimal distance is not reached in all cases, the distances reached by displacement only deviate slightly from the target distance.

(iterations 51 to 100) led to an additional translation of only 0.2 m. In this experiment, the specified minimal distance was only reached entirely between P_1 and P_2. Although two critical distances remain after 50 iterations, these proximity conflicts would probably already be acceptable from a cartographic point of view, as they come very close to the target minimal distance. In practice, results of a snakes-based transformation are always a compromise between the specified constraints (e.g. minimal distance) and the required computing time. For both a faster convergence to the minimal distance and a more efficient use of computer resources, Bader (2001) proposes running two snakes processes, each with half the total number of iterations. Given a total number of 100 iterations, for instance, the result of the first process, consisting of 50 iterations, then serves as input for the second run (again numbering 50 iterations), instead of running a single snakes process with 100 iterations. As the experiments reported in Bader (2001) show, the two-stage approach allows considerably faster convergence and also provides the opportunity to the user to end the process after fewer iterations if the result is already acceptable (though perhaps not optimal).

The following experiments – cf. Figure 5.8 - illustrate how the cartographer can control the result of a snakes-based displacement algorithm by varying the weights assigned to the polygons.

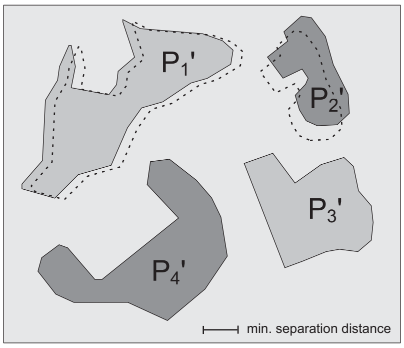


a.



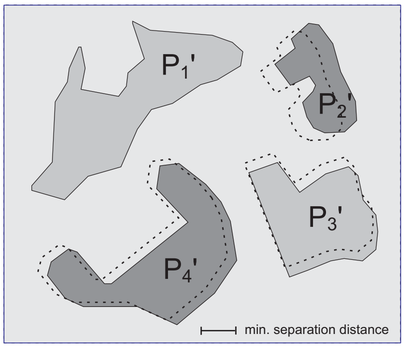
	P_1	P_2	P_3	P_4
weight	1.0	1.0	1.0	0.0

b.



	P_1	P_2	P_3	P_4
weight	1.0	1.0	0.0	0.0

c.



	P_1	P_2	P_3	P_4
weight	0.0	1.0	1.0	1.0

d.

Figure 5.8: Varying the weights assigned to the polygons in snakes-based displacement. **a.** Original situation. **b.-d.** Results of the displacement operation. The weights used are indicated below every figure. The original situation is displayed by dashed lines. (data source: © Swiss National Park)

In order to establish a desired minimal separation distance (150 m) between the disjoint polygons shown in Figure 5.8a the displacement operator is applied. For instance, such a distance is used in map generalization to guarantee the legibility of disjoint polygons despite a smaller display scale. In the original situation only the polygons P_3 and P_4 are far away from each other (175 m), P_1 and P_2 are separated by 88 m, P_1 and P_4 by 107 m, P_2 and P_3 by 90 m. The generalization solutions (Figures 5.8b-d) derived from the original situation result from binary variations of the weights, that is, either a weight of 1.0 (no resistance to deformation) or 0.0 (no deformation) is used in the examples shown here. A snakes process with 25 iterations and a target minimum separation distance of 150 m accomplished all the generalized views. In light of the previous discussion on the convergence of snakes it seems of interest that a minimum distance of 143 m was reached in every solution shown in the Figures 5.8b-d after 25 iterations.

5.4.2 Enlargement and automated propagation

The second series of experiments investigated the differences between the enlargement of polygons based on the snakes method and a scaling approach. The usage of snakes for propagating a change of a polygon geometry directly to all adjacent neighbors (automated propagation) was evaluated. The results of the experiments are shown in Figure 5.9. Figure 5.9a depicts the original situation at the source scale (upper map) and the target scale (lower map) embedding a size conflict of polygon P_2, which requires an enlargement of its area by the factor 2.0 in order to meet the minimal size constraint.

While Figure 5.9b shows the result of a scaling algorithm for enlarging P_2 (with subsequent propagation of necessary displacements by snakes), the situation shown in Figure 5.9d represents the result accomplished by snakes-based enlargement. The comparison of these two figures shows the difference between the two approaches. In contrast to the scaling algorithm (Figure 5.9b) that emanates from the polygon's center of gravity (cog), the snakes-based approach (Figure 5.9d) emanates from the polygon boundary. Thus, an equal enlargement in all directions is achieved and the shape characteristics in the polygon boundary are better preserved. The simple mathematical scaling algorithm is often accompanied by shape distortion (compare the part of P_2 in 5.9b&5.9d, respectively, highlighted by arrows). Hence, although the linear time scaling algorithm is computationally far more efficient than the snakes method, a snakes-based approach for enlarging polygons may be preferred from a cartographic point of view particularly if complex polygons are involved. Of course, a more complex scaling algorithm that calculates buffers for every polyline may lead to results similar to that of the snakes approach but such an algorithm would also require a similarly large amount of computer resources.

Regardless of the algorithm used for polygon enlargement, the polygons in the neighborhood will be affected by the enlargement and thus they will have to be updated. This propagation of the displaced polygon boundaries is cartographically and algorithmically much more complex than the mere enlargement of a single polygon. Simple, sequential algorithms are not feasible for this task. Using snakes, two different approaches for updating the neighboring polygons of an enlarged polygon are possible. The first (Figure 5.9c) integrates propagation in the process of snakes-based enlargement, that is, the force model is set to 0 and the weight to 1.0 for the neighboring polygons (see polygon P_1 in Figure 5.9a). The borders between P_1 and P_3 are adapted iteratively to the altered geometry due to the forces acting on P_2 which result from the enlargement of polygon P_2. The second approach uses two independent snakes-based processes: an iterative one for the enlargement and a single step snakes process for the propagation (see Table 5.1 for the snakes parameters applied).

	Snakes Parameters for snakes-based enlargement	snakes-based propagation
α	0.001	0.0
β	0.2	0.2
γ	2000.0	*not required*
Iterations	10	1

Table 5.1: Snakes parameter derived from empirical testing and used for snakes-based enlargement and propagation displayed in Figure 5.9.

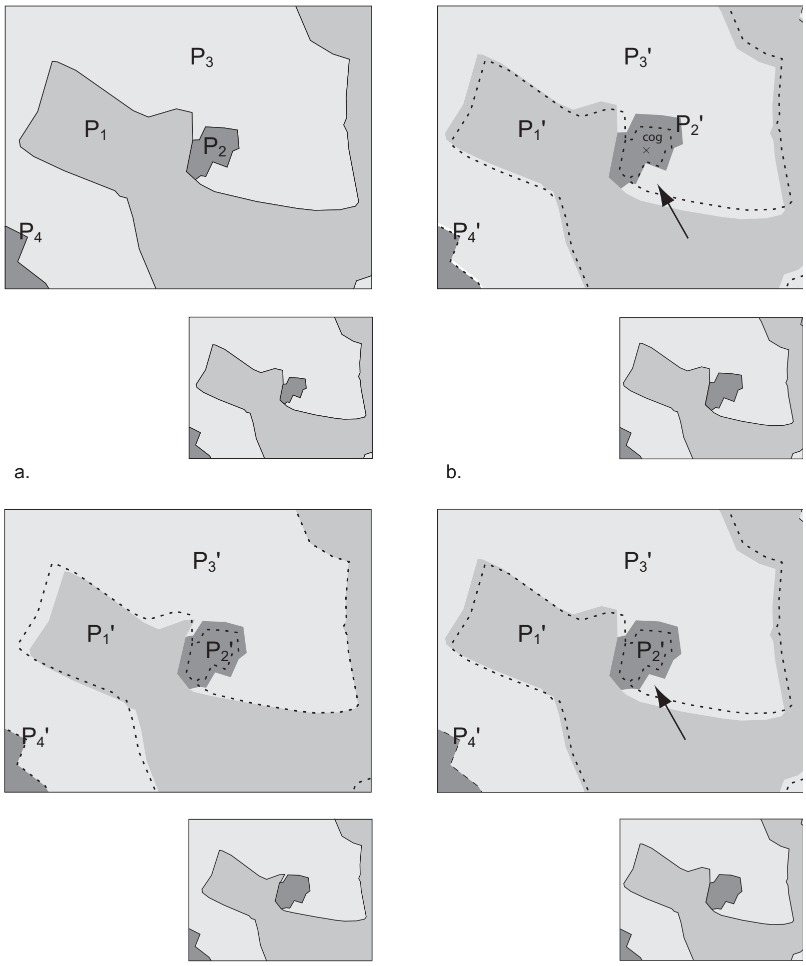


Figure 5.9: Experiments related to enlargement and automated propagation. **a.** Original situation. **b.** Two-step enlargement by scaling, followed by snakes-based propagation. **c.** Enlargement and propagation in a single snakes process. **d.** Enlargement and propagation in two separate snakes processes. In a to d the situation is shown at the source scale in the upper graphics and at the target scale in the lower maps of each figure. In the upper maps of the figures b, c, and d the original situation is shown in dashed lines. (data source: © Swiss National Park)

The single-step snakes process for propagation can also be used in combination with other enlargement algorithms, such as a scaling algorithm. This propagation method was used in the examples represented in Figures 5.9b&d, respectively. Because the borders between P_1 and P_3 were included in the iterative snakes-based enlargement of P_2 the first approach produced a higher overall computational cost than the second approach, where displacements for these geometries were calculated only once. When comparing the images in Figure 5.9c, where enlargement and propagation was performed in one snakes process, and Figure 5.9d, which resulted from two independent snakes processes, differences are noticeable in the way displacements have been accommodated on the boundaries between P_1 and P_3. These differences are due to the way displacements were applied, that is, either by several small displacement vectors with varying length and directions (first approach) or by a single vector (second approach).

In this example both solutions provide similar results. Further tests are required to

- draw conclusions about the cartographic quality of propagation provided by these two possible ways to accomplish automated propagation; and
- establish guidelines for the translation of generalization constraints (e.g., maintenance of shape characteristics or positional accuracy of polygon boundaries) into snakes parameters.

For concave inlets of polygons, the enlargement by snakes may sometimes produce cartographically unsatisfactory solutions and cause conflicts between parts of a polygon. See Figure 5.10b, where the enlargement of the polygon by a factor of 1.3 almost fills in the inlet. Figure 5.10c illustrates the failure of snakes-based enlargement which resulted in a self-intersection of the polygon boundary and the creation of an invalid polygon topology. These errors stem from the implementation of the enlargement operator as a boundary-moving method, which implies that the enlargement is achieved by the displacement of the polygon boundary based on the forces derived from its distance to the virtual push object. A possible solution to this problem is the introduction of an additional force model that can handle such conflicts within a polygon boundary. However, one might also argue that narrow concavities represent a specific case that is best detected and preprocessed by other algorithms which would, for instance, remove such inlets. Hence, they would not provide a risk to the snakes-based enlargement algorithm applied subsequently.

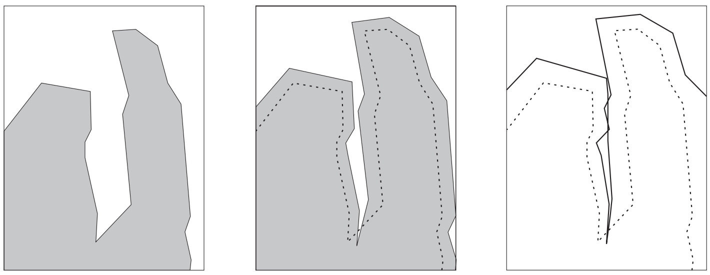
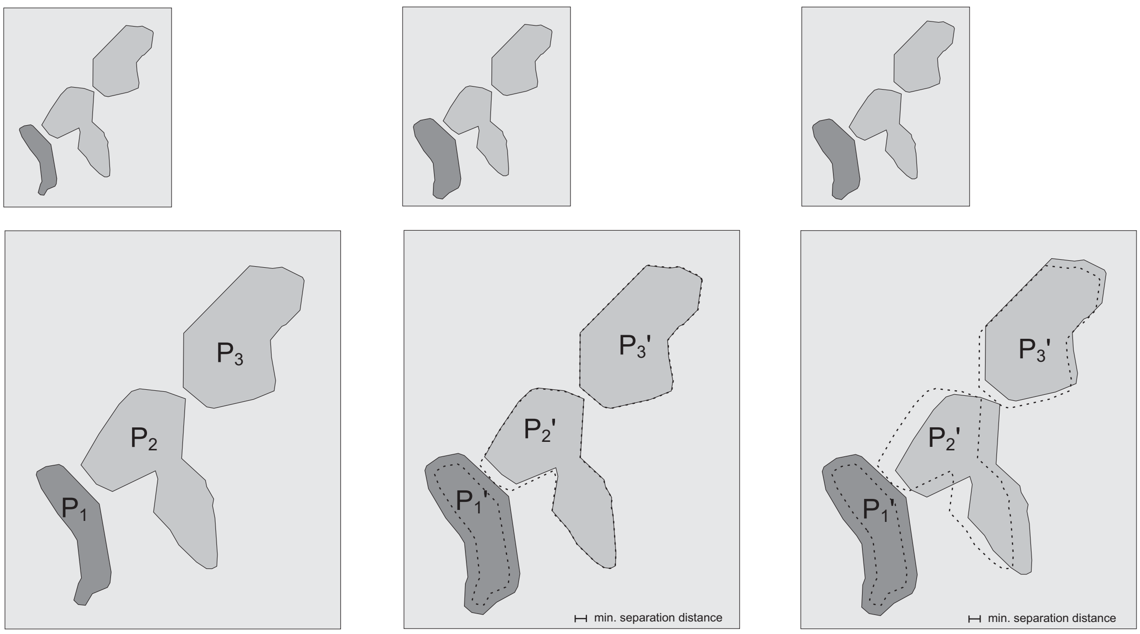


Figure 5.10: Snakes-based enlargement of concave inlets: **a.** Original situation. **b.** Unsatisfactory result of the snakes-based enlargement by a factor of 1.3 (inlet too narrow). **c.** The polygon's enlargement by a factor of 1.5 fails due to self-intersection of the polygon outline. In Figures b&c. the original situation is shown in dashed lines. (data source: © Swiss National Park)

5.4.3 Interplay of different generalization operators

Snakes support comprehensive generalization; i.e., the transformation of one polygon is not calculated in isolation from the whole polygonal subdivision, rather, the polygon interacts with its adjacent and nonadjacent neighbors to find the (mathematically) optimal result. The creation of new proximity conflicts resulting from the enlargement of a polygon is avoided. This aspect, as well as the interplay of different generalization operators enabled by snakes, is the issue of the following experiment.

Figure 5.11a portrays the original situation at the original scale of 1:25,000 (lower map) and the target scale of 1:50,000 (upper map). The dark gray polygon P1 requires an enlargement by a factor of 1.9 to reach the minimal size imposed by the minimal size constraint with respect to the target scale. The other two polygons (P_2 and P_3) are of sufficient size and not part of a proximity conflict with P_1 or with each other. Due to the enlargement of polygon P_1, however, a new proximity conflict between polygons P_1 and P_2 is created – compare the boundary of polygon P_1' and the original boundary of P_2' shown as a dashed line in the lower map of Figure 5.11b. By specifying the weight and force model property of polygon P_2, various scenarios are possible to resolve this new conflict. Figure 5.11b shows the local deformation of P_2 and Figure 5.11c its translation. Note the weights and force models listed below Figures 5.11b&c. This adaptation to the transformations of P_1 was performed without requiring any additional user interaction in a single snakes process. In the same way as P_2 adapts to the altered geometry of P_1, additional conflicts could be solved in the same snakes process, such as the proximity conflict between P_2' and P_3' in Figure 5.11c. The translation of P_3' in Figure 5.11c was driven by the displacement of P_2 which was itself triggered by the enlargement of P_1.



	P_1	P_2	P_3
weight	1.0	1.0	1.0
force model	1.0	3.0	3.0

	P_1	P_2	P_3
weight	1.0	1.0	1.0
force model	1.0	4.0	4.0

a. b. c.

Figure 5.11: Example for the interplay of different generalization operators enabled by snakes. **a.** Original situation at the target scale (upper map) and original scale (lower map). **b.** Enlargement of polygon P_1 and preservation of the minimal distance to P_2 by exaggeration of the 'isthmus' between P_1 and P_2 at the source and target scales. **c.** Enlargement of polygon P_1 and preservation of the minimal distance to P_2, and subsequently between P_2 and P_3, by displacement represented at the target and source scales. **b.&c.** The weights and force-models used for every polygon are listed below the resulting figure. The original situation is depicted in dashed lines. (data source: © Swiss National Park)

The snakes process required 40 iterations for accomplishing the result in Figure 5.11b and 80 iterations for that in Figure 5.11c. The iterations were done to ensure the enlargement of P_1 and the minimal distance of approximately 30 m between all polygons. When using the displacement operator, forces in the opposite direction acting on P_2 emanated from P_1 and P_3. The translation done by P_2' and P_3' in Figure 5.11c is not the best solution conceivable because the relative position of P_2' to P_3' has changed. The combined displacement of both polygons by a single displacement vector would provide a more convenient solution. As discussed previously, forces for displacement are exclusively derived from the shortest connection between the push polygon and displaced polygon. Hence, further enhancements in the force computation may need to be considered, such as an improvement of the computation of the displacement vectors which would enable the combined displacement of several polygons or the consideration of free map space, which a polygon could be shifted to without risking a new conflict.

The process of automatically assigning weights and force models to polygons depends on the need for enlarging P_1 while ensuring the minimal distance between P_1 and P_2 and between P_2 and P_3. If these requirements are present, the weight value of P_1 is set to 1.0 and the force model 'vertex_line' is applied. The force model 'combined' (indicating exaggeration) and a weight of 1.0 for polygons P_2 and P_3 make it possible for these polygons to maintain absolute position and lead to a solution that does not require additional map space (Figure 5.11b). The solution depicted in Figure 5.11c is achieved by a weight of 1.0 and the force model 'vertex_line_max' indicating displacement of polygons P_2 and P_3. This parameter setup is appropriate for preserving the shape of P_2 and P_3, but it should be only chosen if ample map space is available to move P_3 so as to avoid new conflicts with polygons nearby.

5.4.4 Snakes in polygon generalization

In contrast to the previous experiments the following experiment intends to demonstrate the applicability of the presented algorithm in the context of a large set of polygons. Figure 5.12a shows a test data set (2.25 x 2.1 km^2) and its generalization by our algorithm. Given a target scale of 1:75,000, the purpose of the map and its cartographic representation, the following minimal dimensions were taken into account (SSC 2002): the minimal area of the polygon should be 7,500 m^2; its minimal width should be 40 m; and the minimal distance between two disjoint polygons 50 m. . Figure 5.12a shows an overlay of the extract at the source scale 1:25,000 (depicted by dashed lines) and the target scale of 1:75,000 (depicted by solid contours). Figure 5.12b portrays the original situation at the target scale, while Figure 5.12c shows the generalized result at the target scale.

The potential of the snakes method for polygon generalization was already demonstrated in the previous series of experiments. The numbers in circles in Figure 5.12a indicate satisfactory generalizations achieved by the presented algorithm. Situation (1) shows the enlargement of a polygon that was too small and the subsequent propagation to the connected polygon boundaries; situation (2) shows the enlargement of an isolated polygon; situation (3) shows the displacement of two polygons in order to accomplish minimal distance; situation (4) shows the interplay of a displacement and enlargement operation; and situation (5) shows the widening of a polygon that was too narrow.

The specific capacity of snakes to solve conflicts in polygon generalization is restricted to cases where the conflicts can be settled by modifying the boundaries of the involved polygons. Situations that could be solved better by alternative operators or require extensions to the displacement, enlargement, and exaggeration operators enabled by our algorithm are highlighted in Figure 5.12a by letters in circles. For instance, the proposed algorithm is not able to achieve a reduction of the number of polygon, as required by the elimination and aggregation operators. Supposing the island polygon in situation (a) represented a rather unimportant category, the elimination of this polygon may be preferred to its enlargement. An aggregation algorithm that merges several polygons into a new polygon might provide a better generalization for the proximity conflict marked as (c), as this could maintain the width of the passage between the three conflicting objects and the prominent polygon south of this group. Situation (b) depicts a part of a polygon that is too narrow; its width is below the allowed minimal width of 40 m and cannot be widened by the proposed algorithm as the algorithm is not able to consider internal conflicts in a polyline (see also Figure 5.10).

5.5 Conclusions

Our experiments emphasized that snakes are not only well qualified for the displacement of linear objects, but also for selected generalization operations in polygon generalization. Compared with other algorithms for the solution of size and proximity conflicts (Bader and Weibel 1997, Jones et al. 1995) the approach discussed here has two distinct advantages: 1) the possibility of a comprehensive solution of size and proximity conflicts within a group of polygons and 2) the integration

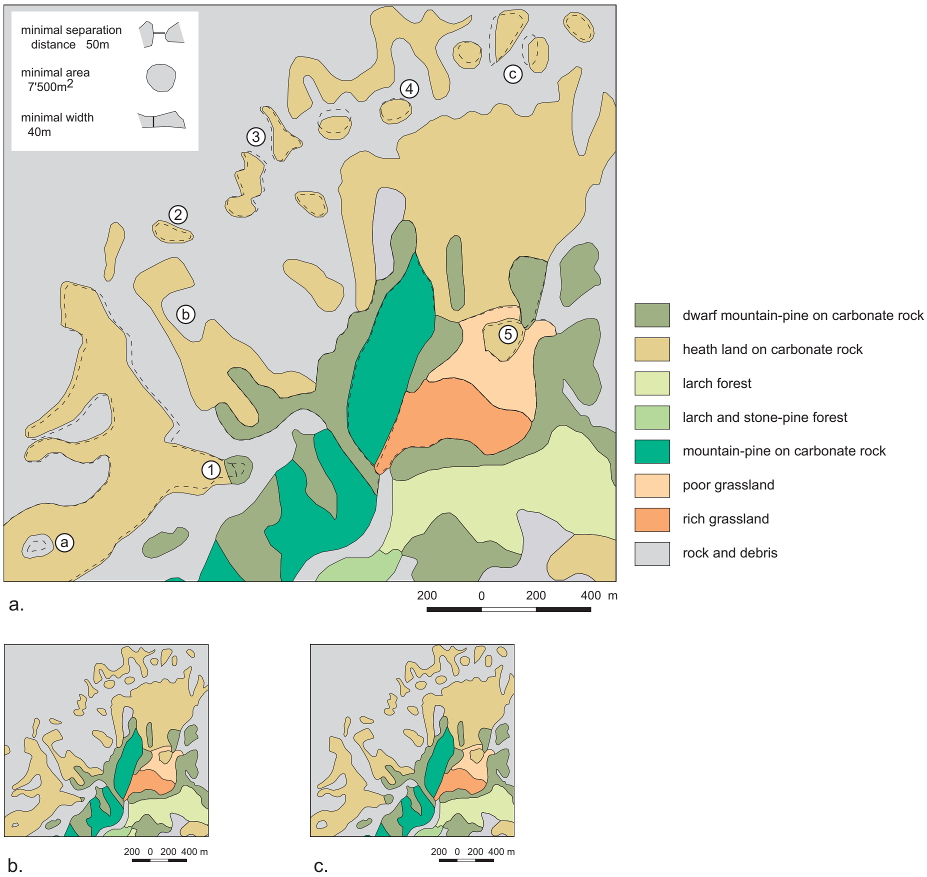


Figure 5.12: The test data set generalized by our algorithm. **a.** The original situation is depicted by dashed lines and the generalized situation is visualized by solid contours (colored for the represented vegetation class). Numbers enclosed in circles indicate successful applications of the snakes method in polygon generalization and letters in circles refer to situations where additional operators or further enhancements of the presented algorithm are required. These are discussed in more detail in text. Figure at scale 1:25,000. **b.** Original data set at target scale. **c.** Generalized data set at target scale. (data source: © Swiss National Park)

of automated propagation in the transformation process. Until recently, independent algorithms (Bader and Weibel 1997, Jones et al. 1995) have been used to solved different conflicts sequentially, and the update of a transformed polygon's adjacent neighbors was done separately. The complexity of handling geometric transformations and their propagation through large polygon mosaics sequentially is overwhelming and is probably one of the main reasons why we have not yet better seen algorithms for polygon generalization. Our comprehensive approach to generalization is enabled by the usage of a single optimization algorithm, which is triggered in such a way that it achieves the displacement, enlargement, exaggeration of a polygon or an arbitrary combination of these operators in one common procedure.

One of the main drawbacks is the high computational cost required by snakes. The computational cost is primarily defined by the number of vertices of the polygons considered, because the snakes method involves the computation of matrices that are proportional in size to the number of vertices. For instance, the displacements for the total of 162 vertices in Figure 5.6 required 45 seconds computing time on a Sun Ultra 30 with 384 MB memory. Although these results are based on a prototype implementation where much room exists for improvement (use of a spatial index for spatial search, sparse matrix computation, etc.), it is recommended to always keep in mind alternative algorithms that could provide similar results faster, especially in the case of less complex conflicts. For instance, the enlargement of a single island polygon is accomplished more easily and efficiently by a scaling algorithm than by snakes. However, propagation of modified polygon geometries to adjacent neighbors must still be solved, even if simpler algorithms are used for sub-tasks. For the propagation of shape modifications, the snakes approach was shown to be a possible candidate that also allows the integration of other, sequential algorithms.

Our experiments also highlighted the algorithm's sensitivity to the setup and tuning of the snakes parameters, that is, the shape parameters α and β, the iteration term γ, and the number of iterations. The choice of appropriate values with respect to the target scale and the interdependencies between the parameters need further investigation. Hence, our experiments were undertaken with the help of empirically derived default values for the specific scale change and type of data. Conversely, the polygon-specific parameters of weight and force model can be assigned automatically by considering the spatial context of the polygon, the spatial and semantic properties of the involved polygons, and the type of generalization conflict encountered.

A starting point for future research could be the further investigation of the influence of the snakes parameters on the algorithm's outcome and of the interdependencies of parameters of the snakes method. Along these lines, the aim should be to formulate generic guidelines for establishing snakes parameters in polygon generalization according to a specific scale change, data type (e.g. geology, land cover) and conflict type (e.g. size or proximity). In doing so, the algorithm's ease of use would significantly increase and the cartographer's control over the algorithm's result could be improved. So far, empirical testing is required, in order to fine tune the snakes' parameters with respect to the desired scale change and the kind of generalized data and to accomplish satisfactory generalization results.

The proposed boundary-moving algorithm is not qualified to establish comprehensive polygon generalization since, in practice, polygon generalization has to consider numerous different generalization constraints, such as the positional accuracy of a polygon boundary or its granularity. Additional generalization operators and algorithms such as semantic generalization (by class aggregation) or shape simplification of polygon objects are thus needed. However, due to the demonstrated potential of the snakes method it is believed that an implementation of the outlined framework (cf. chapter 4&7) should make use of the presented snakes-based algorithm. In doing so, the characteristics of decision making of these two approaches should be considered. While an algorithm based on optimization techniques, such as snakes, aims at the concurrent resolution of different conflicts the agent-based model solves conflicts in turn, in order to be able to control the development of constraints' satisfaction and find iteratively the best solution. Consequently, two strategies for the integration of the snakes-based algorithm into an agent-based framework for automated generalization seem to be reasonable: (1) the snakes-based algorithm always implements exclusively a single cartographic operation, that is, either a displacement, enlargement or exaggeration operation, that is proposed by a violated constraint; and (2) the snakes-based algorithm is applied to a group of polygons or an individual polygon as an universal plan. If an unsatisfying solution is established the iterative agent-based approach is tried for the generalization of the corresponding group of polygons or polygon. Otherwise, the solution generated through snakes is accepted.

Chapter 6

Modelling constraints for polygon generalization

6.1 Introduction

Besides algorithms for conflict resolution constraints are the other main prerequisite for the implementation of the framework for automated, agent based polygon generalization outlined in chapter 4. In this framework, constraints are used to detect conflicts, to control the resolution of conflicts and to evaluate and compare accomplished solutions (Weibel 1996, Ruas 1998).

Previous work such as Weibel (1996), Peter and Weibel (1999a) and Edwardes and Mackaness (2000) discussed constraints for polygon generalization mainly on a conceptual level. Starting from their work a basic set of constraints for polygon generalization should be established. The novelty of this work lies in enriching this set in such a way that the constraints of the set could be easily applied to automated polygon generalization, that is, these constraints should have the potential to control an agent-based generalization process. The resulting requirements are specified in the next section together with a generic discussion on generalization constraints. The remaining sections deal exclusively with constraints for automated polygon generalization and their modelling. Altogether, the chapter provides the basis for the integration of this set of constraints into the framework for agent-based automated polygon generalization outlined in chapter 4.

6.2 Generalization constraints

The concept of constraints was originally transferred from computer science to map generalization by Beard (1991) and later on emphasized by, among others, Mackaness (1995), Weibel (1996), Weibel and Dutton (1998), Ruas (1999) and Barrault et al. (2001). This section intends to provide an overview of both the basic concept of constraints in map generalization and of different taxonomies of constraints relevant to this work.

6.2.1 Concept of generalization constraints

The concept of constraints used in this thesis owes to previous research conducted by Ruas (1998, 1999) and in the AGENT project (Barrault et al. 2001). A constraint designates a final product specification on a certain property of an object that should be respected by an appropriate generalization. Constraints are often implemented as functions of comparison. For instance, the size of a forest polygon should be greater than 1000m^2. In comparison to rules, constraints are neither bound to a single condition – in fact constraints are often related to a synthesis of conditions (Ruas and Plazanet 1996) – nor to a particular action (Beard 1991). Every constraint is linked to the following values and methods (Ruas 1998, 1999, Barrault et al. 2001) – their interdependencies are shown in Figure 6.1:

- a *goal value* that defines the value an object should at least reach or maintain during generalization according to a certain constraint. In other words, goal values are thresholds that objects should respect in order to satisfy a certain constraint.
- a *measure*, i.e. a method for computing the current value of the property that the constraint refers to.
- an *evaluation method* that examines the compliance of an object to a certain constraint, that is, it determines the satisfaction of the constraint according to this object. A so-called severity value describes the level of satisfaction, that is, the discrepancy between the measure's result and the goal value previously defined. The satisfaction of constraints is standardized to a range from 1 ('very bad') over 2 ('bad'), 3 ('medium') and 4 ('good') to 5 ('perfect') in order to allow the comparison of the severity of different constraints.
- a *list of plans* [plan$_1$, plan$_2$, ... , plan$_n$]. A plan designates a list of cartographic algorithms and corresponding parameters that are suited to improve the constraint's satisfaction. In consideration of the severity and the specific situation the same constraint may propose different lists of plans.
- an *importance value* denotes the relative importance of constraints, that is, it represents the importance of a constraint to the quality of a generalization result, in comparison to other constraints. The importance value is essential to compromise between different constraints attached to the same object and to validate the affect of a generalization operation.
- a *priority value*. This value describes the priority of treatment of a constraint in the generalization process, that is, it defines which constraint should be solved first. Note that, constraints of high importance need not to have a high priority as well.

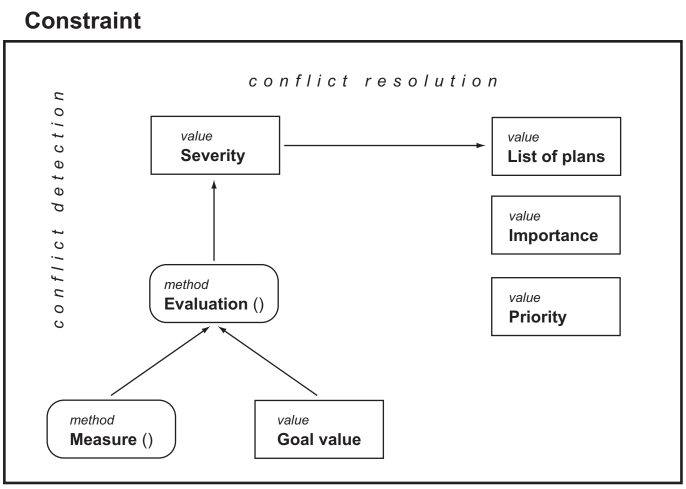


Figure 6.1: The values and methods attached to a constraint and their interdependencies.

Consequently, every constraint of the aspired set must hold a goal value, a measure, an evaluation method, a list of plans, an importance value and a priority value, in order to meet the requirements of agent-based generalization.

6.2.2 Taxonomies of generalization constraints

Since there seems to be no agreement on taxonomies of generalization constraints in the 'generalization community' – compare, for instance, those taxonomies discussed in Weibel (1996), Harrie

(1999) and Ruas (1999) – this section provides a brief overview of those taxonomies of constraints used throughout this thesis. The discussed taxonomies accommodate the fact that every generalization constraint

- is bound to a specific spatial level of a polygonal subdivision.
- is related to a certain aspect of a polygonal subdivision, namely either its graphical appearance, its underlying topology, its spatial and semantic structure or its generalization.
- plays a specific role in the generalization process.

Spatial levels. A generalization constraint is always evaluated with respect to a specific cartographic object. Cartographic objects in polygonal subdivisions occur at four different spatial levels, namely the map, a group of polygons, a polygon and a polygon outline (cf. section 4.2). Hence, every constraint can be associated with one or more of these levels.

Similar aspects. Alternatively, constraints may be subdivided into four groups of constraints that concern similar aspects of a data set (Dutton et al. 1998, Weibel and Dutton 1998, Ruas 1999):

- *Metric (or graphical) constraints* translate human limits of perception (Maliĉ 1998) into metric perceptibility thresholds (minimal dimensions) of map objects. The specification of corresponding goal values depends on the target scale, the output medium (e.g. paper or screen), the map purpose and the cartographic representation (e.g. contours or filled contours). Minimal dimensions of polygonal subdivisions are among others discussed in SSC (2002) and Spiess (2002).

- *Topological constraints* ensure that the topological structure of a polygonal subdivision is maintained or modified consistently (Dutton et al. 1998, Ruas 1999). For instance, self-intersections of a polygon boundary or any intersection of two polygon boundaries must be avoided.

- *Structural constraints* intend to preserve the spatial and semantic structure of a data set in consideration of aesthetics and visual balance. With respect to aesthetics and visual balance some authors, such as Weibel (1996) and Dutton et al. (1998), proposed a further distinction of so-called gestalt constraints. This work abstains from this distinction. Structural constraints are dealt with on the level of an individual polygon (e.g. shape characteristics of a polygon), a group of polygons (e.g. spatial alignment of objects, size ratios of objects) and the whole data set (e.g. relative areas of categories).

- *Procedural constraints* relate to the generalization process itself. So, they influence the sequence of generalization operations and the selection of suitable algorithms and parameters, respectively. Such a constraint may specify a threshold of semantic similarity for merging polygons, for instance, in the course of an elimination or aggregation operation.

Generalization process. Another taxonomy looks at the role of constraints in the generalization process. Hence, constraints are divided into offensive and defensive constraints. *Offensive constraints* define a need for generalization and every appropriate generalization of a data set must meet them, respectively. Thus, the violation of such a constraint always leads either to a spatial or semantic transformation of a data set (e.g. in the case of the violation of a metric constraint) or to the rejection of a generalized data set (e.g. in the case of the violation of a topological constraint). *Defensive constraints* denote such constraints that need not to be met imperatively but strive for optimal compliance. They are flexible, while offensive constraints are strict. For instance, shape distortion or changes in size ratios should be kept as low as possible. While they are not imperative, defensive constraints may help to identify the best solution among different possible solutions which equally satisfy all offensive constraints. With respect to the generalization process, Ruas (1999) proposed a slightly different distinction, namely constraints of maintenance and of generalization. Constraints of maintenance should preserve a property as faithfully as possible but, in contrast to defensive constraints, they can be either strict (e.g. topology) or flexible (e.g. polygon shape). Constraints of generalization are those constraints that must be respected, such as the minimal size of a polygon.

6.3 Constraints for polygon generalization

While the previous section dealt with constraint on a general level this section presents a conceptual discussion of constraints for polygon generalization. Constraints are organized, according to the taxonomy of 'similar aspects', into groups of metric, topological, structural and procedural constraints (cf. section 6.2.2). Additionally, the spatial level that a constraint refers to is annotated in parentheses behind every listed constraint. Subsequent sections examine methods for both the evaluation of the satisfaction of constraints (section 6.4) and the coherence of generalization operators and constraints' satisfaction (section 6.5).

As cartographic knowledge related to map generalization in general (Weibel et al. 1995, Müller et al. 1995b), constraints for polygon generalization may be extracted from

- interviews with *cartographic experts* (Schylberg 1993, Kilpeläinen 2000),
- *existing map series* (Edwardes and Mackaness 2000),
- *textbooks* of (thematic) cartography (Imhof 1972, Dent 1990, Slocum 1999) or map generalization (SSC 2002), and
- guidelines of mapping agencies (McMaster 1991, Landesvermessungsamt Nordrhein-Westfalen 1993).

With respect to polygon generalization these sources often focus either on very specific topics such as the elimination of an individual polygon (Schylberg 1993, Kilpeläinen 2000) or the generalization of a single category such as 'forest' as a layer of a topographic map (Imhof 1972, Arnberger 1993, SSC 2002). Hence, the inventory of constraints for polygon generalization provided here is based on both constraints derived from the sources listed above and those constraints proposed initially by Weibel (1996).

6.3.1 Metric constraints

M1 Consecutive vertex distance (line level). Consecutive vertices of a polygon boundary should be separated by a minimum distance at least. This constraint intends not to trigger generalization but to speed up the generalization process by removing redundant vertices from a polygon boundary while maintaining the polygon shape as faithfully as possible (Visvalingam and Williamson 1995).

M2 Outline granularity (line level). Imperceptible crenulations of a polygon boundary must be eliminated (Figure 6.2).

M3 Distance between boundary points (polygon level). Any non consecutive points of a polygon geometry should be separated by a minimum distance at least.

M4 Minimal area (polygon level). All polygon objects should have at least a minimal area for the given target scale. In general, objects should "be large enough for the reader to see and differentiate areal patterns" (Dent 1990, p. 152).

M5 Respect spatial context (polygon and group level). Individual polygons and groups of polygons should respect their spatial context in conflict resolution. In other words, this constraint prevents the creation of new conflicts between generalized polygons or groups of polygons and other polygons that are not generalized at the same time. For instance, a group of disjoint island polygons should respect their spatial context, that is, the polygon that embeds them.

M6 Object separation (group level). The distance between two disjoint polygons should be not less than a minimum distance.

M7 Number of categories (map level). The number of retained categories is closely linked to the spatial detail of a polygonal subdivision since the more categories are shown the more polygons will be portrayed. The target scale, the map purpose (e.g. a geology map for a tourist vs. an expert in geology) and the map theme determine the concrete number of categories. Due to its dependence on the specific map that needs to be generalized this constraint represents a typical case where no global rules exist but the user will specify the corresponding threshold values (cf. section 6.4).

6.3.2 Topological constraints

T1 Self-intersection (line level). A valid polygonal subdivision of the plane – see Jaakkola (1998) or Frank et al. (1997) for a definition of a polygonal subdivision – must not contain self-intersecting polygon geometries.

T2 Intersection of different polygons (polygon and group level). Intersections of polygon geometries must be avoided since they prohibit the creation of a topologically consistent polygonal subdivision.

6.3.3 Structural constraints

S1 Shape distortion (polygon level). The distortion of a polygon shape should be minimized, that is, shape characteristics such as angularity or intrinsic micro shapes should change as little as possible.

S2 Absolute position (polygon level). The change of an object's absolute position should be minimized.

S3 Relative configuration (group level). Generalization should maintain as best as possible the direction and distance relations of objects (Yaolin et al. 2001). That is, generalization should preserve not only the positions of polygons, relative to each other, but also characteristics in the spatial distribution of polygons such as alignments, clusters and containments.

S4 Size ratios (group and map level). Size ratios should be preserved in a polygonal subdivision on different levels during generalization, for instance, between polygons of an alignment or a cluster, between polygons of a category and between all categories building a polygonal subdivision.

6.3.4 Procedural constraints

P1 Illogical results (line, polygon and group level). Generalization should not produce results that are implausible with respect to the spatial (e.g. a phenomenon occurring in compact polygons shown by long and thin polygons) or the semantic component (e.g. impossible neighborhoods of categories) of the represented theme.

P2 Child entity's constraints (polygon, group and map level). Both the hierarchical organization of the spatial levels of polygon generalization and the agent-based approach (Galanda and Weibel 2002a) make so-called parent entities responsible for the generalization of their child entities. For instance, a polygon cluster supervises the independent generalization of its polygons or a polygon controls the generalization of its boundary. This constraint is attached to every parent entity in order to ensure sufficient satisfaction of all those constraints which are delegated to its individual child entities (Ruas 1999, Barrault et al. 2001, Galanda and Weibel 2002a).

P3 Aggregation similarity (group level). This constraint defines the minimum level of semantic similarity required to merge two polygons of different categories. For instance, a polygon of the category 'nursery' is rather aggregated with a polygon of the category 'forest' than with one of the category 'lake'. In other words, the semantic similarity between the categories 'nursery' and 'forest' is closer than the one between the categories 'nursery' and 'lake'. Semantics play a important role in polygon generalization. Since the spatial and semantic components of a polygonal subdivision are intimately linked, and any treatment of one in isolation to the other will have a high risk of misrepresenting the phenomenon (Mark and Csillag 1989).

P4 Equal treatment (all levels). Ensure that similar conflicts are solved in similar ways across the polygonal subdivision.

6.4 Evaluate constraints

A constraint is always evaluated for an individual object at one of the spatial levels of polygon generalization. The process of evaluating constraints is of great relevance in agent-based approaches to map generalization, since it is responsible for conflict detection, conflict resolution and evaluation of results – see also the discussion of an agent's life cycle in section 4.3. In order to fulfill these tasks every constraint needs to be formalized. The formalization of a constraint designates the transformation of a constraint into a formal description that is interpretable by computers. Formalization is based on *goal values* and *measures* (cf. section 6.2.1).

On a global level, goal values are derived from generalization controls such as map purpose, target scale, limits of perception etc. (Weibel and Dutton 1998) while on a local level the specification of a constraint, that is, the definition of a constraint's goal value is influenced by an object's semantics (Ruas 1999). The specification of goal values directly affects the generalization results, that is, inappropriate goal values can change the intrinsic character of a polygonal subdivision (e.g. too many small polygons are eliminated) or result in an insufficient generalization (e.g. too many details are kept).

The evaluation of a constraint determines the satisfaction of a constraint with respect to a certain object. That is, it compares the result of the measure, calculated on the corresponding object, to the constraint's goal value and derives a degree of satisfaction (=severity) from their discrepancy. However, it is not equally easy to evaluate the satisfaction of different constraints by means of numeric measures. Certain constraints are directly linked to geometric or semantic properties of a polygonal subdivision. Examples include 'M4 Minimal area' or 'M7 Number of categories'. Thus, it is straightforward to establish an appropriate method for their evaluation. Other constraints, such as 'S1 Shape distortion' or 'S3 Relative configuration' are fuzzy and ill-defined (Weibel and Dutton 1998). Hence, they are very difficult to formalize and subsequently to evaluate. More detail on the evaluation of the individual constraints is therefore provided in the following sections.

6.4.1 Evaluation of metric constraints

As mentioned above metric constraints relate to limits of perception, i.e. minimal dimensions of mapping. Hence, maps, textbooks (e.g. Spiess (2002), SSC (2002)) and guidelines of mapping agencies (e.g. McMaster (1991)) provide sufficient information to specify goal values. Goal values depend mainly on the cartographic representation of the polygonal subdivision and the output media (Weibel and Dutton 1998). Thresholds that are valid for paper maps are listed in Table 6.1, along with an illustration by solid contours. Note that, goal values are defined in map units in order to ensure independency of the target scale. The measures used for the evaluation of metric constraints are summarized in Table 6.2.

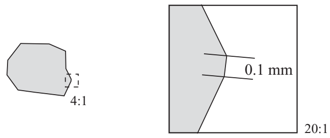
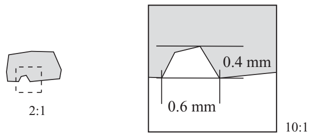
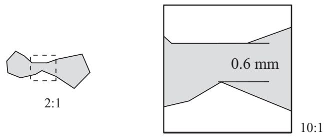
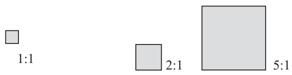
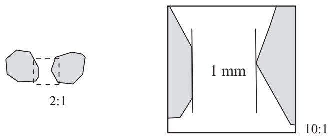
Constraint	Goal value	
M1 Consecutive vertex distance	> 0.1 mm	
M2 Outline granularity		
minimal shape width	> 0.6 mm	
minimal shape height	> 0.4 mm	
M3 Distance btw. boundary points	> 0.6 mm	
M4 Minimal area	> 4 mm^2	
M5 Respect spatial context	TRUE	
M6 Object separation	> 0.6 mm	
M7 Number of categories	varying	

Table 6.1: Goal values of metric constraints (in map units) for polygon generalization with respect to paper maps. Listed goal values stem from SSC (2002), Spiess (2002) and empirical observation.

Evaluate 'M1 Consecutive vertex distance'. The constraint 'M1 Consecutive vertex distance' is evaluated by determining the minimum distance d_{Consec} between any pair of consecutive vertices (v_i, v_{i+1}) along a polygon boundary – cf. Table 6.2. The constraint's goal value was set to a minimum of 0.1 mm through empirical observation and testing in order to remove exclusively redundant vertices.

Evaluate 'M2 Outline granularity'. An excessive granularity of a polygon outline is defined by the occurrence of imperceptible shapes (micro shapes). Hence, the evaluation of this constraint relies on a preliminary identification of such micro shapes. Similar to the concept proposed by Wang and Müller (1998), it is assumed that every polygon boundary consists of a sequence of external and internal shapes. An external/internal shape is built by a set of subsequent vertices of the polygon boundary that include an internal angle lower/greater than 180° as well as a start and end vertex with an opposite angle (internal angle is greater/lower than 180°). In Figure 6.2 a polygon is split up into its external shapes (shape 2, 4, 6) and internal shapes (shape 1, 3, 5).

According to the mapping guidelines presented in SSC (2002) and Spiess (2002) shapes are regarded as imperceptible if their height falls below 0.4 mm and/or their width falls below 0.6 mm. Here, the height of a shape is measured by calculating the maximum distance of any vertex of a respective shape to the straight line that connects the shape's start and end vertex – cf. the inlet

shape		*start*	*enclosed vertices*	*end*
1	int.	v_4	v_5	v_6
2	ext.	v_5	v_6, v_7, v_8, v_9	v_{10}
3	int.	v_9	v_{10}	v_{11}
4	ext.	v_{10}	v_{11}, v_{12}	v_{13}
5	int.	v_{12}	v_{13}, v_{14}, v_{15}	v_{16}
6	ext.	v_{15}	$v_{16}, v_{17}, v_{18}, v_1, v_2, v_3, v_4$	v_5

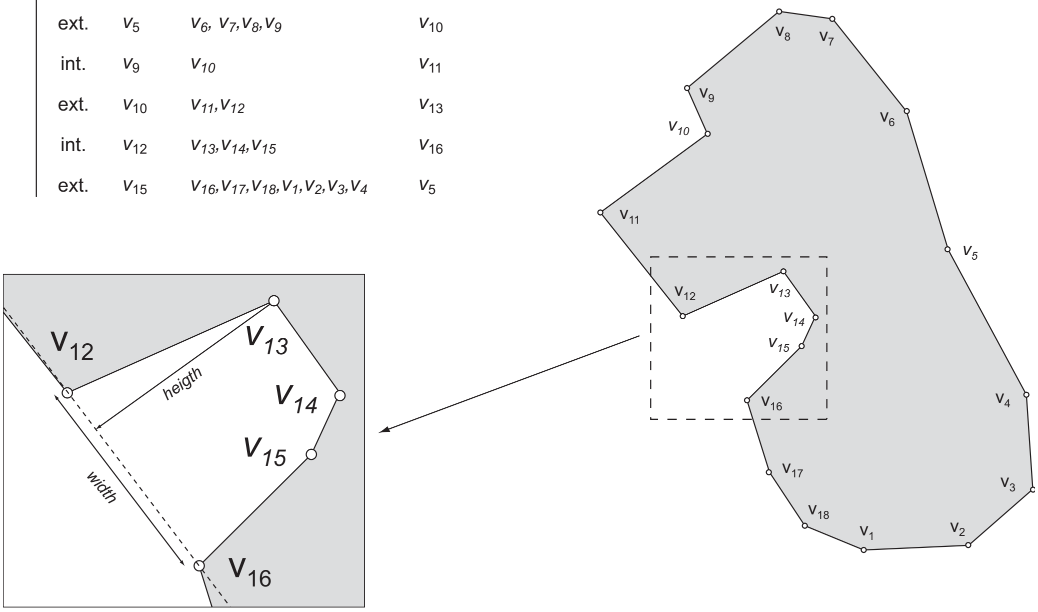


Figure 6.2: A polygon boundary composed of internal and external shapes and the principle of calculating a shape's width and height.

of Figure 6.2 and Table 6.2. The width is defined as the distance between the shape's start and end vertex along a straight line – refer again to the inlet of Figure 6.2 and the Table 6.2.

Evaluate 'M3 Distance between boundary points'. The distance between any points of a polygon geometry is determined by the measure 'detect narrow parts' proposed by Bader and Weibel (1997). This measure identifies narrow parts of a polygon by means of a conforming Delaunay triangulation (Bern and Epstein 1995) built of the points of the polygon boundary. Considering the mapping guidelines of SSC (2002), the goal value of this constraint is set to a value of 0.6 mm.

Evaluate 'M4 Minimal area'. In order to control the satisfaction of the constraint 'M4 Minimal area', a basic area measure is used to calculate the area of a polygon. Considering the most common representation of polygonal subdivisions, that is, solid contours, a minimal area of 4mm^2 is proposed (Maliĉ 1998). The goal value may be varied according to a polygon's semantics, for instance, taking into account whether a polygon belongs to a frequent or rare category or if the polygon's category is considered to be more or less important with respect to map theme and map purpose.

Evaluate 'M5 Respect spatial context'. The evaluation of the constraint 'M5 Respect spatial context' is achieved by intersecting the generalized geometry $geom_G$ of a polygon or a group of polygons with the context geometry $geom_{Con}$, which represents the spatial context of a polygon or group of polygons. In other words, it defines the map space that can be used to establish a generalization of the corresponding polygon or group of polygons without creating a new conflict as a side-effect. Such a geometry can be calculated by using buffering techniques (Boffet and Rocca Serra 2001) while respecting the minimal distance between two polygons – cf. Figure 6.3.

The constraint is satisfied if the generalized geometry $geom_G$ lies completely within the context geometry $geom_{Con}$ – see also Table 6.2.

Figure 6.3: Examples of context geometries for individual polygons (left figure) and a group of polygons (right figure). Context geometries are derived from buffers of the outline of adjacent polygons at the minimal distance between two polygons. The polygons that are generalized are represented by solid contours in dark gray while their adjacent polygons are represented by solid contours in mid-gray . Dashed lines show the derived context geometries.

Evaluate 'M6 Object separation'. The minimal distance between polygons is defined by the shortest distance d_{Obj} that is found between any pair of polygons (O_i, O_j) of the same group – cf. Table 6.2. According to (SSC 2002) the distance between polygons should not fall below 0.6 mm.

Evaluate 'M7 Number of categories'. Due to its strong dependency on target scale, map purpose, map theme etc. an optimal number of categories can not be defined *a priori* for arbitrary scales. Thus, its goal value is defined as 'varying' in Table 6.1.

6.4.2 Evaluation of topological constraints

Topological constraints belong to the group of offensive constraints, i.e. every data set must adhere to them. A polygonal subdivision can only have two different states concerning topological consistency, namely a consistent or an inconsistent one. Thus, a boolean value can describe the satisfaction of such a constraint, i.e. a 'TRUE' denotes topological correctness and a 'FALSE' the occurrence of topological errors. Constraint 'T1 Self-intersection' and 'T2 Intersection of different polygons' are evaluated by geometrical operations checking the generalized object geometries for self-intersections and intersections with other polygons, respectively.

6.4.3 Evaluation of structural constraints

As explained above, structural constraints control the change of aesthetic properties of a polygonal subdivision during generalization. While in conventional generalization these constraints are often met routinely by cartographers, "they are often hard to translate into operational terms" (Weibel 1996, p. A.4) for automated generalization. But, it is difficult to interpret the measures used for evaluating structural constraints with respect to the quality of a generalized data set since the structure of a polygonal subdivision is modified during the resolution of metric conflicts. For instance, it would make no sense to reject a solution because of a violated constraint 'S2 Absolute position' if metric constraints are significantly improved in their satisfaction at the same time. Along these lines, Weibel (1996) proposed to initially assume the fulfillment of structural constraints to be the result of all the metric and topological constraints being met. According to the entire generalization process, the satisfaction of these constraints may not only allow comparison

Constraint	Measure
M1 Consecutive vertex distance	$d_{Consec} = min(\sum_{i=1}^{n} \overline{v_i v_{i+1}})$
M2 Outline granularity	
minimal shape width	$width = min(\sum_{i=1}^{n} \overline{v_1 v_n})$
minimal shape height	$height = min(\sum_{i=1}^{n} max(\sum_{j=2}^{n-1} dist(v_j, \overline{v_1 v_n})))$
M3 Distance btw. boundary points	'detect narrow sections' (Bader and Weibel 1997)
M4 Minimum area	'polygon area' (Laser-Scan 1999)
M5 Respect spatial context	$geom_G \cap geom_{Con} = geom_G$
M6 Object separation	$d_{Obj} = min(\sum_{i=1}^{n} \sum_{j=1}^{m} dist(O_i, O_j))$
M7 Number of categories	n_{Categs}

Table 6.2: Measures for evaluating metric constraints.

of different solutions that equally meet all the other constraints, but also, to maintain the overall visual appearance (Weibel 1996). In other words, the preservation of the overall visual appearance is assumed to result from the satisfaction of all structural constraints.

Evaluate 'S1 Shape distortion'. The amount of shape distortion is measured by comparing shape indices that are calculated for both the original and generalized polygon geometry. The used shape indices are the perimeter-area ratio (FRAGSTATS 1994) and the comparison of a polygon shape to a circular shape of the same area (Peter 2001). Generally, however, shape properties are difficult to describe by numeric values even on the level of an individual polygon (Weibel and Dutton 1998).

Evaluate 'S2 Absolute position'. This constraint is evaluated by calculating the relative area overlap of the original and generalized polygon geometries. Other suitable measures would be the vector and areal displacement (McMaster 1986) or the Hausdorff distance (Hangouët 1995).

Evaluate 'S3 Relative configuration'. The description of relative positions of polygons to each other involves the calculation of auxiliary data such as a Delaunay triangulation (Jones et al. 1995, Ruas 1995, Bader and Weibel 1997) or a Minimal Spanning Tree (Regnauld 1998, Bader 2001). The change in relative position results from a quantitative comparison of these geometric structures before and after generalization. Here again, measures such as the vector and areal displacement (McMaster 1986) or the Hausdorff distance (Hangouët 1995) can be applied.

Evaluate 'S4 Size ratios'. Size ratios on both the group and map level are measured through the calculation of relative area values, for instance, the relative area of a category of the total subdivision or the relative area of a polygon of another polygon.

6.4.4 Evaluation of procedural constraints

Procedural constraints are linked to the generalization process itself rather than to properties of a polygonal subdivision. The evaluation of such a constraint helps to guide the generalization process. That is, it supports decision making, for instance, when examining whether two polygons should be merged or alternatively supervised child entities require generalization.

Evaluate 'P1 Illogical results'. The method used for evaluating the constraint 'P1 Illogical results' depends strongly on the kind of result that would be regarded as illogical and subsequently should be prevented. For instance, an illogical neighborhood such as a lake in the sea is detected by comparing the semantics of adjacent polygons.

Evaluate 'P2 Child entity's constraints'. The satisfaction of this constraint is derived from the average satisfaction of all supervised child entities. The goal value of the constraint 'P2 Child entity's constraints' defines a minimum level of satisfaction that any child entity should at least reach.

Evaluate 'P3 Aggregation similarity'. The semantic similarity between two polygons is calculated by a measure proposed by Yaolin et al. (2002b). This performs similarity evaluation based on the classification schema of the underlying categorical data. A goal value can not be specified *a priori* since it depends exclusively on the class hierarchy of the generalized data.

Evaluate 'P4 Equal treatment'. This constraint relies on the consideration of two situations being similar. In automated generalization a situation is characterized by a number of factors, such as the constraint that is violated, the objects involved and the severity of the conflict. Since automated analysis of a situation is never as holistic as a cartographer's view and conflict detection is based on given goal values that are interrelated between different constraints there is no guarantee *per se* that similar situations are treated equally.

6.5 Plans proposed by constraints

Plans are suggested by generalization constraints in order to propose solutions that can help to remedy conflicts that have been detected. Plans can take different forms. On the one hand, if an object does not satisfy a certain constraint this constraint may propose plans that are able to diminish its violation with regard to that object. On the other hand, a constraint may recommend avoiding plans for an object's generalization that are known to degrade its satisfaction (Ruas 1999). In general, plans denote generalization algorithms and respective parameters that could be applied to an object in order to improve the satisfaction of a constraint (Beard 1991, Duchêne et al. 2001a). A constraint (e.g. a topological constraint) may also suggest rejecting a generalized data set due to its violation.

Plans are always proposed in consideration of a concrete situation, that is, a constraint is expected to recommend different plans for different situations. This selection of plans also allows generalization strategies to be implemented. For instance, in order to preserve or even emphasize polygons of a specific category, elimination algorithms are avoided as a means for their generalization.

6.5.1 Plans of metric constraints

Metric constraints are the driving force of map generalization, that is, from their violation originate the need for map generalization and the subsequent transformations related to them accomplish most of the adaptation to the target scale. While the relationship of metric constraints to generalization operations can be established on a general level, the concrete application of a plan always depends on both those algorithms available in a generalization system and the specific situation (cf. section 7.2.2). Hence, Table 6.3 provides an overview of metric constraints and attached plans that are defined on the level of generalization operations.

	map	group		group/polygon		polygon		line	
	Reclassification	Aggregation	Typification	Displacement	Exaggeration	Elimination	Enlargement	Simplification	Smoothing
M1 Consecutive vertex distance	+	~	~		~	+	+	++	++
M2 Outline granularity	+	~	~		~	+	+	++	++
M3 Distance btw. boundary points	+	~	~		++	+	+	+	+
M4 Minimal area	+	++	++		+	++	++	~	~
M5 Respect spatial context	+	++	++	++	++	+	-	~	~
M6 Object separation	+	++	++	++	++	+	-	~	~
M7 Number of categories	++								

Table 6.3: Metric constraints and plans that are listed on the level of generalization operations. See explanations in the text.

Table 6.3 shows five different types of relations between metric constraints and generalization operations, namely a *directly positive* ('++'), *indirectly positive* ('+'), *indirectly negative* ('-'), *indefinite* ('∼') and *no* (' ') relation. This distinction is based on the foreseeable influence of a generalization operation on the satisfaction of constraints with respect to a certain object.

Directly positive relation ('++'). A *directly positive relation* between a constraint and a generalization operation occurs if this operation is able to improve the corresponding constraint's satisfaction with respect to an object. For instance, the constraint 'M6 Object separation' may apply a displacement or aggregation algorithm to a group of polygons in order to improve its satisfaction according to this group – cf. Table 6.3 and Figure 6.4b&c .

Indirectly positive relation ('+'). An *indirectly positive relation* concerns situations where a constraint's satisfaction with respect to a certain object is improved through a generalization operation triggered by another constraint imposed on that object or one of its child entities. Thus, the generalization operation is not directly related to the property improved as a side-effect. For instance, since an elimination operation removes entire polygons from the subdivision, on the one hand, the removed object needs no longer to meet any constraints. On the other hand, satisfaction of constraints (e.g. 'M6 Object separation') may increase for those groups of polygons to which the eliminated polygon belonged – see Figure 6.4d. Another example is the reclassification operation, i.e. a reduction of the number of categories represented. This semantic transformation usually implies also a reduction in the number of geometric conflicts within a polygonal subdivision (Spiess 1990, Galanda and Weibel 2002b).

Indirectly negative relation ('-'). An *indirectly negative relation* is characterized by the fact that a constraint's satisfaction is deteriorated as a side-effect. For instance, the enlargement of an

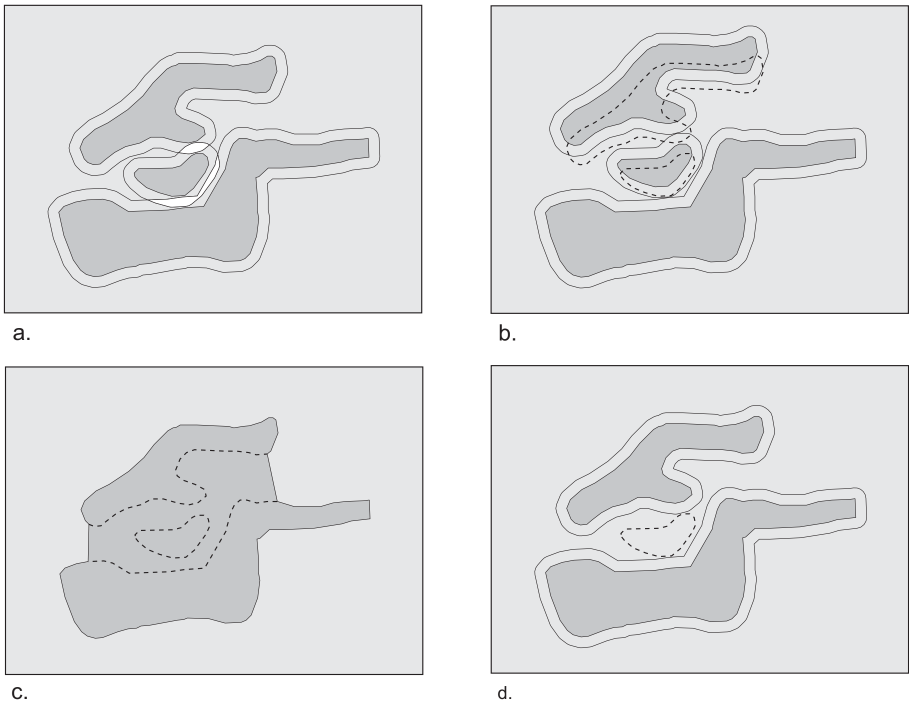


Figure 6.4: Improving the satisfaction of the constraint 'M6 Object separation'. **a.** Original polygons and their buffered outlines at half of the minimal separation distance. Hence, overlaps (white fill color) of buffers mark conflicts. **b.&c.** Solution of the proximity conflict by a displacement and an aggregation algorithm, respectively (directly positive relation). **d.** The conflict is solved indirectly through the elimination of the middle polygon which was triggered by a constraint of the respective child entity(indirectly positive relation). The original geometry of modified polygons is displayed by dashed lines.

individual polygon usually implies an aggravation of the satisfaction of the constraint 'M6 Object separation' with respect to the group the enlarged polygon belongs to.

Indefinite relation ('∼'). Some generalization operations have neither a generally positive nor negative influence on the satisfaction of constraints. Such relations are classified as *indefinite relations*. For instance, there exists no trend as to how the satisfaction of the constraint 'M2 Outline granularity' changes subsequently to an exaggeration operator.

No relation (' '). A relation between a metric constraint and generalization operation is of type *no relation* if the satisfaction of a metric constraint is not influenced by applying the generalization operation. For instance, a displacement operation does not affect the satisfaction of the constraint 'M1 Consecutive vertex distance' since the polygon geometry is translated but its shape remains unchanged.

Figure 6.4 demonstrates three possible solutions for satisfying a violated constraint 'M6 Object separation'. As the generalization of an object is controlled by various constraints the satisfaction of this constraint affects other constraints, too. Thus, the best solution is the situation that best compromises the satisfaction of all constraints attached to an object. For instance, the solution achieved by a displacement operation represented in Figure 6.4b emphasizes the occurrence of three disjoint, independent polygons. This solution is probably the best solution if there is enough space to displace the polygons without creating new proximity conflicts with other polygons and if all the supervised child agents satisfy the constraint 'M4 Minimal area'. The aggregation of disjoint polygons (Figure 6.4c) is restricted in such a way that the constraint 'P3 Aggregation similarity' must be satisfied, that is, the polygons belong to the same category or to semantically similar categories. The aggregation operation additionally allows satisfaction of the constraint 'M5 Respect spatial context' on the group level and the constraint 'M4 Minimal area' on the level of individual polygons (cf. Table 6.3). Figure 6.4d shows the result of an elimination operation. Since the polygon in the middle is removed all conflicts related to this polygon are solved automatically as side-effect (cf. Table 6.3). As long as only metric constraints are considered this solution is easy to establish and perfect but in fact structural constraints (e.g. 'S3 Relative positions' or 'S4 Size ratios') are violated and so this solution is considered as suboptimal.

While metric constraints trigger semantic and geometric transformations of data sets structural and topological constraints control mainly the acceptance of these changes. Plans proposed by these constraints and procedural constraints are discussed in the following section.

6.5.2 Plans of other constraints

Topological constraints. Since a valid state of a polygonal subdivision must not include any self-intersection or intersections between polygon boundaries, every violation of a topological constraint leads stringently to a rejection of a generalized data set. Generalization algorithms may observe such constraints explicitly in order to ensure topological correct solutions. For instance, Edwardes et al. (1998), de Berg et al. (1998) and Saalfeld (1999) proposed such algorithms for the simplification of polygon boundaries. If non-topologically aware generalization algorithms are used, however, an *a posteriori* topological check may establish whether the topological correctness has been maintained.

Structural constraints. Structural constraints belonging to the group of defensive constraints, rather control the maintenance of the spatial and semantic structure than initiate modifications of a polygonal subdivision. It is recommended that their violation stays within certain limits to avoid visually unbalanced results and unnatural size relations (Weibel and Dutton 1998). The identification of meaningful relations between the violation of structural constraints and the necessity of generalization, that is, to meet other constraints, is difficult. Generally, every structural constraint is able to reject a solution provided by the generalization process if its violations exceeds a predefined limit. In practice, structural constraints are mainly used to compare different solutions that equally meet other, mainly metric, constraints (cf. section 6.4.3).

Procedural constraints. Procedural constraints do not have the general characteristic of what type of plans are to be proposed if they are violated. While the constraint 'P4 Equal treatment', that designates a basic principle of generalization, does not suggest any plans the constraint 'P1 Illogical results' is – as topological constraints – able to reject a generalized data set if it is violated. As mentioned above, the constraint 'P3 Aggregation similarity' evaluates if the aggregation of two polygons put forward by another constraint is feasible. That is, if the calculated semantic similarity exceeds the required goal value this plan is triggered, otherwise it is refused. The constraint 'P2 Child entity's satisfaction' is essential in agent-based generalization as it triggers and controls the generalization of supervised child entities. In case of dissatisfaction it can propose triggering the autonomous generalization of a child entity that does not meet all its constraints. For instance, the generalization of polygons is always controlled by a parent entity (cf. chapter 4). That means, a parent entity is responsible to satisfy not only its own constraints but also its child entities.

6.6 Importance of constraints

In practice, constraint based generalization is characterized by the fact that every cartographic object has to meet several constraints. Hence, solutions must compromise between different, and most likely even contradicting constraints. For accomplishing such a compromise importance values on constraints are needed that define if a constraint must be met or could be relaxed in comparison to other constraints (Beard 1991, Ruas 1999).

Generic importance. Topological constraints generally hold a higher importance than metric constraints. They must be fulfilled in order to obtain a sound polygonal subdivision whilst metric constraints need not to be satisfied *per se*, that is, they can be relaxed if necessary. As explained above, it is difficult to define meaningful relations between the violation of structural constraints and the quality of a generalized data set. Additionally, some concepts such as shape are fuzzy and ill-defined (Weibel and Dutton 1998). Thus, structural constraints are assigned the lowest importance although these constraints are from a cartographic point of view highly significant for a fully satisfactory generalization (Weibel 1996). The procedural constraints 'P3 Aggregation similarity' and 'P4 Equal treatment' are not looked at for defining importance of constraints since they are linked to the generalization process itself rather than to individual objects. While the constraint 'P2 Child entity's constraints' should receive a similar importance to metric constraints, the importance of constraint 'P1 Illogical results' is not taken into account for the generic importance of constraints discussed below since its importance depends – as discussed previously – too much on the kind of illogical result that should be prevented.

Importance is only relevant amongst constraints referring to the same spatial level of polygon generalization, that is, importance of constraints must be defined separately for the line, polygon, group and map level. Table 6.4 lists the generic importance of individual constraints that were derived from a priori knowledge, textbooks and empirical testing at these levels of polygon generalization.

Importance on the line level. The constraint 'M1 Consecutive vertex distance' aims only at removing duplicated points, that is, it is not a prerequisite of proper generalization. Hence, the constraint 'M2 Outline granularity' receives a higher importance than 'M1 Consecutive vertex distance', while the constraint 'T1 Self-intersection' holds the highest importance at the line level.

Importance on the polygon level. On the polygon level the constraint 'M4 Minimal area' takes precedence over all the other constraints with the exception of the topological constraint 'T2 Intersection of different polygons'. As long this metric constraint is unsatisfied it makes no sense to consider other constraints since if a polygon is too small to be perceivable other constraints are only of secondary interest. Empirical testing emphasized that the constraints at the line level controlled by 'P2 Child entity's constraints' commonly affect the entire polygon, while the constraint 'M3 Distance between boundary points' has, rather, a local impact. Thus, the procedural constraint obtains higher importance than the metric constraint. The constraint 'M5 Respect spatial context' receives the second lowest importance since proximity conflicts are better solved at the group level. Finally, the lowest importance is assigned to the structural constraints 'S1 Shape distortion' and 'S2 Absolute position'. The same importance is assigned to all structural constraints since their ranking seems to be reasonable only in consideration of user's needs and preferences.

Importance on the group level. At the group level, topological constraints are again of the highest importance, likewise structural constraints again obtain the lowest importance. A preliminary generalization of the child entities, i.e. polygons, attached to a group level is a prerequisite for a meaningful evaluation of the metric constraints 'M5 Respect spatial context' and 'M6 Object separation'. Hence, both constraints are assigned a lower importance than the constraint 'P2 Child entity's constraints'. While the constraint 'M6 Object separation' concerns distances between ob-

Spatial level	Constraints (ordered from highest to lowest importance)
Line level	T1 Self-intersection
	M2 Outline granularity
	M1 Consecutive vertex distance
Polygon level	T2 Intersection of different polygons
	M4 Minimal area
	P2 Child entity's constraints
	M3 Distance between boundary points
	M5 Respect spatial context
	S1 Shape distortion, S2 Absolute position
Group level	T2 Intersection of different polygons
	P2 Child entity's constraints
	M5 Respect spatial context M6 Object separation
	S3 Relative configuration, S4 Size ratios
Map level	M7 Number of categories, P2 Child entity's constraints
	S4 Size ratios

Table 6.4: Importance of constraints on the line, polygon, group and map level, respectively.

jects of the same group the constraint, 'M5 Respect spatial context' refers to distances between the entire group and the spatial neighborhood. Thus, they are assumed to be interrelated and receive equal importance in the generalization process.

Importance on the map level. On the map level the constraints 'M7 Number of categories' and 'P2 Child entity's constraints' hold the highest priority. While the metric constraint controls the semantic generalization, that is, a reduction of the number of shown categories, the procedural constraint initiates adaptations of the polygons' geometries to the target scale if need be. To achieve an appropriate generalization both processes are equally important. The lowest importance is assigned to the constraint 'S4 Size ratios'.

These generic importance values of constraints for polygon generalization must be adapted specifically to the conducted generalization task, which is characterized by the map purpose, the given kind of categorical data, the users' needs and preferences etc. (Ruas and Plazanet 1996).

6.7 Prioritization of constraints

Priorities of constraints allow procedural knowledge to be considered in the generalization process according to which constraint should be satisfied prior to others, that is, an 'optimal' sequence of constraint satisfaction can be indicated (Regnauld 2001). The priorities discussed below and summarized in Table 6.5 are derived from empirical knowledge and testing. As with, the importance of constraints, priorities are only relevant among constraints that refer to the same level of polygon generalization. Additionally, the only constraints need to be considered are those that propose plans that result in transformations of the data set. Note that, whenever a constraint, such as a topological constraint, demands the rejection of a solution a backtrack to the previous state is imperatively performed without taking into account any other plans.

Spatial level	Constraints (ordered from highest to lowest priority)
Line level	M2 Outline granularity
	M1 Consecutive vertex distance
Polygon level	M4 Minimal area
	P2 Child entity's constraints
	M3 Distance between boundary points
	M5 Respect spatial context
Group level	P2 Child entity's constraints
	M5 Respect spatial context, M6 Object separation
Map level	M7 Number of categories
	P2 Child entity's constraints

Table 6.5: Priorities of constraints on the line, polygon, group and map level, respectively.

Priorities at the line level. On the line level the constraint 'M1 Consecutive vertex distance' receives the highest priority. Its satisfaction helps to speed up the resolution of subsequent conflicts such the constraint 'M2 Outline granularity', by removing redundant vertices from polygon boundaries.

Priorities at the polygon level. The constraint 'M4 Minimal area' receives the highest priority on the polygon level as the satisfaction of this constraint may go along with the satisfaction of the constraints 'M3 Distance between boundary points' and 'P2 Child entity's constraints'. For instance, the enlargement of a polygon may also solve conflicts related to supervised line (child) entities such as an excessive granularity of polygon outlines. Since the procedural constraint controls the generalization of the polygon boundary and the removal of redundant points, respectively, it receives a higher priority than the constraint 'M3 Distance between boundary points'. The resolution of this metric constraint can be significantly speeded up by a reduced number of vertices along the polygon boundary initiated by 'P2 Child entity's constraints'. As an evaluation of the constraint 'M5 Respect spatial context' relies on a completed generalization of the polygon geometry it obtains the lowest priority for the polygon level.

Priorities at the group level. The satisfaction of the constraints attached to supervised child agents may also affect distances between objects. Hence, the constraints 'M5 Respect spatial context' and 'M6 Object separation' receive lower priority than the constraint 'P2 Child entity's constraints'. As already discussed above, both metric constraints are interrelated and thus obtain equal priority.

Priorities at the map level. At the map level the constraint 'M7 Number of categories' receives the highest priority since a reduction of the number of classes represented in the target map also implies a reduction in number of polygons and possible conflicts. Thus, child entities are identified and generalized autonomously, that is, the constraint 'P2 Child entity's constraints' holds a lower priority than the metric constraint 'M7 Number of categories'.

The priorities and severities of the constraints attached to a certain object help to detect the best plan for starting the generalization process of the corresponding object. The identification and subsequent triggering of the best plan allows the iterative generalization process to be sped up since it is hoped that a perfect solution (state), that is, a complete satisfaction of all constraints, is reached earlier using this heuristic. For a detailed discussion on the principle of decision making in the AGENT engine refer to section 4.3 and Regnauld (2001), Barrault et al. (2001), Duchêne and Regnauld (2002).

6.8 Conclusions and outlook

The chapter suggested a preliminary set of constraints that intends to cover the basic requirements of polygon generalization. In continuation of previous research (Weibel 1996, Peter and Weibel 1999a) this set was raised and individual constraints discussed at a conceptual level. The novelty of this work lay in the enhancement of the individual constraints by a method for their evaluation based on goal values and measures, by a list of plans on the level of generalization operations, a priority value and an importance value as a basis of compromising between several competing constraints.

The definition of these important values indicates a certain relevance of constraints for the quality of a generalized data set. The assigned importance values propose to emphasize topological and metric constraints rather than structural constraints. Additionally, structural constraints have been considered to be fuzzy and not established sufficiently for providing suitable methods for their evaluation. Although generalization is only controlled by the compliance of objects to metric, topological and procedural constraints and structural constraints are neglected it is expected that a well readable, generalized polygonal subdivision can be achieved. In doing so, it is risked that essential structures (i.e. configurations of polygons) are lost and logical relations between objects are destroyed. However, the properties of a polygonal subdivision controlled by structural constraints, such as shape, visual balance or relative configuration, often rely on subjective interpretation and are usually perceived differently by map users. So, the integration of these constraints into a comprehensive generalization process seems anyway to be difficult and unclear. Hence, it is proposed to initiate future research that studies, on the one hand, structural constraints and their impact on generalization quality at a conceptual level, in order to be able to consider these constraints in the validation of generalization solutions. On the other hand, this research should focus on the formalization of the concepts that underlie structural constraints, such as shape or visual balance.

Besides the discussion on structural constraints, the following points stood out:

- Plans need further specification, that is, the plans listed on the level of generalization operations must be marked down to algorithms and parameters in consideration of available algorithms, generalization controls (e.g. target scale, map purpose, user's needs etc.) and the situation that has to be solved.
- The generic importance and priority values assigned to constraints at the spatial levels of polygon generalization provide a foundation for any implementation of these constraints. Additional testing may allow insight to be gained answering whether (1) the magnitude of the covered scale change or the specific type of polygonal subdivision (e.g. geology, land cover) influences these generic importance and priority rankings of constraints and (2) a variation of the priority values with respect to the severity of a violated constraint can help to achieve a satisfying solution more efficiently and an increase of generalization quality, respectively.
- Although this set of constraints was set up with respect to our general research goal, i.e. the automation of polygon generalization by means of a MAS, it seems to be a valuable starting point for the implementation of any system for automated polygon generalization.

The next chapter describes the integration of this preliminary set of constraints into the proposed framework (cf. chapter 4). In doing the constraints' implementation, the proposed generalization constraints and their evaluation methods are combined with algorithms for polygon generalization and the AGENT engine into a comprehensive, automated generalization process. The prototype will allow empirically testing. The results of the planned experiments intend to achieve, amongst other things, the evaluation of the proposed constraints and their evaluation methods. Furthermore the goal is to arrive at conclusions on the potential of the proposed constraints not only in isolation from each other but also in interaction with each other for automated polygon generalization.

Chapter 7

Implementation and results

7.1 Introduction

The previous chapters provided a theoretical concept for agent-based polygon generalization (chapter 4), innovative algorithms based on an energy minimizing optimization technique (chapter 5) and a preliminary set of constraints for automated polygon generalization (chapter 6). This chapter intends to subsume the presented concepts and methods to a prototype system for automated polygon generalization. Thus, it focuses on the implementation of a prototype system based on the AGENT engine in LAMPS2 as well as the performance of experiments with real world data, that is, the large scale digital landscape model of Switzerland VECTOR25. The qualitative and quantitative evaluation of the experimental results aims at reaching a conclusion on;

- the potential of the set of constraints proposed in the previous chapter, and
- whether the proposed multi-agent approach is an appropriate method to automate polygon generalization.

7.2 Implementation

The implementation of the prototype establishes an extension of the MAS that was developed by the AGENT consortium for the generalization of road networks and urban settlement areas for polygon generalization. Thus, the implementation uses the GIS LAMPS2 as the development environment.

7.2.1 Class diagram

The implementation of the prototype for automated polygon generalization relies on the generic classes of the AGENT engine of LAMPS2, namely the agent and the constraint classes (Duchêne and Regnauld 2002). The *agent class* holds the attributes and methods which refer to the iterative generalization (or life cycle) of an agent, such as, the `states` attribute, i.e. a description of all states an agent passed through – cf. section 4.3. The *constraint class* possesses attributes and methods that allow its instances to control the generalization of an agent, such as the `goal value` or `importance` (cf. section 6.2.1). Since two methods of the agent and constraint class are similar, their differences are clarified briefly: the method `evaluate_constraints()` of the agent class handles the evaluation of all constraints assigned to an instance of an object (i.e. an agent) while the method `evaluate_constraint()` of the constraint class refers to the evaluation of a single constraint (i.e. to an instance of the constraint class). Likewise, the method `propose_plans()` of the agent class suggests plans for the improvement of an agent that are derived from all related constraints, while the method `propose_plans()` of the constraint class proposes plans with respect to a single constraint and a single agent. The method `supervise_child_agents()` of the agent class deals with the supervision of child agents and their generalization. With the help of this method, for instance, a map agent controls the generalization of group agents and instances of the group agent class manage the generalization of attached polygon agents. The association between the agent and constraint class retains a cardinality of [1] to [1...*], that is, every agent relates to one or several constraints. Instances of these two classes do not exist. They only provide generic attributes and methods for subclasses that inherit from the agent or constraint class such as `map agent` or `constraint_A`.

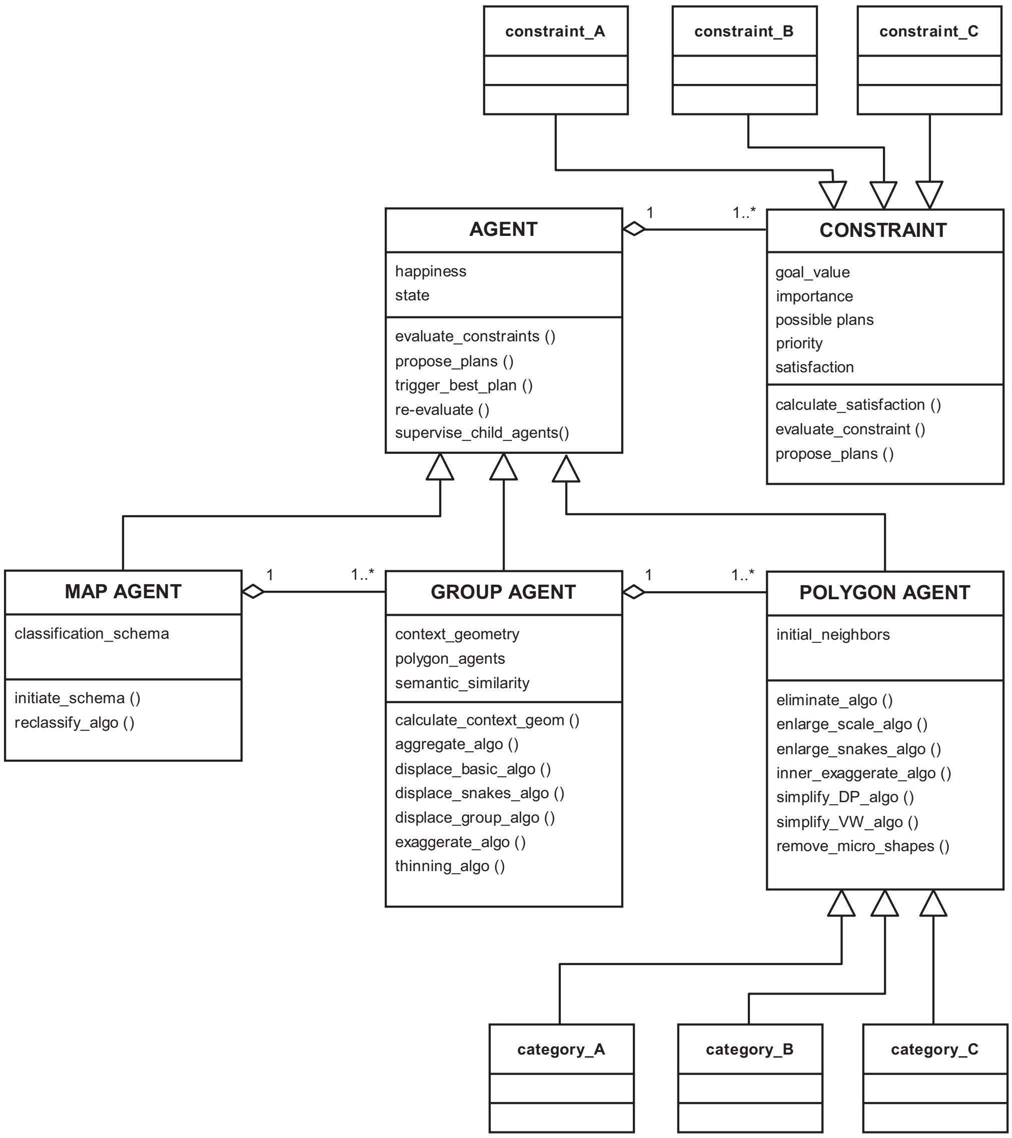


Figure 7.1: Simplified UML class diagram of the implemented prototype.

The following specializations of the agent class refer to the spatial levels of polygon generalization:

- *Map agent class.* The method `initialize_schema()` of the map agent class is responsible for the initialization of the classification schema that underlies a polygonal subdivision. Its outcome is stored as the attribute `classification_schema` on the map agent class. This method and attribute are subsequently used for the semantic generalization (`reclassify_algo()`) of categories performed at the map level. The relationship between an instance of the map agent class (map agent) and instances of the group agent class (group agents) is an aggregation, that is, a map agent is always composed of several group agents.

- *Group agent class.* The group agent class stores the geometry of the group's spatial context, `context_geometry` (cf. 6.4.1), a list of supervised polygon agents (`polygon agents`) and the level of semantic similarity between all supervised child agents (`semantic similarity`) as attributes. The methods of this class refer to the derivation of the context geometry (`calculate_context_geometry()`) as well as several generalization algorithms (`*_algo()`). Again, the relationship between the group agent class and the polygon agent class is an aggregation.
- *Polygon agent class.* While the only attribute of the polygon agent class stores the initial neighbors of an instance (`initial_neighbors`) all its methods refer exclusively to generalization algorithms. The classes of categories that are represented in a polygonal subdivision, then, inherit from the polygon agent class. In other words, every polygon of a polygonal subdivision becomes an instance of the polygon agent class, that is, a polygon agent.

Although a line agent class was proposed in chapter 4 it is not presently implemented in this first prototype due to time constraints of the PhD project. Thus, the key functionality of line agents, that is, the generalization of polygon outlines, is attached to the polygon agent. The evaluation of the experiments' results may indicate whether this functionality needs to be delegated to a proper line agent class or can be satisfactorily handled by polygon agents – cf. sections 7.4, 7.5 and 7.6. Furthermore, a method for the creation of group agents at run time is missing since this process was substituted by an interactive process – see the corresponding discussion in section 7.3.1.

7.2.2 Implementation of constraints

The previous chapter discussed not only characteristics of relevant constraints for polygon generalization but also provided the basis of the implementation of constraints by examining methods for the evaluation of constraints, priorities and importance among constraints and cartographic operations and their influence on the satisfaction of a constraint. In fact, the prototype incorporates all the constraints suggested in chapter 6 with the exception of the following constraints:

- The constraint 'M5 Respect spatial context' is not included due to limitations in its preliminary implementation and missing algorithms to resolve corresponding conflicts.
- The constraint 'P4 Equal treatment' is not considered since further research is required with respect to its formalization (cf. section 6.4.4).

The implementation of the constraints' goal, importance and priority values and measures follows the approach proposed in chapter 6, while the evaluation methods and plans used require further specification with respect to the prototype system.

Evaluation methods determine the satisfaction of a constraint and, thus, describe the compliance of an agent to a constraint by the severity value that ranges from 'very bad' to 'perfect' – for more details see section 6.2.1. In order to translate the detected discrepancy between the measure's output and goal value into one of the 5 levels of severity an empirical approach was chosen. First, the deviation of a goal value was detected by visual examination of test cases that would still allow an acceptable solution even if a constraint is not met completely. For instance, the constraint 'M4 Minimal area' is considered as satisfied whenever the size of a polygon reaches at least 95% of the goal value, that is, the required minimal area. The remaining range of values was then divided into 4 equal classes (Ruas 1999) and later adapted according to the results of the empirical experiments. Table 7.1 shows the levels of severity of the constraint 'M4 Minimal area' with respect to the compliance of an object to the goal value. Furthermore, the severity classification and scores of all the implemented constraints can be found in appendix B.

current value / goal value	severity
< 25 %	1 ('very bad')
25 – < 50 %	2 ('bad')
50 – < 75 %	3 ('moderate')
75 – <= 95 %	4 ('good')
> 95%	5 ('perfect')

Table 7.1: Severity levels derived from the evaluation methods of the constraint 'M4 Minimal area'.

The plans of individual constraints are presented in the following paragraphs. Note that several generalization operations often exhibit the potential to increase the satisfaction of a constraint. For instance, the compliance of a group agent to the constraint 'M6 Object separation' can be improved by either an aggregation, typification, displacement or exaggeration operation – see Table 7.2. Strategies for the automated proposal of plans are discussed at the end of this section.

Plans of metric constraints. Metric constraints and their violation, respectively, are the main trigger of map generalization. The established relationships between metric constraints and plans, at the level of generalization operations (cf. Table 7.2) and algorithms, owe not only to the discussion in section 6.5.1 but also depend on the availability of corresponding algorithms in the prototype system.

		map	group		group/polygon		polygon		
	Generalization operation / Metric constraint	Reclassification	Aggregation	Typification	Displacement	Exaggeration	Elimination	Enlargement	Simplification
map	M7 Number of categories	x							
group	M6 Object separation		x	x	x	x			
polygon	M4 Minimal area		x				x	x	
polygon	M3 Distance btw. boundary points					x			
polygon	M2 Outline granularity								x
polygon	M1 Consecutive vertex distance								x

Table 7.2: Implemented metric constraints and associated generalization operations.

Table 7.3 lists the 15 polygon generalization algorithms of the prototype system. They stem from different sources, namely:

- elementary computational geometry (e.g. vector-based displacement of individual polygons, assignment of polygon to semantically closest adjacent neighbor),

- the GIS LAMPS2 itself (Laser-Scan 2002a, 1999),
- other projects on map generalization such as Visvalingam and Whyatt (1993) or Bader and Weibel (1997), and
- previous own research on snakes-based algorithms for polygon generalization (cf. chapter 5).

The typification operator is substituted here by a simple filtering algorithm that controls the elimination and selection, respectively, of polygons based on their size and semantic importance while the subsequent enlargement and displacement is delegated to the individual polygon agents. The algorithm 'Remove micro shapes' eliminates micro shapes along a polygon boundary whose height and width fall below a pre-defined threshold. Micro shapes are identified by adapting the concept proposed by Wang and Müller (1998) for the characterization and simplification of polylines – see also section 6.4.1.

General. Operation	Algorithm
Reclassification	Change category and merge with adjacent polygons of the same category
Aggregation	Enhanced convex hull (Laser-Scan 1999)
Typification	Simple filtering by size and semantics of polygons and independent generalization of polygon agents
Displacement	Snakes-based displacement (chapter 5)
	Vector-based displacement of individual polygons
	Vector-based displacement of a group of polygons
Exaggeration	Snakes-based exaggeration (chapter 5)
	Widen narrow polygon (Bader and Weibel 1997)
Elimination	Assignment of polygon to semantically closest adjacent neighbor
	Divide up polygon among neighbors (Bader and Weibel 1997)
Enlargement	Simple scaling algorithm (Laser-Scan 1999)
	Snakes-based enlargement (chapter 5)
Simplification	Enhanced Douglas-Peucker algorithm (Laser-Scan 1999, Douglas and Peucker 1973)
	Visvalingam-Whyatt algorithm (Visvalingam and Wyhatt 1993)
	Remove micro shapes (modified after Wang and Müller 1998)

Table 7.3: Overview of algorithms implemented in the prototype system.

Plans of other constraints. Table 7.4 provides an overview of the other constraints implemented in the prototype.

Topological constraints are dealt with implicitly in the generalization process since any violation of the constraints 'T1 Self- intersections' or 'T2 Any intersections' prevents the creation of a valid polygonal subdivision within LAMPS2. In other words, every state that violates such a constraint is imperatively considered as worse and backtracked to the previous state.

Structural constraints define additional qualities that a generalized data set should adhere to if all metric constraints are equally satisfied. In the preliminary prototype structural constraints are implemented but not considered in the decision making (e.g. the comparison of states) since additional research is needed to link their satisfaction to the quality of generalization (cf. section

6.4.3). That is, their satisfaction is currently provided as additional information on every state of an agent. Hence, their evolution can be traced and analyzed, although they are not considered in proposal and evolution of plans.

From the *procedural constraints*, the constraint ‘P1 Illogical results’, ‘P2 Child entity’s happiness’ and ‘P3 Aggregation similarity’ are implemented in the prototype system. It is considered as illogical – with respect to the used test data – if a polygon becomes or is no longer an island polygon of another polygon due to generalization, the constraint ‘P1 Illogical results’ is used at the polygon level to prevent such a modification[1]. In other words, its violation leads to an imperative backtrack. While the constraint ‘P2 Child entity’s happiness’ suggests itself a plan, that is, the generalization of supervised child agents, the constraint ‘P3 Aggregation similarity’, if necessary, validates a plan of another constraint that would lead to the aggregation of agents. Subsequently, this constraint decides according to semantic analysis whether the plan should be triggered or refused.

Agent type	Constraint	Consideration in decision-making
Group agent	T2 Intersection of different polygons (topological)	yes, implicitly
	P2 Child entity's constraints (procedural)	yes
	P3 Aggregation similarity (procedural)	yes
	S4 Size ratios (structural)	no, tracing only
	S3 Relative configuration (structural)	no, tracing only
Polygon agent	T1 Self-intersection (topological)	yes, implicitly
	T2 Intersection of different polygons (topological)	yes, implicitly
	S1 Shape distortion (structural)	no, tracing only
	S2 Absolute position (structural)	no, tracing only
	P1 Illogical results (procedural)	yes

Table 7.4: Topological, procedural and structural constraints implemented in the prototype system.

Proposal of plans. The prototype implements two different strategies for the proposal of plans – see also the discussion of an agent’s life cycle in section 4.3. On the one hand, the framework allows a trial and error approach to be pursued , that is, to trigger different operations and compare accomplished results according to their impact on constraints’ satisfaction. On the other hand, the severity of a constraint, as well as spatial and semantic analysis of a conflict and its context, may indicate that a certain operation is more suited than others to resolve a conflict. That is, the results of spatial and semantic analysis influence the choice of operations. For instance, a conflict with the constraint ‘M4 Minimal area’ can be solved at the level of an individual polygon either by enlarging or eliminating the polygon. Figure 7.2 shows the decision tree for the selection of either an enlargement or elimination operation with respect to such conflicts. The decision making

[1] Of course, a topological constraint that observes the topological relation between constraints can be used in the same way. As the change of polygon containment may be appropriate with respect to other types of polygonal data, the procedural constraint is used to maintain the neighborhood relation.

depends on two criteria, that is, the severity of the constraint and the semantic importance of the polygon. A polygon is considered to be of high semantic importance if it belongs to a category that covers less than 3% of the area of the entire polygonal subdivision. While procedural knowledge incorporated in this decision tree is derived from empirical knowledge and testing on the test data used in this work, the threshold could naturally be changed to better reflect the requirements of other types of data and study areas. Also, it would be possible to incorporate pre-existing knowledge about the semantic importance of relevant categories (e.g. if certain categories were specifically coded to be important).

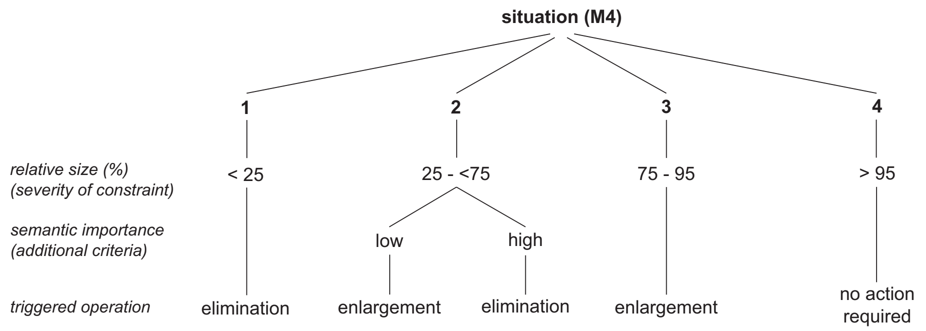


Figure 7.2: Decision tree of the constraint 'M4 Minimal area' for the selection of a generalization operation. See text for explanations.

Besides testing the success of different operations, it is often reasonable to also try different algorithms that implement the same generalization operation but exhibit different qualities with respect to map generalization. For instance, an enlargement operation can be accomplished by a snakes-based algorithm or a simple scaling algorithm – cf. Table 7.3. As highlighted in section 5.4.2, a snakes-based algorithm ensures equal enlargement in all directions but may produce unsatisfactory results for concave inlets, while a simple scaling algorithm leads to shape distortion but is more robust according to different polygon shapes. In running both algorithms, generalization can benefit from the agent-based approach and its capability to calculate and compare different candidate solutions. In other words, the potentially best solution can be found empirically at run time, for the particular data and map at hand.

The plans proposed for polygon generalization in the case of constraint violation are again listed in appendix B.

7.3 Experiments

The experiments and their evaluation intend to achieve a reliable assessment of the potential of a multi-agent system and of the implemented prototype system, respectively, for automated polygon generalization. Therefore, the experiments examine as a test case the generalization of the 'primary surfaces' layer of VECTOR25 to a target scale of 1:50,000 and 1:100,000 (cf. section 7.4 and 7.5). VECTOR25 is the large-scale landscape model of Switzerland, produced by swisstopo, the Swiss national mapping agency. Its content and geometry is based on the Swiss National Map 1:25,000. The 'primary surfaces' layer of VECTOR25 is stored as a polygonal subdivision that distinguishes 28 different categories. Since the content and geometry of the small-scale digital landscape model of Switzerland VECTOR200 is equivalent to a scale of 1:200,000, the automated derivation of a 'primary surfaces' layer from VECTOR25 at intermediate scales such as 1:50,000 and 1:100,000 represents a relevant task in topographic map production as well as in Geographic Information Science.

For conducting the experiments a test area in the South West of Switzerland near the city Nyon and Lake Geneva was chosen. It covers a size of 3 x 3 km^2 and consists of 10 different categories represented by 103 polygons (Figure 7.3). The choice of the test case was motivated by the fact that the polygonal subdivision in this area is very inhomogeneous, that is, it is built of polygons of various sizes and shapes as well as different configurations of polygons. Examples include, groups of disjoint polygons, groups of connected polygons or isolated polygons. Thus, it is fair to assume that this test area is representative for the 'primary surfaces' layer of VECTOR25 in particular and polygonal subdivisions that represent land cover in general.

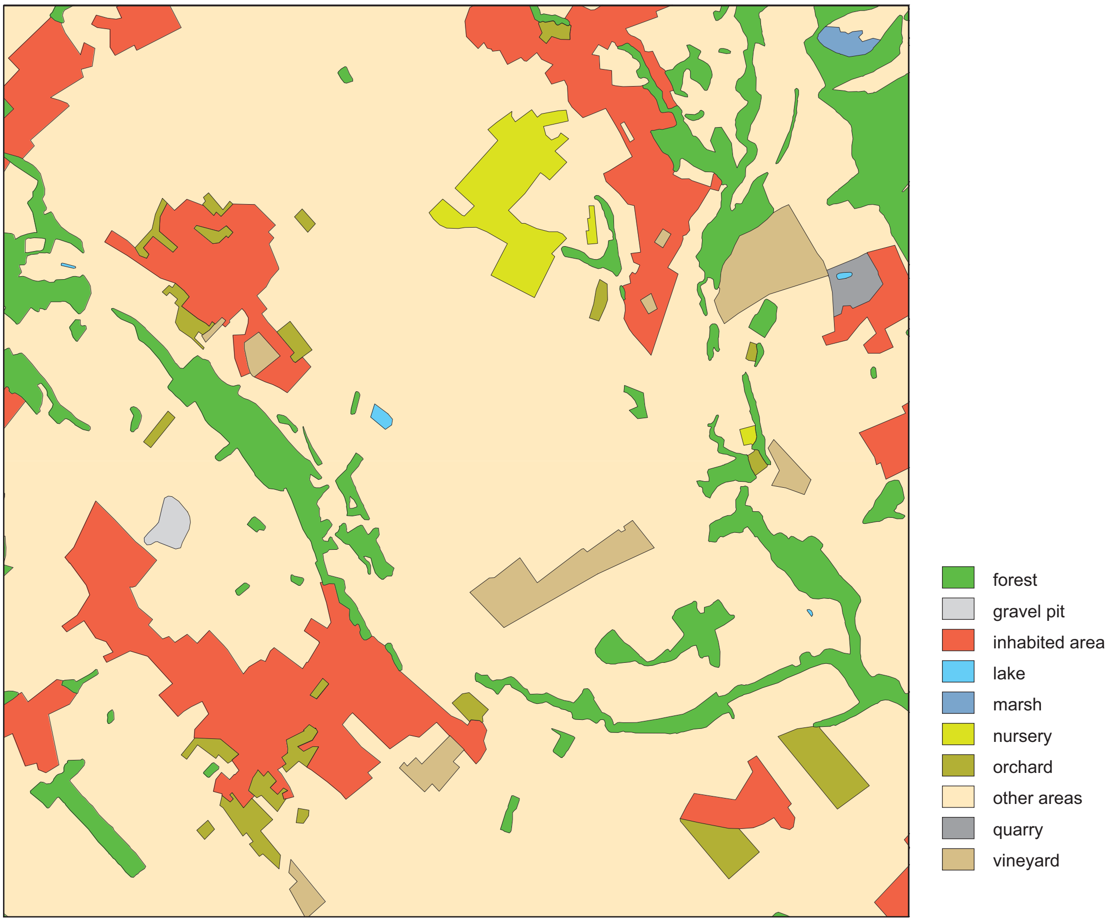


Figure 7.3: The test case that shows an extract of the'primary surfaces' layer of VECTOR25 – figure at scale 1:25,000. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

7.3.1 Generalization process

As outlined in chapter 4, the agent-based generalization of a polygonal subdivision is organized into three main stages, namely a pre-processing; a generalization, and a post processing stage.

Pre-processing. At this stage constraints, are specified, as discussed in chapter 6, and supplementary information on the polygonal subdivision is calculated. That is, for the generalization of VECTOR25, the classification hierarchy that underlies the 'primary surfaces' layer is stored as an attribute of the map agent.

Generalization. At the beginning of the generalization stage the map agent is activated and tries to satisfy its attached constraints. In the presently implemented version of the map agent, discussed here, this is only the metric constraint 'M7 Number of Categories'. For the generalization of VECTOR25, the goal value of this constraint is set in accordance with the Swiss National Map 1:50,000. Since the Swiss National Map at 1:50,000 represents only 8 different land use categories, a reclassification operation is triggered in order to achieve 'perfect' satisfaction of this constraint. In other words, the polygons that belong to the categories 'nursery' and 'orchard' are assigned to the category 'other areas' in consideration of the classification hierarchy of VECTOR25 (Figure 7.5b) and if necessary merged with adjacent neighbors of the same category.

The generalization continues with the assignment of polygons to group agents. A group agent is composed of several polygon agents sharing a common geometric, topological or semantic relation (cf. section 4.3). Possible approaches to the automated identification of group agents are:

- *Cluster agents* based on a spatial distance criteria (i.e. proximity) can be automatically detected by an algorithm based on buffering techniques or a Delaunay triangulation and a spatial distance threshold which can again be derived from the minimal distance between polygons (i.e. the constraint 'M6 Object separation'). Cluster agents based on semantics can be extracted from a polygonal subdivision with the help of a measure of semantic similarity, such as the one proposed by Yaolin et al. (2002b) for categorical coverages with an underlying hierarchical classification schema. Clustering techniques allow the consideration of both spatial and semantic criteria in the grouping process (Jain and Dubes 1988, Han et al. 2001). However, they often require prior information, such as the number of desired clusters or descriptive statistics of the data distribution, that is often missing in map generalization (Anders 2003).
- The identification of *category agents* is straightforward since they relate directly to the categories represented by a polygonal subdivision. For instance, in the reclassified polygon mosaic of the test area a maximum of 8 category agents is possible (Figure 7.5a).
- *Topological partition agents* can be derived by queries on the topology of a polygonal subdivision or additional feature classes. Examples of topological partition agents include all polygons that are connected to a certain polygon (i.e. first order neighbors) or to a linear feature, such as a river or road.
- The assignment of polygon agents to a *geographic partition agent* relies on the availability of geographic data that allow the formation of groups of polygons. Examples of such data are a river or road network. Geographic partitions were so far not studied in the context of polygonal subdivisions.
- The initialization of *pattern agents* usually requires knowledge on the theme represented by a polygonal subdivision, subjective interpretation of a data set through an expert or a well defined set of criteria that describes the characteristics of different patterns. In general, methods of data mining and knowledge discovery (Fayyad et al. 1996, Han and Kamber 2001) may serve as first a starting point for the development of algorithms for the automated detection of pattern agents. A concept for the detection of a specific pattern, that is, the alignment of buildings (i.e. disjoint polygons), was, for instance, proposed by Christophe and Ruas (2002). Other approaches to automated pattern recognition in the context of polygonal subdivisions were, so far, not reported in research.

It is obvious that the difficulty of automated grouping of polygon agents depends on the properties that are significant for a certain type of group agent. In other words, if group agents are specified by well defined spatial and/or semantic criteria this process can be straightforward. Examples include the automated detection of spatial cluster agents, first order or second order neighbors of a polygon (topological partition agents) and category agents. If the structures and criteria underlying group agents are fuzzy and, thus, difficult to formalize the identification of such group agents (e.g. geographic partition agents and pattern agents) often relies on knowledge of the represented theme as well as subjective interpretation. Hence, its automation seems to be more complicated. Since a polygonal subdivision is usually composed of different types of group agents a full automation of the grouping process requires methods for the identification of all potential types of group agents.

The previous paragraph highlighted already possible methods and techniques for the automated detection of group agents within polygonal subdivisions. However, most of these techniques and methods were not yet studied in the context of polygonal subdivisions as well as formalized knowledge of meaningful groups of polygons is missing. Furthermore, preliminary tests emphasized that the automated instantiation of group agents requires significant research effort with respect to methods of spatial and semantic analysis as well as pattern recognition. Along these lines, it was decided to abstain from an automation of the process of building groups of polygon agents. Thus, polygons are interactively assigned to group agents for the sake of the implementation of a comprehensive framework for automated polygon generalization.

That is, polygons are assigned manually after semantic generalization to 26 different group agents by considering their topological relationships, the constraint 'M6 Object separation' and by empirical observation – examples of group agents are shown in Figure 7.4.

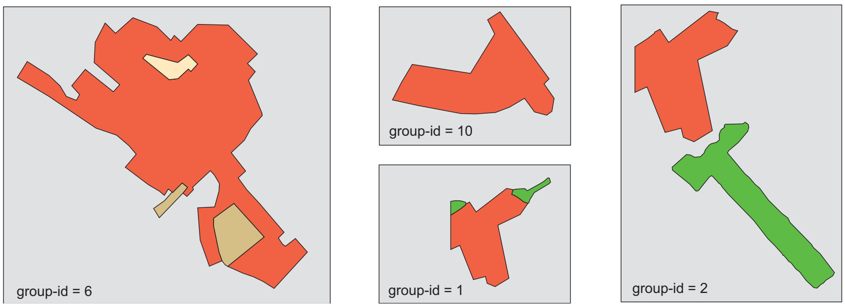


Figure 7.4: Different examples of manually identified group agents in the test area. Note that individual polygon agents may belong to different groups for reasons of efficiency – compare the red polygon in both the right figure and the lower figure in the middle. Group agents are shown at different scales in these illustrations. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

The generalization stage continues by triggering the sequential generalization of the individual group agents that in turn are responsible and trigger the generalization of their supervised polygon agents. The generalization stage ends when all group agents have completed their generalization.

Post processing. As the achieved results are subject to qualitative and quantitative evaluation the post processing stage that performs interactive re-generalization by the user, if necessary, was not run in this experiment. However, output from the agent-based generalization stage could provide useful hints to the post processing stage in the form of: severity values of constraints still not quite satisfied, flags on any agents that do not respect all their associated constraints or indications of plans that failed to resolve certain conflicts.

7.4 Generalized land cover 1:50,000

This first experiment deals with the generalization of the (reclassified) extract of the land cover layer of VECTOR25 to a target scale twice as small, i.e. 1:50,000. This scale reduction implies that the generalized polygonal subdivision has to be represented in a quarter of the original map space. After the grouping of the polygon agents the generalization process is run fully automatically. The generalization has been performed on a laptop with a Pentium4 processor (1.8 GHZ), 512 MB memory and the operating system Windows XP. It took for the target scale of 1:50,000 approximately one hour. Of course, the implementation intends to provide a proof of concept rather than being computationally optimized. The polygonal subdivision, i.e. the extract of the land cover layer of VECTOR25, generalized for a scale 1:50,000 is shown in Figure 7.5c at the target scale 1:50,000 and in Figure 7.6 at a scale 1:25,000.

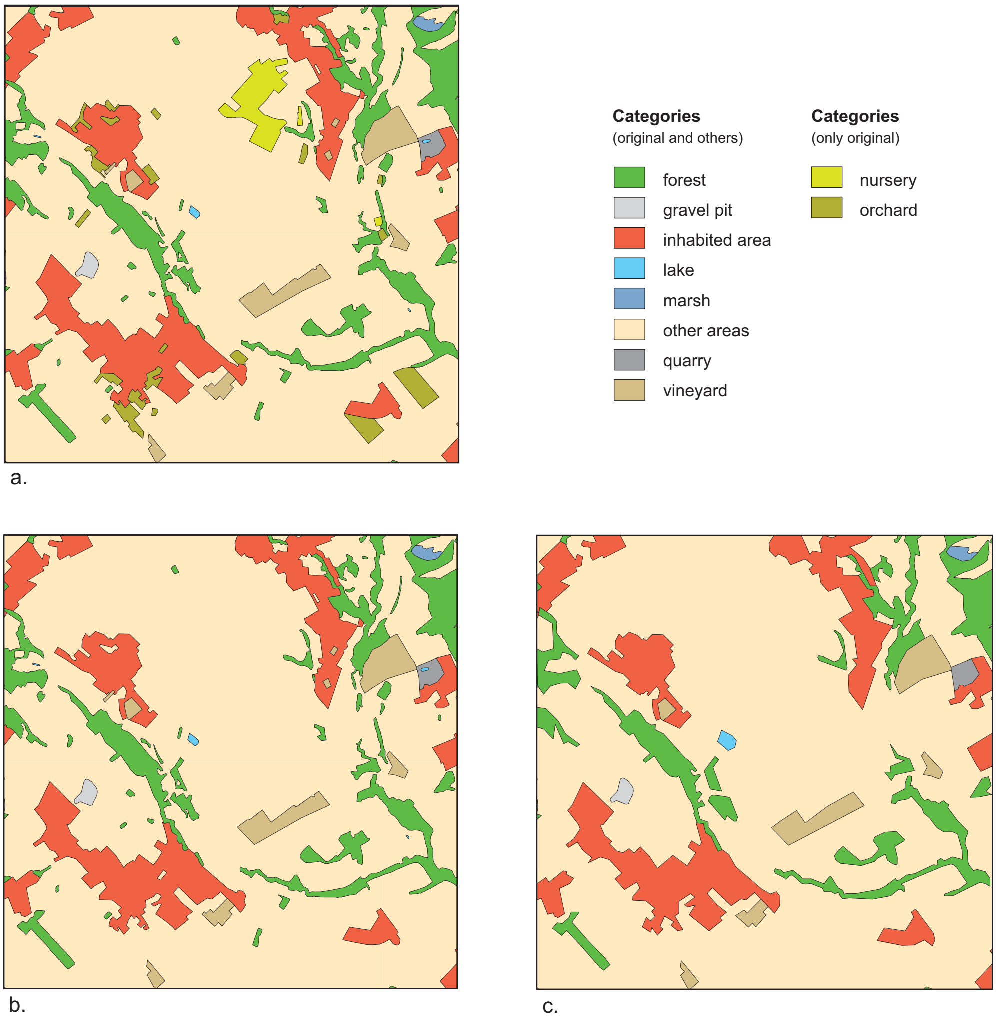


Figure 7.5: a. Original polygonal subdivision. **b.** Polygonal subdivision of the test area reclassified for a scale 1:50,000. **c.** Polygon mosaic of the test area fully automated generalized for a scale of 1:50,000. Figure at scale 1:50,000. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

For the evaluation of the quality of the generalized polygonal subdivision of the test area two methods were chosen:

- a qualitative evaluation of the results, that is, a visual examination of the generalized polygonal subdivision presented in the next subsection (7.4.1), and
- a statistical (quantitative) evaluation of the constraints' satisfaction before and after generalization discussed in section 7.4.2.

Evaluation is the systematic acquisition and assessment of information to provide useful feedback on the results of the conducted experiment (Trochim 2001). Since map generalization in map production is still carried out. to a large part, in an interactive mode only, little research (e.g. McMaster and Veregin (1997), João (1998)) was devoted to the automated evaluation of generalized data. In other words, methods and subsequently software for the evaluation of generalized data have been neglected in generalization research (Weibel and Dutton 1999, AGENT consortium 2000). However, Bard (2003) recently presented a preliminary assessment model of generalization.

7.4.1 Qualitative evaluation of results

Qualitative evaluation examines visually the readability of the generalized polygonal subdivision and the maintenance of geographical meaning in relation to the original data set. That is, the generalized data set should be as close as possible and coherent to the source data. In qualitative evaluation the fact must be considered that there exists no perfect solution to a given generalization task *per se* . So, qualitative evaluation is always influenced by the person who performs it. Besides the original polygon mosaic of the test area (Figure 7.3), Figure 7.6 serves as a basis for qualitative evaluation[2]. In order to avoid confusion from reclassified polygons, this figure shows consciously the reclassified and not the original polygonal subdivision. In doing so, it is hoped that a better qualitative evaluation can be achieved.

In comparison of the original and the generalized polygonal subdivision (Figures 7.5a&c) a significant improvement of readability is noticeable while the geographical meaning is considered to be well preserved by people performing the evaluation. In other words, the main task of map generalization, namely the compromise between ensuring legibility and maintaining the polygonal subdivision's characteristics, seems to be solved satisfactorily across the entire test area. Besides readability some aspects on a more local level need closer attention and discussion. The main issues noticed in qualitative evaluation are summarized in the following paragraphs.

Shape simplification. Several polygons in the generalized polygonal subdivision show spikes along their boundary – compare the labels (S) in Figure 7.6. These spikes give the user an unnatural impression and are considered as an inappropriate generalization of the original polygon or even as an error. In fact, they are side-effects of shape simplification that result from the application of the Viswalingam-Wyhatt (Visvalingam and Whyatt 1993) algorithm for the reduction of outline granularity of polygons. This line simplification algorithm accomplishes a boundary geometry that satisfies well a geometrical criterion, that is, the size of the area spanned by three consecutive vertices, but it does not allow characteristics of the polygon shape to be maintained, emphasized or eliminated. Since the algorithm 'Remove micro shapes' can only remove micro shapes along the polygon boundary it also fails to achieve comprehensive shape simplification in all cases.

The noticed deficits in shape simplification are attributable to the usage of inappropriate algorithms. This failure indicates that new algorithms are required that reduce the outline granularity of a polygon by simplifying the shape of the polygon instead of simplifying the polygon outline. Such approaches that perform shape simplification based on shape characteristics were already established in building generalization (e.g. Rainsford and Mackaness (2002)) and in line generalization for other types of feature classes (e.g. Plazanet (1996), Mustière (1998), Wang and Müller (1998)).

Resolution of proximity conflicts. Proximity conflicts defined by the violation of the metric constraint 'M6 Object separation' are generally solved satisfactorily, that is, the achieved solutions allow good readability. However, it seems that for doing so the displacement operation is exclusively used, although the prototype offers implementations of other generalization operations, such as exaggeration or aggregation as well. For instance, Figure 6.4 in chapter 6 shows, amongst other things, different possible solutions of a proximity conflict. This fact highlights one weakness of the current prototype in relation to methods for spatial and semantic analysis. Such methods could be increasingly used to describe conflict situations in order to propose plans or assign importance values to constraints that take into account the specific situation. For instance, the importance of

[2]In the qualitative evaluation of the experimental results the author was supported by Christian Resch (Cartographer, Lower Austria GIS), Peter Sykora (Cartographer, Institute of Cartography, ETH Zurich) and André M. Winter (Cartographer, Department of Geography, University Innsbruck).

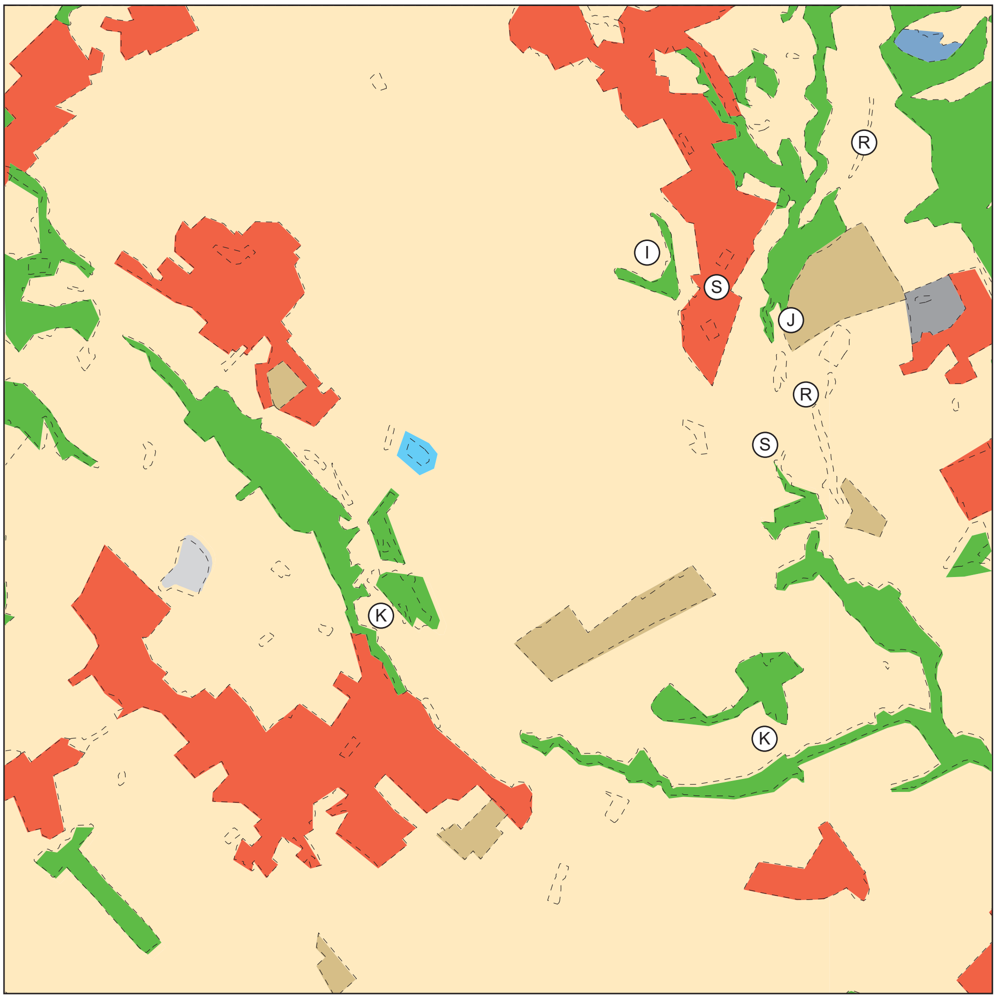


Figure 7.6: Comparison of the polygon mosaic of the test area reclassified (dashed lines) and the polygonal subdivision generalized for a scale of 1:50,000 (solid contours) – figure shown at scale 1:25,000. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

the structural constraints 'S1 Shape distortion' and 'S2 Absolute position' allow the preference of either a displacement operation, that preserves shape rather than positional accuracy of polygons, or an exaggeration operation, that maintains positional accuracy rather than shape of polygons. Due to this deficit plans for the resolution of proximity conflicts are currently applied in a predefined order and importance values of constraints are constant. Thus, conflicts of the same type are solved in the same manner without consideration of the specific structural situation unless the plan with the highest order does not achieve an appropriate solution. In this experiment the displacement operations probably received the highest order of potential plans while the constraint 'S1 Shape distortion' and 'S2 Absolute position' are not considered in decision making given the current settings of the prototype. Hence, proximity conflicts are mainly solved by the displacement operation due to the lack of methods for spatial and semantic analysis, as well as the deficits discussed with respect to structural constraints.

Omission of small polygons. The omission of less important objects is one of the basic operations of map generalization. It achieves a reduction of complexity of a data set and good readability despite a smaller scale, respectively. When looking at the generalized polygonal subdivision (Figure 7.5 and 7.6), a criticism was that too many polygon agents were removed. Note, for instance, the situations marked with the label (R) in Figure 7.6. Given the current settings of the prototype, the elimination of polygon agents is controlled by a spatial criterion, i.e. the size of an object, and a semantic criterion, i.e. the importance of a polygon (cf. section 7.2.2). A straight forward solution to keep more polygon agents in the generalized polygonal subdivision would be to tune the parameters relating to the omission of polygon agents differently . That is, either to decrease the goal value of the constraint 'M4 Minimal area' or to change the thresholds in the constraint's decision tree shown in Figure 7.2. Figure 7.7 portrays, for instance, the influence of the goal value of the constraint 'M4 Minimal area' on the generalization of a group agent. Note that this goal value relates not only to the number of omitted polygon agents but also to the generalization of the complete group agent. While the situation on the right in Figure 7.7 shows the generalized group agent extracted from the generalized test case (i.e. a goal value of 4 mm^2 in map units), the situation on the left represents the same group agent generalized with a smaller goal value for the constraint 'M4 Minimal area' (i.e. a goal value of 2 mm^2 in map units).

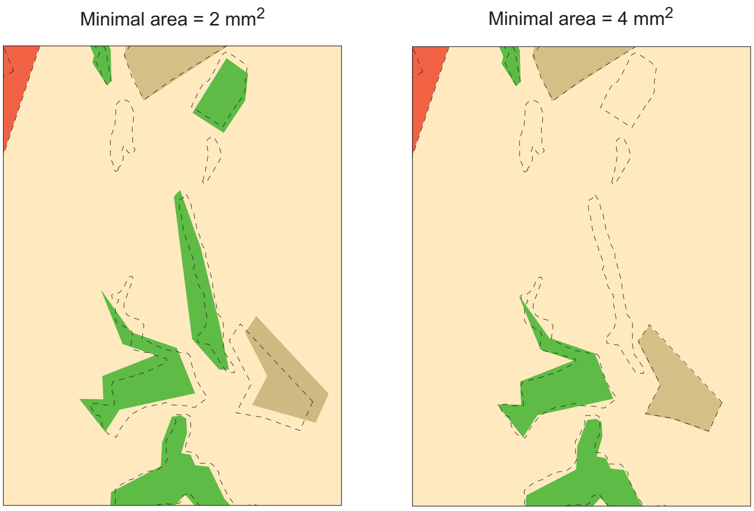


Figure 7.7: Generalization of a group agent with a goal value in map units of 2 mm^2 (left) and 4 mm^2 (right) for the constraint 'M4 Minimal area'. The original solution is represented in dashed lines and the generalization solution by solid contours – figure not at scale. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

The previous paragraph stated already the need for additional methods of spatial and semantic analysis. With respect to the omission of small polygons such methods could help further to obtain better solutions while considering the semantic and spatial context of a polygon agent more thoroughly. Possible methods should tackle, amongst other things, the analysis of whether a polygon is isolated and should therefore be emphasized or the detection whether a polygon belongs to a group of disjoint polygons, and thus the resolution of a size conflict should be delegated to the parent agent. In doing so, aggregation and typification operations could partly replace the independent generalization of polygon agents within groups of disjoint polygons.

Along these lines, the number of omitted polygons could be reduced either through a stronger consideration of methods for spatial and semantic analysis (i.e. a conflict's context) or a changed parameter tuning (Figure 7.7).

Inhomogeneous generalization. Due to the concept of decision making applied in the AGENT engine (Regnauld 2001, Barrault et al. 2001), every agent is updated by the best state (i.e. the best compromise in the satisfaction of attached constraints) ever reached during the generalization process. Of course, this state needs neither to be perfect nor to satisfy all constraints equally well. So the generalization strategy implies that the degree of generalization may appear inhomogeneous across the generalized data set. For instance, the mark (I) in Figure 7.6 points to a polygon where a very narrow part was not widened while mark (J) refers to a similar situation where an exaggeration operation was successfully applied. The marks (K) in Figure 7.6 highlight other examples of inhomogeneous generalization, namely polygons with significantly different outline granularity.

An investigation into the reasons why a polygon agent did not reach a perfect state or does not meet a certain constraint can only be answered by tracing back or reproducing its evolution during iterative generalization[3]. That is, the development of a polygon agent's satisfaction and of the satisfaction of its constraints is analyzed over time. This task can be accomplished by quantitative generalization besides providing some statistical values on the quality of the generalized polygonal subdivision.

7.4.2 Quantitative evaluation of results

In general, quality is defined as fitness for use. Hence, the quality of generalization results can be evaluated by examining if the requirements imposed by scale and users, in the presented framework modelled by constraints, are respected. The satisfaction of constraints is the basis of decision making in the proposed agent-based approach to automated polygon generalization (cf. section 4.3). Thus, an investigation of the evolution of agents' and constraints' satisfaction during the generalization process allows the requirements of quantitative evaluation to be met, namely to represent the quality of the generalized polygon mosaic of the test area by statistical (numerical) values. These values should help at a local level to express the overall quality of the generalized polygonal subdivision of the test area and at a global level help to draw conclusions on the potential of the proposed framework and the implemented prototype. The quantitative evaluation of the achieved results is performed separately for polygon and group agents.

Polygon agents

Quantitative evaluation at the level of polygon agents considers only the polygon agents that are retained in the reclassified representation of the test area (Figure 7.5b). In other words, the 26 reclassified polygons are not taken into in account. Since reclassified polygons do not violate any constraints in the generalized data set they are assumed to have a perfect state. So, their consideration would result in misleading statistics and consequently misleading conclusions on the capacity of the proposed framework and prototype.

The following statistics are based on the metric satisfaction of polygon agents, that is, the rounded average satisfaction of all metric constraints attached to a polygon agent[4]. So, the considered constraints are 'M1 Consecutive vertex distance', 'M2 Outline granularity', 'M3 Distance between boundary points' and 'M4 Minimal area'. Metric satisfaction is chosen since the violation of metric constraints defines the preliminary need of map generalization (cf. section 6.5.1). In other words, if agent-based polygon generalization was applied to a data set that satisfies all metric constraints at the polygon level this data set would be considered to be perfect with respect to the generalization of polygon agents. On the other hand, topological, structural and procedural constraints related to polygon agents always reject solutions if they are violated – see section 6.5.2. Hence, it is obvious that these constraints must be fulfilled *per se* with respect to the polygon

[3]As an example appendix C discusses not only the evolution of a polygon agent during iterative generalization but also shows the system output generated during the generalization of this agent.

[4]The importance of constraints is not considered in the calculation of metric satisfaction since relative importance values such as 'constraint A' is more important than 'constraint B' can not be reasonably translated into numerical weight values for the calculation of a weighted average.

agents of the generalized data set. Along these lines, satisfaction of metric constraints seems to be a legitimate measure to evaluate the success of generalization at the level of polygon agents.

On a scale of 1 ('very bad') to 5 ('perfect') the average metric satisfaction of polygon agents in the test area improved by the agent-based generalization from 2.7 (after reclassification) to 4.5 if the deleted polygon agents are considered as perfect and to 4.1 if only the retained polygon agents are taken into account. Applying the terminology for the description of the satisfaction of constraints (c.f. section 6.2.1), the achieved generalization on the polygon level can be considered as ranging between 'good' and 'perfect'. Table 7.5 presents a cross-tabulation of the rounded metric satisfaction of polygon agents before and after iterative generalization. Only polygon agents retained in the generalized data set are examined. For instance, before generalization 27 polygon agents had a metric satisfaction of 3 ('moderate'). Of these, 9 could not improve at all, 9 improved to a metric satisfaction of 4 ('good') and 9 could even reach a perfect state, i.e. a metric satisfaction of 5.

With respect to metric satisfaction of retained polygon agents, Table 7.5 allows the following conclusions to be drawn:

- 28 of 44 polygon agents (64%) improved their satisfaction while 11 of 44 polygon agents (25%) could not increase their metric satisfaction and 5 (11%) were already perfect.
- 43 of 44 agents (98%) reached at least 'moderate' metric satisfaction (3) and 33 out of 44 (75%) at least 'good' metric satisfaction (4).
- 17 of 44 (39%) polygon agents reached a 'perfect' metric satisfaction while 5 (11%) remained 'perfect', that is, after automated generalization 50% of all agents were 'perfect' with respect to metric constraints.
- None of the polygon agents was degraded by the generalization process[5].

		count of generalized polygon agents (after)					
	metric satisfaction	**1**	**2**	**3**	**4**	**5**	*total*
count of ungeneralized polygon agents (before)	**1**	0					*0*
	2		1	1	1	4	*7*
	3			9	9	9	*27*
	4				1	4	*5*
	5					5	*5*
	total	*0*	*1*	*10*	*11*	*22*	*44*

Table 7.5: A cross-tabulation of the metric satisfaction of polygon agents that are retained in the generalized polygonal subdivision.

In order to identify the reasons why polygon agents did not reach a perfect state it is necessary to go into more detail at the analysis of the constraints' satisfaction, that is, to examine the compliance of polygon agents to the individual constraints.

The Figures 7.8, 7.9, 7.10 and 7.11, respectively, visualize the improvement of satisfaction in relation to metric constraints and polygon agents due to automated generalization. That is, they present the relative frequency of polygon agents at the different levels of severity of a constraint before and after their generalization. Note that bar plots are calculated only for the polygons that are retained in the generalized data set, as a consideration of the deleted polygon agents

[5] The degradation of a polygon agent is excluded by the generalization strategy, that is, if polygon agents can not accomplish an improvement of satisfaction during their independent generalization they remain unchanged and keep their original state, respectively. However, a degradation of a polygon agent can occur if it helps to improve the over-all satisfaction of a parent agent.

would again encourage misleading conclusions. The exception is the bar plot that refers to the constraint 'M4 Minimal area' (Figure 7.8). This shows the relative frequency of all polygons before their generalization (lower bar) and of retained polygons after generalization (upper bar) in order to support the evaluation of the resolution of size conflicts which is mainly achieved through elimination operations and subsequent update of the polygonal subdivision.

M4 Minimal area. As discussed above, the constraint 'M4 Minimal area' receives the highest importance at the polygon level since the compliance of polygon agents to this constraint is essential in order to obtain a readable generalized data set. Due to generalization, the relative number of polygon agents that meet this constraint perfectly was increased from 49% before generalization to 93% after (Figure 7.8). Of course, it is relatively easy to satisfy a constraint if a large number of the conflicts with it are resolved by the elimination of corresponding polygon agents and the subsequent adaptation of the polygon mosaic. Along these lines, compare also the discussion on the omission of small polygons in section 2.3 and in the previous section. 7% of the generalized polygon agents remained at the worst severity. In order to identify the reasons for the failure of the 3 corresponding polygon agents the generalization process was run again interactively for their group agents. The cause of one failure was identified as a bug on the implementation of the elimination algorithm of Bader and Weibel (1997), while the other two polygons were generalized accurately. Thus, the causes could not be reproduced and investigated. That is, further experiments are required to examine the robustness of the prototype in detail.

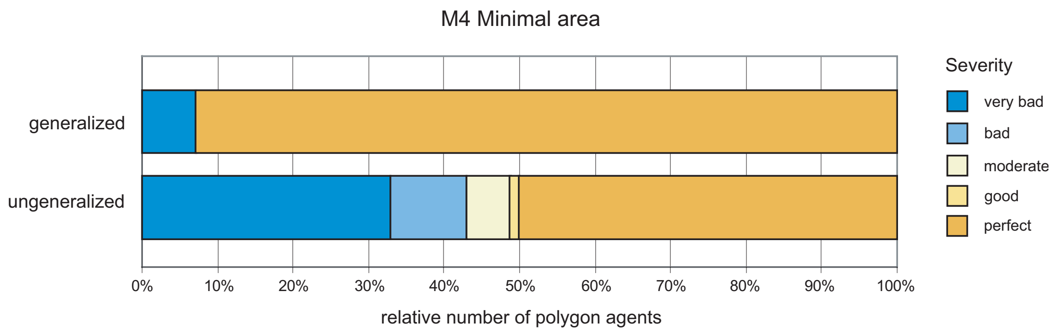


Figure 7.8: Satisfaction of the constraint 'M4 Minimal area' by relative number of un-generalized and generalized polygons.

M1 Consecutive vertex distance. The constraint 'M1 Consecutive vertex distance' aims at the removal of redundant points of a polygon outline (c.f. section 6.3). Every polygon agent tries only once at the beginning of its iterative generalization to meet this constraint. If the constraint is violated a line simplification algorithm, namely the Douglas-Peucker algorithm, is triggered with a conservative simplification tolerance. Visvalingam and Williamson (1995) examined and proved the potential of the Douglas-Peucker algorithm for the deletion of vertices (i.e. conservative coordinate weeding) while the original shape of a geometry is preserved as faithfully as possible. Empirical tests showed that a simplification threshold of 0.1mm in map units for this algorithm ensures a sound compromise between shape maintenance and removal of superfluous points. The severity of this constraint distinguishes only two values, namely a 'perfect' and 'not perfect' satisfaction, due to its underlying concept[6]. Automated generalization established a tripling of the relative number of agents that satisfy the constraint 'M1 Consecutive vertex distance' so that after generalization

[6]In the calculation of metric satisfaction these two values are considered as '5' for a 'perfect' severity and '1' for a 'not perfect' one. Since experiments according to the usage of 5 degrees of severity – as the other constraints do – did not lead to significantly different metric satisfactions a further distinction of severity levels of this constraint 'M1 Consecutive vertex distance' was omitted.

almost two-thirds of the polygon agents respect this constraint – see Figure 7.9. Of course, the application of Douglas-Peucker with a larger simplification parameter could improve the compliance of the polygon agents to this constraint but might also risk a change of polygon shapes as a side-effect. It is more reasonable to control the simplification of the outline granularity of polygon agents by the corresponding constraint, namely 'M2 Outline granularity'.

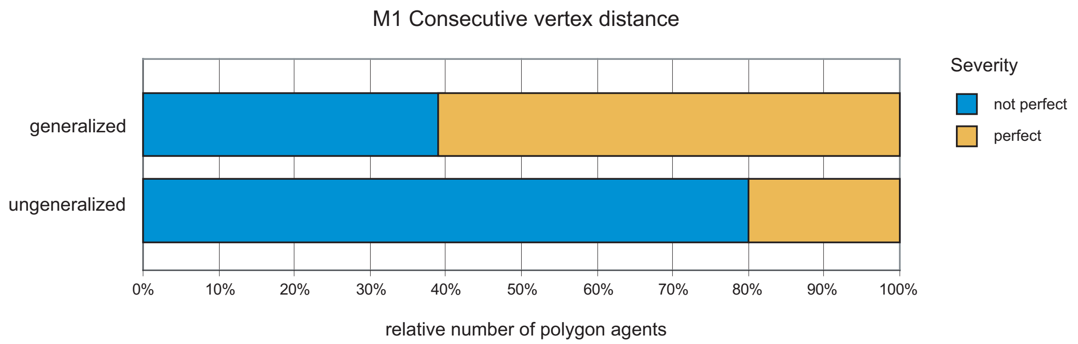


Figure 7.9: Satisfaction of the constraint 'M1 Consecutive vertex distance' by the relative number of retained polygons.

M2 Outline granularity and M3 Distance between boundary points. Approximately two-thirds of all polygon agents respect the constraints 'M2 Outline granularity' and 'M3 Distance between boundary points' in the generalized data set. In other words, generalization achieved a doubling of the polygon agents that satisfy perfectly the constraint 'M2 Outline granularity', i.e. from 32% to 68% (Figure 7.10), and even a tripling of the polygon agents that meet the constraint 'M3 Distance between boundary points', that is, from 18% to 61% (Figure 7.11).

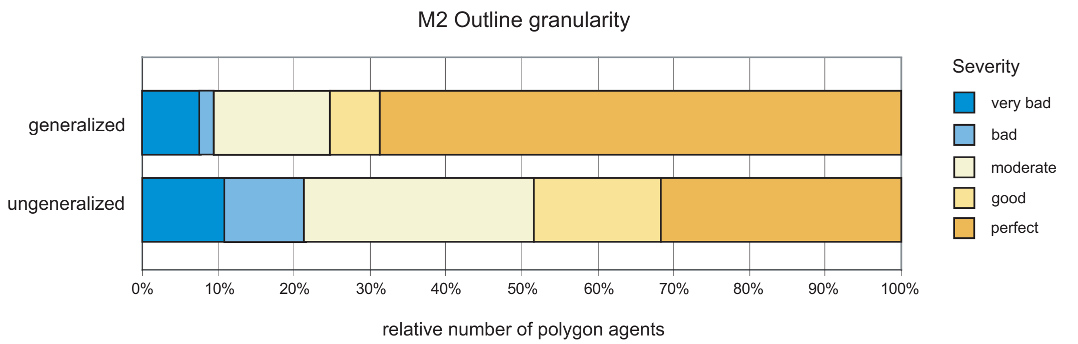


Figure 7.10: Satisfaction of the constraint 'M2 Outline granularity' by the relative number of retained polygons.

In reproducing the generalization process the following reasons were identified why polygon agents do not meet one of these metric constraints:

- The algorithm 'Widen narrow polygon' used for the implementation of the exaggeration operation failed to achieve sufficient improvement in the satisfaction of the constraint 'M3 Distance between boundary points'. In other words, generated solutions were not considered as better than the original state and thus refused – compare the discussion on decision making in an agent's life-cycle in section 4.3. A possible cause is that the algorithm resolves conflicts locally, i.e. at the level of few vertices of the polygon boundary, rather than on a set of consecutive vertices and thus new conflicts may be created as a side-effect of another conflict's resolution.

- The deficits of the chosen shape simplification algorithms were already discussed above in the qualitative evaluation of the generalized polygonal subdivision. In reproducing the generalization process it became again evident that these algorithms miss the capability to solve shape simplification in all situations in such a way that the constraint 'M2 Outline granularity' is met by the corresponding polygon agents.
- Another reason found for dissatisfaction of polygon agents – especially with respect to the constraint 'M3 Distance between boundary points' – was the necessity of polygon agents to compromise between their own needs and those of adjacent polygon agents. Note that given the current settings in the prototype polygon agents are first generalized independently, then all the first order neighbors are adapted to the modified polygon agents and finally the changes are validated on the group level and subsequently the state is accepted or rejected, respectively. Along these lines, the experiments demonstrated that isolated polygon agents achieved a significantly higher average satisfaction of the constraint 'M3 Distance between boundary points', namely 4.63, than polygon agents with more than one adjacent neighbor which hold an average satisfaction of this constraint of 3.85. In contrast, the satisfaction of the constraint 'M2 Outline granularity' seems not to be related to the number of adjacent agents since both groups of polygons obtain a similar value of average satisfaction, namely approximately 3.3.
- Better solutions were rarely rejected because of topological errors. Only the first order neighbors are adapted according to the generalized geometry of a polygon agent. Thus, whenever the second order neighbors were affected, topological errors may demand a rejection of a solution.

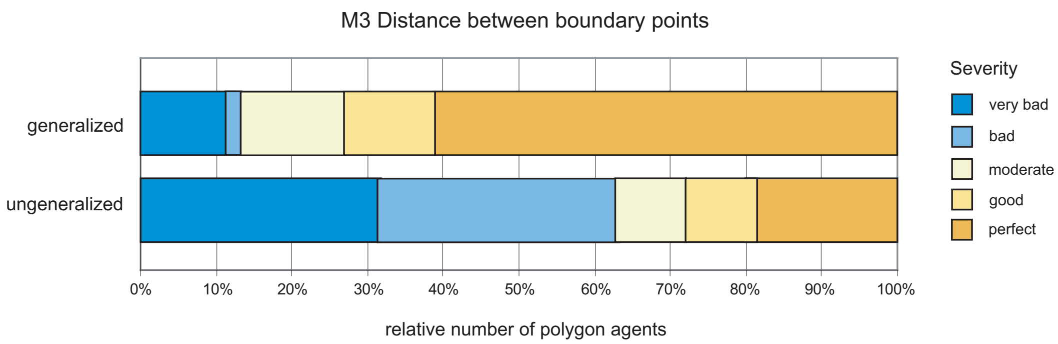


Figure 7.11: Satisfaction of the constraint 'M3 Distance between boundary points' by the relative number of retained polygons.

A discussion of the constraints' satisfaction with respect to the importance of constraints (cf. section 6.6) refers to another principle of map generalization, that is, less important constraints should be relaxed in order to enable a better satisfaction of more important constraints. The constraint 'M4 Minimal Area' is not only – besides the topological constraint 'T2 Intersection of different polygons' – the most important constraint at the polygon level but also the one with the largest portion of 'perfect' agents, namely 93%. As implied by the importance of constraints, more polygon agents satisfy completely the constraint 'M2 Outline granularity' (68% 'perfect agents') than the constraints 'M1 Consecutive vertex distance' (61%)[7] and 'M3 Distance between boundary

[7]At the polygon level the importance of the constraints 'M2 Outline granularity' and 'M1 Consecutive vertex distance' is represented in Table 6.4 by the constraint 'P2 Child entity's constraints' that controls the satisfaction of the constraints attached to line agents. Thus, these constraints receive a lower importance than the constraint

points' (61%). Hence, the constraints with the lowest importance at the polygon level also exhibit the smallest portion of 'perfect' agents. In other words, the importance of constraints at the polygon level is well reflected by the relative number of agents with 'perfect' satisfaction after generalization and considered by the implemented prototype, respectively.

From visual examination of the generalized results the assumption was raised that the shape complexity of polygon agents might correlate with the satisfaction of the constraint 'M2 Outline granularity' and 'M3 Distance between boundary points'. Hence, the shape complexity of every polygon agent was described by a comparison of its shape to a circular object of the same area (Peter 2001). The correlation coefficient of the standardized measure of shape complexity and the satisfaction of the constraint 'M2 Outline granularity' showed no significant correlation (r=0.06). In contrast, the correlation of shape complexity and 'M3 Distance between boundary points' showed a moderate, negative correlation (r=-0.48). That means, the higher the shape complexity the less satisfaction of this constraint can be expected. Of course, the shape complexity of polygons is also correlated with the number of first order neighbors (r=0.63). Thus, the reason for the correlation between shape complexity and the constraint 'M3 Distance between boundary points' may be both due to some weakness of the exaggeration algorithm as well as the necessity to compromise the satisfaction of several adjacent polygon agents.

The generalization of polygon agents is supervised by group agents and, thus, the next step in quantitative evaluation is the examination of the satisfaction of group agents and their associated constraints.

Group agents

Group agents are initiated in the agent-based generalization process after generalization of the map level – see also sections 4.3 and 7.3.1. Hence, the following statistics compare the satisfaction of group agents before generalization (i.e. following their initialization) and after automated generalization. Similarly to metric satisfaction of polygon agents, the need for generalization of group agents is indicated by two constraints, namely the metric constraint 'M6 Object separation' and the procedural constraint 'P2 Child entity's constraints'. Consequently, the basic satisfaction of a group agent is defined as the average satisfaction of these two constraints[8].

From the 26 group agents in the polygonal subdivision reclassified for the target scale (cf. 7.3.1), 21 agents were retained in the generalized polygon mosaic. That is, 5 group agents were not preserved since all their associated polygon agents were removed because of conflicts with the constraint 'M4 Minimal area'. The automated generalization of the test data set achieved an increase of average basic satisfaction of group agents from 2.6 in the reclassified to 4.0 in the generalized polygonal subdivision. If the removed group agents are considered as perfect the average satisfaction of group agents in the generalized data set would reach 4.2.

Table 7.6 describes the improvement of the (integer rounded) basic satisfaction of group agents due to automated generalization, that is, the direct influence of generalization on the group level. The analysis of this cross-tabulation provides evidence of the following facts:

- 17 of 21 group agents (81%) achieved an improvement in their satisfaction while the satisfaction of 4 group agents (19%) remained unchanged.
- All group agents met at least a 'moderate' basic satisfaction and 15 of 21 group agents (71%) even a 'good' basic satisfaction.
- A third of all group agents, i.e. 8 of 21 (38%), reached a 'perfect' basic satisfaction through automated generalization.
- None of the group agents received a de-grading of their satisfaction.

While before generalization none of the group agents hold a 'perfect' basic satisfaction, after

'M4 Minimal area' and a higher importance than the constraint 'M3 Distance between boundary points'.

[8]Similarly to polygon agents, the importance of constraints is not considered when calculating the satisfaction of group agents, due to the difficulty of translating relative importance values into weights of a weighted average.

count of ungeneralized grouo agents (before)	count of generalized group agents (after)					
basic satisfaction	**1**	**2**	**3**	**4**	**5**	*total*
1	0	0	2		1	*3*
2			1	2		*3*
3			3	4	5	*12*
4				1	2	*3*
5						*0*
total	*0*	*0*	*6*	*7*	*8*	*21*

Table 7.6: A cross-tabulation of the (integer rounded) basic satisfaction of group agents that are retained in the generalized polygonal subdivision.

generalization none of them hold a 'very bad' or 'bad' (integer rounded) basic satisfaction. Thus, the quality of the generalization at the group agent is again considered to be encouraging since a significant improvement of basic satisfaction was observed. Similarly to polygon agents, a detailed examination of the improvement of the satisfaction of the individual constraints 'M6 Object separation' and 'P2 Child entity's constraints', that are considered for the basic satisfaction of a group agent, may allow the identification of weaknesses of the implemented prototype.

P2 Child entity's constraints. The discussion of the compliance of group agents to the constraint 'P2 Child entity's constraints' reflects the quality of achieved generalization at the (subordinate) polygon level and its quantitative evaluation. The reason is that this procedural constraint at the group level controls the independent generalization of subordinate polygon agents (cf. section 6.3). Before generalization 81% of the group agents held a 'very bad' and no group agent a 'good' or 'perfect' satisfaction of this constraint, while after generalization a third of all group agents obtained 'perfect' satisfaction with respect to the constraint 'P2 Child entity's constraints' – see also Figure 7.12. However, almost half of the agents remained still at a 'bad' or even 'very bad' level of severity of this constraint. The analysis of why group agents do not respect this constraint refers back to the satisfaction of the supervised polygon agents and its development during generalization, which was already discussed earlier in this chapter.

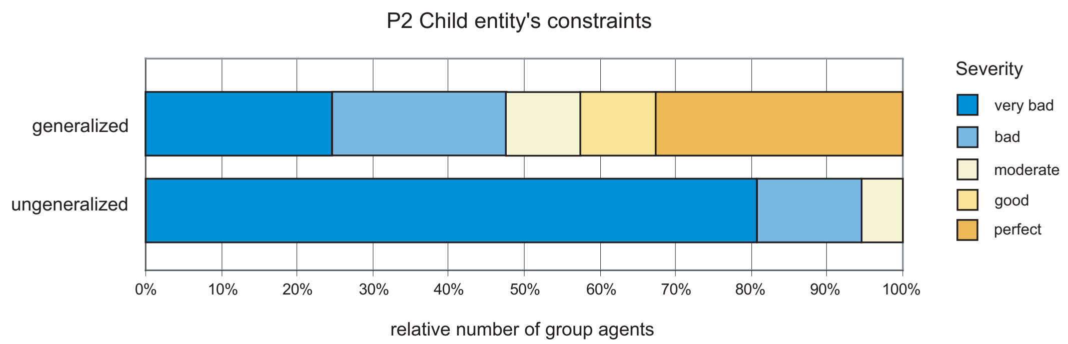


Figure 7.12: Satisfaction of the constraint 'P2 Child entity's constraints' by the relative number of retained group agents.

M6 Object separation. Qualitative generalization pointed out already that proximity conflicts seemed to be resolved well and quantitative statistics of group agents and the satisfaction of the constraint 'M6 Object separation' support this observation. That is, the prototype system for all group agents except one reached a 'perfect' satisfaction of this constraint. For 6 of 7 group agents that did not meet the constraint before generalization, a significant improvement in this constraint's satisfaction was achieved. To be fair, it must be considered that the constraint 'M6 Object separation' is automatically satisfied if a group agent consists only of one polygon. For instance, 4 of the 26 original and 7 of the 21 generalized group agents fall in this category. The only group agent that does not meet this constraint relates to a bug in the update of the polygonal subdivision according to a modified polygon agent, that is, due to the changed geometry of a polygon agent two other polygon agents of the same category become adjacent but are not merged to one polygon. Consequently, the evaluation of this constraint 'M6 Object separation' considers an insignificant distance between polygon agents.

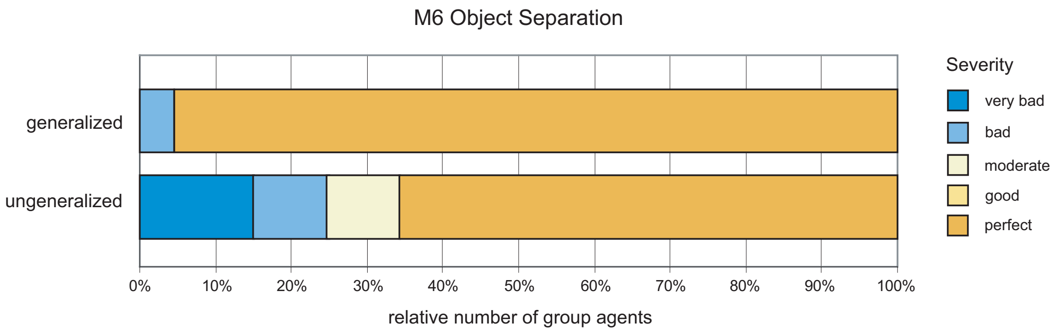


Figure 7.13: Satisfaction of the constraint 'M6 Object separation' by the relative number of retained group agents.

Although the constraint 'P2 Child entity's constraints' exhibits a higher relative importance than the constraint 'M6 Object separation' the satisfaction of the latter constraint is significantly better that the one of the constraint 'P2 Child entity's constraints'. This fact is no direct indicator of the prototype's quality to compromise between the satisfaction of these constraints since, on the one hand, the importance is only relevant if conflicts with respect to both constraints occur in the same group agent. For instance, 7 of 26 group agents (27%) in the original data set embed both types of conflicts. On the other hand, the resolution of proximity conflicts has only an 'indirectly negative relation' to the enlargement operation at the polygon level (cf. Table 6.3). In other words, the satisfaction of the constraint 'M6 Object separation' at the group level is mainly interrelated with the satisfaction of the constraint 'M4 Minimal area' at the polygon level that is again controlled through the constraint 'P2 Child entity's constraints' at the group level. For this constraint 93% of all polygon agents reached a 'perfect' satisfaction while the other 7% failed to do so due to the reasons discussed earlier in this section. Furthermore, it is much easier from an algorithmic point of view to ensure minimal distance between objects than to generalize comprehensively polygon agents. Along these lines, the better compliance of group agents to the constraint 'M6 Object separation' seems to be reasonable.

Synthesis

Bearing in mind that the generalization was fully automatically achieved – besides the manual assignment of polygons to groups which is not related to the generalization process itself – and that generalization is an ill-defined and over-constrained problem (c.f. section 2.1) the results of quantitative evaluation are generally encouraging. It stated that 50% of all polygon agents reached a 'perfect' state and 25% a 'good' state with respect to metric satisfaction. At the group level, 38% of all group agents hold a 'perfect' and 33% a 'good' basic satisfaction after generalization. However, it should be considered that quantitative evaluation relies on the predefined evaluation methods for the individual constraints and a change of these methods would imperatively lead to different results of generalization and quantitative evaluation, respectively.

Results of quantitative generalization may assist the user during the post processing stage of agent-based polygon generalization when they perform a final evaluation of the results and some interactive (re)generalizations, if necessary, by pinpointing the potential problem spots. In practice, all the agents that did not reach a perfect state could be marked and receive an indication of the type of remaining conflicts and a proposal of possible actions for their resolution. However, it must be emphasized again that map generalization is an over-constrained problem and thus a solution is always a compromise between the satisfaction of various constraints (cf. chapter 6).

7.5 Generalized land cover 1:100,000

The second experiment deals with the generalization of the land cover layer of the test area to a scale of 1:100,000. In other words, the readability of the polygonal subdivision must be ensured although the map space is reduced to a 1/16 of the original map space. This experiment intends to achieve an indication of the flexibility of the proposed framework and prototype for automated polygon generalization. Thus, it acts on the same assumptions as the previous experiment (i.e. number of represented categories, used generalization constraints, groups of polygons etc.) – of course, with the exception of the target scale. Since all the parameters required by the prototype for automated polygon generalization, such as the goal values of the generalization constraints, are defined in map units and specified at run time only the target scale needs to be adapted prior to running the second experiment. Figure 7.14 portrays the fully automatically derived land cover data set 1:100,000 at a scale 1:50,000 and 1:100,000 as well as the original polygonal subdivision of the test area at a scale of 1:100,000 while Figure 7.15 shows a comparison of the original data set reclassified for a scale 1:50,000 and the data set generalized for the target scale.

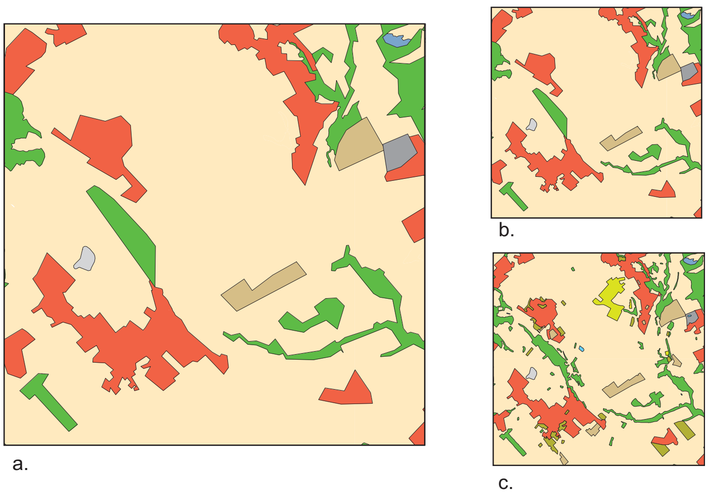


Figure 7.14: Polygonal subdivision of the test area generalized for a scale 1:100,000 **a.** at a scale 1:50,000 and **b.** at the target scale 1:100,000. **c.** Original data set from VECTOR25 at a scale 1:100,000. Figure shown at scale. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

Qualitative evaluation. In qualitative evaluation of the experiment's result the same criticism arises that has been already discussed with respect to the polygon mosaic generalized for a scale 1:50,000. The key points again refer to the unsatisfactory solution of shape simplification, the lack of methods for spatial and semantic analysis and the inhomogeneous generalization – for a detailed discussion on these aspects see section 7.4.1. Unsatisfactory generalizations even appear in the context of the same polygons and groups of polygons. For instance, compare the generalization of the group of disjoint polygons in the east of the test area in the scale 1:50,000 (Figure 7.6) and 1:100,000 (Figure 7.15). A more thorough consideration of methods for spatial and semantic analysis may indicate for both scales a typification or aggregation operation rather than the omission of most of the polygons.

Quantitative evaluation. Since qualitative evaluation highlighted a similar quality in the generalized land cover data sets at 1:50,000 and 1:100,000, it is also assumed that the causes why agents do not respect certain constraints are similar as well. Furthermore, the preliminary aim of this second experiment was to study the flexibility of the proposed approach. Thus, a detailed quantitative evaluation of the generalized land cover 1:100,000 is abstained from and only some basic statistics of quantitative evaluation are presented that give a rough impression of the accomplished quality.

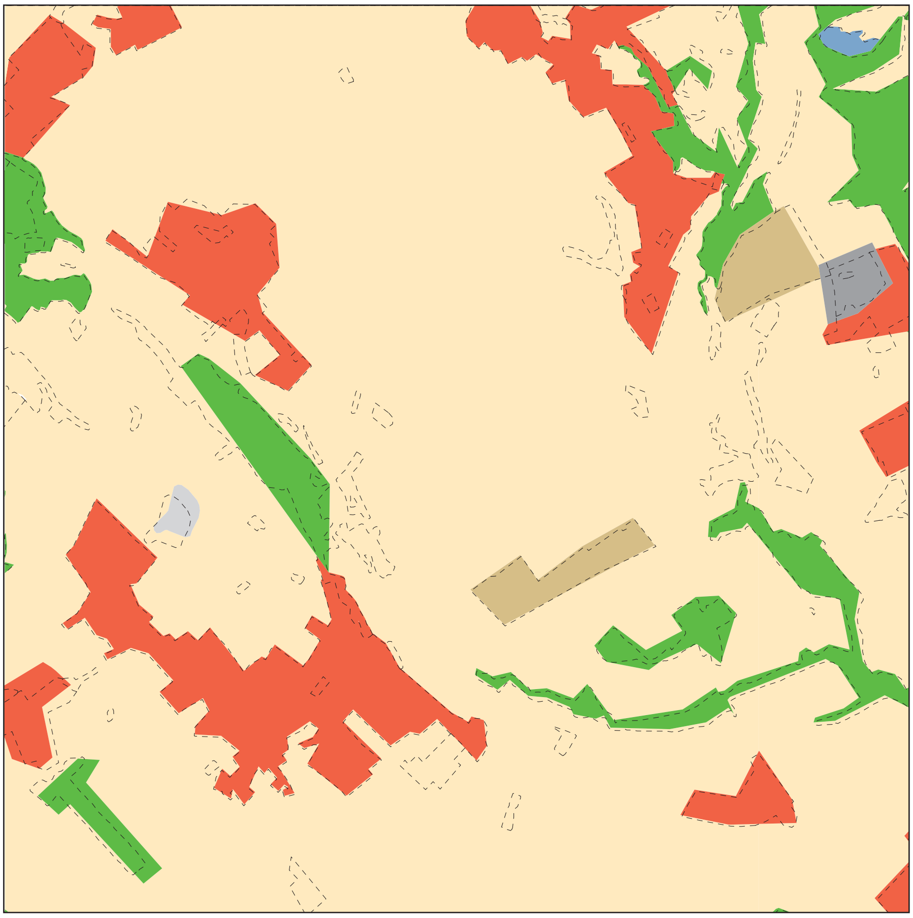


Figure 7.15: Polygonal subdivision of the test area generalized for a scale 1:100,000 shown at a scale 1:25,000 (solid contours) together with the test data set reclassified for a scale 1:50,000 (dashed outlines). Figure shown at scale. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

At the polygon level 25, of the 103 polygons of the original extract of VECTOR25 (24%) are retained. In other words, 26 polygons are eliminated through reclassification (cf. section 7.3.1) and 52 during the generalization of the group and polygon agents. The average metric satisfaction (cf. section 7.4.2) improved from 2.6 (after reclassification) to 3.6 in the generalization result, on a scale of 1 ('very bad') to 5 ('perfect'). 8 of 25 polygons (32%) receive a 'perfect' satisfaction after generalization. Further, 15 of 26 group agents remain in the polygonal subdivision generalized for a scale 1:100,000. Their basic satisfaction (cf. section 7.4.2) increased from 2.4 to 3.8 due to generalization.

With respect to quantitative evaluation, the results for a scale of 1:100,000 are slightly worse than the ones for a scale of 1:50,000. The reason is attributed to the fact that the required generalization effort from a scale 1:25,000 to 1:100,000 is, of course, significantly higher than the one from a scale 1:25,000 to 1:50,000. The result of the second experiment shows the capacity of the proposed framework as well as the implemented prototype to accomplish automated polygon generalization for different scale changes. Qualitative and quantitative evaluation of the generalized result support this statement in showing a generalization quality that is close to the one accomplished in the previous experiment and a good legibility despite the smaller scale. Furthermore, the raised criticism concerns the algorithms used and lacking methods rather than the proposed generalization strategy itself.

7.6 Conclusions

This chapter discussed the implementation of a first comprehensive framework and research platform, respectively, for the automated generalization of polygonal subdivisions. The reported research was conducted in continuation of the AGENT project (Lamy et al. 1999, Barrault et al. 2001). The experiments with VECTOR25, the large-scale digital landscape model of Switzerland, and its generalization for a target scale of 1:50,000 and 1:100,000 demonstrated the capability of a MAS in general and the enhanced AGENT engine in particular for such a complex generalization task. The qualitative and quantitative evaluation of the achieved generalization of the land cover layer 'primary surfaces' of VECTOR25 encourage the following conclusions:

General

- Qualitative evaluation established that the prototype achieved generalized polygonal subdivisions of the test area that visually meet not only the requirements of readability at the target scales (i.e. 1:50,000 and 1:100,000) but also maintain the geographical meaning of the original data set. Minor criticisms were raised with respect to selected aspects of the achieved solution, such as the (in-)homogeneity of generalization or inadequate shape simplification.
- The quality and efficiency of the generalization of group agents should be improved by the implementation of negotiation mechanisms that allow agents to communicate in order to develop common plans that help to establish a solution which reaches the best compromise between the constraints associated with different polygon agents. Duchêne (2003) recently proposed such concepts for the interaction of agents in the generalization of topographic maps and implemented them within the AGENT engine.

Constraints

- The procedural knowledge used in the current implementation to evaluate the satisfaction of a constraint and to choose plans according to a certain situation stems exclusively from empirical knowledge and testing. Mustière (1998) and Mustière et al. (2000) investigated machine learning techniques to improve existing procedural knowledge and gain additional knowledge on cartographic generalization. Applying similar studies to polygon generalization could help to improve the capability of the existing prototype to adapt more flexibly to different situations.

- The development and implementation of methods for semantic and spatial analysis should be emphasized, on the one hand, in order to be able to consider more explicitly the conflict situation and its structure (e.g. number of polygons, their relative sizes, and their configuration) when proposing plans and, thus, to achieve alternative and maybe better solutions more efficiently. On the other hand, such methods can provide the basis for the automated detection of group agents that is not implemented so far.
- With respect to metric constraints the evaluation of the achieved results did not indicate the need for any additional metric constraint on the polygon or group level except the constraint 'M5 Respect spatial context'. It is not yet implemented but it is supposed to be of increased importance for the generalization of groups of disjoint polygons.
- In the developed prototype changes in the structural constraints such as 'S1 Shape distortion' and 'S2 Absolute position' are tracked across the different states of an agent but not used actively in decision making, that is, the comparison of different states according to their generalization quality. The control of the outcome of automated generalization could be improved by their integration in decision making (i.e. into the agent life cycle).

Algorithms

- Both the qualitative evaluation of the granularity of polygon boundaries and the quantitative evaluation, with respect to the improvement of the metric constraint 'M2 Outline granularity', imply that the simplification of polygon shape can not be established satisfactorily in all cases by means of the existing concepts and algorithms. Thus, on the one hand, for reducing the outline granularity of polygons such algorithms should be developed and used that simplify the shape of the polygon itself rather than the polygon boundary. On the other hand, the line agent class was not implemented in the current version but according to its concept – compare section 4.2 – its implementation should allow a better resolution of such conflicts.
- The compliance of polygon agents to the constraint 'M3 Distance between boundary points' might be increased if the algorithms that implement an exaggeration operation are improved. For instance, the snakes-based exaggeration algorithm should be enhanced in such a way that it is able to handle individual polygons as well (cf. chapter 5).
- The reproduction of the generalization process during quantitative evaluation showed that the dissatisfaction of constraints related in some cases to the update of the polygonal subdivision relating to a generalized polygon agent. That is, this process should not only take into account first order neighbors but also second order ones to enable a better and topologically consistent re-integration of a generalized polygon agent into the subdivision.

In conclusion of the implementation of the prototype system, it seems legitimate to state that the concept of agent-based generalization proved its potential for the automated generalization of polygonal subdivisions. This approach was originally suggested by the AGENT consortium (Barrault et al. 2001) for the generalization of topographic maps and extended for polygon generalization in this PhD project. Results on the qualitative and quantitative evaluation of the experiment support this statement as well as the efficacy, i.e. the capability to resolve a given problem, of the proposed framework and the implemented prototype. A detailed discussion on the MAS-based approach to polygon generalization as well as the AGENT engine of LAMPS2 is provided in the sections 8.2.1 and 8.2.2.

While the conclusions here focused on the experiment conducted with the implemented prototype the next chapter intends to summarize the PhD project by discussing the main achievements and insights gained on the subject of polygon generalization, and offer an outlook on open research questions.

Chapter 8

Conclusions

This thesis examined a MAS-based approach to automated polygon generalization. In doing so, the undertaken research work studied possible ways to model decision making in automated polygon generalization, energy minimizing techniques to resolve metric conflicts in polygonal subdivisions and the modelling of constraints for polygon generalization. Finally, a prototype system for automated polygon generalization based on the proposed framework and generalization tools was implemented and evaluated.

This section intends to subsume and highlight the main achievements of this work as well as the insights gained into automated (polygon) generalization. The achievements and insights initiate an outlook on further improvements of the prototype system, on possible future research with respect to the automated generalization of polygonal subdivisions and on the challenge of a comprehensive solution of automated generalization, that allows arbitrary scale changes as well as the comprehensive generalization of different spatially related data types (e.g. road networks and forest parcels in a topographic map).

8.1 Achievements

The achievements are discussed with respect to the main research questions of the work, that is, the development and implementation of a comprehensive framework, the examination of the potential of energy minimization techniques and the identification and modelling of constraints for automated polygon generalization.

Comprehensive framework. Starting from an analysis of possible approaches to orchestration in map generalization, a framework for automated polygon generalization was established. It follows an MAS-based approach to polygon generalization. In doing so, it relies on ideas and concepts developed in previous research carried out by Ruas (1999) and the AGENT consortium (Lamy et al. 1999, Barrault et al. 2001). Such concepts, among other things, are constraints as the engine of generalization, the concept of levels of analysis and an agent's life cycle (cf. chapter 4).

In order to study the potential of the proposed framework, a prototype system was implemented based on the previously outlined ideas and methods. It constitutes the first prototype system for comprehensive polygon generalization ever reported in generalization research. In practice, the implementation is an extension of the original AGENT prototype that was developed for the generalization of road networks and urban settlement areas within topographic maps, for the automated generalization of polygonal subdivisions. Hence, the research work and implementation focused mainly on the specification of new agent types, measures, algorithms and constraints that were especially adapted to the needs of polygon generalization. The implemented prototype system includes 3 agent types (i.e. map, group and polygon agents), 16 generalization constraints, 16 measures and 15 generalization algorithms (implementing 8 generalization operations) as well as procedural knowledge related to automated polygon generalization. The prototype enables a fully automated polygon generalization process with the exception of the identification of group agents (see below).

The prototype is implemented in the commercial GIS LAMPS2 which is used by other institutions of the generalization research community, such as the French, Danish and British NMA. Furthermore, it is very flexible with respect to the extension with additional generalization tools such as algorithms, measures or constraints. Hence, generalization tools for polygon generalization that are developed in future research need no longer be tested independently but can be directly integrated into a framework for polygon generalization. In other words, their fitness for polygon generalization can be determined, on the one hand, as an atomic generalization tool and, on the other hand, as an integrative component of an automated generalization process.

Additionally, the developed and implemented constraints, measures and algorithms are available independently of the AGENT engine as a rich toolbox for polygon generalization within LAMPS2. Thus, they can also be used together with other approaches for orchestration in polygon generalization. For instance, the established constraints may be executed for the evaluation of manually generalized polygonal subdivisions or the snakes-based algorithm may be applied to the resolution of size and proximity conflicts in interactive generalization. Along these lines, this work established an extensive research platform for future research dedicated to polygon generalization and its automation.

Energy minimization techniques. This work studied the concept of energy minimization, i.e. the snakes method, as a basis for the implementation of polygon generalization algorithms. In doing so, it follows the approach proposed by Bader (2001) for line generalization. In practice, a single algorithm that executes boundary moving operations was developed. It can be triggered in such a way that it achieves the displacement, enlargement or exaggeration of polygons or an arbitrary combination of these generalization operations. For doing so, a concept was established that varies the force model and weight properties of the conflicting polygons.

The proposed algorithm allows the simultaneous resolution of size and proximity conflicts while preventing the creation of new conflicts of these kinds. In other words, a holistic approach to generalization is enabled – of course, only with respect to size and proximity conflicts –, that is, the algorithm achieves automatically a compromise between different constraints (i.e. minimal size of polygons and minimal distance between polygons). The algorithm can also propagate modified polygon geometries to the polygonal subdivision either as a part of the transformation process itself or as a separate process that is triggered just after the transformation process. Due to both the holistic solution of size and proximity conflicts as well as the possibility of automated propagation, it is believed that the developed algorithm constitutes a significant improvement compared to both existing algorithms and sequential approaches to propagation.

The algorithm was integrated into the agent-based prototype system as well as the toolbox of generalization tools for automated polygon generalization. The snakes-based algorithm demonstrated its fitness for use in accomplishing appropriate results not only in interactive generalization – as shown in section 5.4 – but also as an integrative component of a comprehensive generalization process. Besides showing the potential of snakes for polygon generalization (see as well the discussion below), this work provided a generic concept for the application of energy minimizing techniques as an example of optimization techniques to polygon generalization. In other words, the concept may be also implemented by other optimization techniques, for instance, least squares adjustment or elastic beams.

Generalization constraints. The work proposed a set of generalization constraints that meet the needs of automated polygon generalization. In other words, the proposed generalization constraints are enabled to control the automated generalization processes of polygonal subdivisions. The constraints are not only specified at a conceptual level but also the methods and properties are provided that are required for the constraints' integration into an automated generalization process. That is, for every constraint a goal value, an importance and priority value, relative to other constraints, as well as methods for measuring the property that the constraint relates to and for determining its satisfaction, are discussed and defined. The constraints are again included in LAMPS2 and embedded into the agent-based framework for automated polygon generalization.

In other words, the framework relies on these constraints for conflict detection, conflict resolution and the validation of generalization solutions. The successful generalization of the test case, that is, a clipping of the 'primary surfaces' layer of VECTOR25 to a target scale of 1:50,000 and 1:100,000, allows the quality of the implemented generalization constraints to be emphasized, since the qualitative and quantitative evaluation considered the results as being promising and satisfying. For instance, in the generalized test data set, 50% of the polygon agents and 38% of the group agents achieved a 'perfect' solution and an additional 25% of the polygon agents and 33% of the group agents achieved a 'good' satisfaction after generalization. So, the proposed constraints demonstrated their fitness for automated polygon generalization.

However, the application of the discussed generalization constraints is neither linked to the implemented prototype nor to an agent-based approach to orchestration in polygon generalization. That is, the proposed generalization constraints as well as their properties (i.e. goal, priority and importance values) and methods (i.e. evaluation methods, measures) may also be used in combination with any other framework for polygon generalization and, thus, serve as a basis for conflict detection, conflict resolution and evaluation of generalization solutions.

8.2 Insights

Besides the achievements discussed in the previous section, the work delivered insights into different aspects of map generalization and its automation by means of MAS technology. The main insights are subsumed in the following sections.

8.2.1 The MAS-based approach to polygon generalization

While the AGENT consortium was able to highlight the feasibility of a MAS-based approach for the automated generalization of road networks and settlement areas, this thesis demonstrated – in extension of the AGENT prototype – the potential of such an approach for the automated generalization of polygonal subdivisions. The reasons are attributable to the following capabilities of such an approach to orchestration in map generalization, namely the capability:

- to combine different generalization tools into a comprehensive generalization process that allows the human way of decision making to be mimicked;
- to adapt to varying generalization controls (e.g. target scale or user's needs) through a different setup of parameters;
- to compromise amongst several constraints of different importance that are associated with an agent;
- to coordinate the generalization of geographic objects at various spatial levels of a polygonal subdivision and, thus, to compromise amongst constraints imposed at different spatial levels of a polygonal subdivision (i.e. agent types);
- to manage side-effects of generalization as an intrinsic component of the generalization process;

With respect to the MAS-based approach to automated polygon generalization the following insights are out standing:

Grouping of polygon agents. The grouping of polygon agents into higher-order features (i.e. group agents) was first identified as an automated task that should be performed by the map agent at run time (cf. section 4). During the implementation of the prototype it was decided – for the sake of the implementation of the entire framework – to substitute this step by an interactive process performed by the cartographer. The reasons owe mainly to two facts, missing methods for spatial and semantic analysis and the need of expert knowledge on the theme represented by the polygonal subdivision. The detection of some group agents relevant to a specific theme, such as spatial cluster agents or topological partition agents seems, on the one hand, to be rather straightforward (cf. section 7.3.1). Otherwise, the identification of group agents (e.g. pattern agents, geographic partition agents) relies to a good deal on the subjective interpretation of a polygonal subdivision by an expert of the theme and should never be exclusively done by a cartographer, in order to avoid the risk of misinterpretation. Hence, methods for the automated grouping of polygon agents (i.e. the identification of meaningful patterns) should be established in cooperation with domain experts. While, for instance, with respect to the grouping of buildings specific methods were already studied – among others, by Regnauld (1998), Bader (2001) and Christophe and Ruas (2002) – methods for the grouping of polygons within polygonal subdivisions were, so far, not investigated. Since this step of grouping polygon agents is crucial for automated polygon generalization and especially the generalization of larger data sets it should be focused on first in future research.

Computer resources. The conducted experiments showed that the MAS-based approach to polygon generalization requires extensive computer resources. For instance, the automated generalization of the test data set, i.e. a polygonal subdivision made up of 103 polygons, to a scale of 1:50,000 took roughly 1 hour although the grouping of polygon agents was interactively done before. Of course, the goal of this PhD project was to study the potential of an MAS-based approach to automated polygon generalization and not to produce a prototype system optimized with respect to its implementation. However, it is believed that such an optimized system will not allow a significant reduction of computational costs since any comprehensive, iterative generalization process – here implemented through a MAS – involves high computational costs. As long map production does not impose time constraints on the generalization process (e.g. on-the-fly generalization in a web application) the aspect of required computer resources seems to be of secondary importance if the achieved solutions and their quality justify the effort and meet the defined requirements, respectively. One possible way to make the MAS-based generalization process more efficient with respect to the usage of computer resources represents the integration of additional procedural knowledge that helps to direct the search for the best plan and, thus, allows a perfect state to be reached earlier.

Procedural knowledge. The definition of the set of constraints as well as the implementation of the prototype system highlighted the necessity of knowledge on polygon generalization. Such knowledge supports the formalization of constraints, the definition of the constraints' evaluation methods and the proposal of plans that take into account the spatial and semantic context of generalization conflicts. From the gained experience about automated (polygon) generalization it seems legitimate to state that the more formalized knowledge is included in a generalization system the better results can be achieved and the more efficient the generalization process can be. The prototype system contains mainly empirical and textbook knowledge. However, additional knowledge about polygon generalization is needed that, for instance, helps to choose the polygon agent of a group agent that should be generalized first. So far, child agents are triggered in turn, that is, neither the severity of the supervised agents nor the semantics of the agents are considered. As a promising alternative to the extensive and tedious empirical testing for acquiring generalization knowledge, Mustière (1998, 2001) demonstrated the potential of machine learning techniques, in order to collect formalized knowledge form experts in cartographic generalization. Doubtlessly, such methods could help, on the one hand, to improve the knowledge about polygon generalization and, on the other hand, to fine tune the parameters required by generalization tools.

Parameter tuning. The empirical testing of the prototype during its implementation and its evaluation showed that the parameter setup and tuning is complex as parameters at various levels of the generalization process (i.e. constraints, measures, algorithms) interact with each other. Thus, it is often considered to be tedious. For instance, the parameter setup and fine tuning of the prototype system involved several days of running extensive experiments. The tuning of the parameters, such as the goal and importance values of constraints, has major impact on the achieved results of automated polygon generalization. Beyond any doubt, the tuning of the parameter requires deep knowledge of the MAS-based framework for polygon generalization and, thus, the ease of use of the prototype is limited at the current state of implementation.

In practice, the fully automated generalization of a clipping of the 'primary surfaces' layer of VECT0R25 for a target scale of 1:50,000 was shown. In doing so, the implementation of the proposed agent-based framework for automated polygon generalization showed its potential and efficacy, i.e. its capacity to solve a given problem. The qualitative and quantitative evaluation of the achieved result emphasize that overall, MAS technology is suited to automate polygon generalization. Of course, experiments on more 'prominent' scale changes, for instance, from 1:25,000 to 1:200,000, as well as on other types of categorical data (e.g. geology, vegetation) are still lacking and, respectively, if successful would allow to amplify the advanced arguments.

8.2.2 The AGENT engine of LAMPS2

Besides the insights with respect to the proposed MAS-based approach to polygon generalization, the work also provided insights into the AGENT engine itself and its implementation in LAMPS2. The AGENT engine used for this PhD project is the one delivered together with the GIS LAMPS2 Release 4.1 (v4-1a-5i-2). It was applied to the generalization of polygonal subdivisions without any modifications of its core functionality (i.e. the life cycle of an agent).

Consideration of constraints' importance. The relative importance values assigned to the generalization constraints define the relevance of constraints in comparison to other constraints for the quality of a generalized data set. These values are taken into account when validating the changes of a data set, that is, a plan triggered by a certain constraint must not decrease the satisfaction of any constraint exhibiting a higher importance. The happiness of an agent is calculated through the average satisfaction of all supervised agents or the minimum satisfaction of any attached constraint. In doing so, the importance of constraints is ignored. Since the happiness of an agent is essential to determine its need for generalization it is believed that a more sophisticated way of calculating an agent's happiness may allow a more differentiated decision making and evaluation of generalization solutions.

Choice of the best plan. The best plan is a heuristic that allows the generalization process to be sped up (cf. section 4.3). Its identification is controlled by the priority values of violated constraints at the first instance and the severity of the violated constraints at the second instance. From the gained experience of agent-based generalization it is supposed that the impact of the best plan could be improved if this plan would be determined at a single instance through a combined criteria derived from both the constraints' priorities and severities. Hence, the efficiency of the agent-based generalization process may improve.

Communication between agents. The AGENT engine does not support the communication between agents. The AGENT project assumed a pyramidical organization of agents, such as in urban areas the town, urban blocks and individual buildings. This implies the distinction of disjoint groups of objects. Consequently, contextual generalization was exclusively delegated to the meso agents. Hence, the need for the communication between agents was not given priority. Due to the fact that polygon generalization acts on a space exhaustive data structure and spatially interrelated objects (e.g. a group of connected polygons) it is expected that agent-based polygon generalization could benefit from a communication facility of agents especially through a better generalization of groups of connected polygon agents. Compromises between competing agents may be found that are not possible now since, so far, polygon agents are independently generalized in turn. Research with respect to the communication of agents was currently reported by Duchêne (2003).

Practical issues. The AGENT engine turned out to be badly documented but well customizable through little programming effort. That is, generalization constraints, algorithms, measures, plans etc. could be added, modified or removed straightforwardly once the implementation of the engine was figured out. In general, customers of LAMPS2 obtain agent-based generalization tools but no direct programming interface to the AGENT engine. Note that our group – as a former project partner – has a specific licence agreement with Laser-Scan Ltd. that provides us access to the source code of the AGENT engine and allows research on the AGENT engine to be done.

Despite these deficits of the AGENT engine the prototype system established generalizations of the test data set that were considered to be satisfying and appropriate representations of the test data set at a reduced display scale. The AGENT engine is currently re-designed and re-implemented by the Laser-Scan Ltd. and, thus, modifications and improvements of the AGENT engine are supposed to be established (Laser-Scan 2002b).

8.2.3 Other aspects of agent-based polygon generalization

Constraints for polygon generalization. The work established and formalized a set of generalization constraints for automated polygon generalization (i.e. 7 metric, 2 topological, 4 structural and 4 procedural constraints). Generalization constraints and their specification (cf. section 6 and Appendix B) relate directly to the potential of the proposed framework for automated polygon generalization since they control the generalization process. As the experiments indicate, the included constraints are a valid specification of the needs of polygon generalization, in order to establish a properly generalized polygon mosaic. While the metric generalization constraints and their satisfaction, respectively, define a preliminary need of map generalization, topological and structural constraints help to validate modifications of a polygonal subdivision. Metric and topological constraints are linked directly to geometric and topological properties of a polygon mosaic and, thus, easier to evaluate than some of the structural constraints that concern fuzzy concepts, for example ‘shape’ or ‘relative configuration’. Consequently, such constraints are difficult to formalize and corresponding knowledge that would allow compromise amongst their satisfaction and the satisfaction of other – mainly metric – constraints is lacking. Thus, structural constraints are only used to describe generalized data in the implemented prototype. However, the evaluation of the generalized test data set emphasized the argument – already discussed in section 6.8 – that comprehensive polygon generalization can be accomplished although structural constraints are not taken into account in the decision making and validation of generalization solutions. Qualitative evaluation of the experiments’ results amplified the good readability of the presented solutions while no critics was raised with respect to the maintenance of essential structures. Of course, structural constraints may be more relevant in relation to different types of polygonal subdivision portraying, for instance, geology. The constraint-based approach demonstrated not only its potential to control a comprehensive generalization process of polygonal subdivisions but also its flexibility according to changing generalization controls. Remember, for instance, that the generalizations of the test data set to the scales of 1:50,000 and 1:100,000 were achieved only through the change of a single parameter, namely the target scale.

Snakes in polygon generalization. Previous research by Burghardt and Meier (1997), Burghardt (2000) and Bader (2001) demonstrated already the fitness of energy minimizing techniques for the generalization of linear features (e.g. roads and rivers). The work provided an insight into the potential of a particular energy minimizing technique, i.e. snakes, for automated polygon generalization. Snakes enable the holistic solution of proximity and size conflicts within a group of polygons. In other words, the application of energy minimizing techniques is restricted to such generalization operations that can be implemented through a boundary moving algorithm. In polygon generalization such operations include the displacement, enlargement and exaggeration of polygons. Otherwise, snakes proved their capability to establish automated propagation either as part of the transformation process itself or as a separate algorithm triggered after the transformation process. So, the propagation capability of the snakes-based algorithm may also be used together with other generalization algorithms, for instance, for adapting a polygonal subdivision to a changed polygon object. However, optimization techniques in the broadest sense are not suited to orchestrate a comprehensive generalization process but are a valuable method for the concurrent solution of several metric conflicts in automated polygon generalization. That is, the agent-based framework for polygon generalization that solves conflicts in turn according to violated constraints should make use of this different approach to conflict resolution, that is, the concurrent solution of different generalization conflicts. The snakes-based algorithm could be applied as an individual plan to a group of polygons or polygon if metric constraints are violated. Due to the reduced number of involved objects the computational costs required by snakes will remain low. If the metric conflicts are solved satisfactorily the agent-based generalization process may be sped up since the calculation of several candidate solutions and eventually a tedious propagation process is avoided. Otherwise, the iterative agent-based process can be alternatively tried out.

Semantics. In polygon generalization semantics have to play a major role (cf. section 2.3) since every polygonal subdivision links intimately a spatial and a semantic component (cf. section 2.2). The implementation of the prototype system as well as the qualitative and quantitative evaluation of the experiments‘ results (cf. sections 7.4 and 7.5) highlighted the importance of semantics – especially in the context of methods for semantic and spatial analysis – for automated polygon generalization. The implemented prototype stores the semantics of a polygonal subdivision (i.e. the classification schema) as a property of the map agent class as a tree like data structure. The agent-based framework allows the consideration of semantics in polygon generalization at two levels, that is, (1) the semantic generalization guided by the metric constraint ‘M7 Number of categories’ and (2) the control of geometric transformations based on the semantic similarity of agents and the importance of a category. The relative area covered by a category of the area of a polygon mosaic is used as an indicator for the rareness and importance, respectively, of a polygon. The fact that map generalization intends to preserve the occurrence of rare objects should be considered in the proposal of plans for the improvement of constraints' satisfaction. For instance, in the implemented prototype a too small polygon of a frequent category is eliminated rather than enlarged while a rare polygon is enlarged rather than eliminated. The semantic similarity of polygons that is calculated after Yaolin et al. (2002b) based on the hierarchical classification schema of a polygonal subdivision influences the agent-based polygon generalization at the following two stages:

- Semantics facilitate the proposal of such plans that consider a conflict's context in conflict resolution. In doing so, the identification of the best plan is more likely and better solutions can be accomplished. For instance, in the prototype system a proximity conflict between polygons of similar categories is solved by means of an aggregation operation rather than a displacement operation while polygons of semantically different categories are displaced from each other rather than being aggregated.
- If a polygon agent is eliminated during agent-based generalization a gap remains that needs to be filled in order to maintain the space exhaustive character of a polygonal subdivision. That is, the area of the eliminated polygon is either assigned to the semantically nearest neighbor or distributed to all adjacent polygons.

The experiments showed that the consideration of semantics in polygon generalization is a necessity, in order to enable efficient and proper generalization results. Hence, a stronger consideration of semantics in polygon generalization based on the ideas pointed out in this paragraph should be stressed. For instance, only some of the decision trees attached to the implemented constraints (cf. Appendix B) take into account the semantics of the involved agents. Semantics of a polygonal subdivision should be also borne in mind when grouping polygon agents – see above.

8.3 Outlook

Both the discussion of the achievements of the PhD project and the insights gained already indicated some aspects of automated (polygon) generalization that require improvement and further research effort. Hence, this section intends to highlight potential enhancements of the presented concepts and methods as well as to motivate and encourage future research at different levels, that is, the implemented prototype system, automated polygon generalization and the consistent generalization of thematic and topographic maps.

Prototype system. The following paragraph lists the main, potential improvements of the implemented prototype system that stem mainly from the quantitative and qualitative evaluation of the generalized data set of the test area (cf. section 7.4)[1]:

- *Experiments* on other categorical data besides land cover and additional scale changes should be conducted in order to achieve, on the one hand, a more reliable evaluation of the potential of the implemented prototype and its underlying framework. On the other hand, such experiments may help to identify limitations and needs (e.g. generalization constraints and algorithms) that are linked to a specific type of polygonal subdivision or scale change.
- As a side-effect of these further experiments, it would be expected that the constraints' goal values and evaluation functions could be *fine tuned* and, consequently, an improvement of the quality of generalizations accomplished.
- The idea of *knowledge acquisition* according to polygon generalization should be resumed since all the generalization knowledge included in the prototype system (e.g. the evaluation functions and the decision trees for choosing the 'best' plan) was derived empirically. Methods of knowledge acquisition, such as machine learning, could help to make the existing knowledge of polygon generalization more robust (Mustière 2001, Regnauld and Mustière 2000) and maybe provide additional knowledge. For instance, improved procedural knowledge could help to increase the performance of automated generalization in providing more detailed decision trees for the resolution of generalization conflicts (cf. section 7.2.2).
- *Structural constraints* for polygon generalization should be examined at the level of individual constraints, in order to formalize fuzzy concepts such as 'shape' or 'visual balance', and with respect to the validation of generalization solutions. That is, the linking of the satisfaction of structural constraints to the quality of generalization in comparison to other constraints. Here, it is expected that empirical tests based on test cases with experts in map generalization as well as map users may help to develop corresponding concepts.
- The *efficacy*, i.e. the capacity to generalize a given set of objects in the best way, was not investigated so far. In such a study deficits could emerge with respect to the validation of solutions, for instance, the rejection of a solution due to restrictive evaluation methods or inappropriate measures.

The agent-based framework. Future research could accomplish enhancements of the agent-based decision making underlying the proposed framework and prototype system. Possible further developments are itemized below.

- A significant improvement of the proposed framework could be achieved by enhancing the map agent in such a way that the assignment of polygon agents to group agents could be automated. Bearing the corresponding problems highlighted in the previous chapter in mind, a short term enhancement could be the implementation of a semi-automatic solution. That is, an algorithm proposes a preliminary grouping based on criteria that are easy to implement such as topological relationships and spatial proximity and, next, an expert refines the grouping with respect to their expertise on the theme represented by the polygonal subdivision. In doing, so groups of polygons could be also established for

[1] Further improvements were already discussed in section 7.6.

data sets significantly larger than the test data set (i.e. roughly 100 polygons) within an acceptable time. In relation to an full automation of the grouping of polygons next steps in research[2] should focus on an inventory of meaningful structures in polygon mosaics, the modelling and formalization of the corresponding expert knowledge, the development of suitable algorithms etc.

- The study of more sophisticated evaluation methods for the determination of the satisfaction of agents with respect to their associated constraints. So far, either the average of the constraints' satisfaction or the minimum satisfaction of all constraints is used, that is, the relative importance is not considered. Alternative methods, for instance, from multicriteria decision making (Malczewski 1999), may help to improve the evaluation of the satisfaction of agents at a local level and the evaluation of generalization solutions at a global level.
- The *snakes-based algorithm*, outlined in chapter 5, is only applied to individual generalization operations, that is, either displacement, enlargement or exaggeration. Additional methods of spatial analysis and procedural knowledge should be sought that allow the full potential of this algorithm to be exploited, that is, the holistic solution of size and proximity conflicts in a group of polygons, i.e. a group agent.
- The integration of *learning mechanisms* that enrich and adjust, respectively, the procedural knowledge contained in the system according to the failure and success of plans might help to improve the efficiency of the system as well as the quality of generalization (Duchêne and Regnauld 2002).

Polygon generalization. Besides the possible improvements of the established prototype system and framework automated polygon generalization provides several leads for future research. The combination of different generalization tools for polygon generalization into a prototype system helps to determine both the completeness of these generalization tools and the potential of the individual tool in a comprehensive generalization process. Hence, the following topics of possible future research according to polygon generalization were identified:

- The *assessment* of generalized data sets usually relies on a cartographic expert who is supported by automatically generated reports and flags on objects that are not sufficiently generalized (Ruas 2001b). Research with respect to the assessment of generalization solutions was neglected in (polygon) generalization (Weibel and Dutton 1999) and, thus, should be emphasized. A preliminary assessment model of generalization was recently proposed by Bard (2003).
- The qualitative and quantitative evaluation of the experiments' result stated the need for improved and additional methods of *semantic and spatial analysis*. These methods should be studied and developed with a focus on categorical data and polygonal subdivisions, respectively, in order to allow an improved consideration of semantics in agent-based polygon generalization.
- The experiments showed that a reduction in the outline granularity of a polygon requires *shape simplification algorithms* that achieve a simplification of the polygon shape rather than the polygon outline. Such algorithms are lacking in polygon generalization. While Bader (1997) outlined a possible approach based on the skeleton of a polygon Rainsford and Mackaness (2002) proposed an algorithm based on template matching for the generalization of buildings. Both works provide ideas that may as well prove to be useful for the shape simplification of polygons.
- Polygon generalization would benefit from the development of a *typification algorithm* that allows the generalization and maintenance of characteristic patterns in the spatial distribution of polygons. The algorithms of Müller and Wang (1992) that establishes the typification of disjoint polygons of the same class, Sester and Brenner (2000) that makes

[2] As a follow-up project to this PhD project a new proposal was submitted to the Swiss National Science Foundation that intends to deal, amongst other things, with this topic.

use of self-organizing maps or Burghardt and Cecconi (2003) based on mesh simplification techniques may serve as a starting point.

Generalization of thematic and topographic maps. Besides continuing the research on (agent-based) polygon generalization the logical continuation of the presented research is the combination of the AGENT prototype and the developed prototype into a common generalization process. In doing so, the aim is to achieve a consistent generalization of linear elements (e.g. road networks, rivers) and polygonal data (e.g. geology, land cover) that is the basis of any thematic and topographic map. For instance, in a geological map categorical data representing geological units are combined with linear features portraying, for example, fault lines or water sheds. This future research is emphasized by the fact that digital landscape models such as VECTOR25 also provide both types of data (i.e. linear and polygonal data). On the other hand, GIS also emerges in all natural sciences dealing with spatial data and, thus, an increased production and use of thematic maps is noticeable. Since the AGENT prototype and the system developed for automated polygon generalization use the same development platform, i.e. the GIS LAMPS2, their combination is relatively simple. However, research is needed that concentrates on the development of new concepts for the consistent generalization of different feature types in thematic and topographic maps, for instance, new constraints, algorithms etc.

8.4 Final remarks

After the success of MAS-based approaches to orchestration in map generalization – demonstrated in the AGENT project for topographic maps and in this work for polygonal subdivisions – the question emerges whether MAS technology is the missing link to a comprehensive solution of map generalization. Compared to other approaches to orchestration pursued in map generalization (cf. section 3) it seems to be evident that MAS technology exhibits a high potential to mimic a cartographer's behavior in map generalization. In other words, it supports holistic decision making, allows to dynamically adapt to different situations, to compromise between several constraints associated with an object and to coordinate the generalization of objects at different spatial levels. However, the dependency of such an approach on algorithms, constraints, procedural knowledge etc. should not be forgotten. For instance, a MAS-based generalization system can only resolve such conflicts that it detects by means of constraints and, furthermore, the quality of an achieved solution relies on the capability of the applied algorithms. Beyond any doubt, this technology is promising but whether it is really the missing link to comprehensive, automated map generalization must be examined and proven, respectively, in the future by further research and exploitation in map production. Other approaches to automated map generalization, such as simulated annealing (Ware et al. 2003) or optimization techniques (Harrie and Sarjakoski 2002, Bader 2001), should also be considered either as stand-alone concepts or as an integral component of an agent-based approach (Nareyek 2001, Russell and Norvig 2002).

The relevance of the output of the PhD project is emphasized by the fact that map production in public and private mapping agencies strives for an automated generalization and production of thematic and topographic maps and, thus, has to deal with polygonal subdivisions (e.g. geology, land cover) or selected categories of land use (e.g. forest and lakes) at least. Since the AGENT engine is already used successfully in map production at the French (Ruas 2001b, Lemarié 2003) and Danish (Bengtson 2001) national mapping agencies with respect to the generalization of road networks and buildings, this PhD project is considered to be a contribution to further automation of map production – especially with respect to polygonal subdivisions. Hence, it is hoped that some of the proposed concepts are included in the Clarity project. This project, that aims at the further commercialization and enhancement of the AGENT prototype, was launched by the GIS vendor Laser-Scan Ltd. as a consequence of the preceding AGENT project (Laser-Scan 2002b).

Once again, a little progress in automated map generalization was achieved and the vision of automated map generalization, that is, the fully automated derivation of arbitrary scales from a single master database, seems to be a little bit closer. This thesis provided answers to some research questions in polygon generalization. However, in doing so it also poses new research questions and emphasized already well known research problems. Thus, it is believed that map generalization still remains one of the most challenging research topics in cartography and geographic information science. This argument is amplified by the fact that is took approximately one hour to generalize the 103 polygons of the test area. Of course, the quality of the results was considered to be satisfying, however, a complete map usually covers a larger area and is made up of several different feature classes (e.g. roads, rivers, labels) that have to be generalized in conjunction with each other. Hence, the independent generalization of individual feature classes, such as polygonal subdivisions, is considered to be an important step towards a comprehensive solution of map generalization and a prerequisite for future research that studies the interplay of different feature classes in map generalization. Besides the comprehensive generalization of geographic data it may seem that map generalization in the context of web mapping or mobile cartography are today's major challenges in generalization research since they add a new constraint to the generalization process, namely time, that is, map generalization should ideally happen in real time. However, the author of this PhD is convinced that technical restrictions in web and mobile cartography (e.g. display resolution, loading time etc.) will be obsolete while comprehensive map generalization, that is, the derivation of any arbitrary scale from a single master database will still be not solved satisfactorily.

Appendix A

Terminology of polygonal subdivisions

This appendix presents those terms used throughout the thesis for describing geometric objects, topological objects and specific polygons and groups of polygons of a polygonal subdivision.

A.1 Geometric objects

Polygon. A polygon is defined as a closed and simple polyline in the plane (Laszlo 1996). A formal definition of a polygon is given, amongst others, by O'Rourke (1995).

Polyline. A simple polyline (chain, polygonal chain) denotes a finite collection of line segments (O'Rourke 1995), which do not 'intersect at any place other than possibly their end-points' (Worboys 1995, p. 100). The polyline enclosing the polygon is referred to as the *polygon boundary*, *polygon outline* or *outer ring*. A closed polyline exhibits no dedicated start and end point.

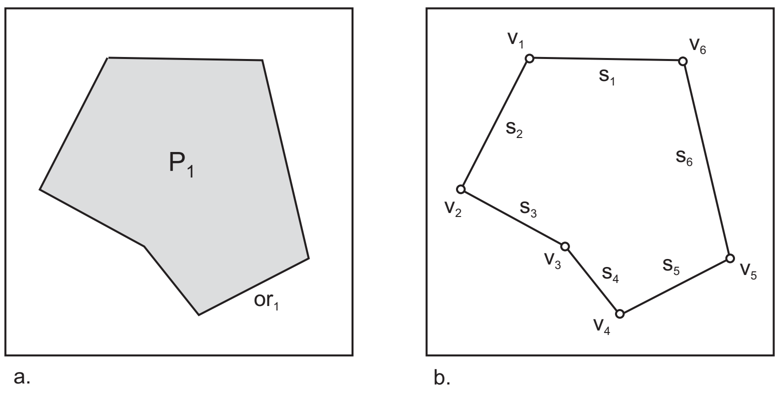


Figure A.1: Geometric objects of a polygonal subdivision. **a.** A polygon and its outer ring. **b.** The outer ring of a polygon and its segments and vertices.

Segment. A line segment denotes the straight connection of two vertices (points) of the plane – cf. the segments $s_{1,2,\ldots 6}$ in Figure A.1b. Two segments are always connected at a vertex.

Vertex. A vertex is a single point of a polyline. The vertices of a polygon's boundary are classified as convex or concave (Laszlo 1996). A vertex is called a *convex vertex* if the interior angle, which is spanned by the vertex and its two adjacent points, measures less than or equal to 180 degrees. Otherwise, it is a *concave vertex*, i.e. its interior angle is greater than 180 degrees. Whilst, for instance, the vertex v_3 of polygon P_1 in Figure A.1b is concave all the other vertices of that polygon are convex.

A.2 Topological objects

Nodes. In topologically structuring a polygonal subdivision[1] at every intersection of polygon rings a node is established. A node is linked to all links that meet at the node and to all polygons that it belongs to. For instance, the node n_6 in Figure A.2 points to the links l_1, l_2 and l_1 as well as to the polygons P_1, P_3 and P_4.

Links. A link is the polyline connecting two nodes, i.e. its from-node and to-node. Additionally, a link in a polygon mosaic is topologically related to two polygons, that is, its left and right polygon. Every ring of a polygon is defined either by a single link or a sequence of links – see, for instance, the outer ring of polygon P_1 and the links l_1, l_2 and l_8 in Figure A.2.

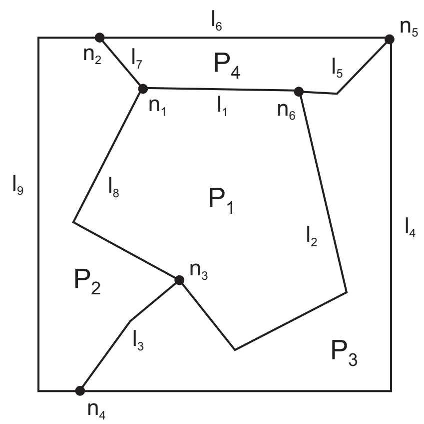


Figure A.2: Topological objects of a polygonal subdivision.

A.3 Specific polygons and groups of polygons

Island polygon. A polygon may have holes, which are bordered by so-called inner rings (cf. ir_{1_1} and ir_{1_2} in Figure A.3). The polygons enclosed by an inner ring are termed island polygons.

Disjoint polygon. A disjoint polygon or island polygon is bounded by a single link, i.e. it is adjacent to exactly one other polygon. The outer ring of an isolated polygon is equal to the inner ring of its surrounding polygon – see for example the outer ring or_4 of polygon P_4 and the inner ring ir_{1_2} of polygon P_1 in Figure A.3. Along these lines, an isolated polygon is always an island polygon.

[1]An introduction to the concepts of topology is found, amongst others, in Worboys (1995), Burrough and McDonnell (1998) and Molenaar (1998)

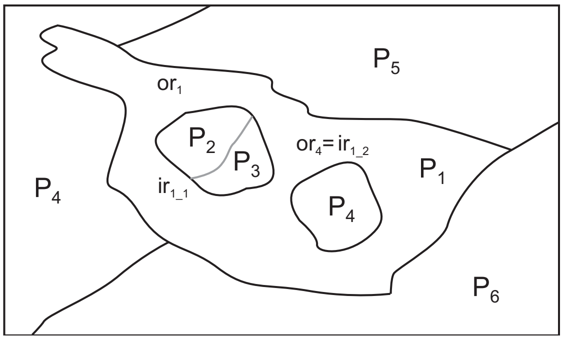


Figure A.3: An island polygon.

Group of disjoint polygons. A group of disjoint polygons denotes several isolated polygons which lie within the same polygon.

First and second order neighbors. All the polygons adjacent to a single polygon are described by the term *first order neighbors*. The *second order neighbors* define the set of polygons which a polygon's first order neighbors and their corresponding first order neighbors make up.

Appendix B

Constraints for polygon generalization

This appendix provides an overview of constraints for polygon generalization and their specification in the developed prototype system for automated polygon generalization. Note that

- The relative importance and priority of a constraint is always given as a rank in relation to the number of ranks at a spatial level (cf. sections 6.6&6.7). *The higher the rank the higher the importance/priority.* For instance, the constraint 'M4 Minimal area' at the polygon level has an importance of '5 / 6', that is, the second highest importance at the polygon level.
- If a situation leads to the proposal of several cartographic operations that should be tried out in turn the number attached to a plan represents the sequence in which these alternatives are run (cf. section 7.2.2).

M1 Consecutive vertex distance

Consecutive vertices of a polygon boundary should be separated by a minimum distance at least. This constraint intends not to trigger generalization but to speed up the generalization process by removing redundant vertices from a polygon boundary while maintaining the polygon shape as faithfully as possible (Visvalingam and Williamson 1995).

Spatial level	Line		
Goal value	consecutive vertex distance > 0.1 mm		
Importance	1 / 3	Priority	1 / 2

Measure

consecutive vertex distance

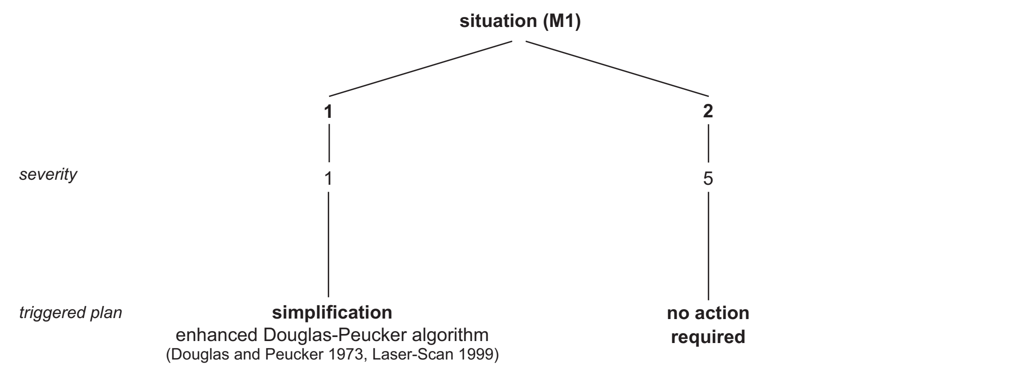
$$d_{Consec} = min(\sum_{i=1}^{n} \overline{v_i v_{i+1}})$$

Evaluation method

severity / *criteria**	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
measure / goal value	< = 98 %				> 98 %

* derived from empirical testing

Decision tree



M2 Outline granularity

Imperceptible crenulations of a polygon boundary must be eliminated.

Spatial level	Line		
Goal value	min shape width > 0.6 mm, min shape height > 0.4 mm		
Importance	2 / 3	Priority	2 / 2

Measure

minimal shape width $width = min(\sum_{i=1}^{n} \overline{v_1 v_n})$

minimal shape height $height = min(\sum_{i=1}^{n} max(\sum_{j=2}^{n-1} dist(v_j, \overline{v_1 v_n})))$

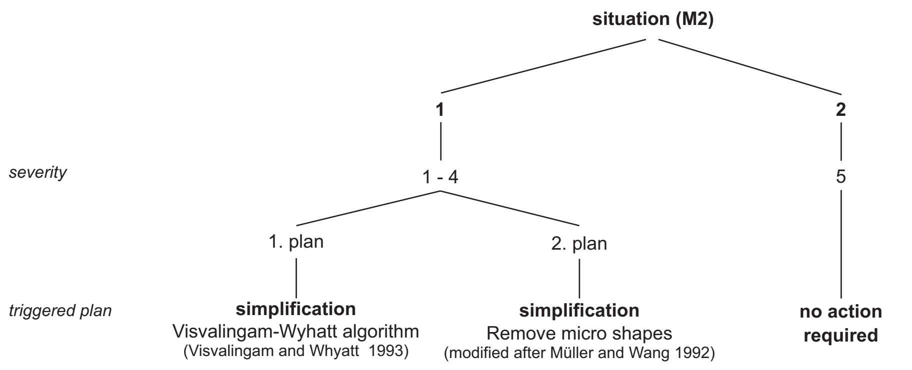
Shapes are defined after Müller and Wang (1992), see also section 6.4.

Evaluation method

severity	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
*criteria**	> = 75 % of MMSS	75% of MMSS - > = 50% of MMSS	50% of MMSS - > = 200% of MD	200% of MD - > = 100% of MD	< 100% of MD

MMSS minimum micro shape size, i.e. either min shape width or min shape height
MD maximal deviation of minimal shape width / minimal shape height to min shape width / min shape height
* derived from empirical testing

Decision tree



M3 Distance between boundary points

Any non consecutive points of a polygon geometry should be separated by a minimum distance at least.

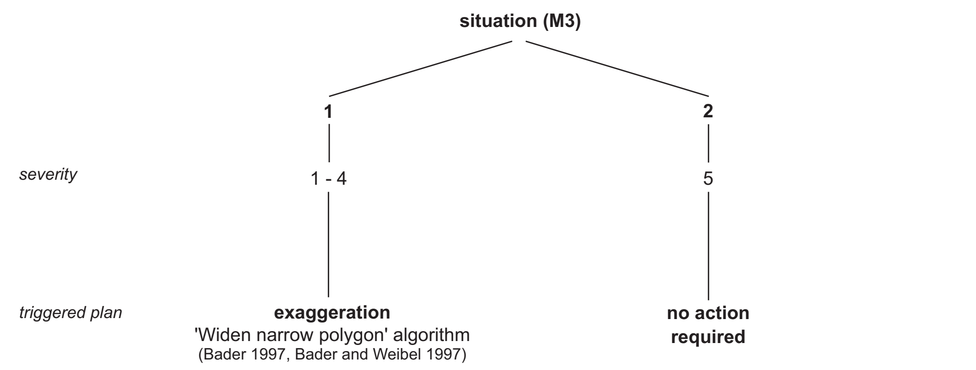
Spatial level	Polygon		
Goal value	minimal distance > 0.6 mm		
Importance	3 / 6	Priority	2 / 4
Measure	Detect narrow sections (Bader 1997, Bader and Weibel 1997)		

Evaluation method

severity / *criteria**	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
measure / goal value	< = 25 %	25 % - < = 50 %	50 % - < = 70 %	70 % - < = 90 %	> 90 %

* derived from empirical testing

Decision tree



M4 Minimal area

All polygon objects should have at least a minimal area for the given target scale. In general, objects should "be large enough for the reader to see and differentiate areal patterns" (Dent 1990, p. 152).

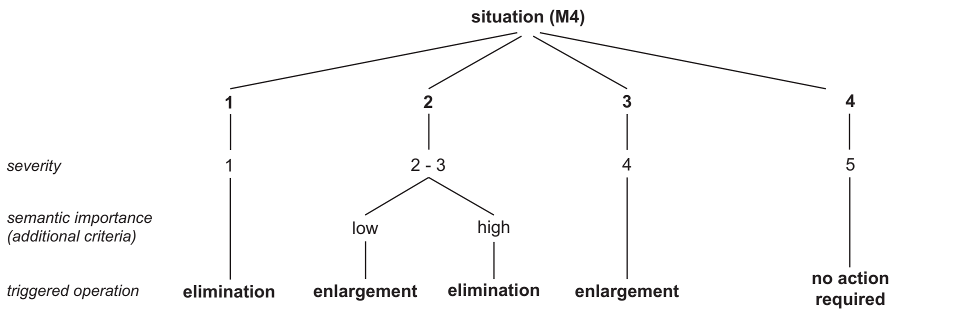
Spatial level	Polygon		
Goal value	minimal area > 4 mm^2		
Importance	5 / 6	Priority	4 / 4
Measure	Polygon area (Laser-Scan 1999)		

Evaluation method

severity / *criteria**	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
measure / goal value	< = 25 %	25 % - < = 50 %	50 % - < = 75 %	75 % - < = 95 %	> 95 %

* derived from empirical testing

Decision tree



elimination *context dependent:* Assignment of polygon to semantically closest adjacent neighbor *or* divide up polygon among neighbors (Bader and Weibel 1997)

enlargement *1. plan* Snakes-based enlargement (Chapt. 5), *2. plan* Simple scaling algorithm (Laser-Scan 1999)

M5 Respect spatial context			
Individual polygons and groups of polygons should respect their spatial context in conflict resolution. In other words, this constraint prevents the creation of new conflicts between generalized polygons or groups of polygons and other polygons that are not generalized at the same time. For instance, a group of disjoint island polygons should respect their spatial context, that is, the polygon that embeds them.			
Spatial level	Polygon, Group		
Goal value	TRUE		
Importance	4 / 6, 3 / 4	Priority	1 / 4, 1 / 2
Measure	Intersection of polygon geometry and context geometry (Section 6.4) $geom_G \cap geom_{Con} = geom_G$		
Evaluation method	NOT IMPLEMENTED		
Decision tree			

M6 Object separation

The distance between two disjoint polygons should be not less than a minimum distance.

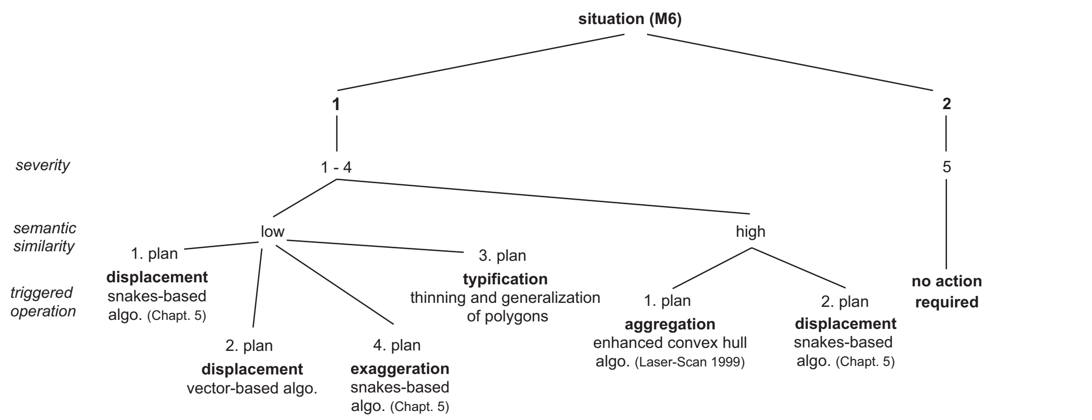
Spatial level	Group		
Goal value	minimal distance > 0.6 mm		
Importance	2 / 4	Priority	1 / 2
Measure	minimum distance between objects $d_{Obj} = min(\sum_{i=1}^{n}\sum_{j=1}^{m} dist(O_i, O_j))$		

Evaluation method

criteria* \ severity	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
measure / goal value	< = 40 %	40 % - < = 60 %	60 % - < = 80 %	80 % - < = 90 %	> 90 %

* derived from empirical testing

Decision tree



M7 Number of categories

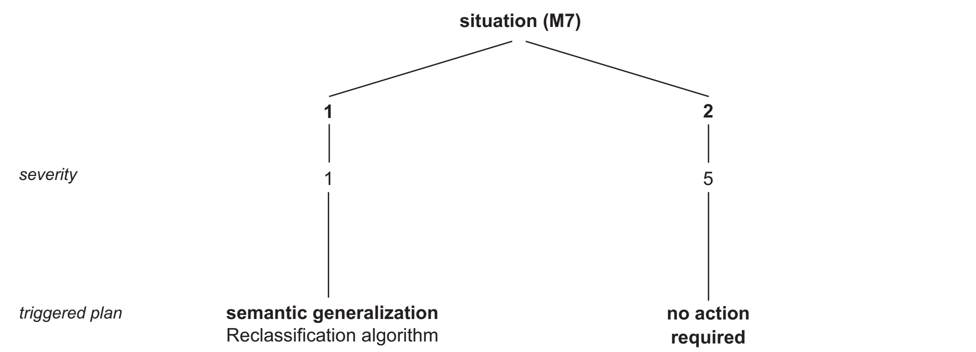
The number of retained categories is closely linked to the spatial detail of a polygonal subdivision since the more categories are shown the more polygons will be portrayed. The target scale, the map purpose (e.g. a geology map for a tourist vs. an expert in geology) and the map theme determine the concrete number of categories. Due to its dependence on the specific map that needs to be generalized this constraint represents a typical case where no global rules exist but the user will specify the corresponding threshold values.

Spatial level	Map		
Goal value	varying		
Importance	2 / 2	Priority	2 / 2
Measure	Number of represented categories (n_{categ})		

Evaluation method

severity	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
criteria	n_{categ} > goal value				n_{categ} <= goal value

Decision tree



T1 Self-intersection

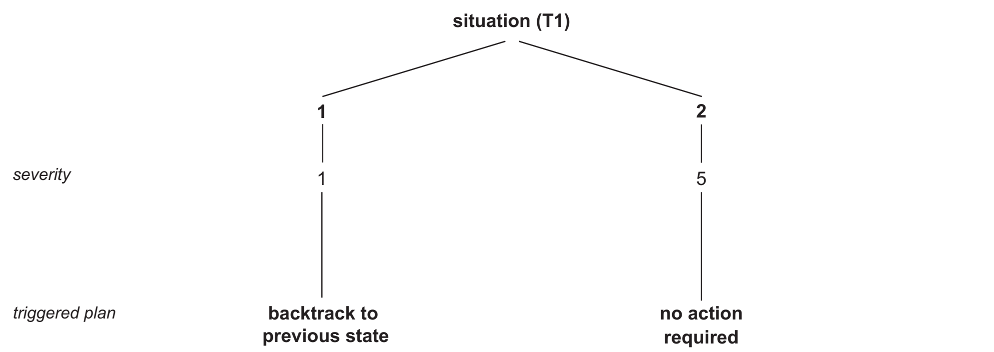
A valid polygonal subdivision of the plane - see Frank et al. (1997) for a definition of a polygonal subdivision - must not contain self-intersecting polygon geometries.

Spatial level	Line		
Goal value	True		
Importance	2 / 2	Priority	
Measure	Test line for self-intersection.		

Evaluation method

severity	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
criteria	invalid topology				valid topology

Decision tree



T2 Intersection of different polygons

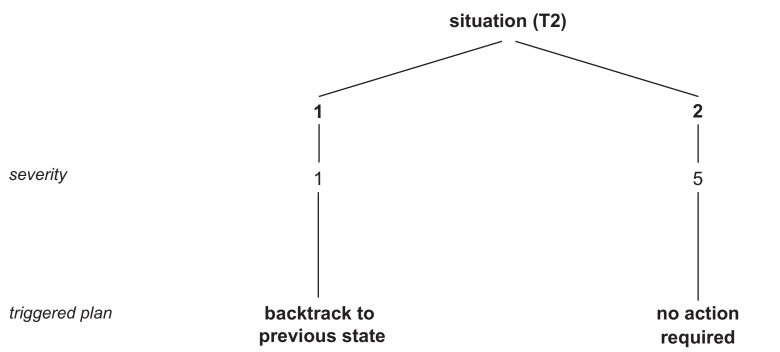
Intersections of polygon geometries must be avoided since they prohibit the creation of a topologically consistent polygonal subdivision.

Spatial level	Polygon, Group		
Goal value	True		
Importance	6 / 6, 4 / 4	Priority	
Measure	Test for intersections of polygon rings and different polygons.		

Evaluation method

severity	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
criteria	invalid topology				valid topology

Decision tree



S1 Shape distortion			
The distortion of a polygon shape should be minimized, that is, shape characteristics such as angularity or intrinsic micro shapes should change as little as possible.			
Spatial level	Polygon		
Goal value			
Importance	1 / 6	Priority	
Measure	Comparison of the original polygon shape to the generalized polygon shape through a perimeter to area ratio. $P : A = P' : A'$		
Evaluation method	Only tracked and not used in decision making.		
Decision tree			

S2 Absolute position			
The change of an object's absolute position should be minimized.			
Spatial level	Polygon		
Goal value			
Importance	1 / 6	Priority	
Measure	Relative area overlap of original and generalized geometry $geom_{orig} \cap geom_{gen} = geom_{common}$		
Evaluation method	Only tracked and not used in decision making.		
Decision tree			

S3 Relative configuration			
Generalization should maintain as best as possible the direction and distance relations of objects. That is, generalization should preserve not only the positions of polygons, relative to each other, but also characteristics in the spatial distribution of polygons such as alignments, clusters and containments.			
Spatial level	Group		
Goal value			
Importance	1 / 4	Priority	
Measure	Preliminary approximation: Number of objects in groups.		
Evaluation method	Only tracked and not used in decision making.		
Decision tree			

S4 Size ratios			
Size ratios should be preserved in a polygonal subdivision on different levels during generalization, for instance, between polygons of an alignment or a cluster, between polygons of a category and between all categories building a polygonal subdivision.			
Spatial level	Group, Map		
Goal value			
Importance	1 / 4, 1 / 2	Priority	
Measure	Relative area values, for instance the relative area of a category of the total subdivision. $A_{categ} : A_{tot} = A'_{categ} : A'_{tot}$		
Evaluation method	Only tracked and not used in decision making.		
Decision tree			

P1 Illogical results

Generalization should not produce results that are implausible with respect to the spatial (e.g. a phenomenon occurring in compact polygons shown by long and thin polygons) or the semantic component (e.g. impossible neighborhoods of categories) of the represented theme.

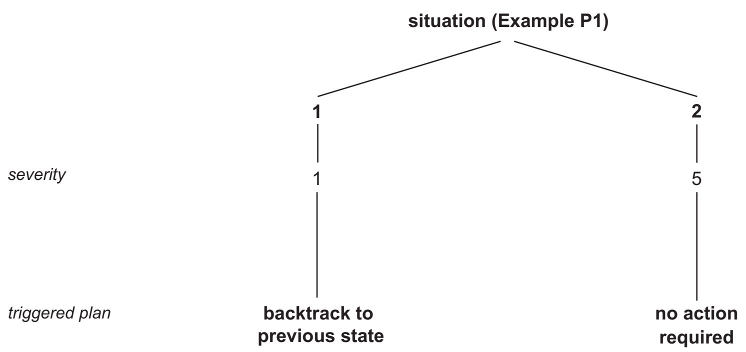
Example:
Maintain island polygons and prevent the creation of new island polygons.

Spatial level	Polygon, Group, Map; *Ex.* Polygon		
Goal value	varying		
Importance	varying, *Ex.* 6 / 6	Priority	
Measure	Varying with respect to the kind of illogical result. *Example:* Count number of island polygons (n_{island}).		

Evaluation method

severity / criteria	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
n_{island}	changed				constant

Decision tree



P2 Child entity's constraints

Both the hierarchical organization of the spatial levels of polygon generalization and the agent-based approach make so-called parent entities responsible for the generalization of their child entities. This constraint is attached to every parent entity in order to ensure sufficient satisfaction of all those constraints which are delegated to its individual child entities (Ruas 1999, Barrault et al. 2001).

Spatial level	Polygon, Group, Map		
Goal value	perfect ('5')		
Importance	4 / 6, 3 / 4, 2 / 2	Priority	3 / 4, 2 / 2, 1 / 2

Measure

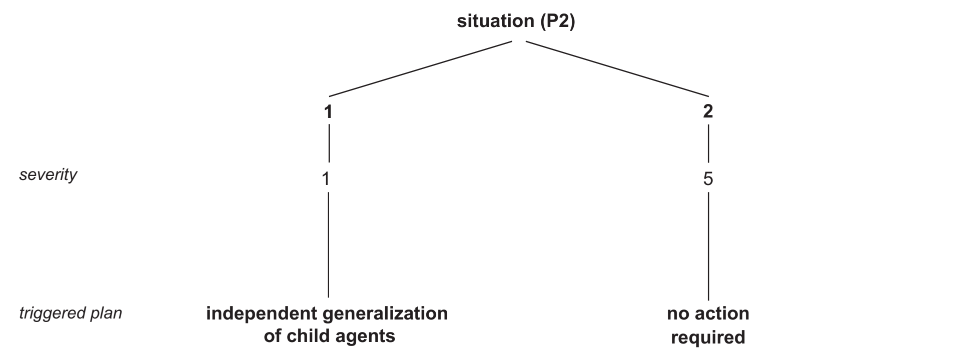
Average satisfaction of child entities

$$satisfaction_{child\_agents} = mean(\sum_{i=1}^{n} satisfaction(child\_agent_i))$$

Evaluation method

criteria \ *severity*	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
average satisfaction of child agents	< goal value				= goal value

Decision tree



P3 Aggregation similarity

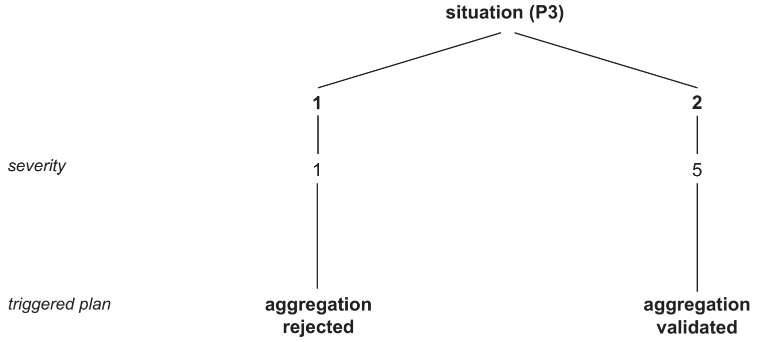
This constraint defines the minimum level of semantic similarity required to merge two polygons of different categories. It is used for the validation of a proposed aggregation operation.

Spatial level	Group		
Goal value	varying		
Importance		Priority	
Measure	Semantic similarity among polygons after Yaolin et al. (2002b).		

Evaluation method

severity / *criteria*	1 'very bad'	2 'bad'	3 'moderate'	4 'good'	5 'perfect'
semantic similarity	< goal value				> = goal value

Decision tree



P4 Equal treatment			
Ensure that similar conflicts are solved in similar ways across the polygonal subdivision.			
Spatial level	all levels		
Goal value			
Importance		Priority	
Measure	NOT IMPLEMENTED		
Evaluation method			
Decision tree			

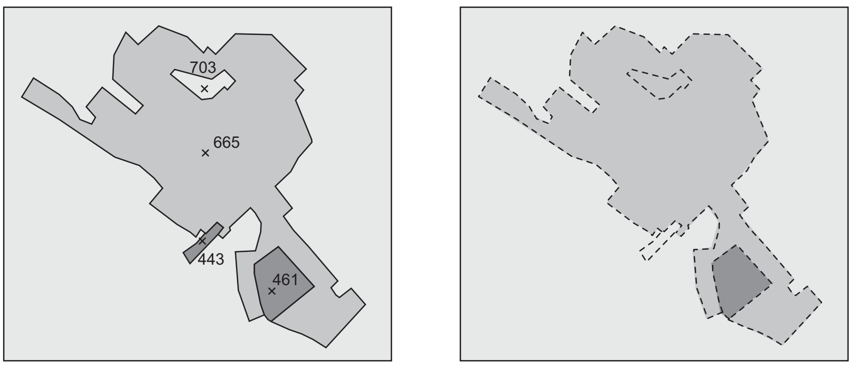
Appendix C

The evolution of two sample agents

This appendix presents the evolution of a group and a polygon agent during iterative generalization, that is each step in their generalization to a target scale 1:50,000 is stepped through and discussed.

C.1 The group agent `6`

The group agent `6` consists in its original state (Figure C.1a) of 4 polygon agents. They make up an island polygon (of the surrounding polygon) and, thus, were assigned to the group agent `6`. Given the current settings in the prototype, the generalization of a group agent is driven by the constraints 'M6 Object separation' and 'P2 Child entity's constraints' – see section 7.2.2. As in a group of connected polygons the minimal separation distance is controlled through the constraint 'M3 Distance between boundary points' at the polygon level the constraint 'M6 Object separation' is not considered in the generalization of the group agent `6`.



a. b.

Figure C.1: The group agent `6` **a.** before generalization with the identifiers of its associated polygon agents, and **b.** after generalization for a scale 1:50,000. The original situation is portrayed in dashed lines and the generalized one in solid contours. Figure not at scale. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

If the constraint 'P2 Child entity's constraints' is violated it proposes the independent generalization of its supervised child agents. In other words – given the current tuning of the prototype system – all the polygon agents are triggered in turn that do not meet their associated constraints.

However, if a polygon agent has once failed to accomplish a perfect satisfaction of its constraints it is not considered to be a potential candidate for independent generalization. After a polygon agent has completed its independent generalization its modified geometry and semantic is propagated to its first order neighbors. Next, the satisfaction of the group agent is re-evaluated and the group agent's life cycle continues – see section 4.3[1].

		states of the group agent 6				
		0	1	2	3	4
satifaction of	Polygon agent 703	*2*	(5)	(5)	(5)	(5)
	Polygon agent 665	*2*	*2*	4	4	4
	Polygon agent 461	*3*	*3*	*3*	5	5
	Polygon agent 443	*1*	*1*	*1*	*1*	(5)
	P2 Child entity's constraints	2	2.75	3.25	3.75	4.75

Table C.1: The satisfaction of the group agent 6 and its associated polygon agents across the states. The satisfaction of polygons prior to their generalization is written in italic while the pseudo perfect satisfaction of polygon agents is listed as '(5)'.

The generalization of the group agent 6 does not include any backtracks – see Figure C.3, that is, the polygon agents are triggered one after the other until all polygon agents have completed their independent generalization. The polygon agents 703 and 443 are removed due to conflicts with the constraint 'M4 Minimal area' at the polygon level. The satisfaction of this group agent improves continuously from 2.0 at the start (*state0*) to 2.75, 3.35 and 3.75 at intermediate states and to 4.74 at the final state (*state4*) – see Table C.1[2]. Figure C.1b depicts the generalized group agent.

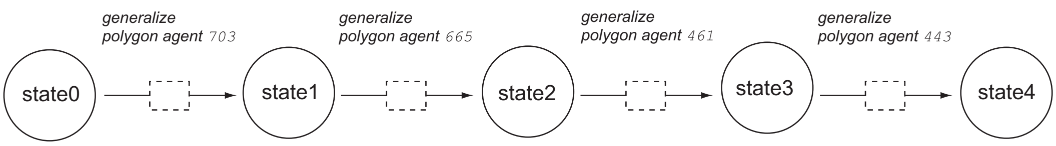


Figure C.2: States of the group agent 6.

C.2 The polygon agent 665

In the prototype system, the generalization of polygon agents is controlled by the constraints 'M1 Consecutive vertex distance', 'M2 Outline granularity', 'M3 Distance between boundary points', 'M4 Minimal area' and 'P1 Illogical results' (cf. chapter 6). The following paragraphs describe and argue about each step in the polygon agent's generalization. The discussion is supported by:

- Figure C.3 that visualizes the state tree of the polygon agent,
- Figure C.4 that illustrates the polygon agent at different generalization states,

[1] In doing so, a polygon agent may be decreased in its satisfaction if it helps to improve the satisfaction of the group agent.

[2] The evaluation method of the constraint 'P2 Child entity's constraints' considers eliminated child agents to have 'perfect'(5) satisfaction.

- Table C.2 that lists the satisfaction of the polygon agent[3] and its attached constraints at the different states., and
- Section C.3 lists the text output generated by the prototype system during the polygon agent's generalization.

		states of the polygon agent 665									
		4	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
satifaction of	M1 Consecutive vertex distance	1	5	5	5	5	5	5	5	5	5
	M2 Outline granularity	4	4	4	3	3	3	3	4	3	5
	M3 Distance btw. boundary points	2	3	5	5	5	3	3	3	3	5
	M4 Minimal area	5	5	5	5	5	5	5	5	5	5
	P1 Illogical result	5	5	5	5	5	5	5	5	5	1
	Polygon agent 665	*1*	*3*	*4*	*3*	*3*	*3*	*3*	*3*	*3*	*1*

Table C.2: The satisfaction of the polygon agent 665 and its associated constraints at the different generalization states. The final state is highlighted through a gray fill color of the corresponding cells.

Since the polygon agent 665 has stepped through states before – in order to adapt to generalized neighbors – its independent generalization starts at the *state4*.

state4. A preliminary evaluation of the constraints associated with the polygon agent determines the initial need of the polygon agent for generalization (cf. section 4.3). The polygon agent 665 does not respect the constraints 'M1 Consecutive vertex distance', 'M2 Outline granularity' and 'M3 Distance between boundary points'. Since the first constraint has the highest priority of all the violated constraints (cf. section 6.7) the generalization process begins with the attempt to improve the compliance of the polygon agent to this constraint. In practice, the enhanced Douglas-Peucker algorithm (Laser-Scan 1999)[4] with a predefined simplification parameter (i.e. 0.1 mm in map units) is executed. It leads to *state5*.

state5. The re-evaluation of the constraints attached to the polygon agent shows that the triggered plan accomplished the desired improvement in the satisfaction of the constraint 'M1 Consecutive vertex distance' - compare Table C.1, that is, this state is better than the previous state. Hence, a new list of plans is proposed for the resolution of the remaining conflicts. Due to the relative priorities of the unsatisfied constraints (i.e. 'M2 Outline granularity' and 'M3 Distance between boundary points'), the constraint 'M3 Distance between boundary points' proposes the next plan. It runs the algorithm 'Widen narrow polygon' (Bader and Weibel 1997) and accomplished the result stored in *state6*.

state6. Again an improvement in the satisfaction of the polygon agent and its attached constraints, respectively, is achieved. Thus, *state6* becomes the best state reached so far. When comparing this state (Figure C.4b) with the polygon's original state (Figure C.4a) only very little changes in the polygon geometry can be observed. However, these modifications established a significant improvement in the polygon agent's happiness. In other words, the polygon agent now meets all its attached constraints with the exception of the constraint 'M2 Outline granularity'. This constraint proposes two plans plan that refer to different approaches of outline simplification,

[3]The satisfaction of a polygon agent is defined, here, as the minimum satisfaction of its attached constraints.
[4]The corresponding references to an algorithms are only given the first time an algorithm is mentioned.

namely the Visvalingam-Whyatt (Visvalingam and Whyatt 1993) and the ‘Remove micro shapes’ algorithm (modified after Müller and Wang (1992)). Due to a higher weight of the first plan – see appendix B – the Visvalingam-Whyatt is triggered.

state7. Although the plan accomplished a – from a visual point of view – simplified polygon shape (Figure C.4a) the satisfaction of the constraint ‘M2 Outline granularity’ even decreases. Since the measure used for the evaluation of this constraint considers the smallest micro shape detected in a polygon outline the occurrence of single micro shape allows the rejection of a state. Hence, a backtrack from *state7* to the previous state, i.e. *state6*, is performed.

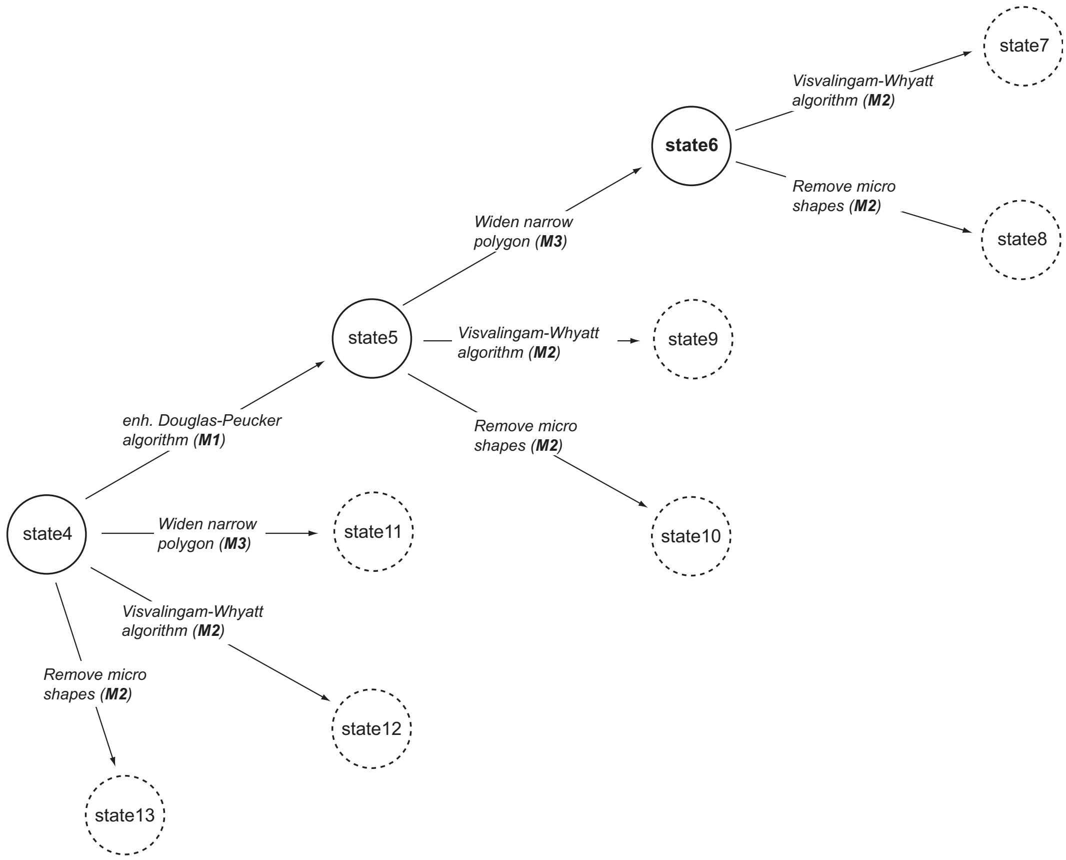


Figure C.3: States of the polygon agent 665. The states that achieved an improvement in the agents' satisfaction are represented by circles with solid outlines, while the backtracked states are illustrated by circles with dashed outlines.

state6. The next plan of the list of plans established previously for this state, that is, the ‘Remove micro shapes’ algorithm) is executed. It leads to *state8*.

state8. As no further improvement of the constraints' satisfaction occurs it is backtracked to *state6*.

state6. At this state no plan to be tried out is left. Thus, a backtrack to its parent state, i.e. *state5*, is performed. However, this state remains the best state reached for the polygon agent 665 so far.

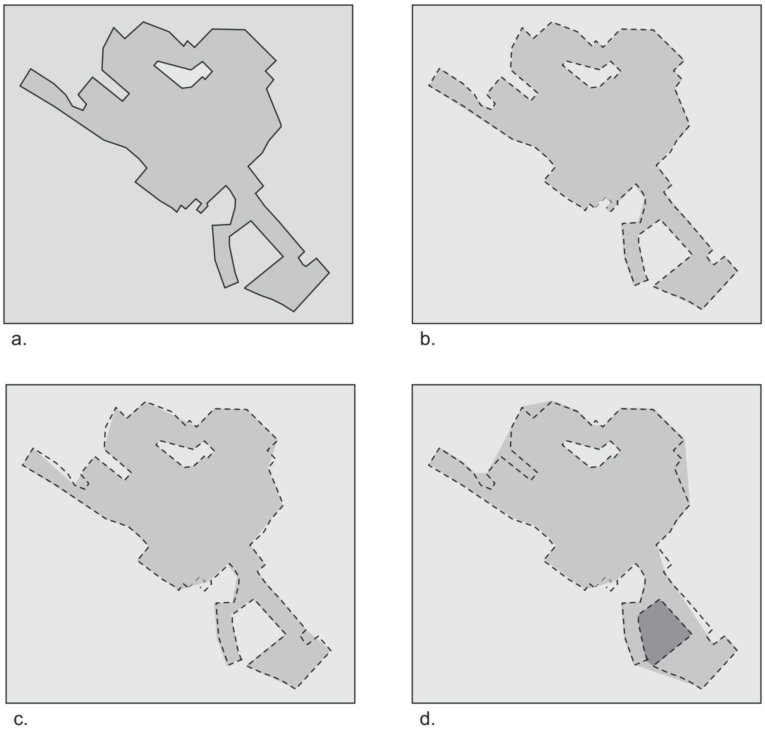


Figure C.4: Selected generalization states of the polygon agent 665. **a.** Original state: *state4*. **b.** Best state reached in generalization: *state6*. **c.** Refused state *state7*. **d.** Refused state *state13*. Data: VECTOR25, reproduced by permission of swisstopo (BA035273).

state5. Since neither the plan running the Visvalingam-Whyatt algorithm (*state9*) nor the plan implementing the 'Remove micro shapes algorithm' (*state10*) lead to an improvement in the agent's happiness a backtrack to *state4* is triggered.

state4. At this state the list of plans contains three more plans. Due to the prioritization of constraints (cf. section 6.7) the next plan triggered is the 'Widen narrow polygon' algorithm proposed by the constraint 'M3 Distance between boundary points'. Its result is stored in *state11*.

state11. Although an improvement of the agent's compliance to the constraints 'M3 Distance between boundary points' and 'M2 Outline granularity' is achieved this state is backtrack because of the modified hill-climbing algorithm that directs the search for the best solution (Regnauld 2001). It approves a new state only if at least one constraint exhibits a better satisfaction in the new state than in one of the previous states (Barrault et al. 2001). In doing so, infinite loops in the iterative generalization process are effectively avoided (Regnauld in press). Along these lines, the generalization process again returns to *state4*.

state4. The next plan at *state4* intends to improve the constraint 'M2 Outline granularity' through a Visvalingam-Whyatt algorithm. It fails to provide an improvement in constraints' satisfaction and, thus, is backtracked. The next plan applies the 'Remove micro shapes' algorithm.

state13. This algorithm achieves even a solution that allows the agent to meet all its associated metric constraints. However, this solution violates the constraint ‘P1 Illogical results’ (cf. section 7.2.2). Since the polygon generated for *state13* would enclose another polygon that has not been enclosed before (Figure C.4d)the constraint ‘P1 Illogical results’ rejects this solution.

state4. No plans to be tried out are left at *state4*. Hence, the independent generalization of the polygon agent 665 terminates. Consequently, the best solution the one in *state6* is used to update the agent. The polygon agent is set to passive again and the control of the generalization process is returned to its parent agent, the group agent 6.

C.3 System output generated during the generalization of the polygon agent 665

This section lists the system output generated by the prototype during the generalization of the polygon agent 665:

```
Constraint ‘areas-micro-agents-constraint’ proposes the following
plans :

Plan: <agent_areas_trigger_and_diffuse>, weight=1. p1=[121,48458795,665]

ORDERING THE PLANS PROPOSED BY THE CONSTRAINTS Execution list of 1
plans>>> 424.50000 ==>
areas-micro-agents-constraint|agent_areas_trigger_and_diffuse

EXECUTING PLAN ‘agent_areas_trigger_and_diffuse’ [  1 of   1] with params:
weight=1.0 p1=[121,48458795,665]

Meso-agent: Execute 665
        ***************************************************
        ** Running agent <Z_Siedl, 665> State = ACTIVE. **
        ***************************************************
state 4 storing geometry with 1

CHARACTERISING AND EVALUATING Micro-agent <665>
************CHARACTERISING AGENT 665 for state ID 4

 Constraint <pagent-poly-min-area-con>, current_value = 153302.81222 157224.04727
 Constraint <pagent-poly-min-area-con>,
 severity = 5.0 (priority <2.00> importance <1.00>)

 smallest consecutive vertex distance found: 1.702940
 Constraint <pagent-poly-consec-vert-dist-con>, current_value =1.70294
 Constraint<pagent-poly-consec-vert-dist-con>,
 severity = 2.0 (priority <1.00> importance <2.00>)

 Constraint <pagent-poly-micro-crenulations-con>,
 current_value = 24.16278 18.55311 30.33764 19.85597 max_below is 0.093247 at 0.700000
 Constraint <pagent-poly-micro-crenulations-con>,
 severity = 4.0 (priority <4.00> importance <1.00>) 504858.100001 142573.199999

 narrowest part is 12.203688
 Constraint <pagent-poly-narrow-parts-con>,
```

```
 current_value = 8.00000 12.20369 min_width/goal = 0.406790,
 Constraint <pagent-poly-narrow-parts-con>, s
 everity = 2.0 (priority <3.00> importance <2.00>)

 Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
 Constraint <pagent-poly-topo-neis-con>,
 severity = 5.0 (priority <1.00> importance <1.00>)

 Constraint <pagent-poly-shape-distort-con>,
 current_value = 0.00000 0.02352 0.02352
 ===============================
 ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.023518
 ===============================
 Constraint <pagent-poly-shape-distort-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 Constraint <pagent-poly-abs-position-con>, current_value = 1.00000
 ===============================
 RELATIVE_COMMON_AREA = 1.000000
 ===============================
 Constraint <pagent-poly-abs-position-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 Compute Happiness of 665 HAPPINESS OF THE OBJECT : 2
 PROPOSING PLANS Plans proposed>>>
 Constraint 'pagent-poly-consec-vert-dist-con' proposes the following plans :
   Plan: <pagent_poly_outline_simplify_algo>,
     weight=0.5 current=-999.0 p1=5.0 p2=0.0 p3=0.0 p4=0.0 p5=0.0
 Constraint 'pagent-poly-micro-crenulations-con' proposes the following plans :
   Plan: <pagent_poly_outline_viswya_simplify_algo>,
     weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0
   Plan: <pagent_poly_remove_micro_shapes_algo>,
     weight=0.5 current=-999.0 p1=31.3 p2=0.0 p3=0.0 p4=0.0 p5=0.0
 Constraint 'pagent-poly-narrow-parts-con' proposes the following plans :
   Plan: <pagent_poly_inner_exaggeration_vec_algo>,
     weight=0.8 p1=0.6 p2=0.0 p3=0.0 p4=0.0 p5=0.0

 ORDERING THE PLANS PROPOSED BY THE CONSTRAINTS Execution list of 4 plans>>>
 111.50000 ==> pagent-poly-micro-crenulations-con|pagent_poly_remove_micro_shapes_algo
 111.80000 ==> pagent-poly-micro-crenulations-con|pagent_poly_outline_viswya_simplify_algo
 231.80000 ==> pagent-poly-narrow-parts-con|pagent_poly_inner_exaggeration_vec_algo
 431.50000 ==> pagent-poly-consec-vert-dist-con|pagent_poly_outline_simplify_algo

 EXECUTING PLAN 'pagent_poly_outline_simplify_algo' [  1 of 4] with
 params : weight=0.5 current=-999.0 p1=5.0 p2=0.0 p3=0.0 p4=0.0 p5=0.0

state 5 storing geometry with 1

 RE-EVALUATION OF THE (MICRO)AGENT >Re-characterise the constraints...
 ************CHARACTERISING AGENT 665 for state ID 5

 Constraint <pagent-poly-min-area-con>, current_value = 154129.64428 158050.87933
 Constraint <pagent-poly-min-area-con>,
 severity = 5.0 (priority <2.00> importance <1.00>)

 smallest consecutive vertex distance found: 1.702940
 Constraint <pagent-poly-consec-vert-dist-con>, current_value = 1.70294
```

```
 Constraint <pagent-poly-consec-vert-dist-con>,
 severity = 5.0 (priority <1.00> importance <2.00>)

 Constraint <pagent-poly-micro-crenulations-con>,
 current_value = 24.16278 18.58729 30.33764 19.85597 max_below is 0.093247 at 0.700000
 Constraint <pagent-poly-micro-crenulations-con>,
 severity = 4.0 (priority <4.00> importance <1.00>)

 narrowest part is 20.273131
 Constraint <pagent-poly-narrow-parts-con>,
 current_value = 3.00000 20.27313 min_width/goal = 0.675771, severity = 3.000000
 Constraint <pagent-poly-narrow-parts-con>,
 severity = 3.0 (priority <3.00> importance <2.00>)

 Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
 Constraint <pagent-poly-topo-neis-con>,
 severity = 5.0 (priority <1.00> importance <1.00>)

 Constraint <pagent-poly-shape-distort-con>, current_value = 0.00017 0.02352 0.02335
 ===============================
 ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.023346
 ===============================
 Constraint <pagent-poly-shape-distort-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 Constraint <pagent-poly-abs-position-con>, current_value = 1.00295
===============================
RELATIVE_COMMON_AREA = 1.002947
===============================
 Constraint <pagent-poly-abs-position-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 >Checking improvement of the handled constraint(s),
   i.e.: in _agent_check_severity_improvement
  Handled constraint pagent-poly-consec-vert-dist-con has improved from 2.0 to 5.0
  >Checking that the other constraints have not been too much damaged...
  in _agent_validate_severity_change
  Constraint pagent-poly-abs-position-con <importance 7.0> has changed from 5.0 to 5.0
  acceptance is true
  in _agent_validate_severity_change
 Constraint pagent-poly-consec-vert-dist-con <importance 2.0> has changed from 2.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-micro-crenulations-con <importance 1.0> has changed from 4.0 to 4.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-min-area-con <importance 1.0> has changed from 5.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-narrow-parts-con <importance 2.0> has changed from 2.0 to 3.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-shape-distort-con <importance 7.0> has changed from 5.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-topo-neis-con <importance 1.0> has changed from 5.0 to 5.0
 acceptance is true
```

```
Compute Happiness of 665 Re-evaluation result: change = Better,
old happiness = 2.0, new happiness = 3.0
Plan improved happiness ref 2.000000 and new 3.000000
Current state compared with the best state...
Current state best so far

PROPOSING PLANS Plans proposed>>>
Constraint ‘pagent-poly-micro-crenulations-con’ proposes the following plans :
 Plan: <pagent_poly_outline_viswya_simplify_algo>,
   weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0
 Plan: <pagent_poly_remove_micro_shapes_algo>,
   weight=0.5 current=-999.0 p1=31.3 p2=0.0 p3=0.0 p4=0.0 p5=0.0
Constraint ‘pagent-poly-narrow-parts-con’ proposes the following plans :
 Plan: <pagent_poly_inner_exaggeration_vec_algo>,
   weight=0.8 p1=0.6 p2=0.0 p3=0.0 p4=0.0 p5=0.0

ORDERING THE PLANS PROPOSED BY THE CONSTRAINTS Execution list of 3 plans>>>
111.50000 ==> pagent-poly-micro-crenulations-con|pagent_poly_remove_micro_shapes_algo
111.80000 ==> pagent-poly-micro-crenulations-con|pagent_poly_outline_viswya_simplify_algo
221.80000 ==> pagent-poly-narrow-parts-con|pagent_poly_inner_exaggeration_vec_algo

EXECUTING PLAN ‘pagent_poly_inner_exaggeration_vec_algo’ [  1 of 3]
with params : weight=0.8 p1=0.6 p2=0.0 p3=0.0 p4=0.0 p5=0.0

state 6 storing geometry with 1

RE-EVALUATION OF THE (MICRO)AGENT >Re-characterise the constraints...

************CHARACTERISING AGENT 665 for state ID 6
Constraint <pagent-poly-min-area-con>, current_value = 154904.58932 158825.82437
Constraint <pagent-poly-min-area-con>,
severity = 5.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 1.702940
Constraint <pagent-poly-consec-vert-dist-con>, current_value =
1.70294 consec_vert_d in mm hhh 0.034059
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 5.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value = 24.16278 18.58729 30.33764 19.85597 max_below is 0.093247 at 0.700000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 4.0 (priority <4.00> importance <1.00>)

narrowest part is 27.829749
Constraint <pagent-poly-narrow-parts-con>, current_value = 2.00000 27.82975
min_width/goal = 0.927658, severity = 5.000000
Constraint <pagent-poly-narrow-parts-con>,
severity = 5.0 (priority <3.00> importance <2.00>)

Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 5.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value = 0.00035 0.02352 0.02317
==============================
```

```
ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.023171
===============================
Constraint <pagent-poly-shape-distort-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

Constraint <pagent-poly-abs-position-con>, current_value = 1.00255
===============================
RELATIVE_COMMON_AREA = 1.002548
===============================
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-narrow-parts-con has improved from 3.0 to 5.0
>Checking that the other constraints have not been too much damaged...
in _agent_validate_severity_change
Constraint pagent-poly-abs-position-con <importance 7.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-consec-vert-dist-con <importance 2.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-micro-crenulations-con <importance 1.0> has changed from 4.0 to 4.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-min-area-con <importance 1.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-narrow-parts-con <importance 2.0> has changed from 3.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-shape-distort-con <importance 7.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-topo-neis-con <importance 1.0> has changed from 5.0 to 5.0
acceptance is true

Compute Happiness of 665 Re-evaluation result: change = Better,
old happiness = 3.0, new happiness = 4.0
Plan improved happines ref 3.000000 and new 4.000000
Current state compared with the best state...
Current state best so far

PROPOSING PLANS Plans proposed>>>
Constraint ‘pagent-poly-micro-crenulations-con’ proposes the following plans :
 Plan: <pagent_poly_outline_viswya_simplify_algo>,
   weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0
 Plan: <pagent_poly_remove_micro_shapes_algo>,
   weight=0.5 current=-999.0 p1=31.3 p2=0.0 p3=0.0 p4=0.0 p5=0.0
ORDERING THE PLANS PROPOSED BY THE CONSTRAINTS Execution list of 2 plans>>>
111.50000 ==> pagent-poly-micro-crenulations-con|pagent_poly_remove_micro_shapes_algo
111.80000 ==> pagent-poly-micro-crenulations-con|pagent_poly_outline_viswya_simplify_algo

EXECUTING PLAN ‘pagent_poly_outline_viswya_simplify_algo’ [ 1 of 2]
with params : weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0 state
7 storing geometry with 1
```

```
RE-EVALUATION OF THE (MICRO)AGENT
>Re-characterise the constraints...
************CHARACTERISING AGENT 665 for state ID 7
Constraint <pagent-poly-min-area-con>, current_value = 154745.58755 158666.82261
Constraint <pagent-poly-min-area-con>,
severity = 5.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 22.545999
Constraint <pagent-poly-consec-vert-dist-con>, current_value = 22.54600
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 5.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value =-1.00000 -1.00000 26.31108 -1.00000 max_below is 0.173778 at 0.700000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 3.0 (priority <4.00> importance <1.00>)

narrowest part is 27.829749 Constraint
<pagent-poly-narrow-parts-con>, current_value = 2.00000 27.82975
min_width/goal = 0.927658, severity = 5.000000
Constraint <pagent-poly-narrow-parts-con>,
severity = 5.0 (priority <3.00> importance <2.00>)

Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 5.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value = 0.00129 0.02352 0.02223
===============================
ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.022226
===============================
Constraint <pagent-poly-shape-distort-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

Constraint <pagent-poly-abs-position-con>, current_value = 1.01452
===============================
RELATIVE_COMMON_AREA = 1.014522
===============================
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-micro-crenulations-con has not improved sufficiently :
previous_severity = 4.0, current_severity = 3.0

Compute Happiness of 665
Re-evaluation result: change = Worse, old happiness = 4.0, new happiness = 3.0
Plan did not improve happiness ... Backtracking

Backtracking state from 7 to 6

Compute Happiness of 665 HAPPINESS OF THE OBJECT : 4

EXECUTING PLAN ‘pagent_poly_remove_micro_shapes_algo’ [  2 of   2]
with params : weight=0.5 current=-999.0 p1=31.3 p2=0.0 p3=0.0 p4=0.0 p5=0.0
```

```
 VALIDATION ERROR but we coninue since considering the outcome as worse - hopefully
 state 8 storing geometry with 1

 Plan did not improve happiness ... Backtracking
 Backtracking state from 8 to 6
 Compute Happiness of 665
 HAPPINESS OF THE OBJECT : 4

Plan did not improve happiness ... Backtracking
Backtracking state from 6 to 5
Compute Happiness of 665
HAPPINESS OF THE OBJECT : 3

 EXECUTING PLAN 'pagent_poly_outline_viswya_simplify_algo' [  2 of   3]
 with params : weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0
 state 9 storing geometry with 1

 RE-EVALUATION OF THE (MICRO)AGENT
 >Re-characterise the constraints...
 ************CHARACTERISING AGENT 665 for state ID 9

 Constraint <pagent-poly-min-area-con>, current_value = 153923.45083 157844.68588
 Constraint <pagent-poly-min-area-con>,
 severity = 5.0 (priority <2.00> importance <1.00>)

 smallest consecutive vertex distance found: 22.545999
 Constraint <pagent-poly-consec-vert-dist-con>, current_value = 22.54600
 Constraint <pagent-poly-consec-vert-dist-con>,
 severity = 5.0 (priority <1.00> importance <2.00>)

 Constraint <pagent-poly-micro-crenulations-con>,
 current_value = -1.00000 -1.00000 26.31108 -1.00000 max_below is 0.173778 at 0.700000
 Constraint <pagent-poly-micro-crenulations-con>,
 severity = 3.0 (priority <4.00> importance <1.00>)

 narrowest part is 20.273131 Constraint
 <pagent-poly-narrow-parts-con>, current_value = 3.00000 20.27313
 min_width/goal = 0.675771, severity = 3.000000
 Constraint <pagent-poly-narrow-parts-con>,
 severity = 3.0 (priority <3.00> importance <2.00>)

 Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
 Constraint <pagent-poly-topo-neis-con>,
 severity = 5.0 (priority <1.00> importance <1.00>)

 Constraint <pagent-poly-shape-distort-con>, current_value = 0.00110 0.02352 0.02241
 ===============================
 ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.022414
 ===============================
 Constraint <pagent-poly-shape-distort-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 Constraint <pagent-poly-abs-position-con>, current_value = 1.01547
 ===============================
 RELATIVE_COMMON_AREA = 1.015471
 ===============================
```

```
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-micro-crenulations-con has not improved sufficiently :
previous_severity = 4.0, current_severity = 3.0

Compute Happiness of 665
Re-evaluation result: change = Worse,
old happiness = 3.0, new happiness = 3.0
Plan did not improve happiness... Backtracking
Backtracking state from 9 to 5

Compute Happiness of 665
HAPPINESS OF THE OBJECT : 3
EXECUTING PLAN ‘pagent_poly_remove_micro_shapes_algo’ [  3 of   3]
with params : weight=0.5 current=-999.0 p1=31.3 p2=0.0 p3=0.0 p4=0.0 p5=0.0

VALIDATION ERROR but we coninue since considering the outcome as worse - hopefully
state 10 storing geometry with 1

Plan did not improve happiness ... Backtracking
Backtracking state from 10 to 5
Compute Happiness of 665
HAPPINESS OF THE OBJECT : 3

Plan did not improve happiness ... Backtracking
Backtracking state from 5 to 4
Compute Happiness of 665
HAPPINESS OF THE OBJECT : 2

EXECUTING PLAN ‘pagent_poly_inner_exaggeration_vec_algo’ [  2 of   4]
with params : weight=0.8 p1=0.6 p2=0.0 p3=0.0 p4=0.0 p5=0.0 504858.100001 142573.199999
state 11 storing geometry with 1

5RE-EVALUATION OF THE (MICRO)AGENT
>Re-characterise the constraints...
************CHARACTERISING AGENT 665 for state ID 11

Constraint <pagent-poly-min-area-con>, current_value = 154115.42640 158036.66146
Constraint <pagent-poly-min-area-con>,
severity = 5.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 1.702940
Constraint <pagent-poly-consec-vert-dist-con>, current_value = 1.70294
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 5.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value = 24.16278 18.55311 30.33764 19.85597 max_below is 0.093247 at 0.700000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 4.0 (priority <4.00> importance <1.00>)

narrowest part is 20.187513
Constraint <pagent-poly-narrow-parts-con>, current_value = 6.00000 20.18751
min_width/goal = 0.672917, severity = 3.000000
```

```
Constraint <pagent-poly-narrow-parts-con>,
severity = 3.0 (priority <3.00> importance <2.00>)

Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 5.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value = 0.00018 0.02352 0.02334
===============================
ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.023343
===============================

Constraint <pagent-poly-shape-distort-con>, severity = 5.0 (priority <5.00> importance <7.00>)
Constraint <pagent-poly-abs-position-con>, current_value = 1.00021
===============================
RELATIVE_COMMON_AREA = 1.000212
===============================
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-narrow-parts-con has improved from 2.0 to 3.0
>Checking that the other constraints have not been too much damaged...
in _agent_validate_severity_change
Constraint pagent-poly-abs-position-con <importance 7.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-consec-vert-dist-con <importance 2.0> has changed from 2.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-micro-crenulations-con <importance 1.0> has changed from 4.0 to 4.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-min-area-con <importance 1.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-narrow-parts-con <importance 2.0> has changed from 2.0 to 3.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-shape-distort-con <importance 7.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-topo-neis-con <importance 1.0> has changed from 5.0 to 5.0
acceptance is true

Compute Happiness of 665
Re-evaluation result: change = Worse,
old happiness = 2.0, new happiness = 3.0
Plan did not improve happiness ... Backtracking
Backtracking state from 11 to 4

Compute Happiness of 665
HAPPINESS OF THE OBJECT : 2

EXECUTING PLAN 'pagent_poly_outline_viswya_simplify_algo' [  3 of   4]
with params : weight=0.8 current=-999.0 p1=602.4 p2=0.0 p3=0.0 p4=0.0 p5=0.0
```

```
state 12 storing geometry with 1

RE-EVALUATION OF THE (MICRO)AGENT
>Re-characterise the constraints...
************CHARACTERISING AGENT 665 for state ID 12

Constraint <pagent-poly-min-area-con>, current_value = 153961.93073 157883.16578
Constraint <pagent-poly-min-area-con>,
severity = 5.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 22.545999
Constraint <pagent-poly-consec-vert-dist-con>, current_value = 22.54600
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 5.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value = -1.00000 -1.00000 26.31108 -1.00000 max_below is 0.173778 at 0.700000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 3.0 (priority <4.00> importance <1.00>)

narrowest part is 20.273131
Constraint <pagent-poly-narrow-parts-con>, current_value = 3.00000 20.27313
min_width/goal = 0.675771, severity = 3.000000
Constraint <pagent-poly-narrow-parts-con>,
severity = 3.0 (priority <3.00> importance <2.00>)

Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 5.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value = 0.00137 0.02352 0.02215
===============================
ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.022149
===============================
Constraint <pagent-poly-shape-distort-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

Constraint <pagent-poly-abs-position-con>, current_value = 1.01632
===============================
RELATIVE_COMMON_AREA = 1.016319
===============================
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-micro-crenulations-con has not improved sufficiently :
previous_severity = 4.0, current_severity = 3.0

 Compute Happiness of 665
 Re-evaluation result: change = Worse,
 old happiness = 2.0, new happiness = 3.0
 Plan did not improve happiness ... Backtracking
 Backtracking state from 12 to 4

 Compute Happiness of 665 HAPPINESS OF THE OBJECT : 2
 EXECUTING PLAN 'pagent_poly_remove_micro_shapes_algo' [  4 of   4]
```

```
 with params : weight=0.5 current=-999.0
 state 13 storing geometry with 1

RE-EVALUATION OF THE (MICRO)AGENT
>Re-characterise the constraints...
************CHARACTERISING AGENT 665 for state ID 13

Constraint <pagent-poly-min-area-con>, current_value = 170146.68183 174067.91689
Constraint <pagent-poly-min-area-con>,
severity = 5.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 8.462861
Constraint <pagent-poly-consec-vert-dist-con>, current_value = 8.46286
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 5.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value = -1.00000 -1.00000 -1.00000 -1.00000 max_below is 0.000000 at 0.000000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 5.0 (priority <4.00> importance <1.00>)

narrowest part is 30.000000
Constraint <pagent-poly-narrow-parts-con>, current_value = 0.00000 30.00000
min_width/goal = 1.000000, severity = 5.000000
Constraint <pagent-poly-narrow-parts-con>,
severity = 5.0 (priority <3.00> importance <2.00>)

Constraint <pagent-poly-topo-neis-con>, current_value = -1.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 1.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value = 0.00688 0.02352 0.01664
===============================
ORIGINAL SHAPE INDEX = 0.023518 CURRENT SHAPE INDEX = 0.016642
===============================
Constraint <pagent-poly-shape-distort-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

Constraint <pagent-poly-abs-position-con>, current_value = 1.01669
===============================
RELATIVE_COMMON_AREA = 1.016690
===============================
Constraint <pagent-poly-abs-position-con>,
severity = 5.0 (priority <5.00> importance <7.00>)


>Checking improvement of the handled constraint(s), i.e.:
in _agent_check_severity_improvement
Handled constraint pagent-poly-micro-crenulations-con has improved from 4.0 to 5.0
>Checking that the other constraints have not been too much damaged...
in _agent_validate_severity_change
Constraint pagent-poly-abs-position-con <importance 7.0> has changed from 5.0 to 5.0
acceptance is true
in _agent_validate_severity_change
Constraint pagent-poly-consec-vert-dist-con <importance 2.0> has changed from 2.0 to 5.0
acceptance is true
in _agent_validate_severity_change
```

```
 Constraint pagent-poly-micro-crenulations-con <importance 1.0> has changed from 4.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-min-area-con <importance 1.0> has changed from 5.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-narrow-parts-con <importance 2.0> has changed from 2.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-shape-distort-con <importance 7.0> has changed from 5.0 to 5.0
 acceptance is true
 in _agent_validate_severity_change
 Constraint pagent-poly-topo-neis-con <importance 1.0> has changed from 5.0 to 1.0
 acceptance is false

 Compute Happiness of 665
 Re-evaluation result: change = Worse,
 old happiness = 2.0, new happiness = 1.0
 Plan did not improve happiness ... Backtracking
 Backtracking state from 13 to 4

 Compute Happiness of 665
 HAPPINESS OF THE OBJECT : 2

 highest state :=13, best state := 6, root state := 4

 Compute Happiness of 665
 Setting state of agent <Z_Siedl, 665> to PASSIVE
 Sending message  [ID= 5] from <Z_Siedl,665> to <agent_areas,4038>.
 Type <INFORM> Content <OK> Received message [ID=    5] from <Z_Siedl,665> to
 <agent_areas,4038>.
 Type <INFORM> Content <OK>
 @@@@@@@@@@@@ 703 udating @@@@@@@@@@2 @@@@@@@@@@@@ 461 udating @@@@@@@@@@

 2 Sending message [ID=    6] from <agent_areas,4038> to <Z_Reben,461>.
 Type <ORDER> Content <agent_set_micro_geometry()>
 Received message [ID=    6] from <agent_areas,4038> to <Z_Reben,461>.
 Type <ORDER> Content <agent_set_micro_geometry()>
        **************************************************
        ** Running agent <Z_Reben, 461> State = REACTIVE. **
        **************************************************
 Next message from buffer is>>> id=6 sender=[198,0,4038]
 recipient=[123,56740904,461] type=2
 content=agent_set_micro_geometry() state 2 storing geometry with 2

 EXECUTING PLAN 'agent_set_micro_geometry' [  1 of   1] with params:

 state 3 storing geometry with 2

 CHARACTERISING AND EVALUATING Micro-agent <461>
 ************CHARACTERISING AGENT 461 for state ID 3

Constraint <pagent-poly-min-area-con>, current_value = 9468.30976 9468.30976
 Constraint <pagent-poly-min-area-con>,
 severity = 4.0 (priority <2.00> importance <1.00>)

 smallest consecutive vertex distance found: 7.324616
```

```
 Constraint <pagent-poly-consec-vert-dist-con>, current_value = 7.32462
 Constraint <pagent-poly-consec-vert-dist-con>,
 severity = 5.0 (priority <1.00> importance <2.00>)
 Constraint <pagent-poly-micro-crenulations-con>,
 current_value = -1.00000 -1.00000 -1.00000 -1.00000 max_below is 0.000000 at0.000000
 Constraint <pagent-poly-micro-crenulations-con>,
 severity = 5.0 (priority <4.00> importance <1.00>)

 narrowest part is 18.597042 Constraint <pagent-poly-narrow-parts-con>,
 current_value = 1.00000 18.59704 min_width/goal = 0.619901, severity = 3.000000
 Constraint <pagent-poly-narrow-parts-con>,
 severity = 3.0 (priority <3.00> importance <2.00>)

 Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
 Constraint <pagent-poly-topo-neis-con>,
 severity = 5.0 (priority <1.00> importance <1.00>)

 Constraint <pagent-poly-shape-distort-con>, current_value = 0.00148 0.03994 0.04143
 ===============================
 ORIGINAL SHAPE INDEX = 0.039944 CURRENT SHAPE INDEX = 0.041426
 ===============================
 Constraint <pagent-poly-shape-distort-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 Constraint <pagent-poly-abs-position-con>, current_value = 1.04940
 ===============================
 RELATIVE_COMMON_AREA = 1.049405
 ===============================
 Constraint <pagent-poly-abs-position-con>,
 severity = 5.0 (priority <5.00> importance <7.00>)

 highest state := 3, best state := 3, root state := 2
 Compute Happiness of 461 Setting state of agent <Z_Reben, 461> to PASSIVE
 Sending message  [ID=    7] from <Z_Reben,461> to <agent_areas,4038>.
 Type <INFORM> Content <OK> Received message [ID=    7] from <Z_Reben,461> to <agent_areas,4038>.
 Type <INFORM> Content <OK>
 @@@@@@@@@@@@ 443 udating @@@@@@@@@@
 2 Sending message [ID=    8] from <agent_areas,4038> to <Z_Reben,443>.
 Type <ORDER> Content <agent_set_micro_geometry()>
 Received message [ID= 8] from <agent_areas,4038> to <Z_Reben,443>.
 Type <ORDER> Content <agent_set_micro_geometry()>

        ***************************************************
        ** Running agent <Z_Reben, 443> State = REACTIVE. **
        ***************************************************
 Next message from buffer is>>> id=8 sender=[198,0,4038]
 recipient=[123,60922672,443] type=2
 content=agent_set_micro_geometry() state 2 storing geometry with 2

 EXECUTING PLAN 'agent_set_micro_geometry' [  1 of   1] with params:

state 3 storing geometry with 2

 CHARACTERISING AND EVALUATING Micro-agent <443>
 ************CHARACTERISING AGENT 443 for state ID 3

 Constraint <pagent-poly-min-area-con>, current_value = 1853.06002 1853.06002
```

```
Constraint <pagent-poly-min-area-con>,
severity = 1.0 (priority <2.00> importance <1.00>)

smallest consecutive vertex distance found: 1.702939
Constraint <pagent-poly-consec-vert-dist-con>, current_value = 1.70294
Constraint <pagent-poly-consec-vert-dist-con>,
severity = 2.0 (priority <1.00> importance <2.00>)

Constraint <pagent-poly-micro-crenulations-con>,
current_value =-1.00000 18.03122 31.72381 3.35571 max_below is 0.332886 at 0.400000
Constraint <pagent-poly-micro-crenulations-con>,
severity = 1.0 (priority <4.00> importance <1.00>)

narrowest part is 18.031223
Constraint <pagent-poly-narrow-parts-con>, current_value = 2.00000 18.03122
min_width/goal = 0.601041, severity = 3.000000
Constraint <pagent-poly-narrow-parts-con>, severity = 3.0 (priority <3.00> importance <2.00>)


Constraint <pagent-poly-topo-neis-con>, current_value = 0.00000
Constraint <pagent-poly-topo-neis-con>,
severity = 5.0 (priority <1.00> importance <1.00>)

Constraint <pagent-poly-shape-distort-con>, current_value =
0.00000 0.12479 0.12479
===============================
ORIGINAL SHAPE INDEX = 0.124789 CURRENT SHAPE INDEX = 0.124789
===============================
Constraint <pagent-poly-shape-distort-con>,
severity = 5.0 (priority <5.00> importance <7.00>)

Constraint <pagent-poly-abs-position-con>, current_value = 1.00000
===============================
RELATIVE_COMMON_AREA = 1.000000
===============================
Constraint <pagent-poly-abs-position-con>, severity = 5.0 (priority <5.00> importance <7.00>)
highest state := 3,
best state := 3, root state := 2
Compute Happiness of 443
Setting state of agent <Z_Reben, 443> to PASSIVE
Sending message  [ID=    9] from <Z_Reben,443> to <agent_areas,4038>.
Type <INFORM> Content <OK>
Received message[ID=    9] from <Z_Reben,443> to <agent_areas,4038>.
Type <INFORM> Content <OK> RE-EVALUATION OF THE (MESO)AGENT
>Re-characterise the constraints...
************CHARACTERISING AGENT 4038 for state ID 4 am i
measuring elements before 4 703:3 removed 703 665:0 461:0 443:0
elements after 3 objects are 461 443 state_id = 4 measure ddict
```

Bibliography

AGENT consortium (2000) “AGENT Workpackage E4 - Evaluation of Laser-Scan Platform”, Technical report, Institut Géographique National France, University of Edinburgh, University of Zurich and Institut National Polytechnique de Grenoble.

Ai T. and van Oosterom P. (2002) “GAP-Tree Extension Based on Skeletons”, in Richardson D. and van Oosterom P. (eds.), *Advances in Spatial Data Handling. Proceedings 10th International Symposium on Spatial Data Handling*, Berlin Heidelberg: Springer, 501–513.

Anders K.H. (2003) “A Hierarchical Graph-Clustering Approach to find Groups of Objects”, in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Armstrong M.P. (1991) “Knowledge classification and organization”, in Buttenfield B.P. and McMaster R.B. (eds.), *Map Generalization: Making Rules for Knowledge Representation*, London, U.K.: Longman, 86–102.

Arnberger E. (1993) *Thematische Kartographie*, Das geographische Seminar, Braunschweig: Westermann, 3rd edition.

Bader M. (1997) *Methoden zur Erkennung und Lösung von metrischen Konflikten in der Polygongeneralisierung*, Master’s thesis, Department of Geography, University of Zurich.

Bader M. (2001) *Energy Minimizing Methods for Feature Displacement in Map Generalization*, Ph.D. thesis, Department of Geography, University of Zurich, URL `http://www.geo.unizh.ch/gis/~mbader/`, accessed: 05/10/2003.

Bader M. and Barrault M. (2000) “Improving Snakes for Linear Feature Displacement in Cartographic Generalization”, in *Proceedings 5th International Conference on GeoComputation*, Manchester, U.K.

Bader M., Barrault M., Regnauld N., Mustière S., Duchêne C., Ruas A., Fritsch E., Lecordix F. and Barillot X. (1999) “AGENT Workpackage D2 - Selection of Basic Algorithms”, Technical report, AGENT consortium.

Bader M. and Weibel R. (1997) “Detecting and Resolving Size and Proximity Conflicts in the Generalization of Polygon Maps”, in *Proceedings 18th International Cartographic Conference*, Stockholm, Sweden, 1525–1532.

Baeijs C., Demazeau Y. and Alvares L. (1996) “SIGMA: Application of Multi-Agent Systems to Cartographic Generalization”, in *Proceedings 7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAA-MAW’96*, Eindhoven, The Netherlands, 163–176.

Bard S. (2003) “Evaluation of generalization quality”, in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Barrault M., Regnauld N., Duchêne C., Haire K., Baeijs C., Demazeau Y., Hardy P., Mackaness W., Ruas A. and Weibel R. (2001) "Integrating multi-agent, object-oriented and algorithmic techniques for improved automated map generalization", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 2110–2116.

Beard K. (1988) *Multiple representations from a detailed database: a scheme for automated generalization*, Ph.D. thesis, University of Wisconsin-Madison.

Beard M.K. (1991) "Constraints on Rule Formation", in Buttenfield B.P. and McMaster R.B. (eds.), *Map Generalization: Making Rules for Knowledge Representation*, London, U.K.: Longman, 121–135.

Bengtson M. (2001) "Design and implementing of automatical generalisation in a new production environment for datasets in scale 1:50.000 (- and 1:100.000)", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 1076–1083.

de Berg M., van Kreveld M. and Schirra S. (1998) "Topologically Correct Subdivision Simplification Using the Bandwidth Criterion", *Cartography and Geographic Information Systems*, 25(4), 243–257.

Bern M. and Epstein D. (1995) "Mesh generation and optimal triangulation", in Du D.Z. and Hwang F. (eds.), *Computing in Euclidean geometry*, World Scientific, number 4 in Lecture Note Series on Computing, 2nd edition, 47–123.

Bigus J. and Bigus J. (2001) *Intelligente Agenten mit Java programmieren. eCommerce und Informationsrecherche automatisieren*, Munich, Germany: Adison-Wesley.

Bobrich J. (1996) *Ein neuer Ansatz zur kartographischen Verdrängung auf der Grundlage eines mechanischen Federmodells*, Ph.D. thesis, Deutsche Geodätische Kommission, München, Reihe C, H. 455.

Boffet A. and Rocca Serra S. (2001) "Identification of Spatial Structures within Urban Blocks for Town Characterization", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 1974–1983.

Böhme R. and Illert A. (2002) "Topographic Cartography", in Anson R.W. and Ormeling F.J. (eds.), *Basic Cartography for students and technicians*, Oxford, UK: Butterworth-Heinemann, 1–23.

Brassel K. and Weibel R. (1988) "A review and conceptual framework of automated map generalization", *International Journal of Geographical Information Systems*, 2(3), 229–244.

Brown N.J., Wright S.M. and Fuller R.M. (1996) "A Technique for the Removal of Outliers During a Computerised Map Generalisation Process", *Cartographic Journal*, 33(1), 11–16.

Burghardt D. (2000) *Automatisierung der kartographischen Verdrängung mittels Energieminimierung*, Ph.D. thesis, Department of Geodasy, Technical University of Dresden.

Burghardt D. (2002) "Smoothing of line objects with optimisation techniques", in *Proceedings Joint ISPRS/ICA workshop on Multi-Scale Representations of Spatial Data*, Ottawa, Canada, URL `http://www.ikg.uni-hannover.de/isprs/wg43-workshop.html`, accessed: 04/07/2003.

Burghardt D. and Cecconi A. (2003) "Mesh Simplification for Building Selection", in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Burghardt D. and Meier S. (1997) "Cartographic Displacement Using the Snakes Concept", in Förstner W. and Plümer L. (eds.), *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, Basel: Birkhäuser Verlag, 59–71.

Burrough P.A. and McDonnell R.A. (1998) *Principles of geographical information systems*, New York: Oxford University Press, 2nd edition.

Buttenfield B.P. (1999) "Sharing Vector Geospatial Data on the Internet", in *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, 581–590.

Buttenfield B.P. and McMaster R.B. (eds.) (1991) *Map Generalization. Making Rules for Knowledge Representation*, London, UK: Longman.

Buttenfield B.P., Weber C.R. and Leitner M. (1991) "How Does a Cartographic Object Behave? Computer inventory of topographic maps", in *Proceedings GIS/LIS'91*, Atlanta, USA, volume 2, 891–900.

CECA-CEE-CEEA (1993) *CORINNE Land Cover - Guide Technique.*

Cecconi A. and Galanda M. (2002) "Adaptive Zooming in Web Cartography", *Computer Graphics Forum*, 21(4), 787–799.

Chavez A. and Maes P. (1996) "Kasbah, An Agent Marketplace for Buying and Selling Goods", in *Proceedings 1st International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, U.K., 75–90.

Christophe S. and Ruas A. (2002) "Detecting Building Alignments for Generalisation Purposes", in Richardson D. and van Oosterom P. (eds.), *Advances in Spatial Data Handling. 10th International Symposium on Spatial Data Handling*, Berlin Heidelberg: Springer, 419–432.

Cook A. and Jones C. (1989) "Rule-based cartographic name placement with prolog", in *Proceedings AutoCarto 9*, Baltimore, USA, 231–240.

Cromley R. (1992) *Digital Cartography*, Englewood Cliffs, New Jersey: Prentice Hall.

De Lucia A.A. and Black R.T. (1987) "A Comprehensive Approach to Automatic Feature Generalization", in *Proceedings 13th International Cartographic Conference*, Morelia, Mexico, 173–191.

Dent B.D. (1990) *Cartography. Thematic Map Design*, Dubuque: Wm.C.Brown Publishers, 2nd edition.

Douglas D.H. and Peucker T.K. (1973) "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Character", *The Canadian Cartographer*, 10(2), 112–123.

Duchêne C. (2003) "Coordinative agents for automated generalisation of rural areas", in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Duchêne C., Barrault M. and Haire K. (2001a) "Road Network Generalization: A Multi Agent Approach", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 2166–2177.

Duchêne C., Cambier C. and Ruas A. (2001b) "Mécanismes de coordination multi-agents pour la cartographie automatique", in *Proceedings JFIADSMA'01*, Montreal, Canada, 331–334.

Duchêne C. and Regnauld N. (2002) "Le modèle AGENT", in Ruas A. (ed.), *Généralisation et représentation multiple*, Paris: Hermes, 369–385.

Dutton G., Weibel R., Peter B., Bader M. and Brazile F. (1998) "AGENT Workpackage A2 - Constraint Analysis", Technical report, AGENT consortium.

Edwardes A. and Mackaness W. (2000) "Modelling knowledge for automated generalization of categorical maps - a constraint based approach", in Atkinson P. and Martin D. (eds.), *GIS and GeoComputation (Innovations in GIS 7)*, London: Taylor&Francis, 161–173.

Edwardes A., Mackaness W. and Urwin T. (1998) "Self Evaluating Generalization Algorithms to Automatically Derive Multi Scale Boundary Sets", in *Proceedings 8th International Symposium on Spatial Data Handling*, Vancouver, Canada, 361–372.

Fayyad U., Piatetsky-Shapiro G. and Smyth P. (1996) "From Data Mining to Knowledge Discovery in Databases", *AI Magazine*, (3), 37–54.

Fisher P.F. and Mackaness W. (1987) "Are Cartographic Expert Systems Possible?", in *Proceedings AutoCarto 8*, Baltimore, USA, 530–534.

FRAGSTATS (1994) *Spatial pattern analysis program for quantifying landscape structure. Documentation of Version2.0.*

Frank A.U., Volta G.S. and McGranaghan M. (1997) "Formalization of Families of Categorical Coverages", *International Journal of Geographical Information Systems*, 11(3), 215–231.

Freeman H.R. and Doerschler J.S. (1992) "A rule-based system for dense map name placement", *Communication of the ACM*, 35(2), 68–79.

Fuller R. and Brown N. (1996) "A CORINE map of Great Britain by automated means. Techniques for automatic generalization of the Land Cover Map of Great Britain", *International Journal of Geographical Information Systems*, 10(8), 937–953.

Galanda M. and Weibel R. (2002a) "An Agent-Based Framework for Polygonal Subdivision Generalization", in Richardson D. and van Oosterom P. (eds.), *Advances in Spatial Data Handling. 10th International Symposium on Spatial Data Handling*, Berlin Heidelberg: Springer, 121–136.

Galanda M. and Weibel R. (2002b) "Algorithmes de généralization des données catégorielles", in Ruas A. (ed.), *Généralisation et représentation multiple*, Paris: Hermes, 273–286.

Galanda M. and Weibel R. (2003) "Using an Energy Minimization Technique for Polygon Generalization", *Cartography and Geographic Information Science*, 30(3), 259–275.

Goodchild M. (1999) "Cartographic Futures on a Digital Earth", in *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, 5–13.

Goodchild M., Sun G. and Yang S. (1992) "Development and test of an error model for categorical data", *International Journal of Geographical Information Systems*, 6(2), 87–104.

Grünreich D. (1985) "Computer-assisted generalisation", in *Papers CERCO Cartography Course*, Frankfurt, Germany.

Han J. and Kamber M. (2001) *Data Mining: concepts and techniques*, Morgan Kaufmann Publishers.

Han J., Kamber M. and Tung A.K.H. (2001) "Spatial Clustering Methods in Data Mining: A Survey", in Miller H. and Han J. (eds.), *In Geographic Data Mining and Knowledge Discovery*, London: Taylor&Francis, 187–217.

Hangouët J. (1995) "Computation of the Hausdorff distance between plane vector polylines", in *Proceddings AutoCarto 12*, Charlotte, USA, 1–10.

Harrie L. (1999) "The Constraint Method for Solving Spatial Conflicts in Cartographic Generalization", *Cartography and Geographic Information Science*, 26(1), 55–69.

Harrie L. (2001) *An Optimisation Approach to Cartographic Generalisation*, Ph.D. thesis, Dept. of Technology and Society, Lund Institute of Technology, Lund University.

Harrie L. and Sarjakoski T. (2002) “Simultaneous Graphic Generalisation of Vector Data Sets”, *GeoInformatica*, 6(2), 233–261.

Herbert G. and João E. (1991) “Automating map design and generalisation: A review of systems and prospects for future progress in the 1990’s. SERRL Working Report 27”, Technical report, Department of Geography, Birkbeck College.

Højholt P. (2000) “Solving Space Conflicts in Map Generalization: Using a Finite Element Method”, *Cartography and Geographic Information Science*, 27(1), 65–74.

ICA (1973) *Multilingual Dictionary of Technical Terms in Cartography*, Wiesbaden: Franz Steiner Verlag.

Imai H. and Iri M. (1988) “Polygonal approximations of a curve-fomulations and algorithms”, in Toussaint G. (ed.), *Computational morphology*, Elsevier, 71–86.

Imhof E. (1972) *Thematische Kartographie*, Berlin, New York: Walter de Gruyter.

Jaakkola O. (1994) “Finnish Corine Land Cover – a Feasibility Study of Automated Generalization and Data Quality Assessment”, Technical report, Finnish Geodetic Institute.

Jaakkola O. (1997) *Quality and Automatic Generalization of Land Cover Data*, Ph.D. thesis, Department of Geography, University of Helsinki.

Jaakkola O. (1998) “Multi-Scale Categorical Databases with Automatic Generalization Transformations Based on Map Algebra.”, *Cartography and Geographic Information Systems*, 25(4), 195–207.

Jain A. and Dubes R. (1988) *Algorithms for Clustering Data*, Englewood Cliffs, New Jersey: Prentice Hall.

Jennings N., Sycara K. and Wooldridge (1998) “A roadmap of agent research and development”, *Autonomous Agents and Multi-Agent Systems*, (1), 7–38.

João E.M. (1998) *Causes and Consequences of Map Generalisation*, Ph.D. thesis, London School of Economics.

Jones C.B., Bundy G.L. and Ware J.M. (1995) “Map Generalization with a Triangulated Data Structure”, *Cartography and Geographic Information Systems*, 22(4), 317–331.

Jost J. and Li-Jost X. (1998) *Calculus of Variations*, Cambridge: Cambridge University Press.

Kass M., Witkin A. and Terzopoulos D. (1987) “Snakes: Active Contour Models”, in *Proceedings First International Conference on Computer Vision.*, IEEE Computer Soc. Press, 259–268.

Kilpeläinen T. (1997) *Multiple Representation and Generalization of Geo-Databases for Topographic Maps*, Ph.D. thesis, Finnish Geodetic Institute.

Kilpeläinen T. (2000) “Knowledge Acquisition for Generalization Rules”, *Cartography and Geographic Information Science*, 27(1), 41–50.

Kilpeläinen T. and Sarjakoski T. (1995) “Incremental generalization for multiple representations of geographical objects”, in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization, GISDATA 1*, Taylor&Francis, 209–218.

Krikpatrick S., Gelath C.D. and Vechi M.P. (1983) “Optimization by simulated annealing”, *Science*, 220, 671–680.

Lagrange P., Landras B. and Mustière S. (2000) “Machine Learning Techniques for Determing Parameters of Cartographic Generalisation Algorithms”, in ISPRS (ed.), *International Archives of Photogrammetry and Remote Sensing*, 718–725.

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

Landesvermessungsamt Nordrhein-Westfalen (ed.) (1993) *Musterblatt für die Topographische Karte 1:25.000*, Bonn: Landesvermessungsamt Nordrhein-Westfalen, 3rd edition.

Laser-Scan (1999) *Gothic Module Reference Manual, MAPGENLIB issue v3-0.*

Laser-Scan (2002a) *AGENT developer's guide.*

Laser-Scan (2002b) “Bringing Clarity to automatic generalisation”, Technical report, Laser-Scan Ltd., URL `http://www.laser-scan.com/clarity`, accessed: 11/13/2002.

Laszlo M.J. (1996) *Computational Geometry and Computer Graphics in C++*, Englewood Cliffs, New Jersey: Prentice Hall.

Le Men H. (1996) “Généralisation cartographique pour l'occupation du sol: application au passage V1 á V2 pour la BD Carto®”, *Revue internationale de gómatique*, 6(2-3), 227–248.

Leitner M. and Buttenfield B.P. (1995) “Acquisition of Procedural Cartographic Knowledge by Revers Engineering”, *Cartography and Geographic Information Systems*, 22(3), 232–241.

Lemarié C. (2003) “Generalisation process for Top100: research in generalisation brought to fruition”, in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Li Z. (1994) “Mathematical Morphology in Digital Generalization of Raster Map Data”, *Cartography*, 23(1), 1–10.

Lichtner W. (1979) “Computer-Assisted Processes of Cartographic Generalization in Topographic Maps”, *Geo-Processing*, 1, 183–199.

Loibl W. and Tötzer T. (2002) “Growth and Densification Processes in Suburban Landscapes - a Spatial Agent - Simulation”, in *Proceedings 5th AGILE Conference on Geographic Information Science*, Mallorca, Spain, 23–30.

Luck M. (1997) “Foundations of Multi Agent Systems: Issues and Directions”, *Knowledge Engineering Review*, 12(3), 307–318.

Mackaness W. (1994) “Issues in resolving visual conflict in automated map design”, in *Proceedings 6th International Symposium on Spatial Data Handling*, Edinburgh, Scotland, 325–340.

Mackaness W. (1995) “A Constraint-Based Approach to Human Computer Interaction in Automated Cartography”, in *Proceedings 17th International Cartographic Conference*, Barcelona, Spain, 1423–1432.

Malczewski J. (1999) *GIS and Multicriteria decision making*, New York: John Wiley & Sons, Inc.

Maliĉ B. (1998) *Physiologische und technische Aspekte kartographischer Bildschirmvisualisierung*, Ph.D. thesis, Department of Cartography and Topography, University of Bonn.

Mark D.M. and Csillag F. (1989) “The Nature of Boundaries on 'Area Class' Maps”, *Cartographica*, 26(1), 65–77.

McGraw K.L. and Harbison-Briggs K. (1989) *Knowledge acquisition, principles and guidelines*, Englewood Cliffs, New Jersey: Prentice Hall.

McMaster R.B. (1986) "A statistical analysis of mathematical measures for linear simplification", *The American Cartographer*, 13(2), 103–116.

McMaster R.B. (1989) "The Integration of Simplification and Smoothing Algorithms in Line Generalization", *Cartographica*, 26(1), 101–121.

McMaster R.B. (1991) "Conceptual frameworks for geographical knowledge", in Buttenfield B.P. and McMaster R.B. (eds.), *Map Generalization: Making Rules for Knowledge Representation*, London, U.K.: Longman, 21–39.

McMaster R.B. (1995) "Knowledge acquisition for cartographic generalization", in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization, GISDATA 1*, Taylor&Francis, 161–179.

McMaster R.B. and Shea K.S. (1992) *Generalization in Digital Cartography*, Washington D.C.: Association of American Geographers.

McMaster R.B. and Veregin H. (1997) "Visualizing Cartographic Generalization", in *Proceedings AutoCarto 13*, Seattle, USA, 174–183.

Molenaar M. (1996) "The role of topologic and hierarchical spatial object models in database generalization", in Molenaar M. (ed.), *Methods for the generalization of Geo-databases (=New Series Nr. 43)*, Delft: Netherlands Geodetic Commission, 13–36.

Molenaar M. (1998) *An Introduction to the Theory of Spatial Object Modelling for GIS*, London: Taylor&Francis.

Monmonier M.S. (1983) "Raster Area Generalization for Land Use and Land Cover Maps", *Cartographica*, 20(4), 65–91.

Morgenstern D. and Schürer D. (1999) "A Concept for model generalization of digital landscape models from finer to coarser resolution.", in *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, 1021–1028.

Morrison J.L. (1994) "The paradigm shift in cartography: the use of electronic technology, digital spatial data, and future needs", in *Proceedings 6th International Symposium on Spatial Data Handling*, Edinburgh, Scotland, 1–15.

Morrison J.L. (1999) "The State of Government Cartography in 1998", *Cartography and Geographic Information Science*, 26(3), 167–199.

Muehrcke P.C. (1986) *Map Use: Reading, Analysis, and Interpretation*, Madison: JP Publications.

Müller J.C., Lagrange J.P. and Weibel R. (eds.) (1995a) *GIS and Generalization. Methodology and Practice. GISDATA 1*, Taylor&Francis.

Müller J.C. and Wang Z. (1992) "Area-Patch Generalization: A Competitive Approach", *The Cartographic Journal*, 29(2), 137–144.

Müller J.C., Weibel R., Lagrange J.P. and Salgé F. (1995b) "Generalization: State of the Art and Issues", in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization. Methodology and Practice. GISDATA 1*, Taylor&Francis, 3–17.

Mustière S. (1998) "Abstraction-Based Machine Learning Approach to Cartographic Generalisation", in *Proceedings 1st GIS'PlaNet conference*, Lisbon, Portugal, CD-ROM.

Mustière S. (2001) *Apprentissage supervisé pour la généralisation cartographique*, Ph.D. thesis, University of Paris VI.

Mustière S. and Duchêne C. (2001) "Comparison of different approaches to combine generalisation algorithms: GALBE, AGENT and CartoLearn", in *Proceedings 4th Workshop on Progress in Automated Map Generalization*, Beijing, China, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Mustière S. and Moulin B. (2002) "What is spatial context in cartographic generalisation?", in *Proceedings Symposium on Geospatial Theory, Processing and Applications*, Ottawa, Canada, CD-ROM.

Mustière S., Zucker J.D. and Saitte L. (2000) "Abstraction-Based Machine Learning Approach to Cartographic Generalisation", in *Proceedings 9th International Symposium on Spatial Data Handling*, Beijing, China, 1a.50–63.

Nareyek A. (2001) *Constraint-Based Agents. An Architecture for Constraint-Based Modeling and Local-Search-Based Reasoning for Planning and Scheduling in Open and Dynamic Worlds*, Berlin, Heidelberg: Springer.

Nickerson B.G. (1988) "Automated cartographic generalization for linear features", *Cartographica*, 25(3), 15–66.

Nickerson B.G. (1991) "Knowledge engineering for generalization", in Buttenfield B.P. and McMaster R.B. (eds.), *Map Generalization: Making Rules for Knowledge Representation*, London: Longman, 40–55.

O'Hare G.M.P. and Jennings N.R. (eds.) (1996) *Foundations of Distributed Artifical Intelligence*, New York: John Wiley & Sons, Inc.

van Oosterom P. (1994) *Reactive Data Structures for Geographic Information Systems*, Oxford: Oxford University Press.

van Oosterom P. (1995) "The GAP-tree, an approach to 'on-the-fly' map generalization of an area partitioning", in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization, GISDATA 1*, Taylor&Francis, 120–132.

Open GIS Consortium (1995) *The OpenGIS Abstract Specification. Topic 6: The Coverage Type and its Subtypes. Version 6*, URL `http://www.opengis.org/techno/abstract/00-106.pdf`.

O'Rourke J. (1995) *Computational Geometry in C*, Cambridge: Cambridge University Press.

Peter B. (1997) *Ableitung von generalisierten Bodennutzungskarten aus der Arealstatistik der Schweiz 1979/85*, Master's thesis, Department of Geography, University of Zurich.

Peter B. (2001) "Measures for the Generalization of Polygonal Maps with Categorical Data", in *Proceedings 4th Workshop on Progress in Automated Map Generalization*, Beijing, China, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Peter B. and Weibel R. (1999a) "Using Vector and Raster-Based Techniques in Categorical Map Generalization", in *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, 1135–1145.

Peter B. and Weibel R. (1999b) "Using Vector and Raster-Based Techniques in Categorical Map Generalization", in *Proceedings 3rd Workshop on Progress in Map Generalization*, Ottawa, Canada, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Peterson M. (2001) "The Development of Map Distribution through the Internet", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 2306–2312.

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

Plazanet C., Bigolin M. and Ruas A. (1998) "Experiments with Learning Techniques for Spatial Model Enrichment and Line Generalization", *GeoInformatica*, 2(4), 315–333.

van Putten J. and van Oosterom P. (1998) "New results with Generalized Area Partitionings", in *Proceedings 8th International Symposium on Spatial Data Handling*, Vancouver, Canada, 485–495.

Rainsford D. and Mackaness W. (2002) "Template Matsching in Support of Generalisation of Rural Buildings", in Richardson D. and van Oosterom P. (eds.), *Advances in Spatial Data Handling. Proceedings 10th International Symposium on Spatial Data Handling*, Berlin Heidelberg: Springer, 137–151.

Ramer U. (1972) "An iterative procedure for the polygonal approximation of plane curves", *Computer Graphics and Image Processing*, 1, 244–256.

Raubal M. (2001) "Ontology and epistemology for agent-based wayfinding simulation", *International Journal of Geographical Information Science*, 15(7), 633–665.

Regnauld N. (1998) *Généralization du bâti: Strucuture spatiale de type graphe et répresentation cartographique*, Ph.D. thesis, Université de Provence, Aix-Marseille 1.

Regnauld N. (2001) "Constraint based mechanism to achieve automatic generalization using agent modelling", in *Proceedings GIS Research UK 9th Annual Conference*, Glamorgan, Wales, 329–332.

Regnauld N. (in press) "Constraint Based Mechanism Applied to Automated Map Generalisation Using Agent Based Modelling Techniques", *Computers, Environment and Urban Systems*.

Regnauld N. and Mustière S. (2000) "Optimising generalisation sequences Using machine learning techniques", in *Proceedings GIS Research UK 8th Annual Conference*, York, UK.

Reichenbacher T. (1995a) *Eine Methode für den Wissenserwerb in der kartographischen Generalisierung durch Interaktionsaufzeichnung und induktives Lernen*, Master's thesis, Department of Geography, University of Zurich.

Reichenbacher T. (1995b) "Knowledge acquisition in map generalization using interactive systems and machine learning", in *Proceedings 17th International Cartographic Conference*, Barcelona, Spain, volume 2, 2221–2230.

Reichenbacher T. (2001) "The world in your pocket - towards a mobile cartography", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 2514–2522.

Reichenbacher T. (2002) "SVG for adaptive visualisations in mobile situations", in *Proceedings SVG Open 2002*, Zurich, Switzerland.

Richardson D. and Mackaness W. (1998) "Computational Processes for Map Generalization", *Cartography and Geographic Information Science*, 26(1), 3–5.

Richardson D. and Mackaness W. (1998, guest eds.) "Special Issue on Computational Methods for Map Generalization", *Cartography and Geographic Information Science*, 26(1).

Rieger M. and Coulson M. (1993) "Consensus or Confusion: Cartographer's Knowledge of Generalization", *Cartographica*, 30(2&3), 69–80.

Ruas A. (1995) "Multiple paradigms for automated map generalization: Geometry, topology, hierarchical partitioning and local triangulation", in *Proceedings AutoCarto 12*, Charlotte, USA, 69–78.

Ruas A. (1998) "OO-constraint modelling to automate urban generalization process", in *Proceedings 8th International Symposium on Spatial Data Handling*, Vancouver, Canada, 225–235.

Ruas A. (1999) *Modèle de généralization de données géographiques à base de contraints et d'autonomie*, Ph.D. thesis, Université de Marne-la-Vallée.

Ruas A. (2000) "The Roles of Meso Objects for Generalisation", in *Proceedings 9th Symposium on Spatial Data Handling*, Beijing, China, 3b50–3b63.

Ruas A. (2001a) *Automatic Generalisation project: Learning Process from interactive generalisation*, OEEPE Official Publication 39.

Ruas A. (2001b) "Automating the generalisation of geographical data: the age of maturity?", in *Proceedings 20th International Cartographic Conference*, Beijing, China, 1943–1953.

Ruas A. (ed.) (2002) *Généralisation et représentation multiple*, Hermes.

Ruas A. (in press) "Developments in a Methodology for Automatic Map Generalisation", *Computers, Environment and Urban Systems.*

Ruas A. and Lagrange J.P. (1995) "Data and knowledge modelling for generalization", in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization. Methodology and Practice. GISDATA 1*, Taylor&Francis, 73–90.

Ruas A. and Plazanet C. (1996) "Strategies for Automated Generalization", in *Proceedings 7th International Symposium on Spatial Data Handling (=Advances in GIS Research II)*, Delft: Taylor&Francis, 6.1–6.17.

Russell S.J. and Norvig P. (2002) *Artifical Intelligence: A Modern Approach*, Englewood Cliffs, New Jersey: Prentice-Hall, 2nd edition.

Saalfeld A. (1999) "Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm", *Cartography and Geographic Information Science*, 26(1), 7–18.

Sarjakoski T. and Kilpeläinen T. (1999) "Holistic Cartographic Generalization by Least Squares Adjustment for Large Data Sets", in *Proceedings 19th International Cartographic Conference*, Ottawa, Canada, 1091–1098.

Schylberg L. (1993) *Computational Methods for Generalization of Cartographic Data in a Raster Environment*, Ph.D. thesis, Royal Institute of Technology, Stockholm.

Serra J. (1982) *Image Processing and Mathematical Morphology*, New York: Academic Press.

Sester M. (2000) "Generalization Based on Least Squares Adjustment", in ISPRS (ed.), *International Archives of Photogrammetry and Remote Sensing.*

Sester M. and Brenner C. (2000) "Typification based on Kohonen Feature Nets", in *Proceedings GIScience 2000. The First International Conference on Geographic Information Science*, Savannah, USA, 21–22.

SFSO (1972) *Die Arealstatistik der Schweiz 1972*, SFSO, Swiss Federal Statistical Office.

SFSO (1992) *Die Bodennutzung der Schweiz, Arealstatistik 1979/85*, SFSO, Swiss Federal Statistical Office.

Sinton D. (1978) "The inherent structure of information as a constraint to analysis: mapped thematic data as a case study", in Dutton G. (ed.), *Havard Papers on Geographic Information Systems*, Reading: Addison Wesley, Sinton/1–Sinton/17.

Slocum T. (1999) *Thematic Cartography and Visualization*, Upper Saddle River, New Jersey: Prentice Hall.

Spiess E. (1990) "Generalisierung in thematischen Karten", in SSC, Swiss Society of Cartography (ed.), *Kartographisches Generalisieren*, Baden: SSC, Kartografische Publikationsreihe Nr. 10, 63–70.

Spiess E. (2002) "Map Compilation", in Anson R.W. and Ormeling F.J. (eds.), *Basic Cartography for students and technicians*, Oxford, UK: Butterworth-Heinemann, 25–84.

SSC (2002) *Topografische Karten. Kartengrafik und Generalisierung*, Kartografische Publikationsreihe Nr. 16, SSC, Swiss Society of Cartography, CD-ROM.

Su B., Li Z. and Lodwick G. (1998) "Morphological Models for the Collapse of Area Features in Digital Map Generalization", *GeoInformatica*, 2(4), 359–382.

Tomlin C.D. (1990) *Geographic Information Systems and Cartographic Modeling*, Englewood Cliffs, New Jersey: Prentice Hall.

Torsun I.S. (1995) *Foundations of Intelligent Knowledge-Based Systems*, London: Academic Press.

Trochim M.K. (2001) *The Research Methods Knowledge Base*, Cincinnati: Atomic Dog Pub Inc., 2nd edition.

Visvalingam M. and Whyatt D. (1993) "Line generalization by repeated elimination of points", *The Cartographic Journal*, 30(1), 46–51.

Visvalingam M. and Williamson P.J. (1995) "Simplification and Generalization of Large Scale Data for Roads: A Comparison of Two Filtering Algorithms", *Cartography and Geographic Information Systems*, 22(4), 264–275.

Wang Z. and Müller J.C. (1998) "Line Generalization Based on Analysis of Shape Characteristics", *Cartography and Geographic Information Systems*, 25(1), 3–15.

Ware J.M. and Jones C.B. (1996) "A Spatial Model for Detecting (and Resolving) Conflict Caused by Scale Reduction", in *Proceedings 7th International Symposium on Spatial Data Handling (=Advances in GIS Research II)*, Delft, The Netherlands, 9A.15–9A.26.

Ware J.M. and Jones C.B. (1998) "Conflict Reduction in Map Generalization Using Iterative Improvement", *GeoInformatica*, 2(4), 383–407.

Ware J.M., Jones C.B. and Nathan T. (2001) "A Simulated Annealing Algorithm for Cartographic Map Generalization with Multiple Operators", in *Proceedings 4th Workshop on Progress in Automated Map Generalization*, Beijing, China.

Ware J.M., Nathan T. and Jones C.B. (2003) "Resolving Graphic Conflict in Scale Reduced Maps: Refining the Simulated Annealing Technique", in *Proceedings 5th Workshop on Progress in Automated Map Generalization*, Paris, France, URL `http://www.geo.unizh.ch/ICA/`, accessed: 05/10/2003.

Waterman D.A. (1986) *A Guide to Expert Systems*, Reading, Massachusetts: Addison-Wesley.

Weibel R. (1991) "Amplified intelligence and rule-based systems", in Buttenfield B.P. and McMaster R.B. (eds.), *Map Generalization: Making Rules for Knowledge Representation*, London: Longman, 172–186.

Weibel R. (1995a) "Map Generalization in the Context of Digital Systems", *Cartography and Geographic Information Systems*, 22(4), 259–263.

Weibel R. (1995b) “Three essential building blocks for automated generalization”, in Müller J.C., Lagrange J.P. and Weibel R. (eds.), *GIS and Generalization, GISDATA 1*, Taylor&Francis, 56–69.

Weibel R. (1995, guest ed.) “Special Issue on Map Generalization”, *Cartography and Geographic Information Systems*, 22(4).

Weibel R. (1996) “A Typology of Constraints of Line Simplification”, in *Proceedings 7th International Symposium on Spatial Data Handling (=Advances in GIS Research II)*, Delft, The Netherlands: Taylor&Francis, 9A.1–9A.14.

Weibel R. (1997) “Generalization of Spatial Data: Principles and Selected Algorithms”, in van Kreveld M., Nievergelt J., Roos T. and Widmayer P. (eds.), *Algorithmic Foundations of Geographic Information Systems*, Springer, 99–152.

Weibel R. and Dutton G. (1998) “Constraint-based Automated Map Generalization”, in *Proceedings 8th International Symposium on Spatial Data Handling*, 214–224.

Weibel R. and Dutton G. (1999) “Generalising spatial data and dealing with multiple representations”, in Longley P.A., Goodchild M.F., Maguire D.J. and Rhind D.W. (eds.), *Geographical Information Systems*, John Wiley & Sons, Inc., volume 1, 125–155.

Weibel R. and Jones C.B. (1998) “Computational perspective on map generalization”, *GeoInformatica*, 2(4), 307–314.

Weibel R. and Jones C.B. (1998, guest eds.) “Special Issue on Map Generalization”, *GeoInformatica*, 2(4).

Weibel R., Keller S. and Reichenbacher T. (1995) “Overcoming the Knowledge Acquisition Bottleneck in Map Generalization: the Role of of Interactive Systems and Computational Intelligence”, in *Proceedings COSIT’95*, Semmering, Austria, 139–156.

Weiss G. (ed.) (1999) *Multiagent Systems. A modern Approach to Distributed Artifical Intelligence*, Cambridge, Massachusetts: The MIT Press.

Werschlein T. and Weibel R. (1994) “Use of neural networks in line generalization”, in *Proceedings EGIS’94*, Paris, France, 76–85.

Wooldridge M. (1999) “Intelligent Agents”, in Weiss G. (ed.), *Multiagent Systems. A modern Approach to Distributed Artifical Intelligence*, The MIT Press, 27–78.

Worboys M.F. (1995) *GIS. A Computing Perspective*, London: Taylor&Francis.

Yaolin L., Molenaar M. and Kraak M.J. (2001) “Spatial Object Aggregation Based on Data structure, Local Triangulation and Hierarchical analyzing method”, in *Proceedings 20th International Cartographic Conference*, Beijing, China, 1625–1640.

Yaolin L., Molenaar M. and Kraak M.J. (2002a) “Categorical database generalization aided by data structure”, in Wise S., Brindley P., Kim Y.H. and Openshaw C. (eds.), *Proceeding GIS research UK 10th Annual Conference*, Sheffield: University of Sheffield, 345–348.

Yaolin L., Molenaar M. and Kraak M.J. (2002b) “Semantic similarity evaluation model in categorical database generalization”, in *Proceeding ISPRS Commission IV Symposium*, Ottawa, Canada, CD-ROM.

Zoraster S. (1991) “Expert Systems and the Map Label Placement Problem”, *Cartographica*, 28(1), 1–9.

Curriculum vitae

Martin Galanda
born October 29, 1973, in Vienna (Austria)
from Austria

Schooling

1980 – 1984	Primary school in Vienna (GTS Wien II).
1984 – 1992	High school in Vienna (Bundesgymnasium Wien XX).
1993 – 2000	MSc in cartography at the University of Vienna with a thesis on 'Visualisierung des österreichischen Lawinenlageberichtes im WWW' ('Visualization of the Austrian avalanche bulletin in the WWW') advised by Prof. R. Weibel.
2000 – 2003	Research Assistant and PhD student at the Department of Geography of the University of Zurich. Dissertation with a thesis on 'Automated Polygon Generalization in a Multi Agent System' advised by Prof. R. Weibel.

Publications

Cecconi A. and Galanda M. (2001) "Multiskalierte Datenbanken und kartographische Generalisierung als Elemente guter Internetkartographie", in Strobl J., Blaschke T. and Griesebner G. (eds.), *Angewandte Geographische Informationsverarbeitung 13, Beiträge zum AGIT-Symposium 2001*, Heidelberg: Wichmann, 78-83.

Cecconi A. and Galanda M. (2002a) "Adaptive Zooming in Web Cartography", in *Proceedings SVG Open 2002*, Zurich, Switzerland.

Cecconi A. and Galanda M. (2002b) "Adaptive Zooming in Web Cartography", *Computer Graphics Forum*, 21(4), 787-799.

Dahinden T., Galanda M. and Neumann A. (2002) "Entwicklungsschub für Grafiken im Web. SVG verspricht Qualität und Interaktivität", *Neue Zürcher Zeitung*, 8.3.2002 (56/2002).

Galanda M. and Kriz K. (1998) "Geodatengewinnung aus dem KM des BEV für den (karto)graphisch anspruchsvollen GIS-Anwender", in Strobl J. and Dollinger F. (eds.), *Angewandte Geographische Informationsverarbeitung, Beiträge zum AGIT-Symposium 1998*, Heidelberg: Wichmann, 99-106.

Galanda M. (2000a) *Visualisierung des österreichischen Lawinenlageberichtes im WWW*, MSc Thesis, Department of Geography, University of Vienna, Austria.

Galanda M. (2000b) "Online (cartographic) visualization of the avalanche bulletin", *Proceedings ICA Workshop on High Mountain Cartography*, Rudolfshütte, Austria.

Galanda M. (2000c) "Interaktive Visualisierung lawinenrelevanter Informationen im WWW", in Schmidt, B. and Uhlenküken C. (eds.), *Visualisierung raumbezogener Daten: Methoden und Anwendungen, 3rd GeoViSC Workshop*, IfGIprints 8, 93-109.

Galanda M. (2001) "Optimization techniques for polygon generalization", *Proceedings 4th ICA Workshop on Progress in Automated Map Generalization*, Beijing, China. http://www.geo.unizh.ch/ICA/ (Accessed: 06/19/2003)

Galanda M. (2003) "Modelling constraints for polygon generalization", *Proceedings 5th ICA Workshop on Progress in Automated Map Generalization*, Paris, France. http://www.geo.unizh.ch/ICA/ (Accessed: 06/19/2003)

Galanda. M. and Cecconi A. (2003) "Adaptives Zoomen in der Internetkartographie", *Kartographische Nachrichten*, 53(1), 11-16.

Galanda M. and Neumann A. (2002) "Scalable Vector Graphics. Ein neuer Graphikstandard für das Internet", *Bulletin des Schweizerischen Elektrotechnischen Vereins*, 9(93), 18-21.

Galanda M. and Weibel R. (2002a) "An Agent-Based Framework for Polygonal Subdivision Generalization", in Richardson D. and van Oosterom P.(eds.), *Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling*, Berlin Heidelberg: Springer, 121-136.

Galanda M. and Weibel R. (2002b) "Algorithmes de géneralization des données catégorielles", in Ruas A. (ed.), *Généralisation et représentation multiple*, Paris: Hermes, 273-286.

Galanda. M. and Weibel, R. (2003) "Using an Energy Minimization Technique for Polygon Generalization", *Cartography and Geographic Information Science*, 30(3), 259-275.

Kriz K. and Galanda M. (1998) "Thematische Aspekte der Hochgebirgskartographie. Lawinengefährdungskarten", in Kriz K.(ed.), *Hochgebirgskartographie. Silvretta'98*, Wiener Schriften zur Geographie und Kartographie, Vol. 11, 115-129.

Loibl W., Galanda M., Kopcsa A. and Zügler J. (1999) "Biodiversität und Klimaänderung. Kleinräumige Modellierung von Klimafaktoren", Technical Report OEFZS–S-0029, Austrian Research Centers.