WEB-BASED ARCHITECTURE FOR ON-DEMAND MAPS - INTEGRATING MEANINGFUL GENERALIZATION PROCESSING

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Wednesday 10 February 2010 at 13.15

by

Theodor Ernst-Christoph Martin Joachim Foerster

born on 6 July 1980
in Munich, Germany

This dissertation is approved by

prof. dr. M.-J. Kraak, promoter
prof. dr. ir. P. van Oosterom, promoter
dr. J. Stoter, assistant promoter

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION, ENSCHEDE, THE NETHERLANDS

ITC Dissertation number 165
ITC, P.O. Box 6, 7500 AA Enschede, The Netherlands

ISBN 978-90-6164-285-5

Abstract

Generating readable maps at a specific scale and for a specific use is a challenge in research and practice. With the advent of the web as a platform for accessing and sharing information between users, maps also became web-accessible. The web provides the means to generate and disseminate maps for specific users (i.e. on-demand). Providing on-demand maps is considered to improve map communication. In this context, automated generalization is one solution to generate these on-demand maps.

Currently, thematic content such as physical plans becomes available on the web and displayed as maps. These maps are available as separate layers without a base map. Thereby they are in need of an on-demand base map, which might be generated regarding the thematic content and the specific user. In the case of physical planning on-demand base maps enhance the communication of the thematic content and thereby improve the communication process between the planning authorities and the public.

However, a web-based architecture to generate and disseminate on-demand base maps has not been proposed yet. Regarding the aspect of disseminating on-demand base maps on the web, the thesis investigates how to formulate user requirements towards the on-demand base map in such an environment. These user requirements are captured in so-called user profiles, which are formalized in UML and XML-models. The on-demand base map is generated according to these user profiles and according to the specific thematic content. As core of the architecture, a so-called generalization-enabled Web Map Service is presented, which consumes these user profiles and generates the base maps accordingly. The architecture is implemented as a proof-of-concept and is applied to the use case of physical planning in the Netherlands.

Besides the aim of disseminating these on-demand base maps on the web, the web is also promising to serve as a platform for web-based generalization. Such web-based processes are performed by Web Services. Establishing web-based meaningful generalization processing, the semantic interoperability of these Web Services has to be enabled, which is considered to be a challenge for research. In particular, this thesis proposes a classification of generalization operators, which is formalized in the Object Constraint Language and deployed using the Web Processing Service interface and so-called WPS profiles.

This research contributes to on-demand web mapping and to automated generalization on the web. Designing user profiles to describe generalization-specific requirements and incorporating them into a web-based architecture is a novelty for web mapping. The user profiles thereby enable the on-demand notion and are a complementary approach to the already established OGC Styled Layer Descriptor documents (for symbolization) and OGC Web Map Context (for map content definition) documents as they define user-specific issues regarding scale and the link between thematic content and the base map. Defining user profiles for the application of base maps is a novel approach in itself. Generating base maps by automated generalization has not been attempted so far. The user requirements define the thematic content as input for the generalization of the base map, which is also a novelty. The generalization-enabled WMS is the component which provides the on-demand base map in a standards-compliant way based on the specific user profile. Additionally, this research contributes to the meaningful integration of generalization processes on the web. In particular, this research contributes to the theory of automated generalization by providing a classification of generalization operators. Further, this research contributes to the issue of semantic interoperability of Web Generalization Services as it formalizes the classification using standardized data models of ISO and OGC and finally proposes the application of WPS profiles and the Object Constraint Language. Using standardized data models, this classification is extensible and comprehensible for future research. The application of WPS profiles and the Object Constraint Language is not only a novelty to generalization research and Web Generalization Services, but also contributes to the aspect of web-based geoprocessing in general.

Samenvatting

Zowel in het onderzoek als in de praktijk is het een uitdaging om leesbare kaarten te genereren in voorafbepaalde schalen en voor specieke gebruiksdoelen. Met de opkomst van het web als platform voor informatieuitwisseling werden kaarten ook beschikbaar via dat web, waardoor het mogelijk werd kaarten te genereren en verspreiden die afgestemd zijn op specifieke gebruikers. Dit noemen we kaarten op afroep, en het wordt algemeen aangenomen dat ze de cartografische communicatie verbeteren. Automatische generalisatie is in dit verband een van de mogelijke oplossingen voor het genereren van zulke kaarten op afroep.

Tegenwoordig komt steeds meer thematische inhoud, zoals ruimtelijke plannen, beschikbaar als kaarten op het web. Deze worden meestal aangeboden als aparte kaartlagen, zonder bijbehorende basiskaart. Er is daarom behoefte aan basiskaarten, aangepast aan de thematische inhoud en de specifieke gebruikers van de kaarten. In het geval van ruimtelijke plannen zou de overdracht van de thematische inhoud, en daarmee de communicatie tussen de planologen en het publiek, door zulke basiskaarten op afroep worden verbeterd.

Op dit moment is er echter nog geen architectuur ontwikkeld om basiskaarten op afroep te genereren en te verspreiden met behulp van web-technologie. In dit proefschrift wordt onderzocht hoe gebruikerseisen kunnen worden geformuleerd voor verspreiding van basiskaarten op afroep in een web-omgeving. Deze gebruikerseisen worden opgenomen in zogenaamde gebruikersprofielen, die geformaliseerd zijn in UML en XML modellen. De basiskaart op afroep wordt vervolgens gegenereerd door de combinatie van deze gebruikersprofielen met de bijbehorende thematische inhoud. Het hart van de architectuur wordt gevormd door een speciale versie van een Web Map Service (WMS) met generalisatiemogelijkheden. Deze interpreteert de gebruikersprofielen en genereert aan de hand daarvan de basiskaarten. De genoemde architectuur is gemplementeerd als een proof-of-concept applicatie met de ruimtelijke planning in Nederland als toepassing.

Naast het gebruik van het web om de basiskaarten op afroep mee te verspreiden, is het web ook geschikt als platform voor generalisatieprocessen. Dergelijke webgebaseerde processen worden door Web Services uitgevoerd. Om generalisatieprocessen zinvol op het web uit te voeren, moeten de Web Services semantisch interoperabel zijn, en dat wordt algemeen gezien als een onderzoeksuitdaging. In dit proefschrift wordt hiervoor een classificatie van generalisatieoperaties voorgesteld, geformaliseerd in OCL, de Object Constraint Language, en gerealiseerd met behulp van de Web Processing Service interface en zogenaamde

WPS profielen.

Dit onderzoek draagt bij aan de vervaardiging van kaarten op afroep op het web en aan geautomatiseerde generalisatie op het web. Een van die nieuwe bijdragen is het ontwerp van gebruikersprofielen om generalisatiespecifieke eisen te beschrijven en ze te gebruiken in een web-gebaseerde architectuur. De genoemde gebruikersprofielen maken het mogelijk kaarten op afroep te maken en zijn daarmee een aanvulling op de al gevestigde OGC standaarden Styled Layer Description (definitie van de symbolisatie) en Web Map Context Document (definitie van de inhoud), want ze definiren gebruikersspecifieke schaalniveaus en de relatie van de thematische inhoud met de basiskaart. Het vastleggen van gebruikersprofielen ten behoeve van basiskaarten is op zich al een nieuwe aanpak, en het gebruik van automatische generalisatie om basiskaarten te genereren is ook een noviteit. Datzelfde geldt voor het inzetten van gebruikerseisen om de thematische inhoud te definiren en op basis daarvan de generalisatie van de basiskaart uit te voeren. Om de basiskaart op afroep te kunnen aanbieden, gebaseerd op specifieke gebruikersprofielen en volgens de relevante standaarden, is de WMS met generalisatiemogelijkheden ontwikkeld.

Daarnaast draagt dit proefschrift bij aan de relevante integratie van generalisatieprocessen op het web. Met name wordt een bijdrage geleverd aan de theorie van geautomatiseerde generalisatie, door de classificatie van generalisatie operaties. Bovendien levert dit onderzoek een bijdrage aan de vraagstukken rond de sematische interoperabiliteit van Web Generalization Services, omdat de eerder genoemde classificatie geformaliseerd wordt met behulp van gestandaardiseerde datamodellen van ISO en OGC, en een voorstel wordt gedaan voor het toepassen van WPS profielen en de Object Constraint Language. Omdat gestandaardiseerde datamodellen zijn gebruikt, is de voorgestelde classificatie begrijpelijk en uit te breiden ten behoeve van toekomstig onderzoek. Het toepassen van WPS profielen en de Object Constraint Language is niet alleen nieuw in het onderzoek naar generalisatie en Web Generalization Services, het draagt ook bij in onderzoek naar geoprocessing op het web in het algemeen.

Acknowledgements

The time for performing this research has been enjoyable, valuable and inspiring since the very beginning. This research would not have been possible without the support of many people. I am deeply grateful to all of them and I want to thank some of them in particular.

First, I thank Prof. Dr. Menno-Jan Kraak for giving me the opportunity to work at ITC as an AIO in the DURP ondergronden project and for being my supervisor and promoter. Additionally, I thank my second promoter Prof. Dr. Peter van Oosterom. Both promoters gave valuable insight into research techniques and supported my research at anytime. I was also privileged to have Dr. Jantien Stoter as my daily supervisor. She discussed my research whenever necessary and helped me striving for scientific quality. She also gave me the chance to spend one month each at the IGN in France and at the FGI in Finland for a fruitful research visit.

I express my sincere thanks to my colleagues at ITC's Geo-Information Processing department (GIP). They supported my research in various ways from the very beginning. Especially, I appreciate the help of Barend Köbben, Dr. Rob Lemmens and Dr. Javier Morales who provided co-authorship for different publications and thereby helped me to develop the research successfully. Additionally, Dr. Rolf de By pointed out various aspects, I would have missed otherwise during my research. Ellen Rijckenberg helped me out regarding any administrative issue and thereby eased the daily work at ITC. During the first half of my research I teamed up with Eddie Poppe, my PhD-mate. We had long discussions about generalization and cartography, which helped me to dive into the topic quickly. My thanks goes also to other ITC staff outside the GIP department: Loes Colenbrander, Job Duim, Harold Borkent and Martin Blankestijn.

I am thankful to 52°North initiative (especially Dr. Andreas Wytzisk and Bastian Schäffer) for providing the platform to publish the developed software and to propagate the research results. I also appreciate the various support of the Institute for Geoinformatics (University of Münster), 1spatial, ESRI Nederland (Dr. John van Smaalen), TU Dresden (Johannes Brauner and Prof. Dr. Lars Bernard), Ordnance Survey of Great Britain (Dr. Nicolas Regnauld and Patrick Revell) and the Centre for Geospatial Science (Dr. Suchith Anand, Dr. Jerry Swan and Prof. Dr. Mike Jackson) in Nottingham, UK.

Regarding the previously mentioned research visits, I am also thankful to Prof. Dr. Anne Ruas, Cecile Dr. Duchene and Julien Dr. Gaffuri from IGN France and to Prof. Dr. Tapani Sarjakoski, Dr. Tiina Sarjakoski and to Dr. Lassi Lehto from FGI Finland for their hospitality. Additionally, I am thankful to Dr. Nicolas Regnauld of Ordnance Survey, who invited me to visit his research team during my research.

For reviewing the final thesis, I appreciate the help of my supervisors and promoters as well as of Jörg, Bastian and Barend.

I thank my family and friends. Without them this all would not have been that enjoyable. Thanks to my family - Mami, Papi, Friedrich and Joachim. Thanks to Flo, Jan, Jörg K., Sebastian and Jörg M.

Finally, I thank Tina for her warm love and all her support during finalizing the research.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

AGENT	Automatic GEneralisation New Technology
API	Application Programming Interface
DCM	Digital Cartographic Model
DLM	Digital Landscape Model
ETSI	European Telecommunications Standards Institute
GBKN	Grootschalige Basiskaart van Nederland
GII	Geographic Information Infrastructure
GIS	Geographic Information System
GML	Geography Markup Language
GUI	Graphic User Interface
HTTP	Hypertext Transfer Protocol
ICA	International Cartographic Association
IMRO	Informatiemodel Ruimtelijke Ordening
INSPIRE	INfrastructure for SPatial Information in Europe
ISO	International Organization for Standardization
MAGNET	Mapping Agencies Generalization NETwork
MDA	Model Driven Architecture
MRDB	Multi Resolution Database
NMA	National Mapping Agency
OCL	Object Constraint Language
OGC	Open Geospatial Consortium

OMG	Object Management Group
SLD	Styled Layer Descriptor
SOA	Service-Oriented Architecture
SVBP	Standard Vergelijkbare BestemmingsPlannen
UML	Unified Modeling Language
UMS	User Modeling System
URN	Uniform Resource Name
WFS	Web Feature Service
WMC	Web Map Context
WMS	Web Map Service
WPS	Web Processing Service
WSDL	Web Service Description Language
WSML	Web Service Modeling Language
XMI	XML Metadata Interchange

Chapter *1*

Introduction

The research described in this thesis covers the topics of automated generalization and on-demand web mapping. These topics can be explained as follows:

Automated Generalization

The process of automatically extracting and emphazising certain aspects of geographic data is called automated generalization [196]. It transforms geographic data into geographic information regarding a specific scale, level of detail and use. The resulting information can be either data or a visual representation of the data (e.g. a map).

On-demand web mapping

The approach of generating and disseminating maps on the web is called web mapping [107]. On-demand in this context means, that the user receives the map according to his/her requirements. This includes specific symbolization, scale, level of detail and selection of layers of geographic data.

This chapter provides the motivation for the conducted research. Based on the motivation the objectives and the scope of this thesis are defined. In addition, the research questions to be answered by this thesis are presented. The research approach selected for this thesis is described in the methodology section. The chapter ends with an overview of the thesis structure.

1.1 Motivation

Producing data or readable maps for a specific use at an arbitrary scale in an automated way is a long standing challenge in geographic information research and practice [123]. Automated generalization provides the tools to generate such data and representations on-demand. New technology and business developments increase the demand for such automated solutions. One of these new technology developments is the establishment of the web as a ubiquitous platform for accessing any information the user demands from any device (such as mobile phones or personal computers). Regarding new business developments, data providers such as National Mapping Agencies (NMAs) have to offer on-demand data products to various clients at a competitive prize scale [108, 114, 177]. Such data products are an important business case for NMAs to compete with proprietary data providers (e.g. TeleAtlas) and community projects (e.g. OpenStreetMap [145]) on the market. Another business development is in the area of e-governance. Disseminating on-demand maps on the web improves information communication such as physical planning maps improve the planning process regarding urban development.

This research identifies two aspects as essential but unsolved for providing on-demand maps on the web generated by automated generalization. Firstly, an approach to formalize user requirements towards the maps is missing, but required to generate on-demand maps on the web by automated generalization. In the following, these formalized user requirements will be called *user profiles* according to ETSI [42] (Section 2.5). Secondly, enabling meaningful generalization processing on the web is required, to support distributed and scalable processing in an automated way. These two aspects benefit from computational capabilities and network bandwidth. Additionally, advances in technology standards and the establishment of Geographic Information Infrastructures (GII) on the web are beneficial [110]. These advances are essential to create an architecture for on-demand web mapping, as they provide standards and best practices stemming from experiences with GIIs. Standards are especially important for the interoperability of the involved components of the architecture.

In the remainder of this section the motivation is explained in more detail concerning user profiles, meaningful generalization processing and the application of base maps on the web.

1.1.1 User Profiles for On-demand Web Mapping

Serving a map regarding a specific level of detail and regarding a specific map configuration is crucial in map communication, as explained by Kraak and Ormeling [109]. User profiles support map communication as they describe the user requirements towards the map. In general, user profiles can be defined as a means to make digital systems aware of specific user requirements. Standards for specifying user requirements regarding on-demand maps exist partially, such as standards for describing symbolization (Styled Layer Descriptor (SLD)) and content (Web Map Context (WMC)). However, a user profile capturing the requirements of the user regarding the level of detail and specific configuration of the cartographic objects and which consequently can be applied to automated generalization is still missing. Therefore, this thesis presents an approach for user profiles describing a map from a user perspective, which defines the transformation of map content regarding scale and level of detail.

Based on the user profile the system can perform the necessary generalization and disseminate the map accordingly. This thesis aims at embedding these user profiles on the web, as this is considered to be the key platform for disseminating information and for communicating within the society in the future [107].

1.1.2 Meaningful Generalization Processing on the Web

Web Generalization Services evolve currently and gain increasing attention by the research community and industry [47, 129]. They provide specific generalization functionality on the web based on Web Service technology and can be used in a semi-automatic way. This interaction is enabled through interoperability, which can be established on a syntactic and a semantic level. Syntactic interoperability is established by common input and output parameters of the specific Web Service interface. Semantic interoperability is established, if the meaning of these parameters is described in such a way that a client can interact meaningfully with the provided functionality. In particular, semantic interoperability is based on common characteristics shared by the components of the architecture. Such common characteristics can be captured in a classification and are the building blocks to create meaningful descriptions of designated functionality. Semantic interoperability is also subject of research in the context of the Semantic Web [13]. To interact meaningfully with Web Services semantic interoperability is a key requirement, which is enabled by formalizing the functionality of the specific Web Service. Semantic interoperability is not yet established for Web Generalization Services and thus meaningful generalization processing is not available yet. This is due to the missing meaningful descriptions of generalization functionality and the loosely-coupled structure of the web. Establishing meaningful generalization processing on the web is interesting, as it allows a client to interact automatically with an unknown number of web-accessible generalization algorithms to achieve a readable map.

This thesis proposes an approach to establish meaningful descriptions of web-based generalization functionality. Such descriptions are a key requirement for on-demand web mapping applications which require web-based generalization functionality. Web-based generalization functionality may enrich the functionality of existing generalization systems and enable distributed processing of automated generalization. Distributing the processing effort on the web for generalization decreases the processing load for a single computing entity and thereby improves the computational performance of the automated generalization. Additionally, meaningful descriptions contribute to the theory of automated generalization, as they are based on classifications, which improve knowledge exchange between experts in research and practice of automated generalization.

1.1.3 On-demand Base Maps on the Web

The thesis addresses the special case of so-called on-demand *base maps*. Base maps are relevant for map communication of thematic content on the web. Thematic content describes the spatial distribution of a particular theme or attribute [170]. Such content becomes increasingly available due to an on-going paradigm shift from analog to digital and from desktop to a web-based information infrastructure in several domains such as physical planning [148]. However, thematic content mostly lacks sufficient information regarding the orientation and localization of the user, as it is sketched and maintained on an existing (topographic) base map, but then detached from this source due to the layer concept. The layer concept is common in existing Geographic Information Systems (GIS) and combines different data separated as layers in one view [201]. On-demand base maps are one of these layers and support the communication of the thematic content.

One popular example of thematic content is physical planning data, which describe development areas. The use case of this thesis aims at physical planning as defined by the *DURP ondergronden project* [37]. The use case of physical planning has a strong relation to developments in society and technology. It aims at improving the communication within public participation and e-governance processes and aims at a web-based strategy to disseminate the planning information to the different users, who may be planners and citizens [25]. The use case of the DURP ondergronden project will be further described in Section 2.6.

1.2 Objectives

This research develops a web-based architecture for the generation and dissemination of on-demand maps, especially with a focus on base maps. In particular, it aims at the following objectives:

1. Developing an approach to formalize the user requirements for on-demand base maps
2. Designing an architecture for generating and disseminating on-demand base maps based on formalized user requirements
3. Designing a concept for meaningful generalization processing on the web.

The first two objectives aim at the aspect of generating and disseminating on-demand base maps on the web. This is achieved by formalizing user requirements as user profiles and by designing a web-based architecture supporting these user profiles. The third objective aims at establishing semantic interoperability of generalization functionality by classifying it and formalizing it for web usage.

The objectives are also supported by the DURP ondergronden project. The DURP ondergronden project addresses usability-related and technological-related research [150]. This thesis focuses on the latter one. The usability-related research identifies some key use and user requirements for the customized base maps, which will be input for the technology-related research. The usability-related research is intended to also yield requirements for the base maps laid down by the level of detail of the physical planning information. For example, a plan with accurate plan information can be portrayed on a detailed base map, whereas a plan that still contains fuzzy description of boundaries should not be portrayed on a highly detailed base map. A complete list of these requirements is still subject to research [40].

1.3 Scope of the Thesis

Based on the objectives (Section 1.2), this thesis focuses specifically on the following issues:

- On-demand web mapping
- Web-based geoprocessing
- Web Generalization Services
- Generalization operators
- Formalization of generalization functionality
- Agent-based generalization.

The following issues are related but outside the scope of this thesis:

- Automated generalization of 3-D topography [102], thematic data and raster data
- User specific research regarding base maps and defining holistic user requirements [40]
- Multi Resolution Databases (MRDBs) and update propagation between scales (i.e. incremental generalization) [105]
- Performance and production considerations
- Multi-source cartography for optimizing the symbolization of different layers in a map [143]
- Vario-scale data structures [142]
- Automated generalization for mobile applications [165]
- 3D, dynamic or temporal generalization.

Overall, this research focuses on the architecture and not on obtaining the optimal map. Appropriate approaches in cartography for obtaining the optimal map are described in Kraak and Ormeling [109] and Slocum [170].

The presented research is relevant to researchers and professionals in the GI-domain, who explore the possibilities of Web Service-based processing especially in the context of automated generalization. Furthermore it is interesting to system architects and service and data providers, who are in need of an automated approach for disseminating their maps and data on-demand. Regarding the applied use case of physical planning, the research will provide insights for decision makers, who have to prepare future strategies for generating and disseminating physical planning maps on the web.

1.4 Research Questions

Based on the identified objectives (Section 1.2) the following main research question can be extracted:

- How can users integrate maps at different scales according to their demands obtained by automated generalization on the web?

This research question is subdivided in the following four sub-questions:

1. How can on-demand base maps be generated?
   Answering this research question is related to the design of the generalization process and the different aspects of the user profile. It thereby links to objective 1 but also to objective 2.

2. How can these on-demand base maps be disseminated on the web?
   This research question is a follow-up on research question 1, enabling the generated maps for web-based dissemination. Answering this research question is related to the formalization of user profiles but also to the architecture. It thereby links to objective 1 and objective 2.

3. How can processes for automated generalization be established on the web?
   Answering this research question aims at establishing web-based processes for automated generalization. This research question is linked to the issue of syntactic interoperability and is related to objective 2 and objective 3.

4. How can interoperability of web-based processes for automated generalization be improved?
   Answering this research question is closely linked to objective 3. Improving the interoperability of web-based processes will enhance the meaningful discovery and invocation of generalization functionality on the web.

1.5 Methodology

The thesis addresses two aspects of generating and disseminating on-demand maps on the web and enabling meaningful web-based processes for automated generalization as presented in Section 1.1. To answer the research questions defined in Section 1.4, the thesis presents a web-based architecture, which is developed in the different phases of 1) design, 2) implementation and 3) evaluation.

1) In the design phase the concept of the web-based architecture is presented. This includes an analysis of the requirements regarding the web-based architecture, but also an analysis of the user requirements towards the base maps. The user requirements are modeled as user profiles and the architecture is modeled using standards from the Open Geospatial Consortium (OGC) [138] such as OGC Web Map Service (WMS) and Web Processing Service (WPS). The core of the architecture is the so-called generalization-enabled WMS, which generates and provides the on-demand maps according to the specific user profile.

2) The architecture is implemented as a proof-of-concept using state-of-the-art software such as the GeoServer application server and 1Spatial Clarity. Designing and implementing the user profiles and the corresponding generalization gives answers to research question 1. Designing and implementing the web-based architecture aims at answering research question 2.

3) Based on the evaluation of the architecture different drawbacks are observed. In particular, the semantic interoperability of web-based generalization functionality to achieve meaningful processing is not yet supported.

The syntactic interoperability is demonstrated by different applications and aims at answering research question 3. To improve the semantic interoperability and to answer research question 4, this research proposes a classification of generalization operators for meaningful integration of web-based generalization. This classification is described in the Object Constraint Language (OCL) and is embedded in the web-based architecture using current approaches for meaningful web-based geoprocessing such as WPS Profiles and XML Metadata Interchange (XMI) format.

Semantics and their formalization can be described for three aspects. Semantics of data, semantics of functionality (encapsulated as Web Services) and semantics of user tasks. In this research, the semantics of generalization functionality (encapsulated as Web Generalization Services) and the semantics of user tasks are investigated. The semantics of the generalization functionality are captured by the concept of generalization operators and their classification. The functionality is finally formalized for the application of Web Services. The semantics of user tasks are formalized through the user profiles for on-demand base maps.

The definitions of geographic data and geographic information are closely related to the definitions of data and information besides their specific focus. Geographic data and geographic information describe geographic phenomena at different levels of abstraction. The terms data and information are not clearly defined in literature. This research follows the definitions of Ackoff [3] and Chen et al. [28]. Ackoff [3] defines data as a set of symbols and information as data which is processed to answer specific questions. Chen et al. [28] applied these definitions to the computational domain by defining data as computational representations of models and attributes of real world or simulated entities (i.e. geographic phenomena). Whereas information is data which have meaning attached and are thereby understandable by computational systems or human users. From a data modeling perspective and taking these definitions into account, data are represented in a source data model and information are represented in a target data model. The target data model incorporates meaning by referring to the context, the human user or the computational system can understand to answer the question.

1.6 Chapter Overview

Chapter 2 - This chapter describes the context of this thesis. It introduces the relevant concepts of automated generalization, Web Services, Web Generalization Services, modeling and user profiles. The sections about current practice of automated generalization and Web Generalization Services also describe some novel work carried out within this research. Regarding Web Generalization Services, the presented novelty of the work in this chapter focuses on proving the applicability of the WPS interface specification. In addition, the chapter describes the use case of on-demand base maps for physical planning, on which the thesis is based on.

Chapter 3 - This chapter presents the design of the architecture. In particular, the requirements of the web-based architecture are examined and the conceptual architecture is presented. It also presents the user requirements towards the base map as applied in this thesis and the design of the user profiles for describing these user requirements.

Chapter 4 - This chapter presents the implementation of the architecture based on the design described in Chapter 3. Based on the implementation this chapter evaluates the architecture and describes several limitations.

Chapter 5 - One of the limitations encountered in Chapter 4 is the lack of semantic interoperability. This chapter presents an approach to overcome this lack of semantic interoperability of Web Generalization Services by proposing a classification of generalization functionality and formalizing it in OCL.

Chapter 6 - This chapter summarizes the answers to the research questions, discusses the findings of this work and provides an outlook for future work.

Each chapter ends with a synopsis concluding the addressed aspects. Table 1.1 presents an overview of the publications achieved by this research in relation to the particular chapter. An overview of the contributions in the specific field of research

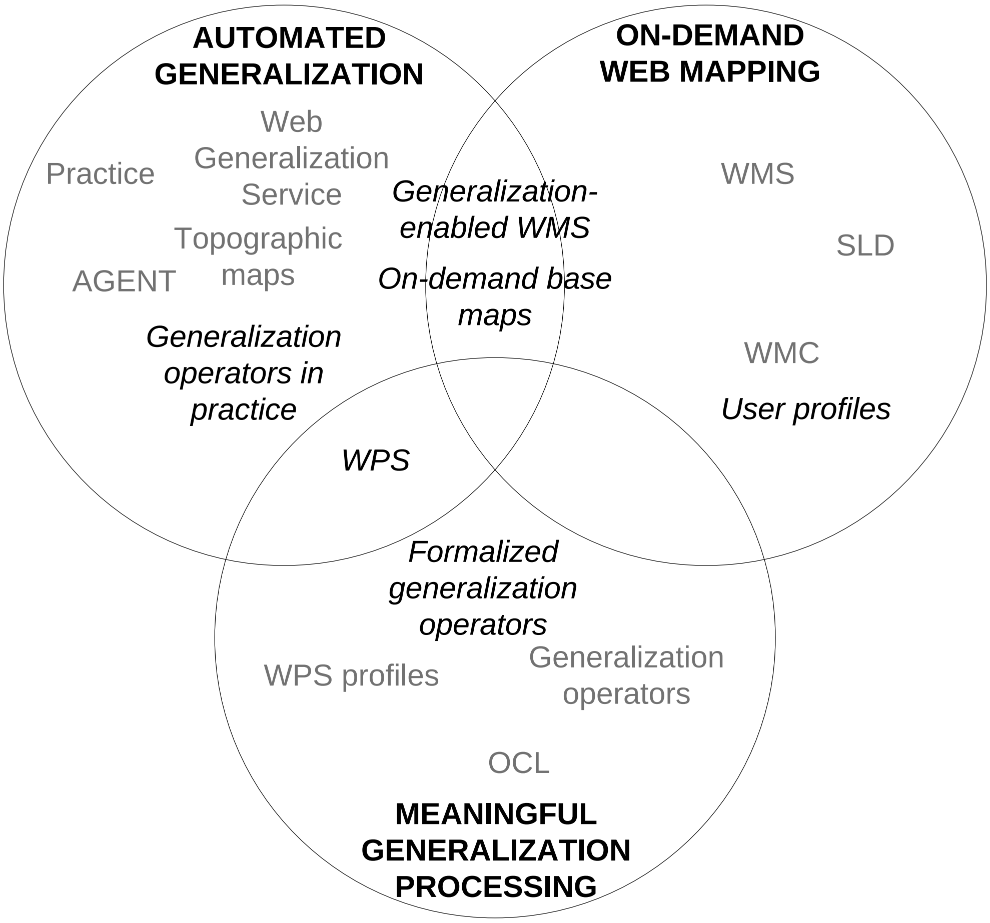


Figure 1.1: Overview of research contributions. The contributions of this research are highlighted in italic font.

as identified in this research is depicted in Figure 1.1. The contributions of this work are highlighted in italic font.

Chapter	Aspect	Publication	Type
Chapter 2 - Context	WPS	[19]	Peer-reviewed full paper - conference
		[50]	Springer book chapter
		[51]	Peer-reviewed IEEE full paper
		[54]	Extended abstract - workshop
		[166]	Peer-reviewed journal
	Generalization in practice	[59]	ISI-journal paper
Chapter 3 - Design	User profiles	[53]	ISI-journal paper
Chapter 4 - Implementation & evaluation	Generalization-enabled WMS	[61]	Springer book chapter
	Grid	[52]	Peer-reviewed journal
	RO-Online	[56]	Peer-reviewed abstract - conference
Chapter 5 - Meaningful generalization processing	Classification of Web Generalization Services	[48]	ISI-journal paper
	Generalization operator classification	[57]	Peer-reviewed full paper - conference
	Operator classification in OCL	[49]	Peer-reviewed full paper - conference
	Ratio-based simplification	[58]	Peer-reviewed full paper - conference

Table 1.1: Overview of publications contributed by this research.

Chapter 2

Context of this Research

Based on the objectives (Section 1.2) and the research questions (Section 1.4), this chapter presents the context of this research. It introduces the relevant concepts and literature which serve as building blocks to design, implement and evaluate the web-based architecture, which has been developed by this research.

For this research automated generalization is the identified means to generate on-demand maps. Thus, Section 2.1 presents relevant literature and concepts of automated generalization. The research aims at a web-based dissemination of these maps by the means of Web Services. Therefore, Section 2.2 presents the concept of Web Services and OGC Web Services in particular. To present the current achievements in the field of Web Services and automated generalization, Section 2.3 reviews the related work and concepts. As the design of the architecture is a key aspect of this research and modeling is the means to achieve it, Section 2.4 presents common approaches for modeling and standards. Section 2.5 reviews the standardization of user profiles outside the GI domain, to provide a foundation for the design of the user profile for on-demand base maps in Section 3.2. The architecture is designed and implemented for the special application of on-demand base maps. This application is exemplified by the DURP ondergronden project, which is presented in Section 2.6. The DURP ondergronden project serves as a use case to test the architecture. The chapter ends with a conclusion of the most relevant implications for this research.

Section 2.1 and Section 2.2 also present some work, which has been part of this research to give more background to the current challenges of automated generalization in practice and to underline the problem and importance of semantic interoperability respectively.

2.1 Automated Generalization

Generalization as applied in the GI-domain is a fuzzy concept regarding its various definitions and applications. In particular, it is deeply influenced by cartography and the application of topographic maps. Originally, generalization was performed on a mapsheet to display a topographic setting on a reduced scale. Manual generalization requires technical skills and scientific interpretation. The cartographer applies generalization according to heuristics and interpretation of the designated theme [88]. Its automation is a challenge for research and practice since digital systems and data became available in the 1970s. At that time, data providers, mostly NMAs, started to shift the manual generalization process to an automated process.

A uniform definition of generalization is hard to give, due to its multi-purpose nature and the different applications it is used for. The International Cartographic Association (ICA) defines generalization as:

> ...the selection and simplified representation of detail appropriate to the scale and /or the purpose of a map... [86, p. 173]

This definition only considers the map as the final output and visual representation as the key aspect. In a digital context however, generalization is also applied to support the following aspects [195]:

- Developing source database based on real-world surveying
- Use computational resources economically (e.g. storage and processing facilities)
- Improve data robustness
- Derive data and maps for multiple purposes
- Optimize visual communication.

Automated generalization can be defined from a user perspective or from a data modeling and computer science perspective, respectively.

From a user perspective automated generalization transforms geographic data into geographic information. In this context, information is defined as an answer to a specific question. The transformation concerns scale, level of detail and use. It is important to note that information does not have to be a map or even a visual representation, but can also be a database table. From a computer science or data modeling perspective, automated generalization can be considered as a transformation of data from a source model to a target model. In this case, the information is actually structured according to the target model. Again, the model can be either describing a visual representation (such as a cartographic model) or data.

Aspects of automated generalization are covered in books [24, 123, 126, 196] and special journal issues with a focus on automated generalization on the web [101, 197]. Additionally, several PhD studies have been carried out lately (since 2006) as for instance:

- Jabeur [97] implemented a Multi-Agent System [200] for automated generalization.

- Neun [129] presented a Web Service architecture for generalization. His work differs from this research, as it did not aim at on-demand web mapping nor did it aim on improving the semantic interoperability of Web Generalization Services. Still, it is a fundamental contribution with a special focus on the syntactic interoperability of Web Generalization Services based on XML-messaging and the publish-find-bind paradigm (see also Section 2.2.1).
- Gaffuri [63] developed an extension of the AGENT model [159] (see also Section 2.1.3).
- Haunert [83] proposed a new approach for aggregating areal features.
- Chaudhry [27] investigated the effects of semantic relations and generalization.

The *ICA Commission on Generalisation and Multiple Representation* [87] and the *ISPRS working group on Multiple Representation of Image and Vector Data (ISPRS WFS II/3)* [96] serve as platforms to discuss advances in automated generalization.

As stated before, automated generalization is a long standing challenge. The simplification algorithm of Douglas and Peucker [34] can be considered as one of the first efforts reported on automated generalization. As observed by McMaster and Shea [126, Chapter 2], a lot of efforts only considered isolated problems (e.g. generalization of linear objects). Since computational power advanced in the late 1980s, researchers such as Smaalen [171] in the context of databases or Beard [11], Ruas [158] in the context of constraints (Section 2.1.2) started to take also contextual information into account. Most lately the application of web technology has gained considerable attention by researchers [129, 165].

Still the problem of automated generalization remains unsolved and it is doubted by some, if this can be solved on the long term. Problems are related to the complexity of data and generalization processing and the requirement to achieve the quality of manual generalized paper maps. The current problems of automated generalization in practice are described in Section 2.1.5. However, as pointed out by Regnauld [153], if the requirements towards the generalized product are limited, automated generalization is applicable and can be achieved by little costs. However, satisfying all requirements is only possible under high costs in comparison to the manual process and is thereby not suitable. Mackaness et al. [124] identified different challenges in generalization research such as data models, interoperability and modeling user requirements. The current challenges and problems of automated generalization in practice are described in Section 2.1.5 based on a survey performed as part of this research.

Scale and Level of Detail

Scale and level of detail are important concepts for automated generalization and are also addressed in this thesis, especially for defining the user requirements towards the base map (Section 3.1.2). Scale is a broad concept, which is not formalized, but implicitly incorporated in any information. In general it refers to the extent of an observation [163]. Cartographic scale, or scale, as it is referred to in this work, can be defined as the ratio between a distance on a map and the corresponding distance on the terrain [109]. Related to scale is also level of detail. Level of detail describes precision and accuracy of an object as captured in the database or portrayed in the map in relation to the real world object. A specific scale on the map implies a specific level of detail. However, the level of detail of different objects on the same map might vary depending on the intended use and user of the map. A formalization of level of detail does not exist.

This chapter highlights the relevant concepts of automated generalization as applied in this thesis. The conceptual models in Section 2.1.1 are presented to define the appropriate approach for generating the on-demand maps. Cartographic generalization has been identified as the key concept in this thesis. To implement cartographic generalization Section 2.1.2 introduces “constraint-based generalization”, which is the common approach to formulate the process of cartographic generalization. Constraint-based generalization is implemented for instance by agent-based generalization which is described in Section 2.1.3. Section 2.1.4 presents the concept of generalization operators, which is one of the building blocks in generalization and is identified in this research as the key concept for meaningful generalization processing on the web. To highlight the current challenges of automated generalization in practice, Section 2.1.5 presents an analysis based on a survey and a workshop performed as part of this research. The results of the analysis put the research into practical perspective.

2.1.1 Conceptual Models for Automated Generalization

Different conceptual models on generalization have been developed [196]. Their aim is to describe the patterns, which are common in automated generalization. They support the design and implementation of systems for automated generalization from a scientific perspective. In the following the main conceptual models are introduced. These different models have different implications for automated generalization. The McMaster and Shea Model and the Brassel and Weibel Model can be applied as parts to the Grünreich Model. Thus they can be used in a complementary way. Whereas, the Grünreich Model is much more fundamental and affects the design as well as the implementation of automated generalization significantly.

McMaster and Shea Model

The McMaster and Shea model [126] divides the generalization process into three aspects: philosophical objectives, cartometric evalulation and spatial and attribute transformations. The first aspect describes the reasons for the generalization process. The cartometric evaluation establishes the different types of conditions, which describe the cartographic requirements. The third aspect addresses the different operations,

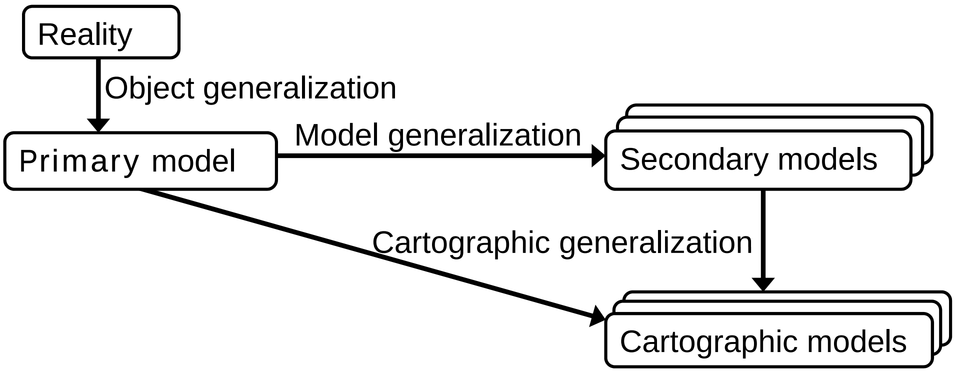


Figure 2.1: ATKIS-model for generalization [75]

which will be performed, if the cartographic requirements are not met.

Brassel and Weibel Model

The Brassel and Weibel model [18] describes the generalization process by the tasks of structural recognition, process recognition and process modeling. The structural recognition perceives the structure of the data and the interrelations of the different objects. This knowledge will be utilized by the process recognition, which compiles a complete generalization process to meet the generalization objectives. The compiled process knowledge builds the basis for the process modeling, which is connected to an environment with the actual generalization operators. The resulting process model will finally be executed.

Grünreich Model

The model of Grünreich separates three different stages of generalization: object generalization, model generalization and cartographic generalization (Figure 2.1). Object generalization captures the real-world objects according to a primary data model. Model generalization describes the transformation of data from the primary data model (source model) into a secondary data model (target model). Cartographic generalization aims at producing maps according to a cartographic model out of data by avoiding cartographic conflicts. The last two aspects of the model define a clear separation of data and representation and is for instance implemented for the production of the German topographic products [15].

The definition of cartographic generalization being applied on symbolized features (i.e. cartographic features) is reflected by the processing scheme for cartographic generalization (Figure 2.2). Symbolization is applied on the features, which are based on

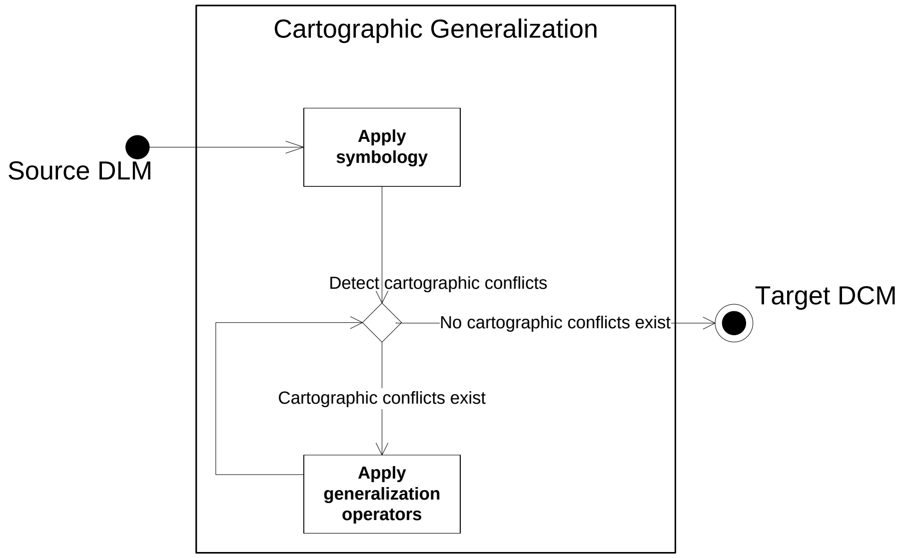


Figure 2.2: The cartographic generalization process

a Digital Landscape Model (DLM), in advance to create the cartographic features according to the Digital Cartographic Model (DCM). The cartographic generalization process performs then on these cartographic features without changing their symbolization. A change in symbolization would result in a change of the cartographic model. This separation is also practiced in Burghardt et al. [20], Cecconi [26] and is applied in standards for web mapping (Section 2.2.3).

2.1.2 Constraint-based Generalization

Beard [11] proposed the concept of "constraint-based generalization" as a new concept to overcome the problems of rule-based generalization [126]. In contrast to rule-based generalization, constraint-based generalization specifies only the final outcome without describing how to achieve it.

This thesis extends the concept of constraint-based generalization, by introducing the more general concept of *map generalization specification.* A map generalization specification consists of constraints and optimization goals. Constraints in this case are adopted from database theory and describe characteristics for database consistency, which have to be achieved or preserved by the generalization process. Optimization goals substitute Beard's concept of constraints and are meant to be guidelines which might be met under certain conditions but may be ignored due to limited map space or other other competing optimization goals.

Cartographic generalization performs the transformations on the map according to

Description	Beard [11]	Steiniger and Weibel [172]	Thesis terminology	
Condition towards an optimal state	constraint	soft constraint	optimization goal	map generalization specification
Fixed condition which has to be met		hard constraint	constraint	

Table 2.1: Terminology overview for constraint-based generalization.

the map generalization specification. By describing the final map, the map generalization specification provides a user oriented-view on automated generalization. This eases the configuration of the generalization process also for non-expert users of generalization systems [11]. To satisfy multiple optimization goals (i.e. the conditions might be met) on the same object optimally, the concept of importance has been introduced by Ruas [158] (resulting in an overall optimization goal based on the relative importance levels). The importance guides the generalization process regarding which optimization goal is more important to be met.

As it appeared in this section, the terminology in the context of constraint-based generalization is overlapping with the concept of constraints in database theory. Steiniger and Weibel [172] for instance refer to constraints as *hard constraints* and optimization goals as *soft constraints*. However, this thesis proposes that the term constraint is used as in the database theory, that is hard constraint and that the term optimization goal is used instead of soft constraint, as optimization is actually involved when solving this type of problem.

Table 2.1 provides an overview of the discussed terminology. The thesis will only distinguish between optimization goals and constraints, if necessary. Otherwise it will use the term map generalization specification to summarize both concepts.

2.1.3 Agent-based Generalization

Cartographic generalization can be considered as an optimization problem [167], as the cartographic objects interfere on the limited map space and the legibility of the cartographic objects has to be maintained as much as possible. Many optimization approaches for cartographic generalization have been developed such as the snakes method [21], elastic beams [8], least square adjustment [81] and simulated annealing [193]. Also an agent-based system [200] for cartographic generalization has been developed within the *Automatic GEneralisation New Technology* (AGENT) project [10, 113]. In this system each of the cartographic objects is controlled by an agent, which tries to maximize the proposed optimization goals. Moreover, the agent-based approach applies an object-oriented model. The interaction of the agents is defined

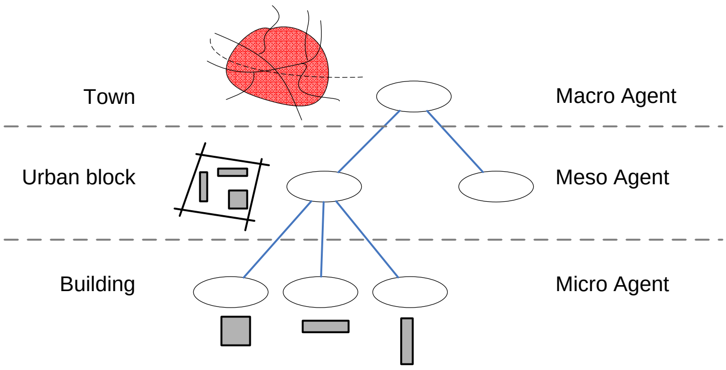


Figure 2.3: Agent model - describing relation between macro, meso and micro agent [159].

by different models [36]. The CartACom model specifies the communication between agents [35]. The AGENT model [159] groups agents for instance by their topological relations. Thereby it allows the generalization process to maintain the original topological relations if necessary. A single cartographic object is controlled by a micro agent and a group of cartographic objects (i.e. a group of micro agents) is controlled by a meso agent. The macro agent controls all cartographic objects of the map display to maximize global optimization goals (Figure 2.3). A macro agent can be considered to be a special type of meso agent, as it also controls a set of objects.

The results of the AGENT project are implemented in the generalization software Clarity of the company 1Spatial [79]. 1Spatial Clarity enables users to specify constraints on cartographic objects and to perform generalization by using the agent-based generalization, as introduced in this section. The software currently supports the production of map products at several NMA's such as KMS Denmark [77] and IGN France [115]. Also in research at NMAs Clarity is applied such as at Ordnance Survey research [155]. Due to the wide-spread application of Clarity at NMAs and its high complexity, a group of NMAs found the *Mapping Agencies Generalization NETwork* (MAGNET) consortium, which is a platform to exchange information about 1Spatial Clarity and a communication channel to formulate requests to 1Spatial [116].

Therefore, in this research the generalization of the base map will be implemented based on 1Spatial Clarity (Section 4.1 and Section 4.2).

2.1.4 Generalization Operators

Generalization operators evolved within the early stage of generalization research and provide abstract descriptions of single actions of the cartographer during manual generalization. Thus a generalization operator is an abstract description of atomic generalization functionality. It is atomic in the sense, that it only affects well-defined and isolated aspects of a single feature or a group of features in an undividable way.

McMaster & Shea		Cecconi		Liu et al.
spatial transformations	Simplification	*<unspecified>*	Thematic selection	Simplification
	Amalgamation		Thematic aggregation	Merge
	Refinement	*individual objects*	Weeding	Amalgamation
	Displacement		Unrestricted simplification	Aggregation
	Smoothing		Enlargement	Classification
	Merging		Exaggeration	Selection
	Exaggeration		Fractalization	
	Aggregation		Smoothing	
	Collapse		Rectification	
	Enhancement	*individual or groups of objects*	Selection	
attribute transformations	Symbolization		Elimination	
	Classification		Displacement	
		groups of objects	Amalgamation	
			Combine	
			Typification	

Table 2.2: Overview of existing classifications.

But it may require several basic computations to get the result such as intersect or compute distance. Nevertheless, being atomic does not imply that such functionality is not without any side effects. Each generalization operator can be implemented by several different algorithms. An example is the simplification operator, which can be implemented by the Douglas-Peucker Algorithm [34] and the Visvalingam-Whyatt algorithm [189]. Generalization operators aim at comparing and classifying different generalization algorithms.

Several initiatives and researchers proposed a classification of generalization operators. An overview of a selection of classifications is provided in Table 2.2.

McMaster and Shea [126] introduced a first classification of generalization operators, which consists of twelve operators in two categories. Their introduced separation of spatial transformations and attribute transformations is trivial in the sense that it classifies classification and symbolization as the only attribute transformations and the others as spatial transformations. Additionally, the classification does not reflect the current practice of data production, as symbolization is mentioned as a generalization operator. In current data production as well as in GI research the visualization and the data are separated to reduce complexity and avoid redundancy.

The AGENT project [113] proposed also a classification of generalization operators. The project focused on enhancing automated cartographic generalization for map production, as already described in Section 2.1.3. The project applied a hierarchy of communicating agents, which solve cartographic conflicts on the level of single features (micro agent) and groups of features (meso agent). Thus, the operators were separated in these two groups, as reported by Cecconi [26]. However, the classification does not cover the operators related to model generalization, because it focuses on cartographic generalization by applying the agent-based approach.

The classification of Liu et al. [121] aims at an object-oriented framework for model generalization operators. But some important operators for geometry type transformation are not covered such as combine and collapse, which are explicitly listed by the other classifications.

Besides the classifications listed in Table 2.2, Kilpelaeinen [105] describes a distinct set of model generalization operators, which is applied in the context of multi-resolution databases. She extracts six model generalization operators. Combining these six operators with the basic database functions `add` and `delete` results in a comprehensive functionality requirement for multi-resolution databases. However, a concept for formalization is not given as well as for the other mentioned classifications.

Also Hake et al. [78] identified some distinct operators, which they did not classify. They state that different operators address one or more of the following generalization tasks: Semantic generalization, geometric generalization and temporal generalization. However, a clear separation or a clear definition of the different generalization tasks is missing.

Most lately Regnauld and McMaster [154] described their view on generalization operators and provided a review on frameworks for generalization. They examined the different operators for common generalization tasks such as, building generalization, line generalization and network simplification. However, they did not aim at providing a holistic classification of generalization operators.

2.1.5 Automated Generalization in Practice

Automated generalization in practice is mostly performed by NMAs to generate topographic data at different scales. According to Keates [103, Chapter 2], topographic data can be defined as geographic data, which describe all the identifiable features on the earth's surface, whether natural or artificial.

Topographic data are mostly stored in topographic databases at multiple scales. They have to support a lot of different applications (e.g. transport, electricity, etc.) with topographic data (e.g. for geographic analysis). Providing topographic data for this broad range of applications through NMAs is mostly motivated by the legal responsibility and interest of each country. To support different applications topographic data is modeled by complex data models, which support complex class hierarchies, complex topological interrelations and different scales. NMAs are mostly in charge of designing, maintaining and providing such databases.

Recently, private companies became competitors for NMAs in providing topographic data such as TeleAtlas (Section 1.1). Their products provide multi-national coverage of mostly road data and are widely used for car navigation, web and mobile applications. Moreover, besides these profit-oriented companies, the OpenStreetMap project provides data free of charge. Their data are produced by volunteers, capturing GPS tracks worldwide. However, the technical challenges related to data modeling and automated generalization remain. The challenges may be of different complexity between countries for instance due to the country's geographic setting or the existing databases and implementations of generalization functionality.

To define NMAs' challenges with respect to automated generalization as starting context of this research a qualitative and a quantitative survey have been carried out. The qualitative analysis especially focused on the trends and policies of automated generalization within NMAs [174, 175]. The analysis is based on a workshop held in 2005 and attended by twelve NMAs. As part of this research, Foerster and Stoter [55] performed a quantitative analysis of automated generalization in practice as performed by NMAs. In particular, it addressed the current challenges of automated generalization on a tangible level, investigating its technical aspects, especially with a focus on generalization operators. It was based on a survey between eleven NMAs from eight European countries. The template of the survey and an extensive discussion of the results can be found in Appendix C and Appendix D. A comprehensive summary of both analyses has also been described in Foerster et al. [59].

From the qualitative analysis it can be concluded that full automation is not applied at any NMA, although some NMAs have made large investments and achieved major steps, of which Denmark is a representative example. Another important conclusion of the workshop is that there is no single approach for the adoption of automated generalization within NMAs. It heavily depends on NMA-specific factors related to scale, specific configuration of the geographic landscape, data models, or organizational aspects.

Ready to use software for automated generalization is therefore not considered as appropriate for automated generalization due to the intrinsic complexity of the generalization task and of the geographic data and due to the mentioned NMA-specific factors. Instead, it will require implementation as well as remodeling efforts of NMAs to introduce automated generalization into their own production lines. To support these NMA specific processes, NMAs need adjustable systems as well as generically applicable generalization functionality. Providing a common view on such functionality, reflecting NMA requirements, may support researchers and software vendors to develop automated generalization solutions for NMAs.

This was the motivation for the quantitative analysis on missing generalization functionality based on the quantitative survey [55]. In particular, the participants of this survey indicated the importance of specific operators and the problematic operators on a scale from 1 (easy) to 10 (difficult) to generate their designated product. Thus, it was possible to provide detailed insights into currently applied strategies towards generalization operators and current problems of generalization operators at NMAs. The analysis demonstrated the relevance of specific generalization operators by combining the importance and problematic (i.e. lacking) aspects of operators (see also Appendix D.1 and Appendix D.2). This showed that the relevance of model generalization operators increases with decreasing scales, but never reaches the relevance level of cartographic generalization operators.

Weighting the relevance measures by importance values of feature types results provides another valuable insight (Appendix D.3. Especially network-based feature types such as rivers, railways and roads are relevant for NMAs in combination with the generalization operators enhancement, typification and elimination (as defined in Section 5.2.1). Overall, contextual generalization operators and generalization operators creating generalized features that inherit a network-based structure are the main challenges for cartographic generalization. This underlines the findings of the qualitative analysis.

The presented results of the qualitative and quantitative analysis describe the long term challenges for NMAs. They may therefore serve as a guideline for NMAs, researchers and software suppliers to better align their activities. The presented work also extends the findings of the OEEPE project [160] and the EuroSDR project [176] as it studies generalization operators not limited to specific generalization solutions or test cases, but as applied and required in NMA production lines.

Mackaness et al. [124] state that research on automated generalization should *connect* to practice for better meeting their requirements and for streamlining research activities. This analysis is an example of obtaining better understanding of NMA requirements for automated generalization and of identifying topics for further research starting from a requirement analysis at NMAs. In addition, exchanging knowledge about generalization operators, the main building blocks of automated generalization, sharpens the terminology in practice and research and thereby improves the interoperability of concepts. This will enable more flexible and effective solutions both in databases as well as on the web. In the future the set of criteria applied in the quantitative analysis could be reassessed to identify the success and remaining problems of automated generalization in practice as for instance by NMAs.

2.2 The Geospatial Web

The term "Geospatial Web" refers to a collection of "Geospatial Web Services", which are queryable and integratable for geographic applications [112]. To enable the vision of the Geospatial Web, standards play an important role.

A GII is an instance of the Geospatial Web and provides a framework on an organizational and technical level. The organizational level is defined by policies and guidelines to enhance data exchange across geographic and organizational boundaries. The technical level is defined by technical specifications, which support the organizational level in achieving the designated goals. According to Groot and McLaughlin [74] a GII can be established at different levels from regional, national, cross national to global scale. An example of a national GII is the Dutch GII organized and maintained by Geonovum. The Dutch GII defines a framework of standards, which are mandatory to any organization participating in the Dutch GII [66]. The currently evolving GII on the European level is defined by the INfrastructure for SPatial Information in Europe (INSPIRE) directive [89]. It provides guidelines on an organizational and technical level, to which all the member countries have to comply, when publishing their geographic data and "Geospatial Web Services" [202].

This section reviews the basic concepts of Web Services and the special breed of OGC Web Services. Web Services are the foundation of the web-based architecture and are described in Section 2.2.1. Additionally, the issue of interoperability is described in this section. OGC Web Services are relevant for this research to interoperate with other geographic applications and to integrate existing geographic data, portrayal and processing functionality (Section 2.2.2). In particular, Section 2.2.3 describes OGC WMS and related standards for web mapping. Additionally, Section 2.2.4 describes Geoprocessing Services, as they are the foundation for establishing web-based processes such as automated generalization on the web.

2.2.1 Web Services

A Web Service can be defined as a software component that provides functionality including access to data sources through a web-accessible interface in a programming language- and platform-independent manner [187]. The Web Service interface is described in a machine-understandable way, which is a fundamental requirement for interoperability. Based on these interfaces Web Services connect readily available software components on the Web in a loosely coupled way [6]. Loosely coupled means, that the service interaction is established during runtime and the services do not know each other in advance. This enables to reuse software components in different applications. Moreover, as Web Services communicate based on platform-independent protocols (e.g. Hypertext Transfer Protocol, HTTP) and exchange formats (e.g. XML), they can be reused by any application written in any programming language and/or running on any operating system. Additionally, Web Services are stateless software components, which means that a Web Service does not expose a specific state to the client and remains stateless before and after client interaction. This is a central design criteria to keep the architecture scalable and flexible for many different applications.

The interaction of web services is based on the publish-find-bind paradigm [72], which specifies three roles: service requestor, service registry and service provider. The course of action is depicted in Figure 2.4:

1. The service provider publishes the service description to the service registry
2. The service requestor finds the desired service by querying the service registry
3. The service requestor binds to the service and retrieves the desired function.

An architecture which supports the publish-find-bind paradigm is called a Service-Oriented Architecture (SOA).

To enable the publish-find-bind paradigm Web Services have to be interoperable. The task of establishing interoperability between Web Services (i.e. web service interoperability) is a challenge, as they are connected in a loosely coupled way. The ISO

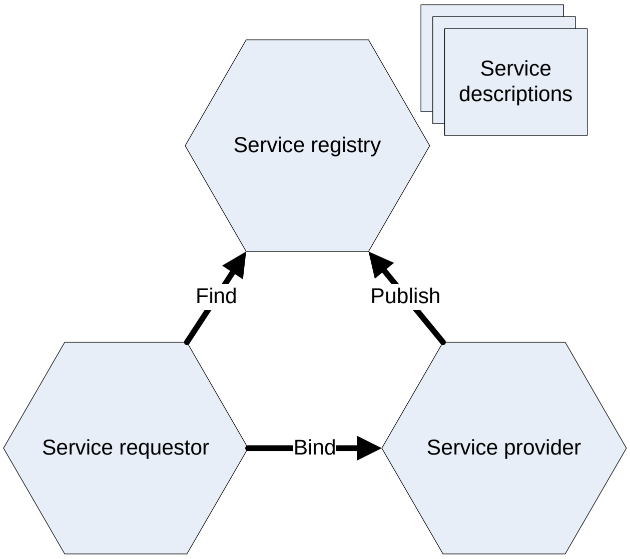


Figure 2.4: The publish-find-bind paradigm.

standard 19119 *Geographic Information - Services* identifies two levels of interoperability for Web Services [95]:

- Syntactic - the Web Services use the same structure and input/output format for the information
- Semantic - the Web Services communicate based on an agreed meaning of the message parameters.

Semantic interoperability and agreed meaning of message parameters is established through common classifications, which are formalized as ontologies [4].

There are also other more fine-grain classifications of web service interoperability, as for instance Nezhad et al. [131] distinguish between syntactic (signature), structural (data structures) and semantic interoperability. The separation between structural and syntactic appears to be standard-specific.

For this work ISO's classification is more suitable, as it states the current situation (i.e. Web Services are interoperable syntactically) and the future perspective (i.e. Web Services will be semantic interoperable).

Web Service interoperability is also crucial for sequencing multiple Web Service instances, i.e. software components, to achieve a designated goal and to create richer functionality. This sequencing of services is called Web Service chaining. In the context of Web Service chaining, three types of user interaction have been classified [5]:

- Transparent - involves full user interaction and requires prior knowledge of the user about the service and the context of the application. The chain is static and preconfigured by the service provider.
- Translucent - the user is aware of interaction within a Web Service chain, but cannot alter the order.
- Opaque - the chain of services is presented to the user as one service, thus the user is not aware of the chain. As in the case of translucent Web Service chaining, the chain is configured by the service provider either in a static or dynamic way.

2.2.2 OGC Web Services

Web Services provide also many advantages for geographic applications as outlined by Kralidis [110]. For this thesis the most important advantages are user-defined data access and data management, which is enabled by specific service interfaces (e.g. WMS). User-defined data access is the basis to realize on-demand web mapping and is thereby the major reason to apply web service technology in the described web-based architecture. Additionally, data management enables integrating data from different sources.

To take advantage of the concept of Web Services and to meet the requirements of geographic applications, such as intrinsic data complexity and specific dissemination requirements for maps and data, the OGC and the International Organization for Standardization (ISO) [91] specified a family of standards. These standards describe exchange formats for data but also Web Service interfaces. These OGC Web Services are also referred to as “Geospatial Web Services” [31]. The specifications of OGC Web Services mainly provide syntactic interoperability, as it is concerned with the encoding of the input parameters, but not with their semantics. Full interoperability, i.e. also addressing interoperability at the semantic level, is still a subject for research and relates to the development of the *Geospatial Semantic Web* [16]. A promising approach to enable the semantic interoperability of Geoprocessing Services is the use of ontologies and semantic service classifications, as introduced by Lemmens [120] and Lutz [122].

The family of standards as specified by the OGC includes:

- data services for providing data through a standardized interface on the web (e.g. Web Feature Service for vector data)
- portrayal services for disseminating maps, mostly being plain images (Web Map Service)
- processing services for publishing and performing web-based processes (Web Processing Service).

Based on these standards it is possible to implement a SOA for geographic applications.

Two differences between mainstream IT Web Services (Section 2.2.1) and OGC Web Services can be observed regarding the interaction pattern and the level of interoperability.

Firstly, mainstream IT Web Services are based on the Web Service Description Language (WSDL), which is searchable through a registry. The WSDL document describes the end point of the service and the operations provided by the Web Service. The WSDL document can easily be changed to describe another service address without changing the information at the registry. Contrarily, each OGC Web Service has a *GetCapabilities* operation (as specified in OWS Commons). This GetCapabilities operation returns the service metadata, which is comparable to the content of the WSDL document. However, as the GetCapabilities operation is associated with a specific OGC Web Service instance, it is required to adjust the registry information of the service, if the specific service changes its address. The registry is not able to update its information, because the service is not reachable using the out-dated address.

Secondly, the level of interoperability is more advanced in the case of OGC web services, as the interfaces describe distinct operations, of which the client knows how to invoke it and what to expect. The meaning of these operations is described textually in the OGC specification documents. Anyway, both types of Web Services are only accessible on a syntactic level of interoperability, but the meaning in the context of OGC Web Services is much more documented than for mainstream IT Web Services based on WSDL documents.

Besides the differences of both type of Web Services, the OGC specifies also exchange formats such as the Geography Markup Language (GML) for vector data. GML provides much more meaning regarding geometry, topology and feature identity, than it is currently supported by mainstream IT standards (e.g. generic XML).

The common and mandatory concepts of OGC Web Services are described in the OGC Web Services Commons Specification [139]. The Web Services Commons Specification ensures interoperability between services and lowers the burden for developers and organizations to adopt new OGC specifications. Designing such concepts is always a trade-off between being generic enough for the different types of services and being specific enough to incorporate specific meaning. An example of such a common concept is the GetCapabilities operation, which is mandatory to any OGC Web Service.

2.2.3 Standards for Web Mapping

Web mapping is covered by the OGC by different standards, the foundation is the OGC Web Map Service (WMS[1]) interface specification. It is commonly used to portray geographic data in form of a map at a certain scale and with a given extent on the web [132]. This map can be based on server-side data or data provided by other vector services. The provided map is a plain image, which can be overlayed with other maps from other WMS instances using WMS-compliant client applications such as the *user-friendly desktop internet GIS* (uDig) [151], *MapBuilder* [125] *OpenLayers* [144] or ESRI ArcMap [146].

The Web Map Service interface specification is the state-of-the-art approach to disseminate maps on the web, especially for public organizations. For instance it is the basis for the INSPIRE view service [182].

The WMS interface specification provides the geographic data according to the layer concept. These different layers are available independently to any client application (i.e. single or multiple layers can be combined from different instances). The metadata about the available layers are provided through the GetCapabilities operation. Based on this information the client can formulate the specific GetMap request to retrieve the designated layer or group of layers. The WMS may provide the different layers with different symbolization (in specification terms called styles). The client can choose the appropriate (pre-defined) style or request a specific layer in a customized style, which is encoded as an SLD document. Additionally, the GetFeatureInfo operation provides access to information about specific features displayed on the map. A basic sequence of actions to retrieve a map is described in Appendix A.2.

Besides the WMS interface specification, two standards specify the symbolization and the content of web maps. They are the SLD and the WMC specification, which will be explained below. The Web Feature Service (WFS) interface specification [134] will not be introduced in detail, as it does not play a significant role for this research, although it is mentioned in the thesis.

OGC Styled Layer Descriptor

The specification of the OGC Styled Layer Descriptor (SLD) describes a way to encode the representation of geographic data, either vector or raster-based. In terms of cartography, the SLD can be considered to be the cartographic model for a specific vector data set. The SLD describes therefore a XML-based encoding for the most essential cartographic attributes of data such as line width and color and specific symbols. The SLD can be used in combination with the WMS to provide on-demand representations of geographic data. An example of an SLD document and an example request, how styles can be dynamically applied to web maps is described in

[1] The Web Map Service is the first official standard of OGC which has been adopted by ISO and is available as ISO 19128.

Appendix A.3.

The application of SLD and WMS has lead to some discussion about its applicability in distributed architectures. In particular, the WMS does not expose the feature types provided in the maps, thus it is difficult for any arbitrary client to define the SLD without prior knowledge of the specific feature types. Therefore, the Feature Portrayal Service interface specification has been lately proposed to overcome this problem [112].

OGC Web Map Context

The OGC Web Map Context specification describes an encoding to formalize the settings of a map, including data sources, symbolization (based on SLD), and extent [134]. It is similar to already existing but vendor-specific approaches of for instance ESRI (map/project file). The specification is supported by various tools available under proprietary or open source software licenses such as ESRI ArcGIS, uDig and MapBuilder.

An example of a WMC document is given in Appendix A.4.

2.2.4 Geoprocessing Services

Web-based geoprocessing has been established in the 1990s, since sufficient network bandwidth and processing capabilities are available. Examples for performing geoprocesses remotely are the GRASS software package [73], the upcoming of remote processing technologies such as CORBA [6] or documented research such as Vckovski [188]. Geoprocesses in general transform geographic data into geographic information. Enabling web-based geoprocessing is the next logical step [104], as geographic data have become largely available through GIIs [74]. Moreover, extracting geographic information from web-based geographic data is an important issue for applications in which decision makers have to integrate multiple sources to answer questions regarding a geographic context. Web-based geoprocessing is also promising to establish distributed platforms for large scale computational calculations such as complex simulation models. In particular, interoperable Geoprocessing Services can be chained to create value-added chains [5].

In the context of web-based geoprocessing, Geoprocessing Services attracted attention by research and industry bodies [41]. Geoprocessing Services have been identified as the means for creating web-based geographic information from available geographic data. The first official release of the Web Processing Service (WPS) interface specification in 2007 also indicates that Geoprocessing Services have become an integrative component of standardized OGC Web Services [139].

Examples of Geoprocessing Services address geostatistics [100], geographic data analysis for specific applications such as bomb threat analysis [173] or water resource management [32].

Because several research projects, different vendors and open source projects address Geoprocessing Services, Brauner et al. [19] conducted a qualitative analysis of these conducted projects. They identified bottlenecks, future challenges and future applications. The current bottlenecks for Geoprocessing Services are related to one of the three research topics:

- Service orchestration
- Semantic descriptions
- Strategies to improve performance.

In the context of web-based generalization processing and with respect to the limitations described in Section 4.3.3, semantic descriptions are crucial for semantic interoperability.

The following two sections will introduce the WPS interface specification, as is the OGC standard for Geoprocessing Services. Additionally, the current approach of enabling semantic interoperability of Geoprocessing Services is introduced.

OGC Web Processing Service

One of the most recent developments regarding “Geospatial Web Services” is the exposition of desktop-based GIS analysis processes through a Web Service interface. This shift is possible due to the increasing availability of network capacity and processing power. But it is also motivated by the increased availability of geographic data served through Web Services. Thus, there is a demand to transform web-based geographic data into web-based geographic information by the means of web-based geoprocesses. Web-based geographic information is promising as it supports decision making on the most up-to-date resources in a distributed environment.

The OGC specified the OGC Web Processing Service interface, which describes a standardized method to publish and execute web-based processes for any type of geoprocess. According to the WPS interface specification, a geoprocess is defined as any calculation operating on geographic data.

In detail, the WPS interface specification describes three operations, which are all handled in a stateless manner: GetCapabilities, DescribeProcess and Execute. GetCapabilities is common to any type of OGC Web Service (Section 2.2.2) and returns service metadata. In case of WPS it also returns a brief description of the processes offered by the specific service instance. To get more information about the hosted processes, the WPS provides process metadata through the DescribeProcess operation. This operation describes all input and output parameters, which are supported by the process. Based on this information the client can perform the Execute operation

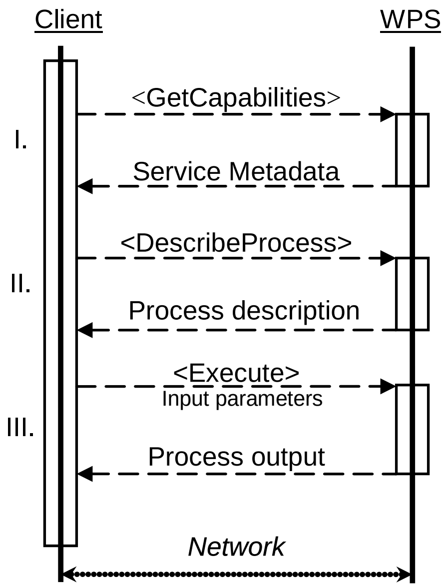


Figure 2.5: Basic WPS communication.

with the specific input parameters upon the designated process. This course of action is depicted in Figure 2.5. Additionally, it is illustrated by a sequence of exemplary requests in Appendix A.1.

Besides this basic communication pattern, the WPS interface provides functionality for scalable processing such as asynchronous processing (implemented using the pull model), storing of process outputs and processing of process inputs referenced as URLs. The application of such URL references as input for specific processes limits the volume of data sent between client and service and allows the service to apply caching strategies. The service retrieves the data once and reuses them multiple times, by using the reference as an identifier for the data. In case of applying caching strategies, the WPS instance still appears stateless to the client, but stores retrieved process inputs and process outputs if possible. Such caching strategies are application specific and have to be chosen carefully [169].

Semantic Interoperability of Geoprocessing Services

The semantic interoperability of Geoprocessing Services is examined by the example of the WPS interface specification and its WPS Profiles. These WPS profiles have been proposed to address several comments about the WPS interface, as for instance by Neun [129]. The WPS interface has been considered by the reviewers being too general as processes can only be described on an abstract level and lacking semantic interoperability in particular. For example, the WPS interface specification does not predefine any specific process, which is mandatory for any WPS instance.

These WPS profiles allow the client to identify syntactically and semantically equal processes provided by WPS instances. WPS profiles are referenced by process descriptions and describe the input and output parameters of a process. The WPS profiles provide a common definition of geoprocesses and can be referenced by other WPS instances, which provide a similar geoprocess (syntactically and semantically). To give an example, let us assume there are two different implementations of a buffer algorithm and both published as WPS processes sharing the same interface (i.e. same input and output parameters). As both processes refer to the same WPS profile, they become interoperable (i.e. sharing the same interface) but also their functionality becomes comparable to the client. The client can select the appropriate WPS process based on the quality of the process output and the performance of the process. However, matching the semantics of the offered process with the WPS profile is the responsibility of the service provider. From a technical perspective, WPS profiles are WPS process descriptions (i.e. defining input and output parameters), which are web-accessible and are identified by an OGC Uniform Resource Name (URN).

Nash [127] specified an initial set of these WPS profiles describing the most common GIS operations. Ostländer [147] also used WPS profiles to describe WPS processes in a GII for spatial decision support.

The trade-off between simplicity and expressiveness and between accuracy and flexibility regarding the specification of Web Services remains, as pointed out by Vckovski [188]. Selecting the right level of abstraction for such a specification is a hard task and has to be verified by different applications. Also an agreed classification of functionality formalized as ontologies is required to enable semantic interoperability as pointed out in Section 2.2.1. An overview of projects and an agenda on web-based processing for geographic applications is presented in Brauner et al. [19].

2.3 Automated Generalization and the Web

Based on the advancements in Web Services technology and standardization with a special focus on geographic applications as explained in Section 2.2 also the demand to provide generalized data (or a representation of it) on the web increased. The provision of generalized data or its representation on the web can be realized in two ways by providing: 1) limited control of the generalized output (e.g. selection of layers, selection of symbolization) and 2) full control of the generalized output (e.g. selection of specific generalization parameters). The first option is applicable to scenarios which focus on non-expert users. The latter option is applicable to scenarios which focus on expert users or involve automated systems.

Generalization on the web is mainly achieved through Web Services. An exception for instance is the MapShaper client application of Harrower and Bloch [82] which is only usable through a specific browser-based client.

Section 2.3.1 describes the concept of Web Generalization Services, which is the foundation for this research to enable meaningful generalization processing on the web. As this research addresses on-demand maps on the web, Section 2.3.2 presents related concepts about on-demand web mapping in the context of automated generalization.

2.3.1 Web Generalization Service

The concept of a Web Generalization Service has been introduced by Edwardes et al. [38]. They described the desire of the research community to develop a common research platform by means of Web Services. This platform was intended to facilitate the reuse and exchange of generalization knowledge (i.e. generalization algorithms) within the generalization research community. Later on Sarjakoski et al. [165] and Edwardes et al. [39] extended this idea to provide generalization functionality on the Web, either as an atomic or a complex process or even as an all-encompassing generalization process. Web Generalization Services are not only applicable to share generalization functionality across organizational and geographic boundaries, but they are also applicable for distributing the complexity and process load of generalization.

Burghardt et al. [23] present an overview of the evolution of Web Generalization Services. Since then, specification programs of the OGC and the research community have drawn more attention on Web Generalization Services.

Edwardes et al. [39] introduced a classification of Web Generalization Services to improve interoperability of Web Generalization Services (Figure 2.6). Originally to each type of service a service interface or a Graphic User Interface (GUI) or both has been attached to indicate the possible interaction modes (i.e. computer-to-computer interaction or human-to-computer interaction). The *Generalization Support Service* is at the bottom level of the classification. It provides basic functionality for enriching the data with structures needed by the generalization process (such as triangulation). The *Generalization Operator Service* is at the next level of the classification and provides functionality on the level of a generalization operator (Section 2.1.4). The Generalization Operator Service and the Generalization Support Service are accessed by the *Compound Generalization Service* that drives the generalization process and automatically evaluates the results.

In the original hierarchy Edwardes et al. [39] suggested to attach only a GUI to the Compound Generalization Service and only a service interface to the Generalization Support Service. In the extension as proposed by Foerster et al. [60] and as a result of this research, both types of services have both types of interfaces attached. This is due to the fact that Compound Generalization Services might indeed be used in a automated fashion (through a service interface) and Generalization Support Services might be used in an interactive fashion (through a GUI) as well. This is an important extension to enable the full flexibility of Web Generalization Services. Figure 2.6 portrays the extended classification. The proposed extensions are depicted in gray.

For demonstrating syntactic interoperability of Web Generalization Services various

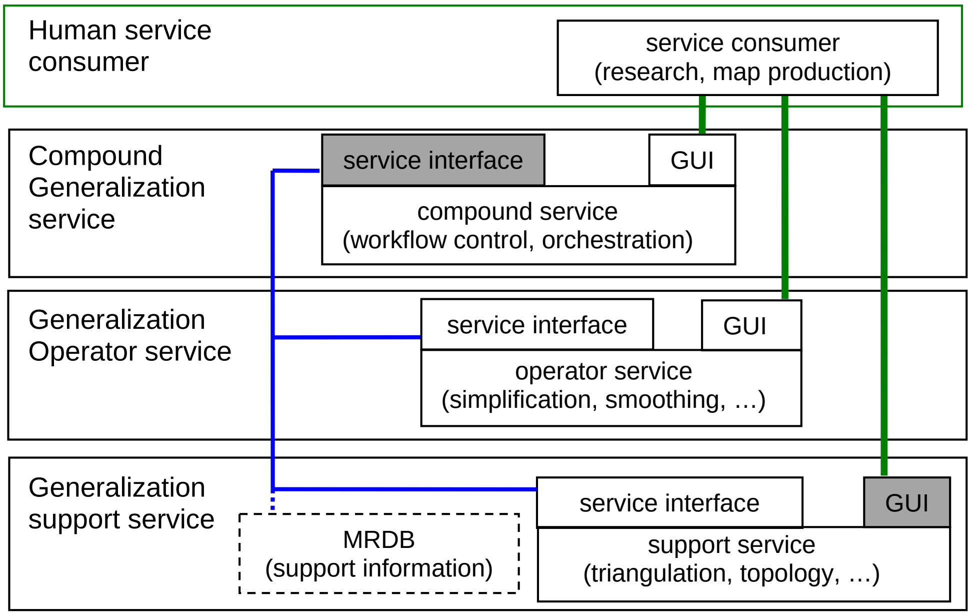


Figure 2.6: Web Generalization Service classification originally adopted from Edwardes et al. [39, Figure 2] and adjusted according to Foerster et al. [60]. The adjustments are highlighted with a gray background.

frameworks have been developed. Neun and Burghardt [130] proposed a Web Service-based framework called WebGen. It allows a client application to perform remote generalization processes through simple XML-messaging. WebGen has been developed in Java and features a central registry, for searching and accessing remote services. Foerster and Stoter [54] also implemented a Web Service-based framework for automated generalization, but based it on the WPS interface specification. They also implemented a client and server application and demonstrated thereby how generalization functionality can be published in an OGC-compliant way.

Web Generalization Services currently do not support semantic interoperability. They are still used predominantly for a single remotely performed operation configured by a human user and not as automatically chained operations to perform complex generalization involving multiple services. Burghardt and Neun [22] presented an example of automated service chaining in a pre-configured environment. Currently the configuration of the generalization processes always requires human reasoning, because the semantic aspects of the description of the generalization algorithms are only available as human-readable text.

Besides the development of Web Generalization Services, which focused on sharing generalization functionality and distributing processing load, *progressive transfer* of vector data became an application and additional goal for automated generalization on the web [14, 142]. Progressive transfer of vector data is inspired by the principle of raster data transfer, which transfers at first less detailed data and more details during further transfer. Thus data, which incorporate the most relevant information, will be transfered first and the less relevant information will be transfered afterwards. Progressive transfer is realized by generalization operators and data structures, which categorize the relevance of the data and take care of the topology of the data. It has been investigated to enable smooth zooming in web-based applications and to address limited bandwidth issues for mobile applications. To enable progressive transfer for Web Services, Vries and Oosterom [190] proposed an extension of the WFS (called WFS-R, R stands for refinement), which allows clients to query objects regarding their specific level of importance through a Web Service interface.

Web Processing Service Interface for Generalization

As pointed out in the previous section, various frameworks for Web Generalization Service have been developed. To enable the interoperability between these frameworks and to be accessible for other clients outside the generalization community, a working group of the ICA Commission on Generalisation and Multiple Representation was recently set up, consisting of research institutes, NMAs and software vendors, to specify a Web Generalization Service based on the WPS interface specification. The working group decided to adopt the WPS interface specification as a basis for a standard-compliant Web Generalization Service as this has been considered to be the state-of-the-art approach for establishing geoprocesses on the web such as generalization. As mentioned earlier, also Foerster and Stoter [54] demonstrated some results to publish generalization functionality through the WPS interface.

The working group formed a technical task force to define a technical specification of a WPS-compliant Web Generalization Service, which meets the special requirements of generalization processing [47]. Expertise acquired during this research also contributed to this task force. The task force specified the following extensions:

- Registry for WPS
- Standardized data model for Web Generalization Services
- Extension of the DescribeProcess response documents of WPS.

The extensions have been implemented as the WebGen 2.0 framework, consisting of a WPS-compliant Web Generalization Service and a WPS-compliant client application, which is able to interoperate with the registry for WPS instances. The implementation of the WebGen 2.0 framework has been financially supported by the Ordnance Survey, Great Britain and is currently used as a demonstrator for Web Generalization Services by the ICA Commission on Generalisation and Multiple Representation.

Client Applications for Web-based Generalization

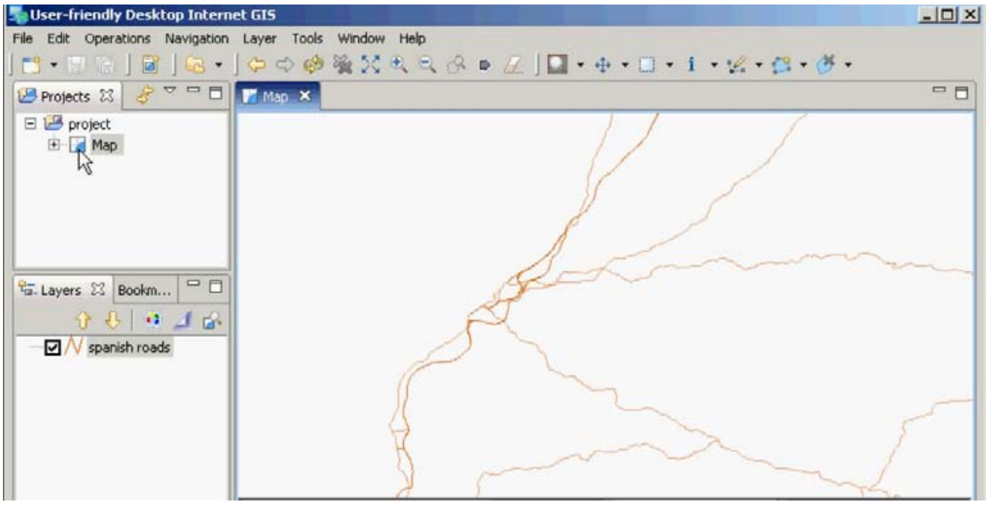
Besides the work about Web Generalization Services as presented in this section and web-based generalization as for instance presented by Harrower and Bloch [82], a client application for web-based generalization has not been proposed yet that incorporates different OGC Web Services. Such a client application being fully web-enabled and supporting the access of Web Services is promising to take full advantage of real-time generalization.

As part of this research, Foerster and Schäffer [50] presented a client plug-in for uDig to show the applicability of web-based generalization processing with a focus on real-time and distributed data access. It is the first client supporting the configuration of WPS-based processes with referenced WFS data by the user. The enhanced uDig client is thereby able to trigger any WPS-based process on the web, by sending WFS-based data via reference to the actual WPS instance. This limits the data transfer between client and server for improving the performance of the client application. The client plug-in has been applied to a real-time processing scenario for risk assessment, which involved real-time data and processes (Figure 2.7 and Figure 2.8). In particular, it applied the Douglas-Peucker algorithm to simplify a road network and to improve visual risk analysis. The original and the generalized data as displayed in the client application are depiced in Figure 2.7a and Figure 2.7b respectively. A screenshot of the final map, depicting the area of Northern Spain with buffered forest fire areas and a simplified road network is presented in Figure 2.8.

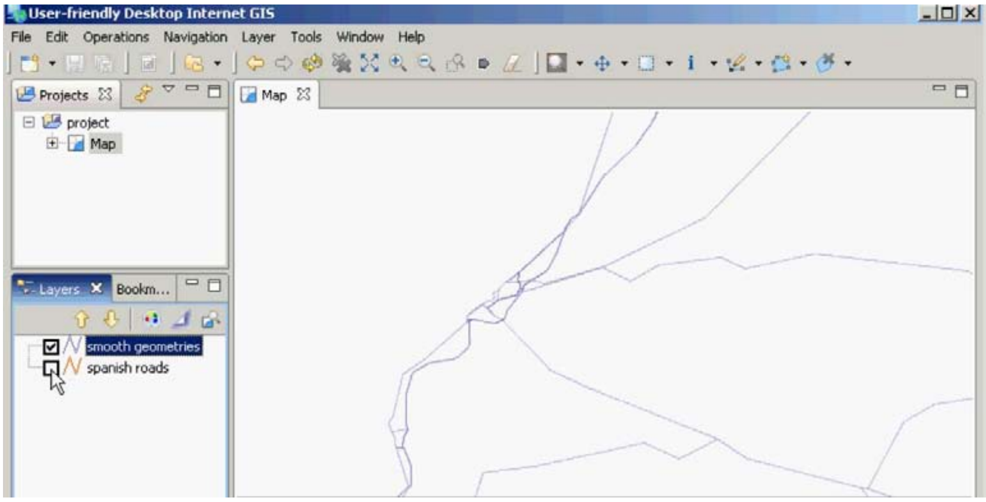
Also as part of this research, Foerster et al. [51] enabled Google Earth™to access WPS. In particular, uDig was enhanced to export KML files, describing pre-configured processes. These pre-configured processes (i.e. HTTP-GET ExecuteRequest to WPS-based process) were included in the `NetworkLink` tag of the KML file. Accessing the KML file through Google Earth allows the user to retrieve real-time information generated by WPS instances. Integrating WPS-based processes into mass-market applications such as Google Earth, enables ordinary users to access real-time information. This is especially interesting for risk management scenarios, in which ordinary users need to access the latest information to decide appropriately.

2.3.2 On-demand Web Mapping

As already defined, the approach of generating and disseminating maps on the web is called web mapping [107]. On-demand in this context means, that the user receives the map according to his/her requirements. This includes individualized symbolization, scale, level of detail and selection of layers. On-demand maps are a contrary concept to the concept of multi-purpose maps such as the classic topographic maps at a fixed scale and with a fixed set of object classes. The need for on-demand maps increases, as the Web is used to access information individually. An example of an increasing need for on-demand maps is given by Neogeography, which defines a new way of sharing user-generated content between ordinary users through easy-to-use



(a) Original road data.



(b) Generalized road data.

Figure 2.7: Screenshots of the uDig client portraying the original data (a), the generalized data (b) used for the risk managment scenario.

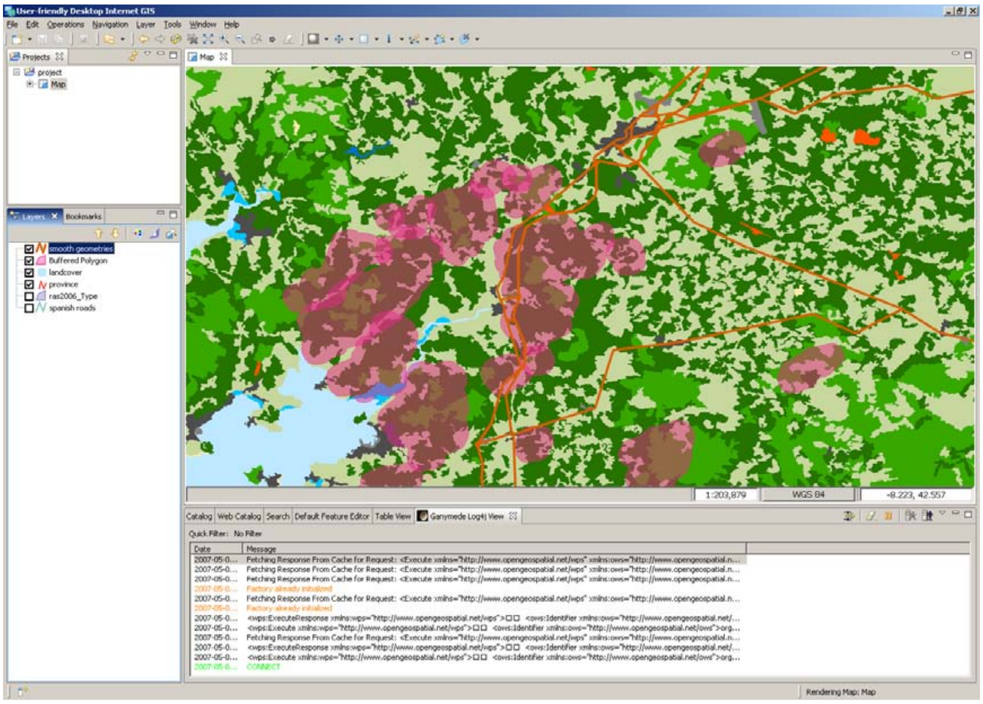


Figure 2.8: The final map for the risk management scenario including the generalized road data (Figure 2.7b and additional CORINE data served from different sources (WPS and WMS) [50].

browser-based applications [185]. Ordinary users need to view their data (such as geotagged photos or geotagged locations) on a map combined with other data on-demand. This involves individual symbolization as well as individual selection of layers as described previously. This on-demand character is technically supported by the WMS specification as well as by so-called *Mash-ups*.

As pointed out by Sabo et al. [162], the concept of on-demand information is currently closely linked to the concept of generating information on-the-fly (also called real-time). This is possible based on web technology and the availablity of up-to-date information. On-the-fly thereby describes the process of immediately generating information, which does not have to be on-demand (i.e. a multi-purpose batch process, which is triggered whenever any user requests the specific information). In web mapping applications it is the overall goal to generate maps on-the-fly and on-demand.

The foundation for web mapping is the WMS interface specification. Formulating user demands in the context of the web is achieved by standards. In particular, the WMC documents describe the geographic data which are displayed on the map and the SLD documents describe the representation of the geographic data.

To meet the demand for readable maps on the web, generalizing maps on-demand gained considerable attention in research. Cecconi [26] for instance proposed a derivation-oriented view for on-demand web mapping by introducing intermediate scales, which are based on traditional fixed map scales and need only small adjustments to be suitable for the final target scale. He claims that this approach would significantly increase the performance of on-demand web mapping. Sabo et al. [162] also investigated on-demand web mapping and proposed Self-Generalizing Objects. These objects are based on a pre-process data enrichment to capture relevant patterns. Based on the enriched database it is possible to perform on-the-fly generalization. Additionally, other projects have been carried out to develop suitable web-based architectures for automated generalization such as the WebPark project [20] and the GiMoDig project [164].

2.4 Modeling and Data Standards

To support the design of the architecture in Chapter 3, modeling is essential. This section presents relevant concepts and standards. Modeling in general is an integral part of the software development process [99]. Modeling is a key aspect for developing comprehensive databases such as for geographic applications. It captures the aspects of the problem domain by abstraction and is the foundation of the implementation. A popular approach to generate those models, is object-oriented modeling [17]. It describes the problem domain by a set of objects modeled through classes. The objects communicate with each other through messages, which invoke specific methods defined by partiular classes. The four elements of object oriented modeling are [17]:

- Abstraction - This addresses the essential characteristics of an application and results in an object model with clearly-defined conceptual boundaries.
- Encapsulation - This hides the implementation of details and is also referred to as the black-box paradigm.
- Modularity - This decomposes the model into a set of cohesive and loosely coupled units.
- Hierarchy - This structures the abstractions using part-of hierarchies (aggregation) and is-a hierarchies (inheritance).

To describe the models in more detail for seamless implementation, the Model Driven Architecture (MDA) has been developed. The idea is to generate runnable program code of the specified models automatically. The MDA thereby closes the existing gap between the specified model and the actual code. According to Warmer and Kleppe [194] the MDA has the following benefits:

- Portability - This increases application re-use and reduces the cost and complexity of application development.

- Productivity - This improves the performance of the development team and reduces development costs.

- Cross-platform interoperability - This makes the model applicable to any platform required by the application.

- Easier maintenance and documentation - MDA is realized in a platform independent way and thereby the effort of writing platform specific code is limited.

Overall, these benefits can be summarized as increasing the sustainability of the models and increasing the efficiency of the software development process by reducing the amount of manually produced code. Although the MDA provides some concepts for designing models which are ready for implementation, the gap still exists and the transformations from the models to the implementation platform remain incomplete.

The following sections describe aspects of modeling, which are applied in this research. The Unified Modeling Language (Section 2.4.1) is identified to describe the structure of the design and the implemenation as presented in Chapter 3 and Chapter 4. OCL is introduced (Section 2.4.2), as it is applied by this research to formalize the proposed classification of generalization operators (Section 5.2). To exchange these formalized models and to exchange information about the formalized generalization operators in this research, the XML Metadata Interchange format is presented (Section 2.4.3). Section 2.4.4 and Section 2.4.5 describe the standardized data models of ISO and OGC, which provide the basis for formalizing the generalization operators.

2.4.1 Unified Modeling Language

To support the software development process and especially the development of data models, the Object Management Group (OMG) [140] specified the Unified Modeling Language (UML), which describes the structure and lifecycle of data models. UML provides a set of standardized tools to specify static and dynamic aspects of the models and supports thereby the software development process in a unified way. The most relevant tool for specifying data models is the class diagram. Class diagrams depict the structure of classes through inheritance (is-a relation) and aggregation (part-of relation). The class diagram thereby captures the structural information of the data model.

An instance of a class diagram is depicted in Figure 2.9. It illustrates a basic relation of classesx (inheritance and aggregation) forming a basic geometry model as defined in ISO 19125 [94]. Geometry is the parent class with the attached method `dimension()`, which returns the dimension of the specific geometry object. Point and LineString are modeled as subclasses of Geometry and thereby inherit from Geometry class. They also inherit the defined methods of Geometry and define their specific methods (`x()`, `y()` and texttttnumPoints()). The LineString class is defined as an aggregate of two or more objects of class Point.

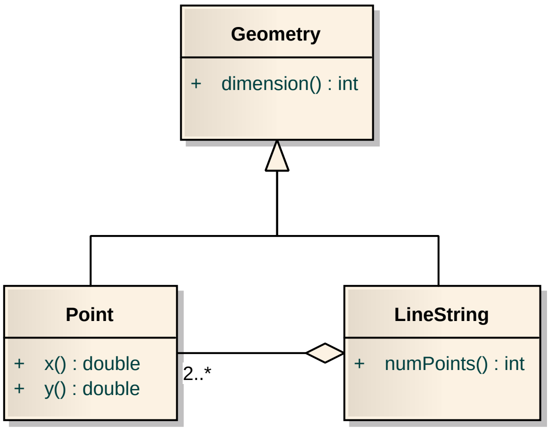


Figure 2.9: Example of a class diagram illustrating a basic geometry model according to ISO 19125 [94].

UML is currently widely used to model and document data models and software, as for instance illustrated by a survey of Dobing and Parsons [33]. Moreover, UML is used for describing standards such as those of the ISO 191XX family. The UML models presented in this research are created with Enterprise Architect of Sparx Systems [181].

2.4.2 Object Constraint Language

The Object Constraint Language (OCL) has been designed by the OMG to add semantic concepts to UML. The final aim was to enhance the MDA and to enable a seamless integration of models and software code (i.e. software code is generated through models). Thus OCL is a crucial part of UML to make the notion of MDA work and to fully benefit from it. Additionally, OCL can be used to check and validate existing models and thereby to verify the implementation [12]. It is a declarative and strongly typed language to query and describe additional semantics of object-oriented models. It can be attached to different types of UML diagrams such as class diagrams and state diagrams. It allows one to specify invariant states of objects and attributes (keyword: `inv`). Methods can be defined via pre and post conditions (keyword: `pre` and `post`)or via a body expression. Additionally, OCL defines a comprehensive mechanism of object querying, which is basically realized by different set operations (indicated with $\rightarrow$). Examples of OCL expressions are given in Section 5.2.2, Section 5.2.3 and Section 5.3.1.

OCL has become a part of UML and gained already some interest in industry as well as research [64]. Pinet et al. [149], Werder [198] described extensions of OCL for geographic applications. Pinet et al. [149] described the so-called *SpatialOCL*, which defines topological operations for checking data consistency and has been prototypically implemented for SQL. Werder [198] defined the so-called *GeoOCL* for expressing optimization goals, but did not provide any proof-of-concept. However, the support of UML in current software modeling tools is not strongly supported, as demonstrated by the work of Hespanha et al. [84]. They developed a OCL plug-in for Enterprise Architect to support data modeling in the land administration domain.

There are several tools available for creating and validating OCL descriptions, such as the open source modeling tool ArgoUML [7] and the Dresden OCL2 tool for eclipse [29].

2.4.3 XML Metadata Interchange

To exchange the data models, developed in UML between different applications in a distributed environment, the OMG specified the XML Metadata Interchange format (XMI). XMI is based on XML and has been adopted by ISO [92].

2.4.4 General Feature Model

The General Feature Model is part of the ISO 19109 model [93]. It defines an object-oriented model for geographic information using UML. It is important to note, that UML is only an example of modeling geographic information. The General Feature Model could be defined in any conceptual schema language such as EXPRESS [90].

The main building block of the General Feature Model is the feature type, which consists of feature attributes, feature associations and feature behaviour. The General Feature Model can thereby be used to express feature types or behaviour on feature types.

2.4.5 GO-1 Application Objects

The GO-1 Application Objects [133] provides a light-weight and implementation-independent model for describing, managing, rendering and manipulating geographic objects. One of the aspects of the model are the Graphical Data Objects, which provide graphical primitives and styling options. The Graphical Data Objects thereby can be applied to represent cartographic features.

The object-oriented model of the GO-1 Application Objects is described in UML and an implementation specification for Java is available.

2.5 Data Standards for User Profiles

User profiles have received a lot of attention in research and development in the wider community of web-based information systems, mobile telecommunication, and other modern IT applications in which different types of human users are involved. There are specific conferences and journals devoted to the topic of user (profile) modeling; for example the annual conference series *User Modeling, Adaptation, and Personalization* (UMAP) and the journal *User Modeling and User-Adapted Interaction* (UMUAI). The aim of user (profile) modeling is to develop interactive computer systems that can be adapted or adapt themselves to their current users, and on the role of user models in the adaptation process. One example of a generic User Modeling System (UMS) is described by Fink and Kobsa [46]. It consists of a dictionary component which includes a user model (users' interests and preferences), a usage model (persistent storage of interface events), a system model (application domain taxonomy) and a service model. The UMS they presented also includes a user learning component to support the acquisition and maintenance of user interests and preferences from usage data, and updates individual user models (including using predictions for missing values in individual user models from models of similar users and applying domain inferences). The UMS architecture supports external clients for both providing information about the user to the UMS, and also retrieving current information about the user from the UMS via an access control system.

Also the *European Telecommunications Standards Institute* (ETSI) performs research for user profiles [43]. The ETSI defines a user profile as *the total set of user-related information, preferences, rules and settings, which affects the way in which a user experiences terminals, devices and services* [42]. The user profile enables the customization of an interface (in combination with the service behind the interface). The ETSI Specialist Task Force 342 organized a workshop at the start of 2009 with the name *Personalization and User Profile Management Standardization*. Its goal was to produce two documents (standards): User Profile Preferences and Information [45] and Architectural Framework [44]. The standardization of user profiles is important, because different clients and services should work well together to implement user-driven applications. Traditionally, the preferences that can be set by users are not consistent between different services. More specifically, the user profile is organized as shown in Figure 2.10: personal information (about or related to the user), human centered preferences (overall preferences that might apply across the user's usage of a wide variety of services), service category related information and preferences (related to service categories and specific services), and device related information and preferences (related to device categories and specific devices). These four issues are adopted in this research as possible types of user profiles and will be referred to as ETSI categories.

One important requirement of the Architectural Framework of the ETSI was to support personalization and profile management. User profiles can contain a large number of settings and preferences. When users first create profile specifications, the creation task can be greatly simplified if the profile specifications are created from templates

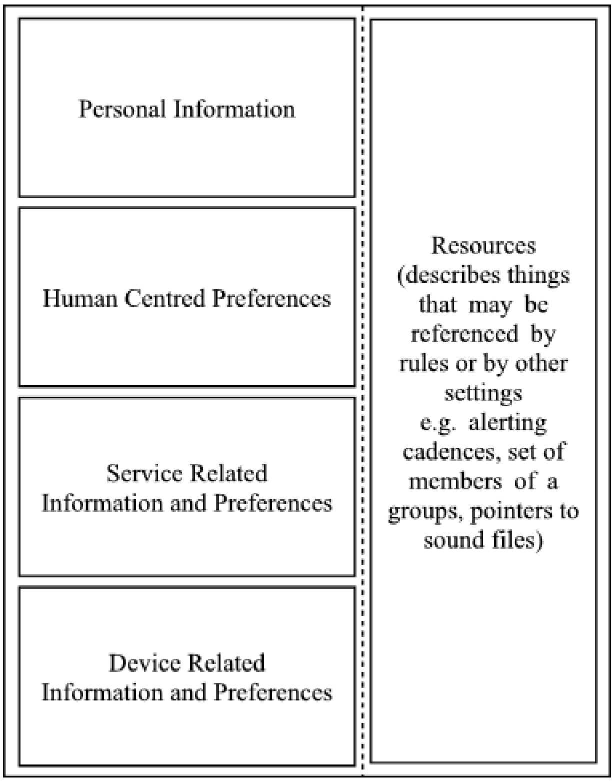


Figure 2.10: General structure of a user profile as defined by the ETSI [45].

(inherited from other existing profiles), which can be then further amended by the user to suit their individual needs. A number of operations support the maintenance of profiles; e.g. create new, modify, copy & paste and delete. The profile data may be manipulated by different actors such as service providers and end users.

2.6 The DURP Ondergronden Use Case

Currently a shift towards e-government is performed in modern society, such as in the Netherlands. Therefore the Dutch ministry for physical planning and environment (VROM) launched a project to develop an *Informatiemodel Ruimtelijke Ordening* (IMRO)[2] for making the content more available over the web [148]. IMRO has been developed by the project for Digitale Uitwisseling in Ruimtelijke Processen (DURP) and is maintained as a standard by Geonovum [68].

[2]Informatiemodel Ruimtelijke Ordening is a Dutch term and stands for Information model physical planning.

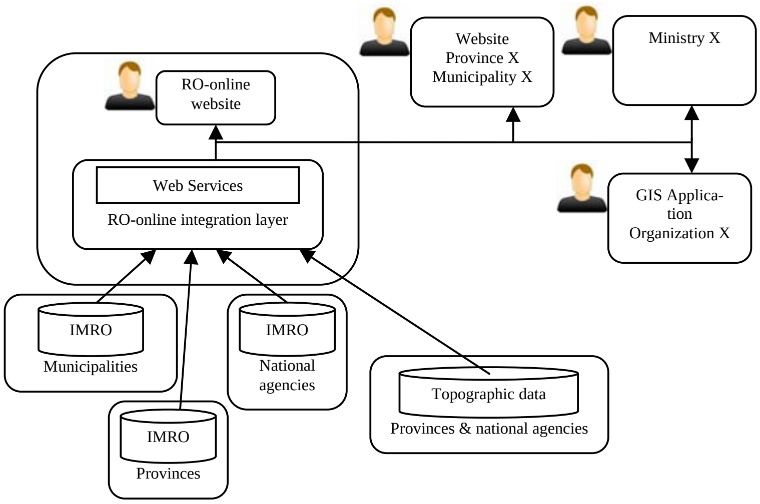


Figure 2.11: The Web Service-based architecture for RO-Online.

Based on the DURP project, the Ruimtelijke Ordening[3] online project (RO-Online) [192] develops a web portal to disseminate the physical plan information based on the IMRO standard. The RO-Online portal provides instant web-based access to physical planning maps, as required by law. These maps are compiled of the digital plan data (compliant to the IMRO model) plus available topographic data. The portal is realized by an interoperable Web Service architecture for the dissemination of the maps (Figure 2.11). RO-Online can be considered as a subset of a GII for physical planning, because of its technological framework, organizational structure and legal commitment [74]. It has been developed concurrently with and independently from the DURP ondergronden project [37].

The base map that is currently incorporated for physical plans in RO-Online stems from a topographic dataset at a fixed scale (for example TOP10NL or TOP50NL) depending on the scale of the plan (municipal, provincial or national government level). The base map might not be optimally applicable to all user groups, as well as to new uses such as collaborative mapping which are possible by the new environment compared to the paper plans. For example, different physical plans can be combined in the portal and they can also be accessed in an interactive way (i.e. zooming). Zooming out for instance might result in a dense topographic base map possibly causing that the thematic content of the map is not interpreted by the user correctly (Figure 2.12). At least, the usability of such a map could be increased if the base map is adjusted according to the zoom scale. The focus of the DURP ondergronden project is to generate base maps on the web by the means of automated generalization based on a technology and usability research (Section 1.2) [150].

[3]Ruimtelijke Ordening is a Dutch term and stands for physical planning.

Figure 2.12: A Dutch municipal physical plan with static topographic data (GBKN) as currently disseminated in RO-Online (map scale approx. 1:5000). Base map appears too detailed, buildings might be aggregated to building blocks to improve the readability of the map.

The DURP ondergronden project is sponsored by the *Ruimte for Geo-Informatie*[4] framework program (RGI) [156]. The DURP ondergronden project is also referenced as RGI-002.

In the following sections, the Dutch data are presented, which are applied in this research. The topographic data for the base map are described in Section 2.6.1. The physical planning data are presented in Section 2.6.2, for which the base maps are generated.

2.6.1 Dutch Topographic Data

Topographic data are captured and maintained at different scales in the Netherlands. An overview of the available topographic data models is depicted in Figure 2.13. The recently published topographic data models are object-oriented and replace the former models, which are geometry-based. The following sections will describe the different data models, which are relevant for this research.

NEN3610 is the base model geographic information in the Netherlands describing any geographic related aspect from topography, to transport (e.g. railway) and physical planning on an abstract level [128]. Specific domains define application models based

[4]Ruimte for Geo-Informatie is a Dutch term and stands for space for geographic information.

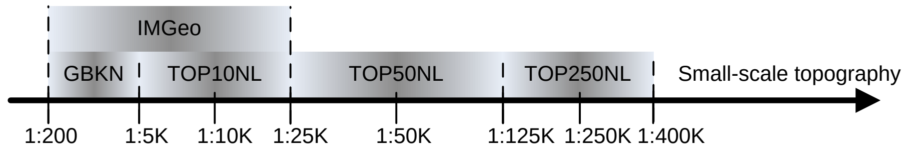


Figure 2.13: Overview of available topographic data in the Netherlands

on NEN3610 such as the IMRO model for physical planning. The TOP10NL and IMGeo define an application model of NEN3610 for topography.

Currently, a data model for multi-scale topography in the Netherlands is under development, which facilitates the semantics and relations between object at multiple scales [178].

TOP10NL

TOP10NL is the Dutch topographic database for a scale range between 1:5000 and 1:25000. TOP10NL is object-oriented and consists for instance of buildings, railways, roads, relief and areas of interest. TOP10NL is, among other formats, also available in GML format [9].

IMGeo and GBKN

Currently, the Informatie Model Geografie[5] (IMGeo) is established as a object-oriented model to exchange large-scale topographic data in the Dutch GII [67]. IMGeo will be populated by the object-oriented version of the Grootschalige Basiskaart van Nederland[6] (GBKN), which is the large scale topographic database consisting of buildings, roads and other objects. Currently, IMGeo-based GBKN data are only available for limited areas. The existing GBKN data are based on a geometry-based model [65]. GBKN is applicable at a scale range from 1:500 to 1:5000. Overall, IMGeo focuses more on management applications, which require more detailed information, whereas TOP10NL addresses for mapping applications, as explained by Stoter et al. [179].

2.6.2 Dutch Plan Data

The recently published Dutch Spatial Planning Act foresees two different kinds of plans [191]:

- Plans defining legal commitment (Bestemmingsplannen)

[5]Informatie Model Geografie is a Dutch term and stands for information model geography.
[6]Grootschalige Basiskaart van Nederland is a Dutch term and stands for the large scale base map of the Netherlands.

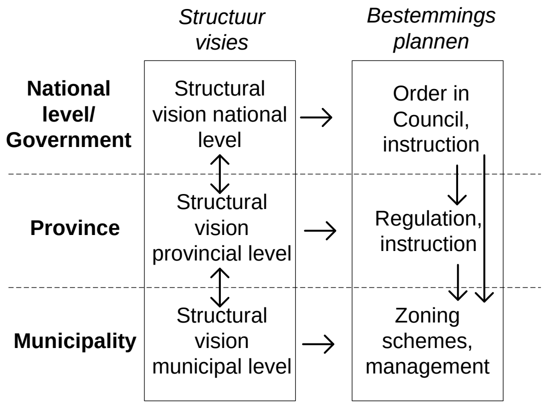


Figure 2.14: Types of physical plans and their functions based on VROM [191].

- Informal plans, defining legal commitment on a higher level of abstraction (Structuurvisies).

Both types of physical plans are developed on municipal, provincial and national level. As depicted in Figure 2.14, Bestemmingsplannen (i.e. physical plans with legal commitment) have legal implications from the national level downwards. The Structurvisies (i.e. physical plans with informal commitment) have legal functions but on a higher level of abstraction. Both types of plans are available through the IMRO data model. The *Standard Vergelijkbare BestemmingsPlannen* (SVBP[7]) defines a unified symbolization for Bestemmingsplannen [69].

2.7 Synopsis of the Research Context

This chapter presents the context and the concepts as they are applied in the thesis. The review of the concepts for generalization has several implications for this research. Cartographic generalization is identified as the key approach as it addresses the generation of on-demand base maps. In this respect "constraint-based generalization" and agent-based generalization are applicable to formulate the generalization requirements and to implement the process of automated generalization respectively. Especially the AGENT model is applicable for implementing the generalization of the base map according to the specific thematic content. Using the thematic content as an input for the generalization of the base map, as applied in this research, is a new aspect in generalization research. The generalization of the base map is implemented based on 1Spatial Clarity, as it is the most advanced implementation of agent-based generalization. The analysis of current challenges in practice by the example of NMAs

[7]Standard Vergelijkbare BestemmingsPlannen is a Dutch term and stands for unified comparable physical plans.

has demonstrated, that cartographic generalization is still a major issue and remains unsolved. Reasons for this are manifold, such as complexity of the generalization process or lack of formalization.

Different initiatives addressed a classification of generalization operators and considered further research being essential for generalization theory. None of these classifications provide a formal description of generalization operators nor do they provide a holistic classification covering all the aspects of generalization (i.e. for cartographic generalization and model generalization). Moreover, it might be the case, that such a holistic classification is not possible. However, it might already be sufficient to provide a classification, which is based on formal models and comprehensible by users.

The review on Web Services and Web Generalization Services demonstrated, that syntactic interoperability is established successfully by several standards for geographic data retrieval, portrayal and processing functionality. Especially, as demonstrated by different implementations [50, 54], carried out as part of this research, syntactic interoperability of geoprocesses on the web and Web Generalization Services in particular can be achieved through WPS. The applicability of WPS interface for Web Generalization Services has also been acknowledged by the ICA Commission on Generalisation and Multiple Representation [47]. However, semantic interoperability of Web Generalization Services is still an unsolved issue and is therefore addressed as part of this research in Chapter 5.

The use case of the DURP ondergronden project is an example for providing on-demand base maps for thematic content on the web. Other applications may be related to soil mapping as an instance of traditional thematic mapping or related to the new field of Neogeography [185]. Finally, using Web Services (such as WMS) and standards (such as IMRO and IMGeo) is essential to integrate the developed architecture into RO-Online.

Chapter 3

Design of a Web-based Architecture for On-demand Base Maps

This chapter presents the design of the architecture for generating and disseminating on-demand base maps on the web. The design of the architecture will be the foundation for the implementation and evaluation in Chapter 4.

In particular, the chapter examines the requirements in Section 3.1, which are defined by the DURP ondergronden use case and the application of on-demand base maps on the web, as described in Chapter 2. Based on the architectural requirements and the user requirements (Section 3.1) the model of the user profiles is described in Section 3.2. Finally, the design of the architecture is presented, which integrates these user profiles to disseminate on-demand base maps (Section 3.3).

3.1 Requirements Analysis

The requirements analysis is based on the use case of the DURP ondergronden project as described in Section 2.6. The requirements aim at an architecture for generating base maps and thereby supporting and improving the communication of the specific thematic content on the web. Additionally, the requirements address a sustainable and extensible architecture, which is applicable to other use cases of on-demand base maps for thematic mapping, such as soil maps. This involves the correct selection of base map classes and the selection of the applicable level of detail of the base map objects. The following sections will examine these requirements in more detail.

To present a valid design and to also evaluate the developed architecture, a thorough analysis of the requirements is essential. The requirements act as criteria for evaluating the architecture in Section 4.3. For this research, the requirements are twofold: requirements addressing the architecture (Section 3.1.1) and the user requirements towards the base map (Section 3.1.2). The user requirements are based on the example of physical planning maps as defined by the DURP ondergronden project. These user requirements are then illustrated for two exemplary users (Section 3.1.3) and are used to create exemplary map samples (Section 4.3.1), generated by the developed architecture.

3.1.1 Architectural Requirements

Regarding the application of on-demand base maps the architecture has to meet the following requirements:

- Perform cartographic generalization
- Consume user profiles
- Integrate Web Services.

These architectural requirements are explained below in more detail.

As the focus of this research is to disseminate on-demand base maps generated by automated generalization, the architecture needs to incorporate a generalization system for cartographic generalization. This will be the key aspect of the architecture. The generalization system for cartographic generalization has to be aware of the symbolization of the base map and the thematic map to perform cartographic generalization sufficiently. The symbolization allows the system to create cartographic features and to evaluate the cartographic generalization (i.e. requirements for symbolized features). Based on the requirement for cartographic generalization the architecture separates symbolization and generalization as also explained in Section 2.1.1 and Figure 2.2.

Additionally, the architecture has to serve different users and user groups at the same time in a flexible way on-demand. Flexible in this context means that the architecture needs not to be pre-configured for specific users. In fact, the architecture has to provide a means to formulate the user requirements in a machine-understandable format. The architecture has to be able to consume these formalized requirements and has to generate the map accordingly (i.e. on-demand). These formalized requirements are described in so-called user profiles and demonstrate the on-demand character of the architecture. Consequently, the user profiles drive the generalization process of the base map. User profiles also enable the flexibility and sustainability of the architecture, as new user groups can be easily exposed to the architecture and the architecture does not need to be changed internally.

To incorporate other types of thematic content, which might be required for other use cases, such as soil mapping, the architecture applies Web Services to access these geographic data. Web Services enable to integrate data ad-hoc into the architecture without changing its internal configuration. They thereby improve the flexibility and sustainability of the architecture. Finally, Web Services will enable to integrate the proposed architecture into the architecture of RO-Online (Section 2.6; Figure 2.11).

To enhance the functionality of the incorporated generalization system, Web Generalization Services need to be integrated into the architecture for improving the generation of the base map regarding quality and performance (Section 2.3.1).

3.1.2 User Requirements for Base Maps for Physical Planning

The aim of the DURP ondergronden project is to disseminate on-demand base maps, generated for a specific user or user group. The base map thereby has to fit the thematic content to support the communication of the thematic map. The user requirements define which contextual information is necessary to optimize or at least improve the communication of the thematic content. The user requirements are the input for the generalization process. Many potential requirements can drive such a generalization process. It is important to note, that this research will not provide a complete list of these potential user requirements, but examine the architecture based on two fundamental user requirements. They provide a reasonable test case for the concept of on-demand base maps (i.e. user profiles, Section 3.2) and for the presented architecture (Section 3.3). A more complete list of user requirements will be compiled by usability research, which is connected to the project related to this research [40] (Section 1.2 and Section 2.6).

The architecture and the identified user requirements aim at providing a readable map by adjusting the base map content and not the thematic content. Generalizing the thematic content is outside the scope of this research, but might be interesting for future research especially for the case of generalizing physical plans. Automatically generalized physical plans might improve the consistency of physical plans, which are currently created independently from each other at different levels (Section 2.6.2 and Figure 2.14).

The user requirements, as identified for this research, are the following:

- Preserve topological relationship between base map and thematic content
- Adjust the level of detail of the base map objects with respect to the type of thematic content the user is interested in.

These requirements are explained below in more detail.

The first requirement depends on the preference of the user, as (s)he may want to choose if the initial topological relationship between the base map and the thematic map should remain the same. Topological consistency between the base map and the thematic content helps the user to link the situation on the map to the real world situation and thereby improves map communication. According to the mental model of Kuipers [111], topological properties of maps can be considered as being more important in map communication than metric distances or geometric shapes. Automated generalization of the base map independent from the thematic map might harm this requirement by applying generalization operations such as aggregation,

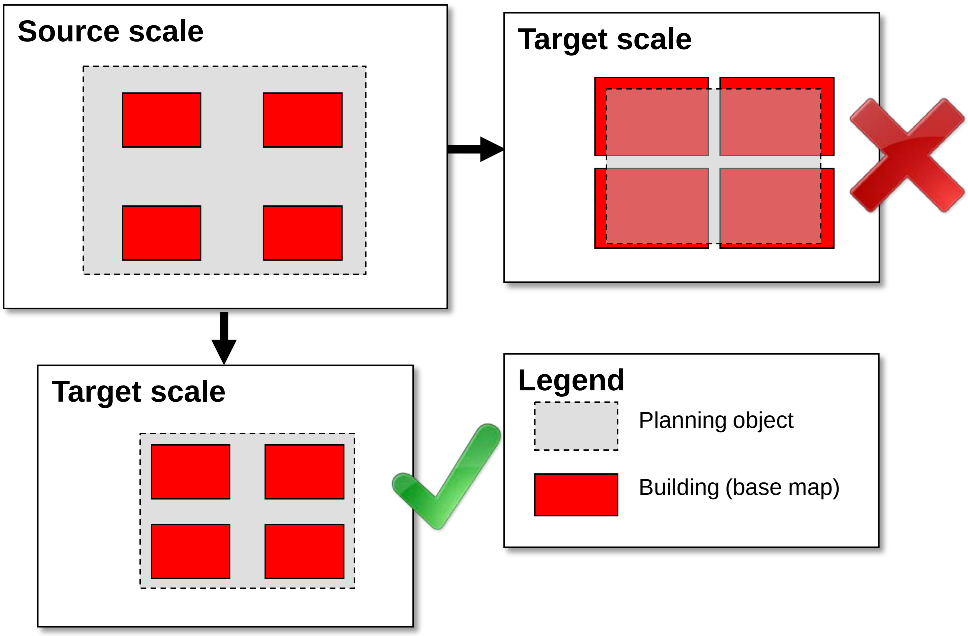


Figure 3.1: Example of the topological requirement - The base map object has to maintain the original topological relation with the thematic base map object, the planning object is graphically scaled.

enlargement or simplification. An example of such a map situation is schematized in Figure 3.1. The physical planning objects are graphically scaled in the target scale. Two different generalization solutions are exemplified, however due to the topological requirement of the user, the enlarging of the buildings is not applicable, as it would violate the initial topological relationship of the base map objects with the thematic content. In some cases the topological requirement may limit the possibility to create the optimal solution regarding other user requirements and additional optimization goals. This is for instance the case, if the thematic objects represent only informal (or even fuzzy) boundaries (such as an informal physical plan; e.g. Structuurvisie in Section 2.6.2, Figure 2.14).

Besides maintaining topology between base map and thematic map, the internal topology of the base map should be maintained during the generalization process to avoid misinterpretation of the map. However, this issue is outside the scope of this research and is for instance addressed by the research about the tGAP tree structure [141, 142].

The second requirement defined for the user profile in this research is that the user should receive a base map adjusted to his/her specific information needs regarding the thematic map. In particular, (s)he should receive a more detailed base map for the parts of the thematic content (s)he is interested in. This has two consequences:

1. Two different users, retrieving the same extent of the thematic content, will get two different base maps, due to their different interests (which imply different relevance of base map objects) in the thematic content.

2. Different objects of the same class may be displayed at a different level of detail (allowed amount of change/generalization): more detail is maintained below thematic classes that are of interest for the user. The assumption is that this higher level of detail better meets the information demand of the user. The user research related to this project has to test this hypothesis. The described concept is different, but related to the idea of the variable scale map [80]. A variable scale map displays the objects located in the center of the map at higher detail. The difference in our approach is that the level of detail is not determined by the location on the map (distance to map center), but is determined by other factors, such as the overlaying thematic object class.

The presented requirements will be modeled in the user profile and will determine the automated generalization process (Section 3.2). Apart from these requirements, the zoom level of the map display drives the generalization process.

3.1.3 Exemplary Users

In a workshop with planners organized as part of the DURP ondergronden project in 2008 an *investor* and an *ecologist* have been identified as potential users of the physical planning map. These potential users have different requirements towards the base map due to different information needs. The investor seeks for future commercial areas and the ecologist investigates nature-related areas on the physical plan to identify compensation areas. Consequently, their base maps will be different although they consult the same type of plan (Section 3.1.2). In particular, the investor will receive a base map, which supports the portrayal of commercial areas by a higher geometric level of detail regarding the base map objects. Contrarily, the ecologist will receive a base map, which supports the portrayal of nature areas by a higher geometric level of detail regarding the base map objects. Due to the legal nature of physical plans, the topological requirement will be the same and all base map objects have to maintain their original topological relation with the physical plan.

In the example maps depicted in Figure 3.2, the same physical plan is presented to the two potential users with different base maps. In fact, the base maps objects inside the specific type of plan area of interest (nature area vs. commercial area) are depicted with higher level of detail. These exemplary users are the use case for the architecture and will be applied to the map samples presented in the evaluation (Section 4.3.1).

Finally, designing different requirements for two different users, as explained in this section, is questionable, when the given users are communicating with each other. They receive different maps, which might cause some confusion and be a limitation at a certain stage of the physical planning process. Designing different base maps for different users is a requirement stemming from the DURP ondergronden project

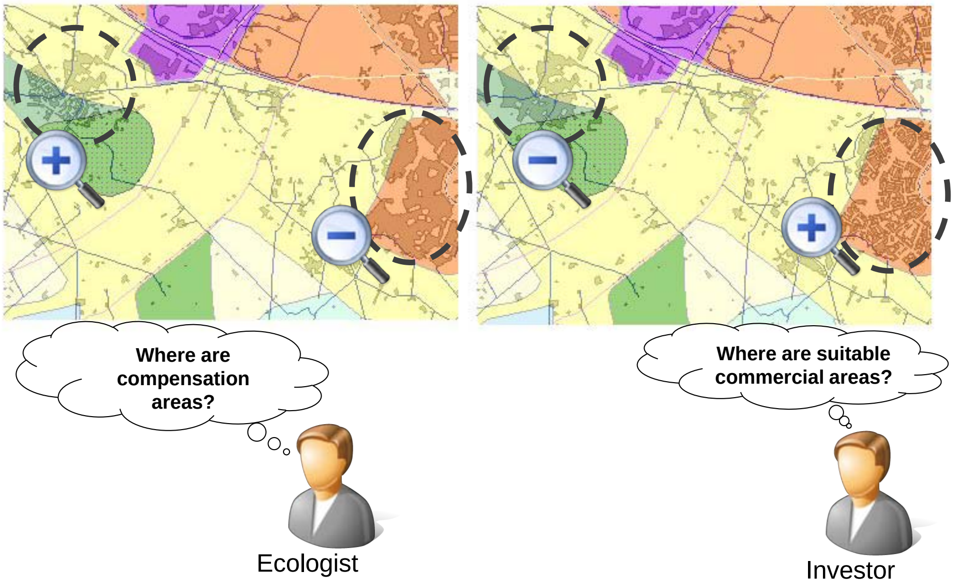


Figure 3.2: Example of a map for an ecologist and an investor. Same map extent and physical plan, but different base map. Areas with differences in level of detail are highlighted respectively.

and it is outside the scope of this thesis to discuss it thoroughly. Usability research may give final answers about the applicability of on-demand base maps for physical planning in the future [40].

3.2 User Profiles

Based on the requirements analysis in Section 3.1 this section describes the design of the user profiles. These user profiles model the user requirements towards the base map. They communicate the user requirements in a way, that they are analysed by the web-based architecture for generating the on-demand base maps accordingly. User profiles are a common concept in mainstream IT [42] (Section 2.5) and describe the customization of a user interface (including selected data and offered functionality). In this research, the thematic map is the user interface as it provides user access to the thematic content. The user can interact with the map by zooming, panning or clicking on the map to retrieve information about a specific area or a specific object. For this research, the physical plans are supplied at fixed scales and are not generalized. While zooming, the physical plan objects are only scaled graphically (i.e. no detail added or removed). The user profiles specify the user requirements regarding the base map related to a specific use of the thematic content. They are described in a machine-understandable way to enable the web-based dissemination approach and to support the on-demand character of the web-based architecture.

The model of the user profiles is described by introducing their key aspects (Section 3.2.1). These key aspects are based on the user requirements (Section 3.1.2). The key aspects of the user profiles are examined using an example user profile (Section 3.2.2), describing one of the exemplary users as introduced in Section 3.1.3. The user profile is the input for the generalization process of the on-demand base map, which is described in Section 3.2.3.

3.2.1 Key Aspects of the User Profile

The user profile describes the user requirements towards the base map and reflects thereby the concepts as introduced in Section 3.1.2. In particular, the user profile addresses the following aspects:

- Selection of the applied foreground/background classes (i.e. thematic content vs. base map content)
- User-specific base map symbolization
- Map generalization specification
- Awareness of topological relationship between base map and thematic map (as generic requirement towards the base map)
- Generalization requirement regarding the topological relationship between the base map and the thematic content
- Inheritance of user profiles.

These aspects are explained below in more detail.

Which object classes are part of the thematic content and which are part of the base map is described in the fore/background selection. This information is important for the generation of the base map, as it identifies the relevant thematic classes and the relevant base map classes.

To support the generation of the on-demand base map, the base map symbolization might be user-specific. Therefore, the user profile contains the cartographic model of the base map linked as a SLD document (introduced in Section 2.2.3). The generalization system detects the cartographic conflicts on the map based on the cartographic model. In a more general setting (of multi-source Internet cartography), one could imagine that there is also a second or third symbolization specified for every object class (layer) in case there are conflicts with the symbolization of other object classes (which have higher importance, i.e. are less allowed to change).

Also the map generalization specification is part of the user profile as it provides an essential input to the automated generalization process. As already mentioned

(Section 2.1.2), the map generalization specification describes the constraints and optimization goals which are specific to the designated user of the user profile. The constraints and optimization goals are modeled as elements of the map generalization specification.

Additionally, the user profile models the requirements of preserving the topological relationship and adjusting the level of detail of the base map objects regarding the type of thematic content (Section 3.1.2) through a *topology awareness list* and a *generalization matrix* respectively. The topology awareness list defines if the initial topological relationship between specific thematic classes and specific base map classes should be preserved. The generalization matrix describes the level of detail (allowed amount of change/generalization) of each base map class in relation to each thematic class by weighting the specific constraints and optimization goals supplied in the map generalization specification. Both concepts will be explained in more detail in this section.

Finally, to model user hierarchies and to avoid redundant user profiles, it is possible to use a so-called parent user profile. The child user profile inherits all attribute values of such a user profile (e.g. general constraints, symbolization) or might overwrite specific attribute values if necessary. This single inheritance model is chosen for the user profiles, because it is less error-prone than multiple inheritance [17] and it is also proposed by ETSI, as described in Section 2.5.

It is important to note, that the user profile does not specify (fixed) portrayal information regarding the thematic content (i.e. thematic symbolization), as this content is included separately in the final map display according to the layer concept. Moreover, in the context of physical plans, symbolization is prescribed by law. Thereby the generalization system is pre-configured with this information and can apply it, when it is necessary during the generalization process. The cartographic representation of the physical plan objects is important as input for the generalization of the base map objects, as it defines the extents of the physical plan objects on the map and is necessary to detect any conflicts with the base map objects regarding the user requirements and the additional map generalization specification.

The UML model of the user profile is given in Figure 3.3. The topology awareness list and the generalization matrix are modeled as sets of tuples defining the relation between the thematic classes and the base map classes. The elements included in the map generalization specification have a name and a value property and two operations. The *weightMapGenSpecElem()* operation allows the generalization matrix to weight the map generalization specification element. The weighted map generalization specification element is then applied using the *performMapGenSpecElem()* operation.

The base map specific aspects of a user profile are incorporated in the topology awareness list and the generalization matrix. These base map specific aspects will be introduced in the following paragraphs in more detail.

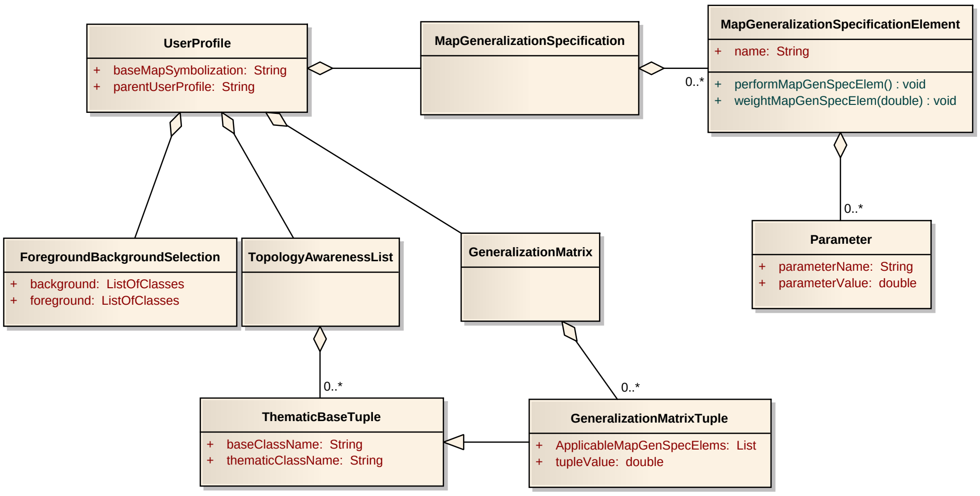


Figure 3.3: UML model of the user profile.

Topology Awareness List

The topology awareness list defines at class level which base map objects have to maintain the original topological relationship regarding the specific thematic content. It allows one (e.g. the provider of RO-Online) to model the topological commitment of base map objects towards the thematic classes. This enables to meet the user requirement of maintaining the topological relationship as it was defined for this research (Section 3.1.2).

Generalization Matrix

The generalization matrix refines the supplied map generalization specification to adjust the map generalization specification of specific base map objects according to the specific thematic content (Section 3.1.2). It relates any thematic class to any class of the base map in a matrix and assigns to each of these relations a specific map generalization specification weight (msw). This weight updates the constraint or optimization goal value (v) assigned to a specific base map object, which is topologically inside a specific thematic class. The map generalization specification value (msv) of the updated map generalization specification element is computed by Equation 3.1.

$$msv = v * msw \tag{3.1}$$

The meaning of the value of msw (i.e. weight) is analog to a magnifying glass or the zoom in/out tools in maps:

- If $msw > 1$, the level of detail will increase (more important)
- If $msw < 1$, the level of detail will decrease (less important).

To actually implement the meaning of msw, a weighting function is attached to each map generalization specification element (`weightMapGenSpecElem()` in Figure 3.3).

As an example, let us assume that there is a map generalization specification element which describes that two buildings should always have a minimum distance of 50 map units between each other. Applying now a weight of 0.5 decreases the scale. Thus, the modified map generalization specification element applies subsequently a distance of 100 map units as the value for the minimum distance between two buildings. In the given example the `weightMapGenSpecElem()` function of the specific map generalization specification element applies internally an inverted value to realize the decrease of scale. To keep the generalization matrix more flexible, it is possible to attach to each map generalization specification weight a set of applicable map generalization specification elements. Thereby it is possible to weight map generalization specification elements differently.

The introduced map generalization specification weight is different from the concept of importance of optimization goals (Section 2.1.2), as it does not change the importance of the optimization goals. Weighting the map generalization specification element has a real effect on the scale of an object (i.e. amount of allowed change to a certain object), whereas the importance of an optimization goal only plays a role in particular map situations in which two optimization goals compete with each other and influences the scale of an object not directly. Thus the two concepts can be applied complementary.

Overall, the advantage of the generalization matrix is that the more relevant (important) objects are kept longer (i.e. changed less) and that the needed space is created by more generalizing less relevant objects. An alternative might be that the user also ranks (sort, order) the object classes for more to less important in an alternatively structured user profile, instead of specifying explicit weights. This might be easier for the user of for instance RO-Online than assigning weights.

3.2.2 Example of the User Profile

This section presents an exemplary XML document of the user profile. XML has been chosen as an appropriate encoding, as it is the de-facto standard for encoding documents on the web [6]. In an ideal situation, the structure of the XML document (i.e. the XML schema) would have been extracted from the UML diagram automatically (Section 3.2.1 and Figure 3.3) using the MDA approach. Due to lack of supported tools, the XML-schemas were created manually for this research. The presented example is also applied in the web-based architecture (Section 4.1). The aim of presenting such an exemplary XML document is to clarify the different aspects of the user profile. However, the example is not meant to define a specific notation of the user profile itself.

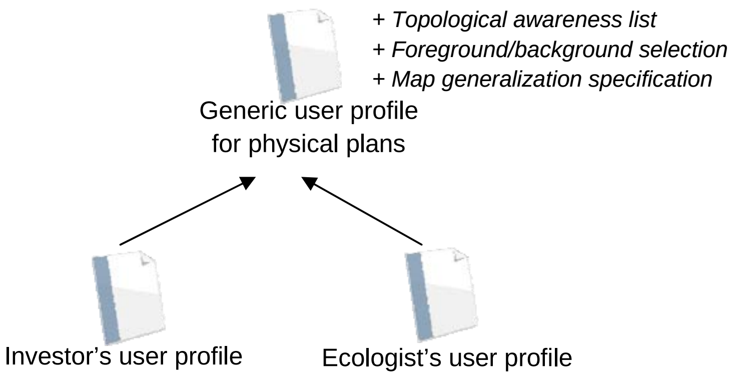


Figure 3.4: Inheritance example for physical planning maps.

It is assumed that the values of the specific user profile document are specified by the provider of the physical planning portal (i.e. RO-Online). The portal provider is aware of the specific user requirements and designs the user profiles according to cartographic criteria. Firstly, the portal provider specifies a parent profile, which describes the common properties of a certain group of user profiles for physical planning. Secondly, the provider specifies user profiles for an ecologist and an investor, which inherit from the generic user profile for physical planning. This results in a user profile hierarchy as illustrated in Figure 3.4.

The parent user profile describes the foreground and background selection, the topology awareness list and an initial map generalization specification (Listing 3.1). The base map symbolization is defined by a remotely referenced SLD document. This allows the generalization process to create the cartographic objects. Creating these cartographic objects is essential for the generalization process, as the boundaries of the cartographic objects might differ from those of the original objects in the database. For instance point and line objects become area objects on the map and thereby require a certain space on the final map. This is one of the main reasons for conflicts on the final map display and symbolization is thereby a crucial aspect for the generalization process.

The foreground and background selection is denoted in the parent profile, as the selected classes are common to the investor and the ecologist. The selected classes are specified as references to remote data services (e.g. WFS) serving the specific vector data (encoded as GML). Each selected class defined in the user profile has an id attribute attached, which is used as reference for the other elements in the XML document (e.g. thematic base tuple and generalization matrix tuple). In the given example the parent user profile selects the area objects of the municipal plan (Commercial Districts and Natural Districts) as foreground and large scale roads (GBKN Roads) and buildings (GBKN Buildings) as base map objects (background) to be included in the final map. The topological awareness list is common to all physical plan with legal meaning such as a Bestemmingsplan (Section 2.6). Preserving the topological relation is only applicable in such cases, in which the boundaries of the thematic objects are relevant regarding the base map objects. In this example the topology awareness is therefore defined as a common property in the parent user profile. In this example the topological awareness is defined between the area planning objects and the area base map objects. Finally, the parent user profile defines an initial map generalization specification. In particular, the parent user profile defines a `preserveBuildings` element. It has to be noted, that the map generalization specification element described in the user profiles are well-known to the generalization system in advance. It identifies the map generalization specification element based on the name attribute described in the XML-element. From the example it becomes already clear, that applying the concept of inheritance to user profiles is convenient, as it avoids defining any user profile from scratch. Additionally it improves the maintainability of the user profiles.

Listing 3.1: Example of the parent user profile document.

```
<UserProfile baseMapSymbolization="http://myserver.com/sld.xml">
  <ForegroundBackgroundSelection>
    <Foreground>
      <ListOfClasses>
        <Item id="Commercial_Districts"> http://myserver.com/wfs?
            Commercial_Districts</Item>
        <Item id="Natural_Districts">http://myserver.com/wfs?
            Natural_Districts </Item>
      </ListOfClasses>
    </Foreground>
    <Background>
      <ListOfClasses>
        <Item id="GBKN_Roads">http://myserver.com/wfs?GBKN_Roads</Item>
        <Item id="GBKN_Buildings">http://myserver.com/wfs?GBKN_Buildings</
            Item>
      </ListOfClasses>
    </Background>
  </ForegroundBackgroundSelection>
  <TopologyAwarenessList>
    <ThematicBaseTuple baseClassName="GBKN_Buildings" thematicClassName="
        Commercial_Districts"/>
    <ThematicBaseTuple baseClassName="GBKN_Buildings" thematicClassName= "
        Natural_Districts"/>
  </TopologyAwarenessList>
  <MapGeneralizationSpecification>
    <MapGeneralizationSpecificationElement name="preserveBuildings">
      <Parameter parameterName="toleranceValue">
        <value>50</value>
      </Parameter>
    </MapGeneralizationSpecificationElement>
  </MapGeneralizationSpecification>
</UserProfile>
```

The examplary (child) user profile describes the requirements of an investor towards the base map for a specific plan. The investor requires a map, which portrays the base map objects inside commercial areas at much higher detail, than the rest of the base map (Section 3.1.3). The example in Listing 3.2 describes requirements towards a base map served to an investor consulting a municipal plan (i.e. Bestemmingsplan).

Listing 3.2: Example of a user profile document for an investor.

```
<UserProfile parentUserProfile= "http://myserver.com/parentUserProfile.xml">
  <MapGeneralizationSpecification>
```

```
    <MapGeneralizationSpecificationElement name="anotherConstraint">
      <Parameter parameterName="anotherParameter">
        <value>42<value>
      </Parameter>
    </MapGeneralizationSpecificationElement>
  </MapGeneralizationSpecification>
  <GeneralizationMatrix>
    <GeneralizationMatrixTuple baseClassName="GBKN Buildings"
        thematicClassName="Commercial Districts">
      <tupleValue>2</tupleValue>
      <ApplicableMapSpecElems>
        <ApplicableMapSpecElem>preserveBuildings</ApplicableMapSpecElem>
      </ApplicableMapSpecElems>
    </GeneralizationMatrixTuple>
    <GeneralizationMatrixTuple baseClassName="GBKN Buildings"
        thematicClassName="Commercial Districts">
      <tupleValue>0.5</tupleValue>
      <ApplicableMapSpecElems>
        <ApplicableMapSpecElem>preserveBuildings</ApplicableMapSpecElem>
      </ApplicableMapSpecElems>
    </GeneralizationMatrixTuple>
  </GeneralizationMatrix>
</UserProfile>
```

The user profile indicates its parent user profile (first line in Listing 3.2). The content of the parent user profile is described in Listing 3.1. When the user profile is being processed, the generalization system automatically takes the values of the parent user profile into account. In the given example, the user profile defines an additional map generalization specification element. This has been inserted to demonstrate the inheritance mechanism. The generalization process will apply the map generalization specification of the parent user profile as well as the map generalization specification of the specific user profile.

Finally, the user profile defines the generalization matrix, which reflects the individual generalization requirements of the base map regarding the thematic content. The user profile weights the `preserveBuildings` element for base map objects of type buildings regarding commercial areas differently than regarding natural areas. According to Equation 3.1 in Section 3.2.1, this affects the parameter value of the `preserveBuildings` element. The base map objects inside a commercial area will be generalized with a map generalization specification value of 100. Whereas base map objects inside a natural area will be aggregated with a tolerance value of 25.

3.2.3 The Automated Generalization Process for On-demand Base Maps

The generalization process of the base map takes into account the following aspects:

1. User profile
2. The specific thematic content
3. The specific zoom level (scale) of the map display.

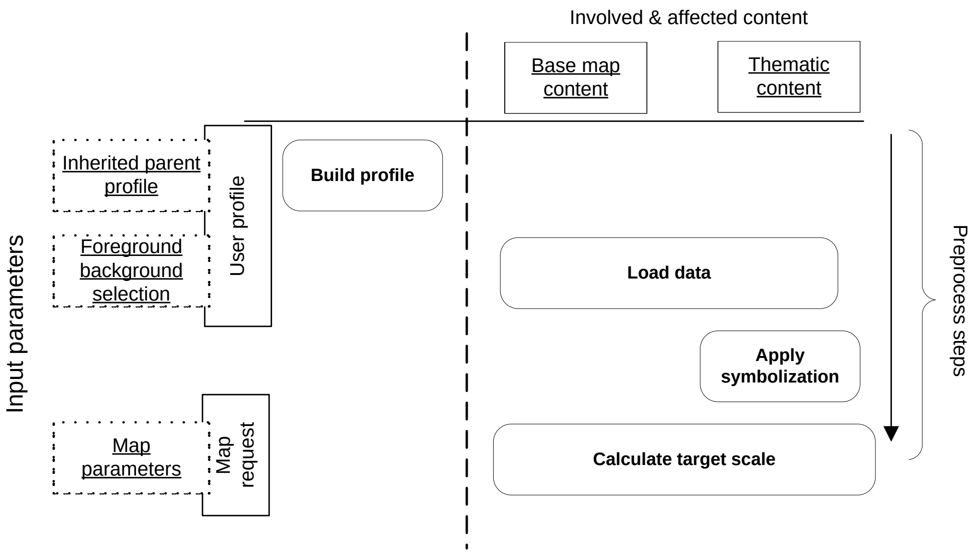


Figure 3.5: Generalization preprocess to initialize the user profile and load the data.

This section presents the different stages of the generalization process and examines at which stage the listed aspects steer the generalization process.

The generalization process has to meet specific prerequisites to handle the user profile. This is achieved by preprocessing the specific user profile (Figure 3.5). Firstly, the process initializes the user profile by analyzing the indicated parent user profile. The process adds the inherited properties of the parent user profile to the properties of the specific user profile. Thereafter the process loads the data for the foreground and the background of the map according to the described selection in the user profile. This allows the process to link the two types of content and to customize the base map with respect to the thematic content. The symbolization of the thematic content is defined in advance and is not subject to change regarding a specific user. An important property of the symbolization is the cartographic line width for calculating the shape of the specific objects on the map. This will enable the generalization process to detect any conflicts with a base map object, if necessary. Before generalizing, the zoom level of the map has to be determined by using the extent (bounding box) and the size (width, height) of the requested map. The zoom level is important to calculate the map units for sufficiently performing and evaluating the generalization process leading to the best result.

Figure 3.6 depicts the actual generalization process as described below. As a first step, the process applies the symbolization provided in the user profile to generate cartographic base map objects. The generalization matrix weights the map generalization specification according to the thematic overlay. The generalization process initializes the set of micro and meso agents (according to the AGENT model; Barrault et al. [10]) for the generalization matrix and attaches the supplied map generalization

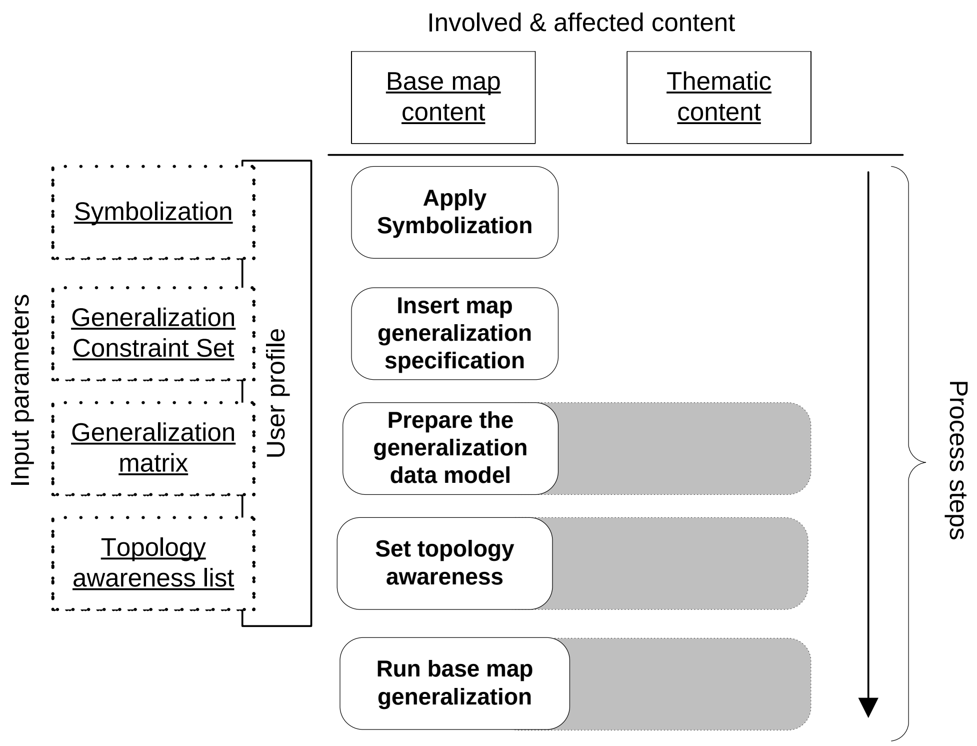


Figure 3.6: Generalization process for on-demand base maps.

specification elements to them. In particular, the AGENT model is adapted to model the relation between thematic content and base map objects. The meso agents are derived from the thematic objects and group the base map objects to steer the generalization of the base map objects. This process results in an internal configuration of agents as depicted in Figure 3.7. The map generalization specification elements attached to each of the base map objects will be weighted based on the values of the generalization matrix. Additionally, based on the topology awareness list certain meso agents are configured to preserve the topological relations between the thematic content and the base map objects. Based on the configuration of the agents, the process performs the generalization of the map by optimizing the portrayal according to the map generalization specification.

3.3 The Web-based Architecture for On-demand Base Maps

The design of the proposed architecture is based on the architectural requirements as presented in Section 3.1.1. The architecture applies OGC Web Services (Section 2.2.2), which either provide geographic data (WFS), portray functionality (WMS)

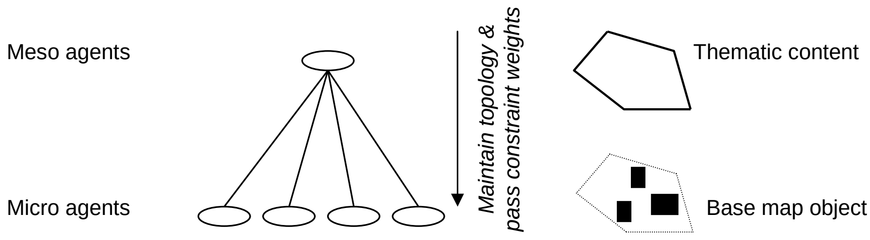


Figure 3.7: AGENT model for on-demand base maps based on the user profile.

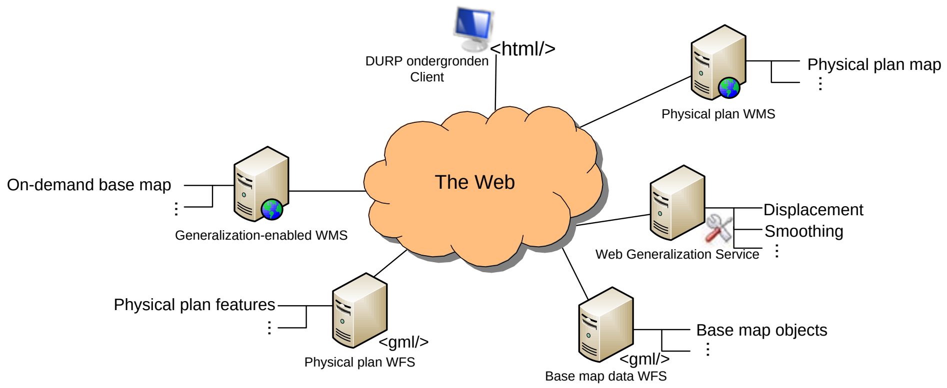


Figure 3.8: The architecture to disseminate on-demand base maps on the web.

or geoprocessing functionality (WPS). To generate and disseminate on-demand base maps on the web generated by automated generalization, this research extends the WMS interface to make it generalization enabled. This generalization-enabled WMS is the core of this architecture and a key contribution of this research. The WMS interface has been chosen, as it is the state-of-the-art approach for disseminating maps on the web such as physical plans (Section 2.2.3). The generalization-enabled WMS disseminates on-demand base maps according to a specific user profile by applying the generalization process described in Section 3.2.3.

The generalization-enabled WMS retrieves the necessary geographic data (base map data and plan data) from WFS instances. The geographic data as well as the user profile are processed on-the-fly at server side. The generalization process embedded within the WMS is capable of executing Web Generalization Services (Section 2.3.1) using generalization algorithms of others. This is beneficial to improve the quality of the generated base maps. The Web Services incorporated in this architecture are depicted in Figure 3.8.

In the architecture the base maps and the thematic content are accessed by a browser-based client application (i.e. the DURP ondergronden client). The client application allows the user to select the correct user profile in advance through a separate web page. Thereafter the client application performs the appropriate request based on the chosen user profile, extent and zoom level to retrieve the desired map. In particular the client application retrieves the base map and the thematic content as separate layers from different WMS instances and overlays them locally (using semi-transparency). This allows the user to combine the layers with different geographic data and it thus enables the flexibility of the architecture.

The final map with its map layers is specified in the WMC document, which is internally linked by the client application to one of the user profiles. The WMC document and the user profile reference SLD documents. The combination of the available standardized user models (WMC and SLD) with the user profile enable to describe a map in a comprehensive and flexible way (the specification of the final maps can be changed without changing the architecture). This combined approach specifies the layout of the map (without the actual data), including extent, selected layers, symbolization and map generalization specification.

The architecture covers the most important user profile categories as specified by the ETSI (see Figure 2.10, Section 2.5). The proposed user profile and the SLD document are related to human centered preferences. The application of WMC documents in the architecture allows the service provider to specify service related information and preferences. This also demonstrates that the combination of these aspects provides a means to specify the preferences of the user towards the map comprehensively. Figure 3.9 depicts the described relation between those aspects and their relation to the ETSI categories. This combined approach is sufficient, if the architecture has only to be integrated into a single type of client application (e.g. a web browser application). In case of a more heterogeneous environment with different types of clients (e.g. mobile phones) device related information and preferences might become necessary as well.

As the architecture is based on standards, it can easily be integrated into existing web-based architectures of organizations such as NMAs. For instance, this architecture can be embedded into RO-Online (Section 2.6). The architectural workflow and the implementation of the generalization-enabled WMS are described in Section 3.3.2 and Section 4.1 respectively.

Section 3.3.1 describes, how the user profiles are propagated to the generalization-enabled WMS. This is essential to provide the user profile as input for the incorporated generalization process. To finally present the interaction between the Web Services as introduced in this section (data services, the generalization-enabled WMS and Web Generalization Services) Section 3.3.2 describes the resulting workflow.

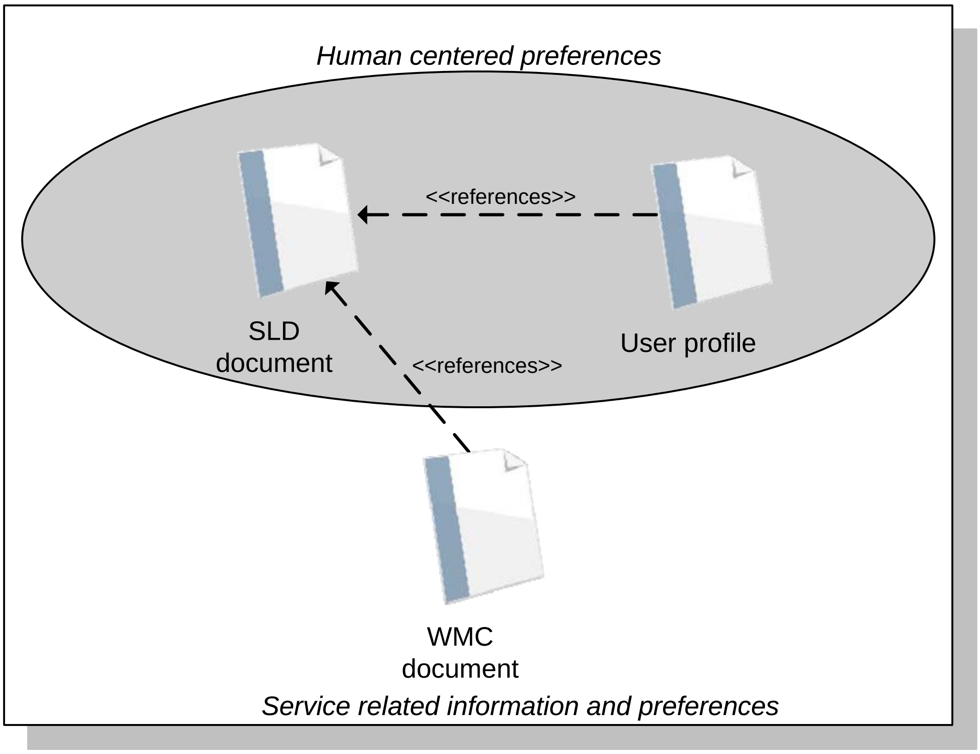


Figure 3.9: The combined approach for on-demand web mapping, as applied in the architecture.

3.3.1 Embedding User Profiles in the Architecture

As the generalization-enabled WMS serves different base maps for different types of users, the user profiles have to be propagated at the same time the base map is requested. Additionally, the requirement to process user profiles instantly is related to the stateless nature of Web Services [95] (Section 2.2.1). Thus, all the required information (i.e. the user profile, zoom level and extent, see also Section 3.2.3) has to be gathered at the same point of time at which the request is received. Based on the user profile, the base map is generated on-the-fly and sent back to the client. The requirement for instant processing of the user profile has some implications for the WMS interface, which therefore had to be extended.

According to the WMS interface specification, any map is retrieved using the GetMap operation (Section 2.2.2, exemplary requests in Appendix A.2). The GetMap operation is extensible by so-called vendor-specific parameters, which can be defined as optional input of the service interface. However, these parameters must not be mandatory and the service must not stop working if such an optional parameter is missing in the specific request.

Consequently, a vendor-specific parameter has been specified for the generalization-enabled WMS. This parameter references the user profile according to which the base map is generated. The generalization process incorporated in the WMS retrieves the user profile using this location reference and processes it as described in Section 3.3. An example of such an extended GetMap request is shown in Listing 3.3. The request specifies the base map as a required layer (indicated by `LAYERS=BaseMap`), as the physical plan is requested seperately from the Physical plan WMS (as explained in Section 3.3.2). An example of a user profile document has been presented in Section 3.2.2.

Listing 3.3: Sample GetMap request incorporating a reference to a user profile for a base map layer

```
http://myWMS?Request=GetMap&
SERVICE=WMS&
BBOX=...&
LAYERS=BaseMap&
USERPROFILE=http%3A//anotherServer/thisUserprofile.xml
```

Including a reference to the user profile instead of including the specific content of the user profile in the request avoids a conflict with any character limitations of the internet protocol. In case of HTTP, there is no practical character limit, but several web-browsers such as Microsoft Internet Explorer and web servers reject HTTP-GET requests beyond 2000 characters. Additionally, using references enables caching and is helpful when considering an implementation in a production environment. The reference to the user profile can be used as an identifier for already processed user profiles. It allows performing the generalization process once, but reusing the result multiple times. However, it has to be made sure, that an update of the user profile is also propagated to the generalization process (incorporated in the generalization-enabled WMS). This requires specific update strategies, which check after a specific timeout or a specific event, if the user profile has not changed yet. Additionally, as the generalization process accesses remote data services, which might be updated content-wise frequently, the cached generalization results might become outdated. Thus selecting a specific caching strategy requires careful analysis [169].

The GetMap operation provides the extent and the width and height of the map, which can be used to calculate the scale of the map. Based on the scale the generalization process determines the map units. The map units are relevant to perform the generalization process correctly.

The information about the initial extent of the map and the incorporated layers is provided by the WMC document, which is internally linked in the client application to the user profile. The WMC document is only a source of information for the client to retrieve the initial map. In the course of user interaction with the client application (zooming, selecting layers), the map is changing and the client application uses its internal state to specify the applicable requests to the specific services.

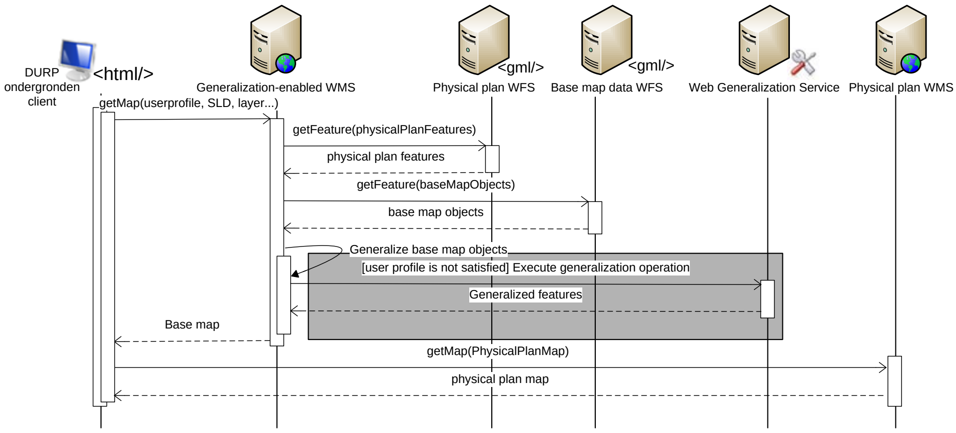


Figure 3.10: Architecture workflow - the gray-shaded box marks the complex generalization processing as described in Section 3.2.3. Workflow is simplified, not showing the link between Physical plan WFS and Physical plan WMS, which is established automatically whenever a plan is retrieved by the client.

3.3.2 The Architecture Workflow

The workflow of the involved Web Services (Section 3.3) is depicted in Figure 3.10. The client application creates the final map by requesting separately the on-demand base map served by the generalization-enabled WMS and the physical plan served by another WMS. The two requested layers will be overlayed inside the client and presented to the user on the map display. This leads to communication overhead for the client application, as it has to trigger two services, but increases its flexibility. Additionally, the client can already display the thematic data while waiting for the base map to be customized. However, note that the generalization process generating the base map has to be aware of the thematic content.

Based on the client request the generalization-enabled WMS retrieves the thematic data and the base map data from the WFS instances as referenced in the user profile to generate the base map. After receiving the geographic data, the generalization-enabled WMS performs the generalization process (indicated as a gray box in Figure 3.10). Section 3.2.3 and especially Figure 3.6 describe this process in more detail. This process also triggers Web Generalization Services, providing additional generalization functionality. Finally, the generalization-enabled WMS serves the base map layer according to the user profile without including the thematic data. As already mentioned, the thematic layer is requested separately by the client application from the Physical plan WMS.

Retrieving the on-demand base map and the physical plan from different services has some implications on the consistency of the data on the final map. In Figure 3.10, the WFS and the WMS for the physical plan data are detached from each other, which is a simplified situation. In fact, the WMS may retrieve geographic data from the WFS, render them and send it to the client, which might solve the issue of data inconsistency, in an optimal sitation. But in practice this might still result in inconsistencies, as the data might have changed during the request of the base map and the physical plan. The problem of data consistency also remains, due to caching strategies and the consequence of retrieving out-dated data from the cache. Therefore, further research on applicable caching strategies is required. Finally, to avoid data inconsistency, the on-demand base map and the physical plan may be served as one layer by the generalization-enabled WMS, which results in a static map and makes it impossible to combine the base map with other content.

3.4 Synopsis of the Design

This chapter presents the design of the architecture and the user profile based on the results of the requirements analysis. The user profile is a means to communicate the user requirements to the architecture and to serve thereby as an input for the generalization-enabled WMS, which has been designed as part of this research. Main aspects of the user profile are the topology awareness list and the generalization matrix, which together realize the user requirements. Based on the architectural and user requirements, the generation of the base map is implemented by the AGENT model using the thematic content as an input. The on-demand base maps are generated according to the specific user requirements based on the user profile, the content of the map (described in a WMC document) and the symbolization (described in an SLD document). This combined approach covers all the ETSI categories, except for the device related information. For this research, a web-browser application is assumed to retrieve the on-demand base map. The architecture is based on Web Services to integrate data services, such as those stemming from RO-Online. Additionally, the architecture incorporates Web Generalization Services to enhance the functionality of the generalization-enabled WMS.

For this research, the user profiles are provided to the generalization-enabled WMS as a vendor specific parameter. In the future, this additional parameter as well as the user profiles should become (part of) official standards at OGC or ISO respectively. Defining these concepts in cooperation with a standards body is beneficial to proof the applicability of the presented concepts in practice.

Based on the design the architecture will be implemented using the described standards (WMS, SLD, WPS, WFS, WMC), as described in Chapter 4. The generalization-enabled WMS will incorporate a generalization system, providing cartographic generalization supported by the agent-based approach (Section 2.1.3). The presented requirements are the basis to evaluate the implemented architecture (Section 4.3). Different map samples will be presented according to the requirements of the exemplary users (Section 3.1.3). One of the discovered limitations, as also stated in Section 2.3.1, is the lack of semantic interoperability to create meaningful generalization processing on the web. Therefore, the implementation described in Chapter 4 incorporates pre-defined constraints (parameterized by the user profile) and does not access Web Generalization Services. Chapter 5 will present an approach for semantic interoperability of generalization functionality to especially address the latter issue.

Chapter 4

Implementation and Evaluation of the Architecture

This chapter presents the implementation of the architecture based on the design presented in the previous chapter. The implementation is used to proof the design of the architecture and to evaluate it regarding specific requirements.

The core of the architecture is the generalization-enabled WMS. Section 4.1 will explain the implementation of this component in more detail. The implementation of the automated generalization for generating the on-demand base map is presented in Section 4.2. The implemented generalization process is incorporated into the generalization-enabled WMS and realizes the process designed in Section 3.2.3. The architecture is evaluated in Section 4.3 by presenting map samples and by validating the requirements of Section 3.1. Based on the encountered limitations, this chapter will suggest some solutions regarding lack of performance, GII integration and meaningful integration of Web Services.

4.1 The Generalization-enabled WMS

The generalization-enabled WMS makes use of 1Spatial Clarity to access agent-based generalization functionality [79]. 1Spatial Clarity has been selected for this research, as it has demonstrated some promising results in the automated production of national topographic maps at different scales [115, 153] (Section 2.1.3).

The generalization-enabled WMS is based on the GeoServer application server [30, 70]. The reasons for choosing GeoServer are two-fold. Firstly, it is written in the Java programming language [180] and thereby can be linked to 1Spatial Clarity seamlessly through an Application Programming Interface (API). Secondly, it can be extended easily through so-called datastores, which connect to databases or middleware components.

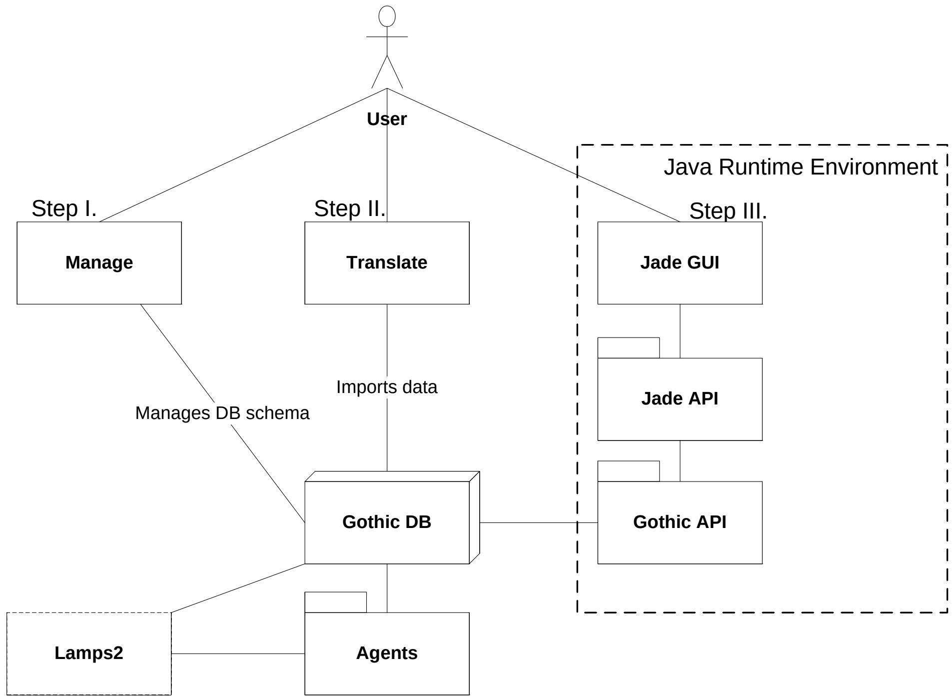


Figure 4.1: Overview architecture of 1Spatial Clarity.

The integration of 1Spatial Clarity into GeoServer is realized via the so-called *Clarity Datastore*.

Section 4.1.1 and Section 4.1.2 will introduce the architecture of 1Spatial Clarity and GeoServer. The Clarity Datastore is presented in Section 4.1.3.

4.1.1 Architecture of 1Spatial Clarity

1Spatial Clarity is built on top of the object-oriented database Gothic, which incorporates the agent-based generalization and a solid implementation of topology.

Figure 4.1 presents the main components of 1Spatial Clarity. A common process consists of deploying the data model (Manage), importing the data (Translate) and then configuring the generalization process within the Jade GUI. However, all the functionality, which is provided through Manage and Translate, can also be directly executed by a native Java call through the Gothic API.

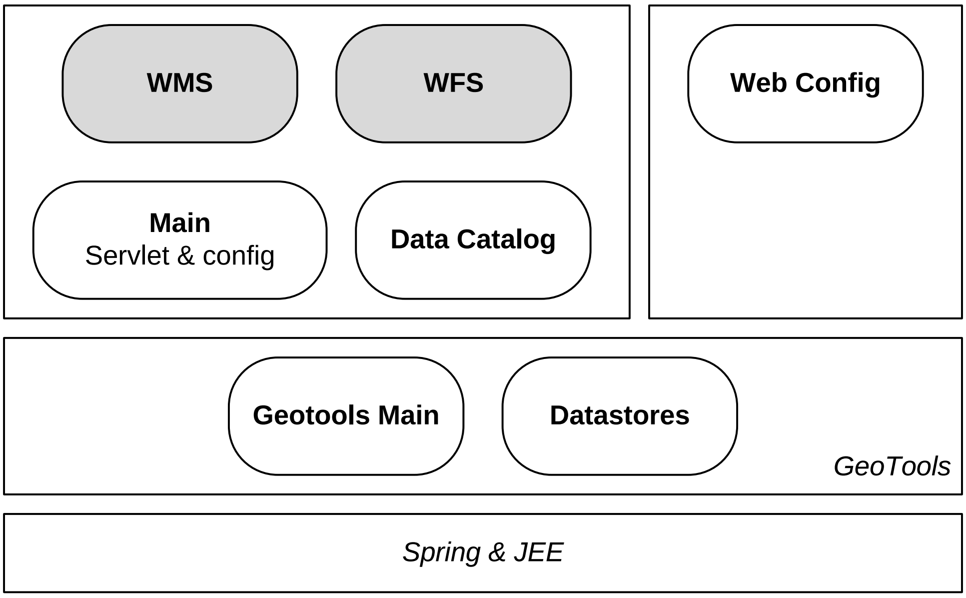


Figure 4.2: Overview of the GeoServer architecture.

4.1.2 Architecture of GeoServer

GeoServer is an application server providing access to geographic data via for instance WMS interface and WFS interface. GeoServer is based on Java Servlet technology and the Spring application server framework and runs in several servlet containers such as Apache Tomcat or Jetty. One of the main building blocks of GeoServer is GeoTools [71], which implements the OGC simple feature model (also known as ISO 19125, [94]) and for instance supports WMC and SLD documents [186]. Additionally, GeoTools provides the concept of datastores, which allows GeoServer to access databases and middleware components in a unified way. Currently, GeoTools provides datastores for file (e.g. ESRI shapefile) and database (e.g. PostGIS, Oracle) access. GeoServer and its datastores can be easily configured through a browser-based user interface. The overall architecture of GeoServer is depicted in Figure 4.2.

4.1.3 The Clarity Datastore

The generalization-enabled WMS is realized as a combined software component of 1Spatial Clarity and GeoServer in this research. Integrating Clarity into GeoServer is possible due to the following reasons:

1. Clarity and GeoServer are written in the Java programming language
2. Clarity can be loaded directly from another application through the underlying

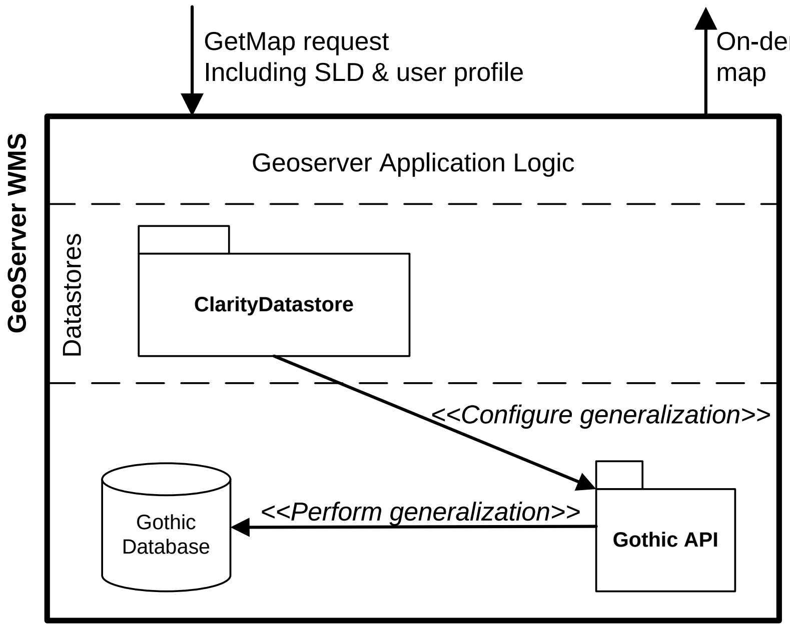


Figure 4.3: Implementation architecture of the generalization-enabled WMS.

Gothic API

3. GeoServer allows embedding external software applications by the means of datastores (e.g. for database access).

The connection between GeoServer and 1Spatial Clarity is realized by the concept of datastores. The implementation architecture of the generalization-enabled WMS and its core component, namely the *Clarity Datastore* is depicted in Figure 4.3. Based on the concept of datastores 1Spatial Clarity can be configured with the request parameters of the client application (user profile, SLD) to perform the generalization on-demand. The Clarity Datastore configures and accesses the Gothic database through the Gothic API.

In particular, the Clarity Datastore is based on three interfaces of the GeoTools application layer, which realizes the Datastore concept: `DatastoreFactorySpi`, `AbstractDatastore` and `FeatureReader` (Figure 4.4). The Clarity Datastore itself then is realized by four classes:

- `ClarityDatastoreFactory`
- `ClarityDatastore`
- `ClarityFeatureReader`
- `ClarityGeneralizationProcess`.

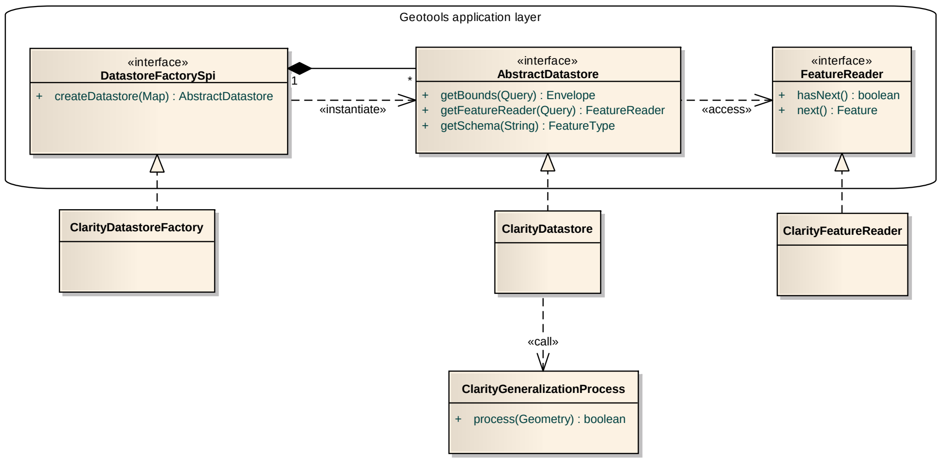


Figure 4.4: Class diagram of the Clarity Datastore.

The `ClarityDatastoreFactory` implements the `DatastoreFactorySpi` and maintains the connection to the Gothic database which is represented by the `ClarityDatastore` class. A `ClarityDatastore` is created by `createDatastore()`. The `ClarityDatastore` represents the `AbstractDatastore` and provides access to the actual data hosted in the Gothic database. The methods `getBounds()` and `getSchema()` provide metadata to the GeoServer implementation to create for instance the GetCapabilities document for the WMS. The `getFeatureReader()` provides access to the generalized features. In particular, `getFeatureReader()` performs `process()` of a `ClarityGeneralizationProcess` based on the input of the user profile and the symbolization. Additionally, `getFeatureReader()` creates a `ClarityFeatureReader`, which provides access to these generalized features in an iterative way. This sequence of actions is also depicted in Figure 4.5.

The full course of action incorporated in the `ClarityGeneralizationProcess` is described in Section 3.2.3 and its implementation is described in Section 4.2. The result of the process is accessible for GeoServer through the `ClarityFeatureReader`, which implements the `FeatureReader`. The resulting features are finally rendered inside GeoServer as a map and sent back to the client.

The resulting map can be displayed in an OGC-compatible WMS client such as depicted in Figure 4.6. The screenshot of the DURP ondergronden client displays a municipal physical plan on top of the on-demand base map containing generalized buildings (and for illustration purpose only, it also shows the original outline of the buildings). The DURP ondergronden client is based on MapBuilder [125] and allows the user to incorporate different map layers offered by different WMS instances into any browser-based application. Additionally, it can be easily configured using WMC documents. Both aspects were crucial, when selecting this framework for the presented research. An examplary WMC document, which is used to configure the

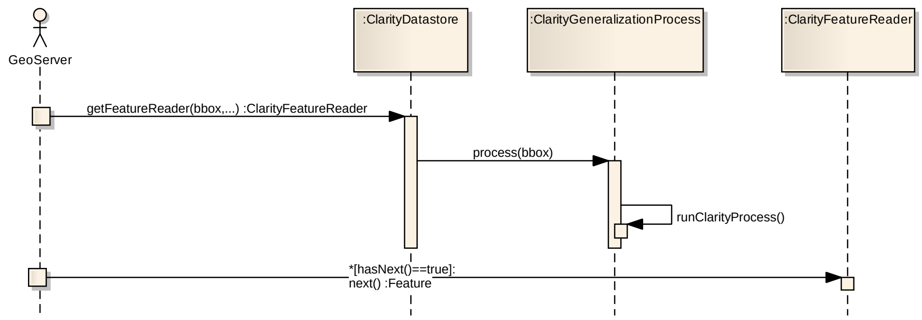


Figure 4.5: Sequence diagram of the Clarity Datastore. This sequence of actions is performed whenever a map is requested.

DURP ondergronden client is described in Appendix A.4.

4.2 The Generalization Process for On-demand Base Maps

This section describes the implementation of the designed generalization process of Section 3.2.3. The generalization process is called by the WMS, whenever it reveives a GetMap request specifying user profile, SLD (symbolization), location and scale. In particular, this generalization process is implemented in the `ClarityGeneralizationProcess` class (see Figure 4.4 and Section 4.1.3).

The process configures the Gothic database through the Gothic API by applying the symbolization, the agent data model and the actual map generalization specification. This information is described in the user profile. The symbolization is generated based on the referenced SLD in the user profile. The configuration of agents depends on the thematic content. The thematic objects serve as meso agents, driving the generalization of the micro agents (i.e. representing base map objects), as already described in Section 3.2.3. In case of 1Spatial Clarity the agent data model is based on inheritance, as depicted in Figure 4.7. In particular, the thematic content (physical plan areas of the Bestemmingsplan, `BP_Polygon`) has to inherit from 1Spatial Clarity's meso agent class and the base map classes (in this example `GBKN_Building`) have to inherit from 1Spatial Clarity's micro agents. Additionally, each meso agent knows its micro agents and vice versa via the `controlling_meso` and the `subagents_ref` attribute respectively. Based on this information, each map generalization specification element can weight its value according to the overlaying thematic type. The map generalization specification weight is applied, whenever a plan is proposed to the agent by the attached constraint object (in this example `PreserveConstraintBuilding` with `proposePlans()`). This allows the generalization process to configure the agents individually with specific map generalization specification weights, which are extracted

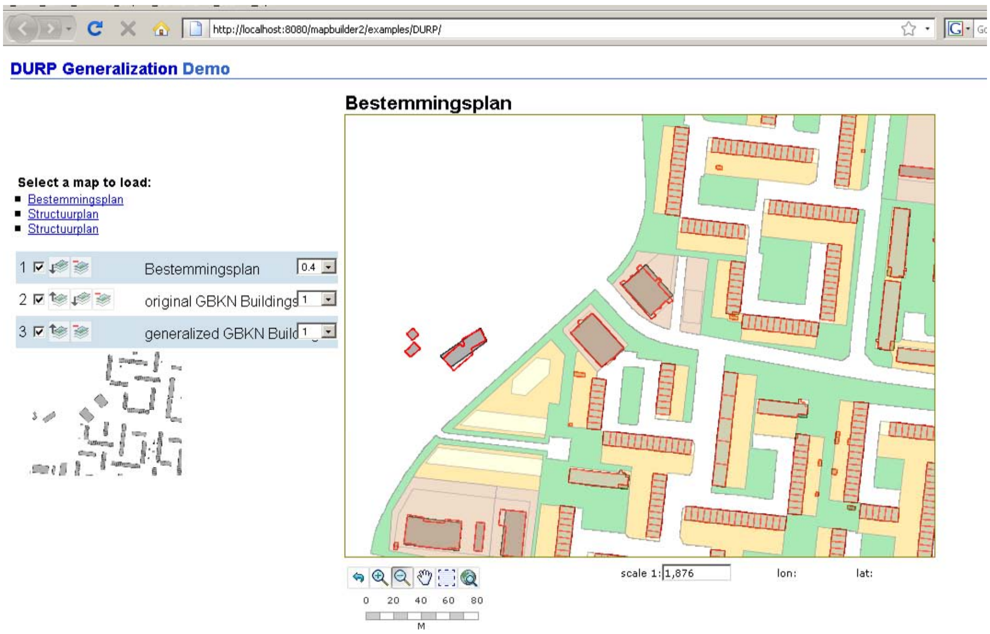


Figure 4.6: Screenshot of the DURP ondergronden client. This map depicts red outline shapes of original buildings, gray shapes of the generalized buildings and a physical plan on municipal level. The final map presented to the user will not contain red outlines of the original buildings, which have only been included here to show the effect of the generalization process.

from the generalization matrix (Section 3.2.1). Based on the topology awareness list each meso agent is configured with a special topology constraint to maintain the topological relationship between the base map and the thematic overlay. All the relevant information of the user profile is internally stored in Clarity's `AgentMapSpecs`. This particular data structure is activated during the generalization process (Line 1, Listing 4.1) and is accessed whenever an agent requires information about optimization goals and constraints.

Listing 4.1: Code fragment of `ClarityGeneralizationProcess`.

```
1    ams.setCurrentMapSpec(mapSpecName);
2    Set objects = Quadx.inArea(v, agentClass, bbox);
3    AgentScheduler scheduler = new gothic.descriptor.AgentScheduler(v);
4    for(int i = 1; i<=objects.numberOfElements(); i ++ ) {
5        GothicObject obj = (GothicObject)objects.getElement(CollectionWhich.
             POSITION, new Integer(i));
6        obj.setVersion(v);
7        Agent.initialise(v, obj);
8        scheduler.addEntry(obj);
9      }
10     AgentSchedulerActivateRV activaterv = scheduler.activate(null);
11   }
```

To apply the generalization process to the objects inside the requested map extent,

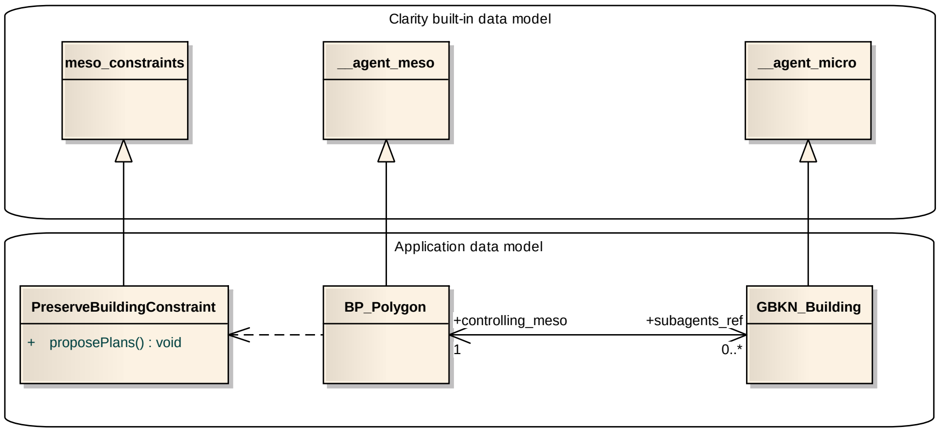


Figure 4.7: Generalization data model in 1Spatial Clarity for on-demand base maps.

the process selects the applicable agents to be generalized (Line 2). This limits the amount of agents to be processed. Clarity's `AgentScheduler` is created (Line 3) and the selected agents are pushed on top of the scheduler (Line 5-9), which is finally activated (Line 10).

The presented generalization process is implemented using data installed locally on server hosting the generalization-enabled WMS. This is contrary to the architecture workflow designed in Section 3.3.2. 1Spatial Clarity is currently not able to connect to such Web Services, but it has been prototypically extended by Regnauld [152] to connect for the instance to Web Generalization Services. Thus, connecting 1Spatial Clarity to Web Services can be considered as an aspect of implementation, which is subject for future work. Connecting to Web Services in such a way requires syntactic interoperability, thus such an integration is not meaningful and can only be achieved in a pre-configured way using known Web Service addresses. The meaningful integration of Web Services into the architecture is extensively discussed in Section 4.3.3 and Chapter 5.

4.3 Evaluation

This section evaluates the architecture to verify the design and the implementation. Additionally, it demonstrates the contribution of this research to web mapping and automated generalization.

Based on the exemplary users (Section 3.1.3) Section 4.3.1 gives some map samples generated by the architecture and the designed generalization process for on-demand base maps in particular. The map samples demonstrate also the effect of the user re-

		Ecologist		Investor	
	Physical planning types	Natural Areas	Commercial Areas	Natural Areas	Commercial Areas
Base map types	Buildings	2	0.5	0.5	2
	Roads	2	0.5	0.5	2

Table 4.1: Simplified generalization matrix of two user profiles for a fictive ecologist and investor. Weight > 1 - more important; weight < 1 - less important.

quirements as described in the user profiles (Section 3.1.2). Based on the requirements the advantages and the limitations of the architecture are evaluated in Section 4.3.2 and Section 4.3.3 respectively. The main limitation is the missing semantic interoperability of Web Generalization Services for meaningful generalization processing. Chapter 5 will elaborate on this limitation in more detail and propose a solution.

4.3.1 Generated Maps for the Selected Users

Based on the potential (types of) users of the developed architecture described in Section 3.1.3, this section presents some generated map samples to illustrate the outcomes of the architecture. The requirements of these potential users are modeled in the architecture through user profiles. One user profile models the requirements of an investor, who is interested in commercial planning areas. The other user profile models the requirements of an ecologist, who is interested in nature planning areas. Subsequently, they have different user profiles resulting in different base maps. As physical plans mostly consist of area objects, the implemented process focuses only on polygons for the geometric overlay.

Table 4.1 presents the generalization matrix, which has been set up for the two user profiles (i.e. the investor and the ecologist). The values of the generalization matrix are applied to the `preserveBuildings` constraint. In this example the values show that the natural areas for the ecologist are more important than the commercial areas. The opposite is true for the investor. The base map is generated from the GBKN database, which is the large-scale topographic database of the Netherlands (Section 2.6.1).

In this example the user profile inherits the same map generalization specification as defined in the parent profile (see Section 3.2.2 and Listing 3.1). The different maps result from the different generalization matrices.

Figure 4.8 shows the resulting maps. In particular, it shows four samples of the same extent (scale 1:2000): the original base map information (IMGeo-based GBKN, source scale 1:1000) (Figure 4.8a), the municipal physical plan with the original base map information (Figure 4.8b), the map (physical plan and generalized base map) as generated for the ecologist (Figure 4.8c) and the map (physical plan and generalized base map) as generated for the investor (Figure 4.8d). The base map objects are retrieved and generalized from the GBKN according to the supplied user profiles. In the sample of Figure 4.8, the base map objects for the ecologist's map are more generalized (less detailed) than for the investor's map. In this example, the `preserveBuilding` constraint is satisfied by aggregating the building geometries. In the ecologist's map all the buildings inside the commercial area are aggregated to one polygon geometry, whereas in the investor's map the buildings are aggregated to a set of three geometries. In both cases, the `preserveBuilding` constraint simplifies the building geometries. The presented example only shows one solution to the `preserveBuildings` constraint. In a more advanced scenario, the buildings could have also been displaced to satisfy the constraint.

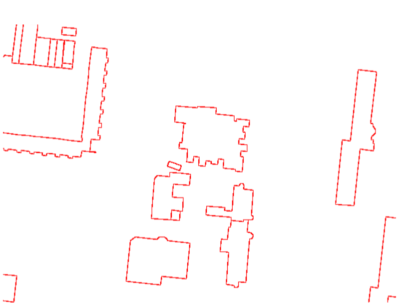
4.3.2 Advantages of the Architecture

The implemented generalization-enabled WMS fullfills many requirements. Firstly, it enables dissemination of on-demand maps, which are generated by cartographic generalization. Secondly, the generalization-enabled WMS serves different maps to multiple users by incorporating a generalization system (i.e. 1Spatial Clarity), which can perform generalization without pre-configuration. The generalization-enabled WMS translates and forwards the user profile on-the-fly to the generalization system and triggers an appropriate generalization process. Meeting the requirements is also possible due to the WMS interface. In fact, the WMS and SLD documents allows the generalization-enabled WMS to separate generalization and symbolization. The generalization system is able to process cartographic features, which are generated based on the symbolization encoded in the SLD. The architecture also meets the requirement of using Web Services. In fact, it is based on WMS, WFS and WPS.

Regarding the user requirements as described in Section 3.1.2, the architecture supports both requirements. Firstly, the user requirements are reflected in the user profile by the topology awareness list and the generalization matrix. Secondly, they are supported by the generalization system, as the AGENT model was adapted to use thematic content as meso agents and the base map objects as micro agents. By adapting the AGENT model it was possible to scale the objects regarding the requirements of the user but also to maintain the topological relationship between the thematic content and the base map objects.

4.3.3 Limitations of the Architecture

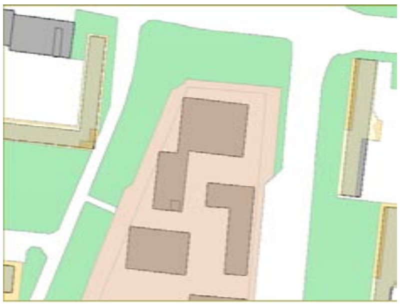
This section evaluates the architecture against the requirements described in Section 3.1. Based on the requirements the architecture has been designed and implemented.



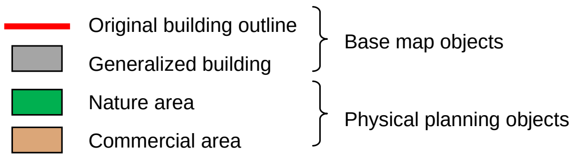
(a) Original base map (source scale 1:1000) displayed at 1:2000.

(b) Original base map (source scale 1:1000) with physical plan displayed at 1:2000.

(c) Ecologist's map displayed at 1:2000.



(d) Investor's map displayed at 1:2000.



(e) Legend

Figure 4.8: Map samples (all displayed at 1:2000) - the original base map information (GBKN, source scale 1:1000) (a), the municipal physical plan with the original base map information (b), the map (physical plan and generalized base map) as generated for the ecologist (c) and the map (physical plan and generalized base map) as generated for the investor (d).

Although the architecture meets the requirements as pointed out in the previous section, the design and the implementation did not solve all the issues. In fact, the design and the implementation confirmed certain limitations.

In particular, the following limitations require additional consideration and are explained in the following paragraphs:

- Lack of performance
- The architecture is not (yet) integrated in an existing GII (i.e. RO-Online).
- Lack of meaningful integration of Web Services for data access and generalization

The following sections present possible solutions for enhancing performance, integrating the architecture in a GII and integrating Web Service meaningfully.

Performance Considerations

The performance aspects of the architecture are outside the scope of this thesis, but to evaluate the applicability of an implementation the performance as such is still interesting. The generalization-enabled WMS, as implemented for this thesis has locally installed data and two pre-configured optimization goals (preserving the building and simplifying the building). The generalization process, as performed by the generalization-enabled WMS takes 30 seconds to generalize a base map consisting of 100 objects and optimizing against two optimization goals. This test was performed on a server system with two CPUs (@2.13 GHz) and 2 GB of memory. The delay of 30 seconds is not acceptable for a web map application accessed by ordinary users, where already a delay of 5 seconds is considered as inappropriate [85]. Additionally, it is expected that this lack of performance becomes bigger, if multiple users access the architecture concurrently.

The encountered problem of performance is caused by the computational complexity of the generalization process. This complexity is caused by the agent-based approach, which optimizes the map situation by calculating different solutions multiple times. Based on the complexity of the incorporated plan of the agents, the optimization of the map setting might result in a computationally intensive task. There are two possibilities to improve the performance of the architecture. One possibility is to enable caching of map results and to apply tiling of map layers. However, in a scenario with different user requirements and the requirement to process up-to-date data, caching might only partially solve the lack of performance. The other possibility is to improve the performance of the generalization process itself by enhancing the agent-based generalization process with Grid Computing technology [62].

To take the full advantage of Grid Computing, the Grid infrastructure requires processing tasks consisting of input data with small memory footprint. Disseminating input data to the Grid infrastructure is considered to be the most expensive aspect. Small memory footprint is mostly achieved by dividing a problem in smaller sub-problems. As the applied AGENT model provides such a division of the generalization problem into smaller processing tasks, the integration of Grid Computing and agent-based generalization is promising.

An integration of Grid Computing and agent-based generalization is highly applicable due to the following reasons:

1. The AGENT model divides the generalization problem into small sub-problems by partitioning the map space, which can be offered as small generalization tasks to the Grid infrastructure. An agent is attached to each of the partitions. The agents configure generalization tasks, which have a small memory footprint (i.e. one physical plan feature plus the included base map objects).

2. Besides dividing a problem, it is also important to merge these sub-problems again to one result. This is possible in the case of agent technology based on the agent's identity and the agent's location.

3. The generalization tasks can run in parallel, as they are configured as atomic and do not interfere with other tasks. From a Grid Computing perspective, both aspects are considered to be crucial to use the Grid infrastructure efficiently.

4. As the generalization system runs multiple iterations to find the most applicable solutions, the generalization system submits these small generalization tasks to the Grid infrastructure and thereby benefits from the processing power of the grid multiple times.

Applying Grid Computing leads to an architecture for a gridified agent-based system for automated generalization (Figure 4.9). Each agent creates a specific generalization task and submits it as a process job to the Grid infrastructure. The created task, consisting of process (executable code) and data (the parameters), is then handled by a grid node and the result is returned to the generalization system. According to the agent cycle the Grid infrastructure is configured by many tasks at the same time and used iteratively until all the agents have reached the most satisfying state. During the execution of the generalization task on the grid, the agents are not able to communicate with each other. The evaluation of the generalization result and the communication of the agents is implemented inside the generalization system.

For the case of physical planning maps, the physical planning objects provide a partition of the base map. This partition of the base map is used to set up the agent hierarchy consisting of meso agents (defined by the extent of the physical planning object) and the micro agents (defined by the base map objects). A generalization task consists of meso agents representing the physical plan and the underlying base map objects represented by micro agents. This task is then submitted to the Grid infrastructure and processed as described above.

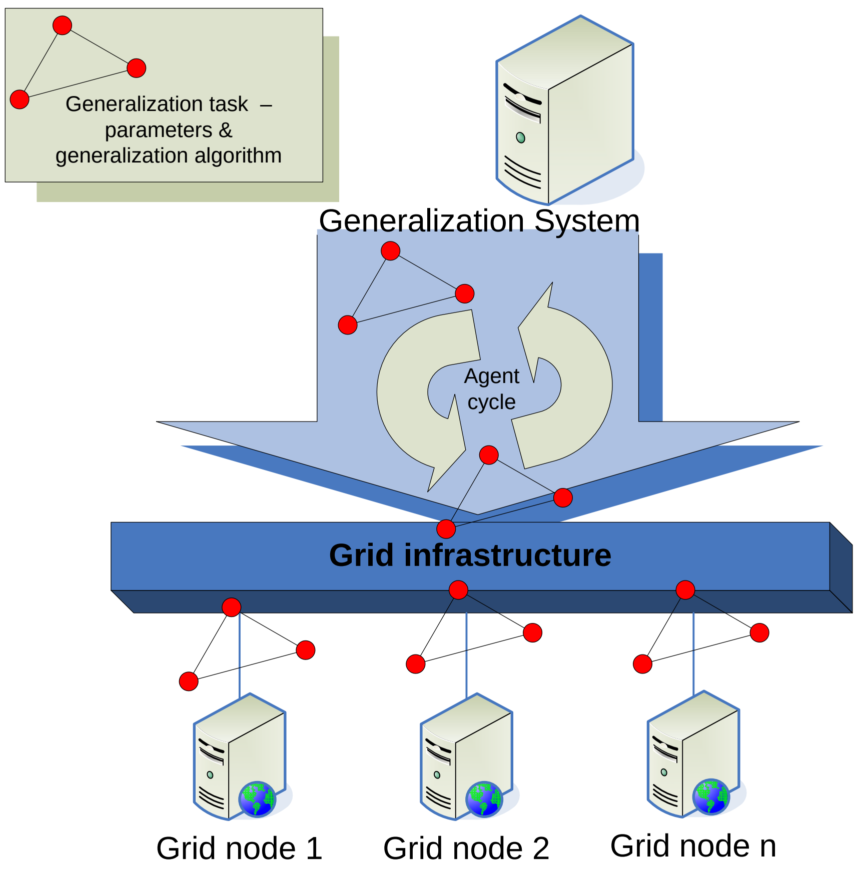


Figure 4.9: Architecture enabling Grid Computing for agent-based generalization.

The presented concept of enabling Grid Computing for agent-based generalization has not been implemented during this research. Further work is required in the case of 1Spatial Clarity to create multiple generalization tasks at one time inside 1Spatial Clarity and to run them in parallel on the Grid infrastructure.

Besides incorporating Grid Computing into the generalization-enabled WMS to improve the performance of the agent-based generalization process, it might also be applicable to replace the agent-based approach with other generalization approaches (i.e. non optimization-based). Such approaches are promising to perform faster or to be at least more scalable. Examples of such processes have been implemented on the web already by Vries and Oosterom [190] using dynamic data structures (tGAP) or by Lehto and Sarjakoski [119] using batch-oriented generalization (implemented by XML transformations). As a consequence, the results of such approaches might lack of cartographic quality, as they for instance do not take the symbolization into account nor do they incorporate a concept for on-demand web mapping such as achieved

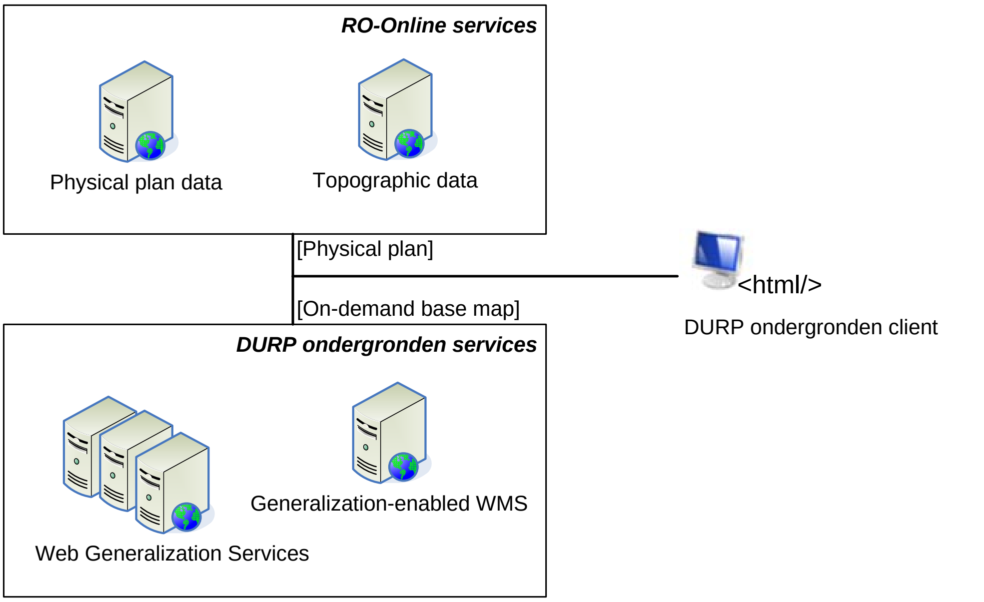


Figure 4.10: Integration of RO-Online and the developed architecture.

in this research by the user profiles.

Integrating the Architecture into RO-Online

The architecture of RO-Online is described in Section 2.6 (Figure 2.11). As RO-Online is based on Web Services, it is possible to integrate the services of the DURP ondergronden project into RO-Online conceptually, as examined by Foerster and Stoter [56] for this research. RO-Online services provide currently physical plan data and topographic data. In an integrated architecture (Figure 4.10), the DURP ondergronden service can access the RO-Online services to generate the on-demand base map. In particular, the DURP ondergronden services access the topographic data from which the generalization-enabled WMS generates the on-demand base map. Additionally, the generalization-enabled WMS retrieves the physical plan data to generate accordingly. The Web Services interoperate based on common standards (WFS and WMS). A WMS-compliant client overlays the physical plan (served by RO-Online) and the on-demand base map (served by the generalization-enabled WMS).

The integration of RO-Online and the DURP ondergronden services acts as a test case for integrating DURP ondergronden services into existing GIIs. The described integration could not be implemented during this research, as the RO-Online architecture is still under development and the necessary RO-Online services only provide limited data coverage due to missing physical plan data (based on the IMRO model). Future work has to verify such an integration. In this context, difficulties are expected especially regarding the performance of the integrated architecture. Thus incorporating Grid Computing as explained in the previous section should be considered. Additionally, it might be helpful to apply caching strategies, to limit the communication between RO-Online services and DURP ondergronden services. However, caching has to be handled with care to provide the latest data to the user as explained in Section 3.3.1.

Meaningful Integration of Web Services

Although the architecture is able to generate and disseminate on-demand base maps based on user profiles, it lacks of concepts to incorporate Web Services meaningfully. Web Services are beneficial for the architecture in two ways:

1. providing up-to-date data
2. providing additional generalization functionality in a distributed manner.

Meaningful integration of Web Services is achieved by semantic interoperability (Section 2.2.1). The semantic interoperability is based on meaningful description of the Web Services and does not only aim at automated integration of Web Services but also aims at integration based on human interaction. Human users also benefit from such meaningful descriptions as such descriptions improve the usability of Web Services. Although, human experts may be able to integrate Web Services meaningfully by consulting external sources (e.g. specifications, manuals or other human experts). The syntactic interoperability for especially Web Generalization Services has been described in Section 2.3.1.

To integrate data services meaningfully, their data has to be described semantically. This is achieved by ontologies for geographic data [4]. Such ontologies are then used to annotate the data services, as for instance presented by Klien et al. [106] and the application of the *Web Service Modeling Language* (WSML). In the given scenario of physical planning the ontologies could be extracted from the data model for physical planning IMRO and the data model for large-scale topography IMGeo (Section 2.6.1 and Section 2.6.2).

Enabling the meaningful description of remote generalization functionality (such as provided by Web Generalization Services) is also required, as it can be used by existing generalization systems to improve their functionality or distribute the processing effort and thereby improve the performance of the generalization process. The semantic description of such functionality however requires more conceptual modeling to integrate the data services meaningfully. What is already described in data models, such as the semantic knowledge and the structuring of the data, is not yet available for processes, this is true especially for generalization functionality. To integrate distributed generalization functionality a classification of generalization functionality is required, as exposed for instance by generalization operators. Until now no classification for generalization operators is available, which is approved by the generalization community, and consequently no formalization has been developed (Section 2.1.4). To integrate automated generalization meaningfully on the web, the following chapter presents a classification of generalization operators, its formalization and its application to Web Services.

4.4 Synopsis of the Implementation and the Evaluation

This chapter presents the implementation and the evaluation of the architecture based on the design as described in Chapter 3. The generalization-enabled WMS is realized on top of GeoServer accessing the generalization functionality of 1Spatial Clarity through the Clarity Datastore. The Clarity Datastore has been developed as part of this research. The implemented generalization process uses the AGENT model to realize the link between the thematic content and the base map. The map samples as generated by this architecture demonstrate how the different user requirements (captured in user profiles) change the base map. One of the advantages of the architecture is that generalization functionality is integrated into the existing WMS interface specification to provide on-demand base maps, but also to provide on-demand maps in general. The evaluation also encounters several limitations regarding a lack of performance, the missing integration into RO-Online and the missing meaningful integration of Web Services in general. The issue of meaningful integration of Web Services into the architecture is twofold: meaningful integration of data services and integration of Web Generalization Services. As this research aims at generalization and only little research has been accomplished (and described in literature), Chapter 5 will elaborate on this issue and present an approach to enhance meaningful generalization processing on the web. The developed approach includes the formalization of generalization (using OCL) and the meaningful description of these generalization operators on the web by the means of WPS profiles.

Chapter 5

Towards Meaningful Generalization Processing on the Web

One of the limitations encountered in the evaluation of the architecture (Section 4.3.3) and also reported by other researchers as explained in Section 2.3.1, is the lack of meaningful integration of Web Services. As described before to enable any meaningful integration of Web Services interoperability is crucial. Interoperability can be either achieved on a syntactic or semantic level (Section 2.2.1). Syntactic interoperability is a prerequisite to achieve semantic interoperability and has been demonstrated for the case of Web Generalization Services in Section 2.3.1. This chapter addresses the problem of meaningful generalization processing on the web (i.e. semantic interoperability), as this a key aspect of the architecture and only little research in this domain has been done.

In particular, this chapter will present a proposal to improve the semantic interoperability of Web Generalization Services by establishing a holistic classification of generalization functionality, classifying generalization functionality at an abstract level and at a more fine grain level of generalization operators. The presented classification is a new approach in research, as it provides a holistic view on generalization functionality and as it is based on standardized data models for defining the generalization operators. At an abstract level, a classification of so-called *Content Transformation Services* is proposed (Section 5.1). Content transformation combines the concepts of schema transformation [117] and generalization [196]. Moreover, to ensure the interoperability of generalization functionality on the level of generalization operators, Section 5.2 proposes a classification of generalization operators enhancing previous classifications (Section 2.1.4). The classification is formalized in OCL. Section 5.3 demonstrates the applicability of the formalized classification and shows how to publish such formalized generalization functionality on the web, by the example of ratio-based simplification. WPS profiles are used to describe such generalization functionality on the web as this research applies the WPS interface specification for syntactic interoperability (Section 2.2.4 and Section 2.3.1).

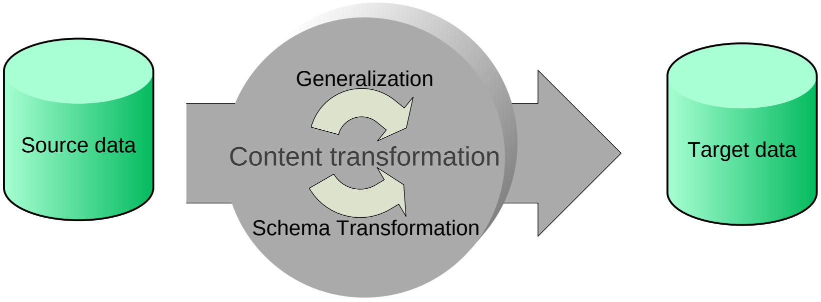


Figure 5.1: Concept of a content transformation process.

5.1 A Web Service Classification for Content Transformation Services

As explained in Section 2.3.1 and also as a result of the evaluation of the architecture (Section 4.3.3), semantic interoperability of Geoprocessing Services (Section 2.2.4) and Web Generalization Services (Section 2.3.1) in particular is still an unsolved issue. As a consequence, digital systems cannot interact with the web-based processes meaningfully as well as to a large part human users. In some cases experienced human users might be able to consult other users or other information sources (such as text books and manuals) to interact with the Web Service meaningfully. To achieve semantic interoperability, a service classification is required. An example of such a service classification is presented in ISO 19119 [95]. Also Lemmens [120] used service classifications to enable semantic interoperability.

To build a classification it is required to group geoprocesses according to their common characteristics. Two of the most important characteristics of geoprocesses are functionality and granularity. Consequently, based on service functionality and service granularity a classification for Content Transformation Services is proposed in this section.

Generalization is for this purpose categorized as a type of content transformation and provides a classification for Content Transformation Services. Related to content transformation is also the so-called schema transformation [118]. Schema transformation means to transform data from a source to a target model describing a different context. Instances of schema transformation are attribute renaming and coordinate transformation. Generalization is dedicated to the transformation of data from a source into a target model regarding scale or level of detail and use. Schema transformation and generalization cover the complete content transformation. The relationship between schema transformation and generalization and how they match each other is depicted in Figure 5.1. The application of content transformation processes is especially relevant in the context of GIIs, in which several datasets coming from different data models have to be integrated into a common data model [199].

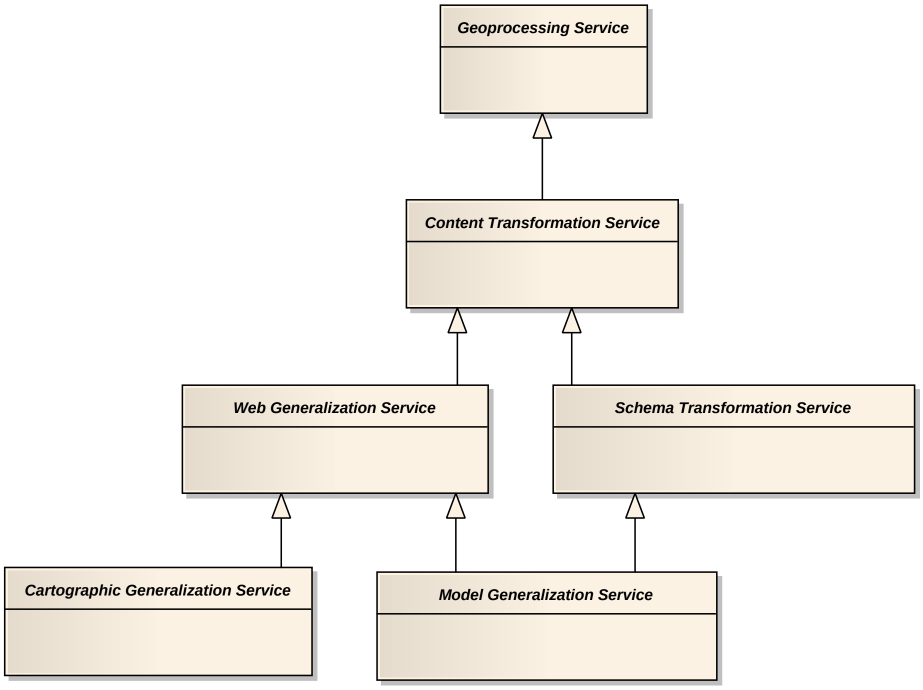


Figure 5.2: Classification of Content Transformation Services according to their functionality.

Service functionality

This refers to the type of functionality, which a specific Web Service provides to the client. An example of a classification of functionality is given by ISO 19119 [95]. Such information is crucial for the client (being human or computer application) to determine the semantic meaning of the specific process.

The model in Figure 5.2 describes Web Generalization Services and Schema Transformation Services according to their functionality. A Content Transformation Service is a type of Geoprocessing Service (Section 2.2.4). Generalization Services can be divided into Model Generalization Services and Cartographic Generalization Services, according to the model of Grünreich [75] (Section 2.1.1). However, the separation between model generalization and schema transformation is not strict due to the missing formalization of level of detail. Additionally, the operators for model generalization and schema transformation overlap. Some operators in schema transformation are applied in model generalization. Generalization is for instance applied by selection, which is also a basic operator in schema transformation. Thus, model generalization functionality can be modeled as a specialization of schema transformation.

Service granularity

This indicates the level of complexity of the exposed process, that is whether the process is implemented through a single service or is composed of a set of fine-grained services. The granularity is thereby seen from a service provider view point, not from a service consumer perspective, as pointed out by Haesen et al. [76]. Still, this information is interesting to clients, as it provides an insight about the complexity and provides thereby a sort of translucent service chaining (i.e. the consumer is aware of the services, which are involved in the specific service chain) (Section 2.2.1).

Apart from classifying Content Transformation Services according to their functionality, Content Transformation Services can be classified by their granularity. This was previously done for Web Generalization Services by Edwardes et al. [39], as shown in Section 2.3.1. Schema Transformation Services can also be divided according to their granularity into compound and operator services. Combining the two classifications leads to a matrix, presented in Table 5.1.

According to the matrix, a service can be classified as compound or operator service depending on its granularity, but it can also be classified according to the functionality it provides: model generalization, cartographic generalization and/or schema transformation. Furthermore, the operator services as proposed here might be classified in more detail by using the classifications of schema transformation operators [117] and generalization operators (see Section 5.2.1) to assign more meaning to the operator service. Consequently, an operator service providing *collapse* functionality for instance might be published in a more meaningful way through the additional classification of generalization operators. Thus, a Web Service providing collapse functionality might be described as an operator service (granularity) and as a model generalization service (functionality) providing collapse functionality formalized in an operator classification in Section 5.2.1.

5.2 A Formalized Classification of Generalization Operators

This section extends the abstract classification of Content Transformation Services, as presented in the previous section, in a more fine-grain manner, where generalization operators play a key role, as they are the building blocks of complex generalization processes. Thereby, a classification of generalization operators is essential to enable meaningful generalization processing on the web.

As pointed out in Section 2.1.4, a classification also contributes to the theory and general understanding of automated generalization.

Section 5.2.1 presents the proposal of a classification of generalization operators. This classification is not intended to be complete. However, the aim is to create a classification, which is extensible and may be adjusted in the future. The classification has been adopted during this research also to the WPS-compliant Web Generaliza-

		Granularity	
		Compound Service	Operator Service
Functionality	Model Generalization		
	Cartographic Generalization		
	Schema Transformation Service		

Table 5.1: Classification matrix for Content Transformation Services.

tion Service (Section 2.3.1). Based on the classification the different generalization operators are formalized (Section 5.2.2), to avoid any ambiguous interpretation of the specific generalization operator. The actual application of these formalized descriptions to a specific generalization algorithm is demonstrated by the example of Douglas-Peucker algorithm (Section 5.2.3).

5.2.1 Classification of Generalization Operators

The proposed classification of generalization operators is based on the model of Grünreich [75], see Section 2.1.1. This model has been chosen, as it provides a clear separation regarding the result, being either geographic data or a map.

Each of the operators has to be classified according to the model of Grünreich. The operators proposed in this research are based on literature review (Section 2.1.4). The basic criteria to decide about the operator's primary affiliation in the Grünreich model is, if it can be defined and applied homogeneously on a feature type or feature instance level. An operator is applied individually on the feature instance level, if a cartographic conflict has to be solved due to scale transition and the specific map generalization specification. Model generalization operators are applied globally upon a dataset. Cartographic generalization operators are guided by globally defined map generalization specifications, but are applied individually upon a single cartographic feature or a group of spatially related cartographic features. However, the impact of both operator types is always local. The cartographic generalization process is previously defined in Section 2.1.1 and depicted in Figure 2.2. According to the model of Grünreich, model generalization operators can be performed in advance to

Model generalization	Class Selection
	Reclassification
	Collapse
	Combination
	Simplification
	Amalgamation
Cartographic generalization	Enhancement
	Displacement
	Elimination
	Typification
	Enlargement
	Amalgamation

Figure 5.3: Classification of generalization operators.

perform cartographic generalization in a next step.

The proposed classification of generalization operators is depicted in Figure 5.3. It has to be noted that the classification is based on literature review and is not intended to be complete. It is formalized using standardized data models and is thereby comprehensible for future generalization research and possible standardization. This classification was applied as part of this research to the WebGen 2.0 framework as published in Foerster et al. [47]. Additionally, this classification was the foundation for the survey of generalization operators in practice [59] described in Section 2.1.5 and Appendix D. In the following, the generalization operators identified for the classification will be described in more detail according to the separation of model and cartographic generalization.

Model Generalization Operators

An example of each of the model generalization operators is depicted in Figure 5.4.

Class Selection

This operator selects the specific instances of a specific class, which should appear in the target data model. It also includes some filtering of the feature type properties according to the target data model (such as a database query). However it does not influence the feature type hierarchy such as reclassification. An example of class selection could be: select all the buildings, which have a minimum size and have a minimum distance to another building. It is important to note, that this class selection has no impact on the spatial attribute of the feature itself. This operator is called class selection to stress the difference to elimination (also known as selection),

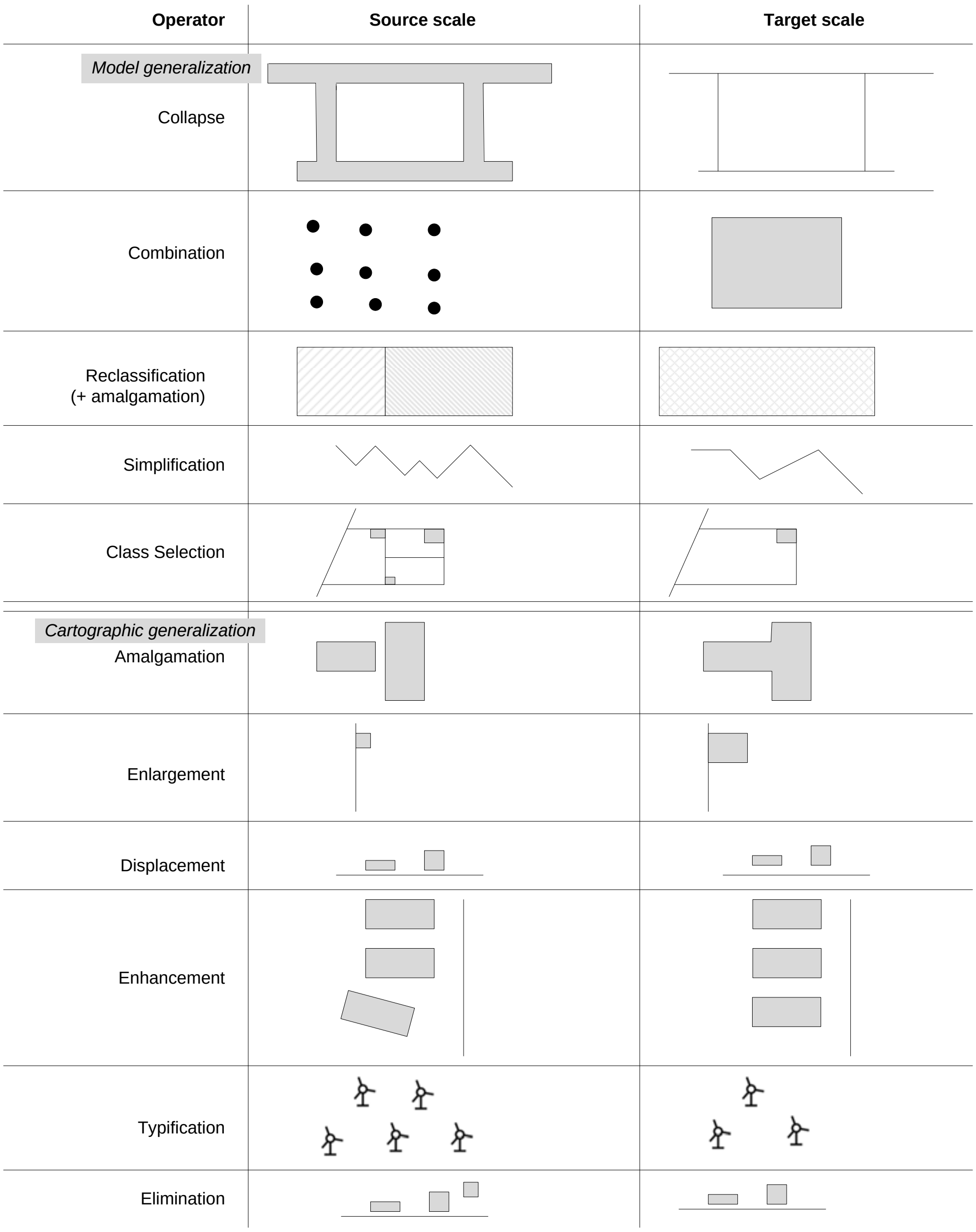


Figure 5.4: An exemplary overview of the effect of the generalization operators.

which is a cartographic generalization operator.

Reclassification

This is an elementary generalization operator, but it does not address spatial aspects (i.e. has no impact on the geometric attribute). However, it is an important operator, as it can cast certain features to become member of other feature types according to the target data model, based on spatial characteristics. Additionally, it can change the attributes of features according to the target model. Reclassification drives or is followed by operators such as amalgamation, combination and collapse, because they can reflect the reclassification also for the geometric attributes (amalgamation) and change the geometric attribute data type according to the target data model (combination and collapse). This operator has an equivalent in schema translation, if the reclassification is not based on or does not require any transformation of the geometric attribute data types. The example shows a reclassification of geological areas, which are reclassified from Holocene and Peistocene to Quarternary age. This reclassification goes in this case - as in most cases - along with amalgamation.

Collapse

This is a complex operator, which decreases the dimensionality of the geometric type of the feature type (i.e. collapses the geometry from polygon to line or to point). One of the most dominant applications for this operator is the collapse of road polygon geometries to road line geometries representing the road's center line.

Combination

Combining a group of features with lower dimensionality to one feature with higher dimensionality has a strong impact, which not only changes the geometric attribute data type, but also goes along with a change of the feature type as well. Combination is thereby the result of a reclassification, in which the geometric attribute data type of the object is changed. For example reclassification of sites of type Leisure (modeled as point) to feature type tourist attraction (modeled as area). As the geometric attribute data type of the feature has been changed, combination is involved. This operator is related to amalgamation, but it is more invasive, as it has to create a new geometric attribute data type based upon the original attribute data type. As in literature this research separates amalgamation and combination.

Simplification

This is an operator which is used to reduce the amount of data. As modeling might aim at reducing the data volume, this work suggests to keep it as a model generalization operator. Simplification is in most cases not invasive upon the feature, because it only deletes aspects of a geometry based on a certain criteria. However, simplification might lead to changes in topology. In the illustrated example, the geometry of a road is simplified.

Amalgamation

This is a special operator, as it can be applied globally upon feature type level for model generalization (e.g. amalgamating a group of buildings to a building block) and locally upon a group of cartographic features (cartographic generalization). It is about amalgamating a group of spatially adjacent (or in close proximity) geometries (of the same geometric type and member of the same feature type) - topologically connected or non-connected - into a single geometry. This operator constructs a new outline boundary for the new geometry. In the context of model generalization it mostly goes together with reclassification, as the geometric attribute should also reflect the applied classification. So for instance several adjacent forests of different type (e.g. coniferous and deciduous) are reclassified to forest area. It is necessary to amalgamate the geometries of the original features to a new geometry and assign this geometry to the reclassified feature. Amalgamation can be subdivided into fusion and merge. Fusion amalgamates connected geometries of a set of features to a new geometry. Features are connected, if their geometries touch. Contrarily, merge amalgamates non-touching geometries to a new geometry. So those features are only located to each other in proximity. The example illustrates a dominant case, the amalgamation of buildings. More specifically this example shows merge, as both geometries are detached originally.

Cartographic Generalization Operators

An example for each of the cartographic generalization operators is depicted in Figure 5.4.

Enhancement

This operator modifies specific geometric parts of a cartographic feature to produce a more pleasing representation or to emphasize an object. This includes smoothing of lines or squaring of area features (e.g. buildings). So on an object-level such enhancements modify specific parts of the geometry (i.e. set of coordinates). In the presented example a cartographic feature representing a building has been enhanced to fit its neighboring cartographic features by squaring and rotating it.

Displacement

This operator moves the complete cartographic feature by applying the same offset to each part of the cartographic feature. The final result is an object with a changed location but still preserving the original shape, also in absolute terms. The example illustrates that the houses are displaced from their origin to be better readable on the target map. The deformation of designated parts of a feature, as displacing parts of line, is addressed by the enhancement operator.

Elimination

The operator removes the cartographic feature from the map display. This operator is somehow the equivalent of the class selection operator, as both operators result in a set with a reduced number of objects. However their level of definition and application is different. Elimination is performed upon a feature instance level and not on a global level such as class selection. In the example one of the objects has a low significance on the target scale map, thus it has been eliminated.

Typification

This is a complex operator, which replaces a set of cartographic features with a smaller set of cartographic features. The operator has to determine the applicable set of new cartographic features and then arrange them by preserving the original pattern. This operator is not atomic in the original sense, but it is impossible to separate the operator in an appropriate way, as the actions applied are highly depending on each other and have to be performed as a whole. Also generalization literature has identified this operator as one. In the illustrated example a group of windmills is represented by a smaller group of windmills to preserve readability.

Enlargement

This operator preserves the shape and the original character of the cartographic feature while scaling it up. This ensures that the cartographic feature is not being deleted on the target map. In the example, a building is enlarged to be preserved on the target scale of generalized map.

5.2.2 Formalized Generalization Operators

To define the different operators unambiguously, each operator type is specified based on a specific data model to describe input and output but also to describe its internal functionality. To do this, the General Feature Model [93] serves as a basis to define model generalization operators and the OGC GO-1 Application Objects model [133] serves as a basis to define cartographic generalization operators. These models enable one to define the abstract functionality of the generalization operators by describing their impact. These models have been chosen, as they provide a standardized view on geographic data and visual primitives, which represent the cartographic features.

An Object-oriented Model for Generalization Operators

The foundation to formalize the generalization operators is an object oriented model described in a UML class diagram reflecting the generalization operator classification as it has been described before. To represent the basic separation of model and cartographic generalization operators, the model specifies two interfaces `ModelGeneralizationOperator` and `CartographicGeneralizationOperator` (Figure 5.5). Interfaces define the methods (i.e. the syntax), which have to be implemented by the subclasses. In an appropriate design, all classes implementing the same interface share the same semantics regarding the inherited methods. Both interfaces have the method `process()`. Depending on the impact of the operator, the method has one input parameter representing a set of features (General Feature Model) for model generalization or a set of cartographic features (OGC GO-1 Application Objects model) for cartographic generalization. The features based on the General Feature Model differ from the cartographic features in a way, that they do not include symbolization. Their boundaries differ from boundaries resulting from cartographic symbolization. As an example, a symbolized line feature on the map already appears as an area, which has specific implications for the generalization process. The cartographic features are always linked to their source features (General Feature Model) from which they are created. During the cartographic generalization process, these source features are not affected.

The actual generalization operators have been designed as separate abstract classes and not as interfaces because interfaces do not allow one to specify any functionality. Each operator implements one of the interfaces regarding its affiliation to model or cartographic generalization. A special case of inheritance is Amalgamation, which implements both interfaces, as it might apply to both model and cartographic generalization. Based on the abstract classes, each algorithm is able to inherit from a specific operator.

It is important to note that the model does not specify any parameters apart from the parameters representing the set (i.e. General Feature Model or GO-1 Application Objects). Parameters are dependent on the specific algorithm and its implementation. To show an example Section 5.2.3 illustrates how to handle algorithm-specific

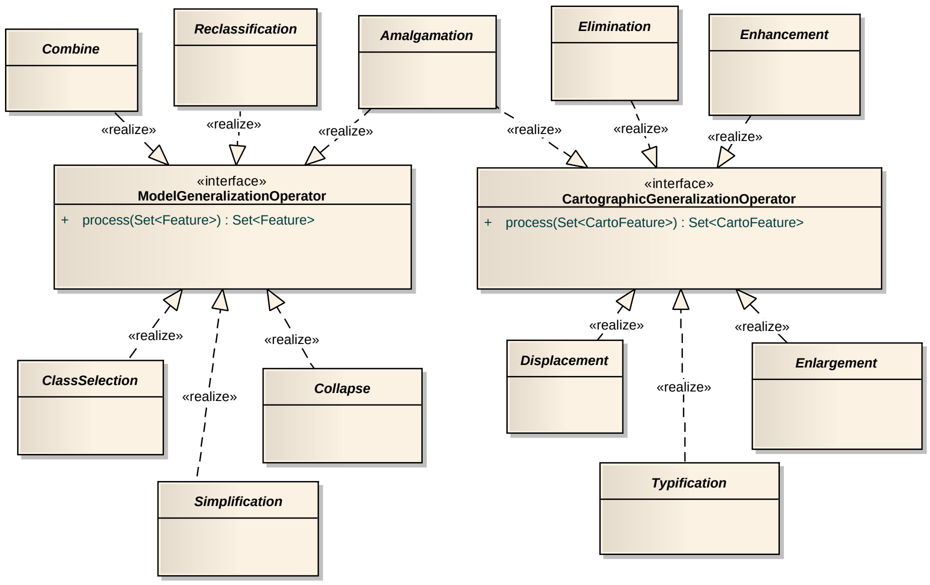


Figure 5.5: UML class diagram of the generalization operators.

parameters.

The abstract functionality of the operators is specified via pre and post conditions. The generalization algorithms, which implement the generalization operators, are then specified via the body OCL expression to represent the functionality. It depends on the specific system, how the algorithm will be coded.

A Formal Representation of the Generalization Operators in OCL

Based on the model of operators depicted in Figure 5.5 this section describes prominent examples of generalization operators formally in OCL to illustrate the formalization approach. Details about the syntax of OCL can be found in Warmer and Kleppe [194]. The syntax of the OCL statements presented here has been validated using ArgoUML.

Model Generalization Operators in OCL

As explained before, the formalization of the model generalization operators and their effect on the features is based on the General Feature Model.

Collapse The functionality of collapse is defined in Listing 5.1 by a pre and a post condition. The pre condition ensures that the features passed to the operator contain a collapsible geometry, that is a line or polygon geometry. The post condition defines that the resulting geometry must have a lower dimensionality with respect to the initial geometry.

Listing 5.1: OCL description of the Collapse operator.

```
Context  Collapse::process(fc: Set<Feature>) : Set<Feature>
  pre:   fc->forAll(Feature f | geometry.dimension>=1)
  post:  result->forAll(Feature f | f.geometry.dimension < fc.getByFid(f.fid).
       geometry.dimension)
```

Simplification This operator is used to reduce the amount of data. It deletes specific vertices (point members) of a geometry, but it does not reposition the geometry nor does it add any vertices.

Listing 5.2 defines a pre and a post condition. The pre condition specifies that all features consist of a geometry, which is at least of type linestring. The post conditions define that coordinates should not be changed or added to the geometry. Therefore, a `changeAllowed` attribute is defined for the `Coordinate` (`changeAllowed=false`) and an `addAllowed` attribute is defined for the `Geometry` (`addAllowed=false`). Additionally, the post conditions define that the number of coordinates has to be decreased. The OCL statements refer to `Geometry` and `Coordinate`, both are taken from the General Feature Model.

Listing 5.2: OCL description of the Simplification operator.

```
Context  Simplification::process(fc: Set<Feature>) : Set<Feature>
  pre:   fc->forAll(Feature f|f.geometry.dimension>=1)

  post:  result->forAll(Feature f | f.geometry.nGeometry()< fc.getByFid(f.fid)
       .geometry.nGeometry())
         fc->forAll(Feature f|f.geometry.addAllowed = false)
         fc->forAll(Feature f|f.geometry->forAll(Coordinate c| c.changeAllowed
              = false))

Context  Coordinate
  def:   changeAllowed: Boolean = false

Context  Geometry
  def:   addAllowed: Boolean = false
```

Cartographic Generalization Operators in OCL

As explained before, the formalization of the cartographic generalization operators and their effect on the cartographic features is based on the OGC GO-1 Application Objects model.

Displacement This operator moves the complete cartographic object by applying the same offset to each part of the object. The final result is an object with a changed location but still preserving the original shape, also in absolute terms. Note that the deformation of designated parts of a cartographic feature, as displacing parts of a line is addressed by enhancement. Listing 5.3 defines for displacement a post condition that determines an offset for an arbitrary coordinate pair in the result geometry. This offset is then compared for all coordinate pairs and should remain constant.

Listing 5.3: OCL description of the Displacement operator.

```
Context Displacement::process(fc :Set<CartoFeature>) : Set<CartoFeature>
  post: let offset: TupleType(x: Real, y:Real)=TupleType{x: result-> first.
      geometry.x - fc.getByFidId(result->first.fid).geometry.x), y: result->
      first.geometry.y - fc.getByFidId(result->first.fid)result->first.
      geometry.y)}

   result->forAll(Feature f | f.geometry.x - result.getByFid(f.fid).geometry.
      x = offset.x and f.geometry.y - result.getByFid(f.fid).geometry.y =
      offset.y)
```

5.2.3 Example of a Formal Description of the Douglas-Peucker Algorithm in OCL

To demonstrate how the OCL definitions (Section 5.2.2) can be used to specify the implementing algorithms this section introduces an OCL description of the Douglas-Peucker algorithm [34]. The Douglas-Peucker algorithm has been selected, as it is one of the well-known algorithms for generalization and as simplification is considered to be the most important operator in practice (see Appendix D.1.1 and Foerster et al. [59]) Following the idea of generalization operators and generalization algorithms, the Douglas-Peucker algorithm, being a simplification algorithm, inherits from the Simplification class (Figure 5.6).

According to the principle of inheritance, all the pre and post conditions defined for Simplification are valid for any subclass such as the Douglas-Peucker algorithm. The Douglas-Peucker algorithm requires a threshold parameter for selecting the appropriate points for simplification. In particular, the algorithm works recursively, by at first finding the most significant point of a line and then subdividing the line at that point into two lines, which are then processed accordingly. The point is selected for being kept in the final result if the distance between the line connecting the two endpoints of the line exceeds the specific threshold. Saalfeld [161] presents a thorough analysis of the Douglas-Peucker algorithm and proposes a methodology based on convex hulls to generate topologically consistent results.

Listing 5.4 describes the Douglas-Peucker algorithm in OCL. Based on its recursive manner, the algorithm calls itself several times. To cater for this, `process()` has a call to an atomic function called `simplifyGeom()`. This does not affect the conditions defined in `process()` of the superclass.

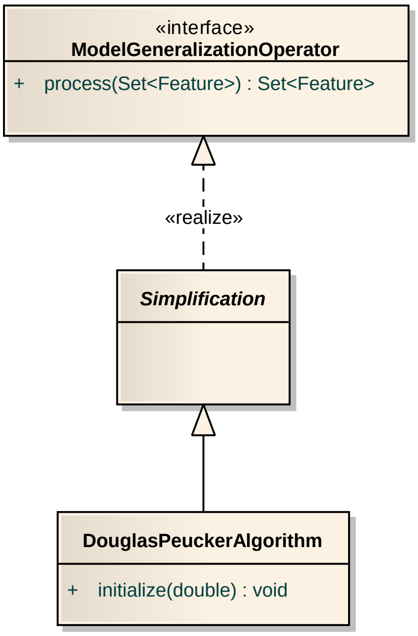


Figure 5.6: UML model of Douglas-Peucker algorithm inheriting the simplification operator.

Listing 5.4: OCL description of the Douglas-Peucker algorithm.

```
Context DouglasPeuckerAlgorithm::initialize(dpTolerance: double): Set<Feature
    >
  inv:  dpTolerance > 0
    self.dpTolerance = dpTolerance

Context DouglasPeuckerAlgorithm::process(fc: Set<Feature>)
  body: result = fc->forAll(Feature f|
                           let resultGeom: Geometry
                           resultGeom.add(f.geometry.first)
                           resultGeom.add(Geometry(simplifyGeom(f.geometry)))
                           resultGeom.add(f.geometry.last)
                           f.setGeom(resultGeom))

Context DouglasPeuckerAlgorithm
  def:  simplifyGeom(geom:Geometry) Set(Coordinate)
  body:
    if(geom.size >2)
      then
      let cand: Geometry = geom.coordinates->sortBy(c.distance(Line(geom.
          first,geom.last)))->at(1)
      if(cand.distance(Line(geom.first,geom.last))>dpTolerance
        then
        let coords: Set(Coordinate)
        coords.add(cand)
        coords->prepend(simplifyGeom(geom(first,cand))
        coords->append(simplifyGeom(geom(cand,last)))
        result = coords
      endif
    endif
```

5.3 Formalized Generalization Operators on the Web

To use the formalized operators on the web, the formalized descriptions have to be web-accessible. A possible approach is to convert the OCL descriptions into a semantically-rich service description, which can be used by clients to perform the specific function meaningfully. Timm and Gannod [183] describe such a conversion of OCL descriptions into OWL-S (i.e. a semantically-rich service description). However, in the context of Geo Web Services, semantic service descriptions (such as OWL-S) are still subject to research. Also in the broad world of mainstream-IT a de-facto standard for semantic service descriptions does not exist, yet [168]. The tool support for such descriptions is prototypical. Moreover, in the context of "Geospatial Web Services", only a few efforts have been reported which resulted in sufficient and successful semantic descriptions, such as the SWING project [157].

Currently a lot of data models and software are developed in UML and become part of international standards (Section 2.4.1). As OCL is a part of the UML standard and is considered to enrich the semantic meaning of these UML models, it is promising to integrate OCL and UML into semantic Web Service descriptions. This direct integration of OCL and UML into Web Service descriptions does not require any translation, which might otherwise cause semantic gaps. Such a seamless integration of models reduces errors and improves thereby the semantic interoperability. This work proposes to expose the generalization functionality using the export format for UML models XMI (Section 2.4.3) and WPS profiles (Section 2.2.4).

As explained in Section 2.1.4, generalization operators provide already an abstraction level for the semantics. A specific operator describes the functionality of the associated algorithms on an abstract level. From a syntactic perspective and also a more detailed semantic view however, the algorithms are totally hetereogenic regarding a specific operator (difference in input and output parameters and difference in the generalization functionality). Consider for instance the simplification operator, as described before. The Visvalingam-Whyatt algorithm [189] and the Douglas-Peucker algorithm [34] can be considered as prominent examples of the simplification operator. They both implement the same functionality (i.e. simplification), but their interfaces differ. They especially differ on a semantic level. The input parameters of both algorithms have a different meaning and a total different effect on the result. Thus, it is necessary to find a common syntactic and semantic abstraction for the interface to enable an interchangable use of the algorithms on the web. Such a common abstraction can be specified through WPS profiles. WPS profiles are chosen for this research, as they enable to define semantics for Geoprocessing Services in a OGC-standard compliant way (i.e. WPS). This research has already identfied WPS as a suitable way to publish generalization functionality on the web syntactially (Section 2.3.1).

This section designs a common interface for the instance of the simplification operator (Section 5.3.1). Based on the design, this common interface is implemented using WPS profiles and OCL (Section 5.3.2). Section 5.3.3 provides a conclusion of the presented approach. *The example of simplification has been chosen, as it is considered* *the most important operator for generalization (see Appendix* D.1.1 and Foerster et al. [59]).

5.3.1 Design of a Formalized Generalization Operator on the Web

To enable meaningful generalization processing on the web on the basis of generalization operators, this section presents the design of a formalized generalization operator by the example of *ratio-based simplification.* Ratio-based simplification is considered as a common measure for simplification and has been successfully demonstrated by Foerster et al. [58] as part of this research.

The simplification ratio is defined as the ratio between the number of vertices (i.e. number of intermediate points plus start and end point) in the dataset before and after simplification. This ratio is comparable to the *Radical law* of Töpfer [184] as both quantify the aspect of generalization. The Radical law describes the number of objects which should appear on a derived scale. In an extended scenario the Radical law may serve as a guideline for configuring the proposed simplification ratio.

The simplification ratio can be applied to several algorithms implementing the simplification operator (even with different syntax), such as the Douglas-Peucker algorithm and the Visvalingam-Whyatt algorithm. The simplification ratio is thereby suitable to be modeled as a common interface for the simplification operator. It can be converted automatically to the corresponding parameter value of the algorithm (by comparing the amount of data of the original with the result). Foerster et al. [58] reported this observation based on the experience of implementing the ratio-based simplification prototypically.

The ratio-based simplification can be described in OCL as shown in Listing 5.5. The ratio-based simplification is defined by a post condition, describing that the ratio of points (intermediate and end and start points) of input and output has to meet the configured ratio of the input. As described in the listing, the ratio is defined as a measure for the points of all the features before and after the simplification. This increases the freedom of the algorithm to achieve the result.

Listing 5.5: OCL description for ratio-based simplification.

```
Context ratio-basedSimplification::initialize(ratio: double): Set<Feature>
  inv: ratio > 0
    self.ratio = ratio

Context ratio-basedSimplification::process(fc: Set<Feature>)
  post: result.features->sum(geometries->sum(coordinates.size))/fc->sum(
      geometries->sum(coordinates.size)) = ratio
```

As the ratio-based simplification provides a common interface for algorithms, which provide simplification functionality, this interface is also applicable to Web Generalization Services and WPS respectively. This common interface is used to implement semantic interoperability based on WPS profiles.

5.3.2 Implementation of the Formalized Generalization Operator through WPS Profiles

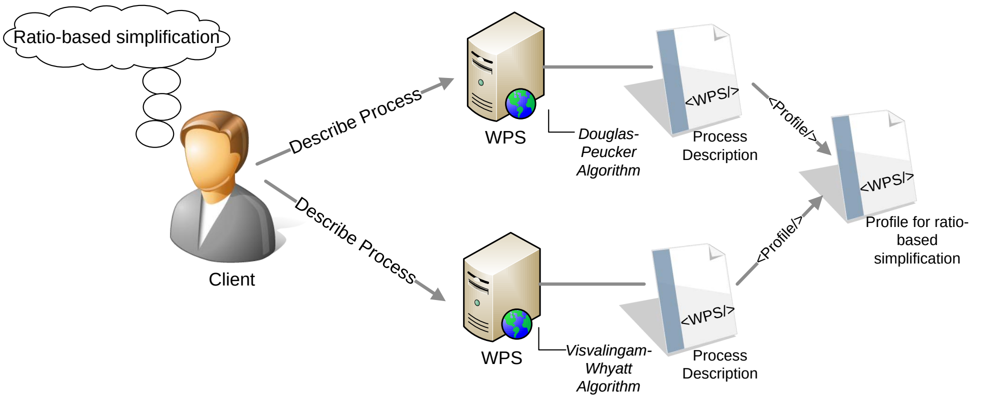
Based on the design of the common interface using a simplification ratio, it is possible to develop a WPS profile (explained in Section 2.2.4 for ratio-based simplification (Appendix B.2). The WPS profile describes two input parameters. One input parameter specifies the ratio-based measure and the other input parameter specifies the data to be processed. Different service providers are able to reference this WPS profile in their process descriptions. An example of such a process description referencing a WPS Profile is shown in Appendix B.1. The client is able to discover and interact with any service referencing this WPS profile for ratio-based simplification meaningfully. The meaning is implied by the referenced WPS profile. The services are distributed on the web, provided by different entities and based on different algorithms (e.g. Douglas-Peucker algorithm and Visvalingam-Whyatt algorithm). However, based on the referenced profile, the client can identify them as equal and select the most appropriate service, by testing such as described in Foerster et al. [58]. Based on the WPS profile, also an intelligent framework is able to automatically select the most appropriate service by running such tests. This is currently possible, based on the WPS interface specification.

In a more extended scenario, the WPS profile references the OCL description serialized in XMI, which specifies the ratio-based simplification, as explained in the previous section (Listing 5.5). Appendix B.3 portrays such a WPS profile, which links the XMI document with the OCL description for a more enriched description. In particular, the XMI document with the OCL description is linked in the metadata element of the WPS profile. Based on this referenced XMI document any client can access the OCL description and apply reasoning.

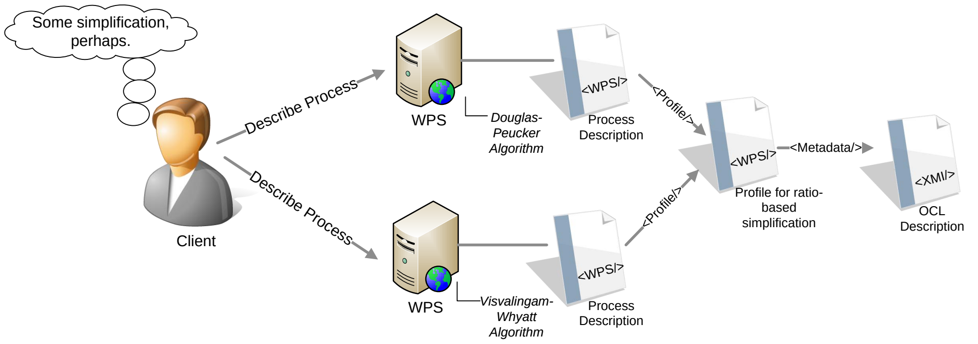
Thus, this research identifies two levels of complexity how to apply WPS profiles for Web Generalization Services and how to enable an interoperable architecture for ratio-based simplification (Figure 5.7). The client seeks for simplification functionality provided through WPS interface on the web. In case of a real world application, the client interacts with a central service registry (such as the OGC Catalog Service), but this component is left out for simplicity reasons.

- Simple scenario (Figure 5.7a) - The client is aware of the concept of ratio-based simplification and checks whether the process references the WPS profile for ratio-based simplification. The client configures the process based on the prior knowledge. This knowledge is coded (in case of automated discovery) or is part of human knowledge (in case of manual discovery).

- Extended scenario (Figure 5.7b) - The client (having probably little knowledge about generalization) discovers the algorithm, inspects the WPS profile and consumes the attached XMI document, describing the ratio-based simplification. Based on this description, the client is aware of the meaning of the algorithm and is able to apply reasoning and to configure the process accordingly.



(a) Simple scenario (no OCL), client requires knowledge about ratio-based simplification.



(b) Extended scenario (with OCL), client requires less knowledge about generalization, but has to understand OCL descriptions.

Figure 5.7: An interoperable architecture for ratio-based simplification based on WPS profiles.

This research claims that referencing an OCL description or any semantic description directly in a process description is not sufficient. WPS profiles and OCL descriptions need to be used complementary to each other, as the client might not be able to reason over a specific OCL description (e.g. due to time constraints or missing reasoning capabilities), but is aware of the WPS profile and its implied meanings. Additionally, WPS profiles are still beneficial for service providers, as they do not need to create their own semantic description (e.g. based on OCL) which is an extensive and error-prone task. Thus, using WPS profiles and OCL descriptions as explained in the extended scenario is promising to enable an interoperable architecture without sacrificing the flexibility.

5.3.3 Reflection on Formalized Generalization Operators on the Web

Formalized generalization operators are essential for meaningful generalization processing on the web. The WPS interface specification has been identified as a starting point and also implemented as part of this research to publish generalization functionality on the web (Section 2.3.1). The semantic interoperability is achieved by WPS profiles, but requires the design of a common interface for specific generalization functionality. Generalization functionality is grouped through generalization operators, which are thereby the foundation for semantic interoperability, but do not define common interfaces. Thus, it is necessary to design a common interface such as the presented ratio-based simplification interface. The definition of ratio-based simplification in a WPS profile allows the user to identify common functionality provided by different Web Generalization Services. However, this simple scenario requires prior knowledge of the human user or the digital system to apply the functionality meaningfully. To achieve meaningful generalization processing, this simple scenario has been extended by attaching formal descriptions, which specify meaningfully the generalization functionality. In the implementation the ratio-based simplification has been formalized in OCL and encoded as XMI. The XMI document is then referenced by the WPS profile and can be used for further reasoning. A reasoning system consuming such WPS profiles and OCL statements is still required to apply the generalization functionality meaningfully. This research claims, that WPS profiles and formalized descriptions (e.g. by OCL) of web-based processes are beneficial to be used in a complementary way.

5.4 Synopsis of Meaningful Generalization Processing on the Web

Meaningful generalization processing on the web is a long standing challenge and has only received little attention by research (Section 2.3.1). Whereas the syntactic interoperability of Web Generalization has been presented in Section 2.3.1, semantic interoperability is still an unsolved issue. Missing semantic interoperability has also been encountered as one of the limitations of the architecture (Section 4.3.3) as it was shown by the implementation. To enable meaning between two entities, such as in web-based architectures, establishing a common classification is part of the solution. Therefore, a broad classification of Content Transformation Services is proposed. To enable the meaningful generalization processing on the level of generalization operators, a classification of generalization operators is presented. This classification is a proposal, which is not intended to be complete nor are the single generalization operators fully-distinct. The generalization operators are formalized in OCL to describe them further and to make the classification comprehensible for future researchers. These formalized descriptions are then deployed on the web using WPS profiles. As the generalization functionality on the web is based on algorithms, the interfaces of the algorithms are heterogeneous. Thus, the formalized operators are not applicable per-se. This requires to extract a common interface for a group of algorithms, which implement the same operator, as shown by the example of ratio-based simplification. Section 5.3.2 demonstrates two possibilities (i.e. with and without OCL) how to include the formalized ratio-based simplification using WPS profiles. It proposes a scalable solution regarding the reasoning effort for integrating this specific functionality meaningfully into a generalization system.

Chapter 6

Discussion and Outlook

This chapter summarizes the findings of the thesis by answering the research questions. Additionally, it discusses the findings of the thesis and provides recommendations for future work.

6.1 Answers to Research Questions

The thesis investigates the main research question of *"How can users integrate maps at different scales according to their demands obtained by automated generalization on the web?"*. To cover all aspect of this main question it has been subdivided into four research questions, which are answered in the following.

How can on-demand base maps be generated?

On-demand base maps are important for thematic map communication. The research has demonstrated, that on-demand base maps can be generated by automated generalization. These on-demand base maps are generated according to specific user requirements. The user requirements chosen for this research aim at the link between the thematic content and the base map. Based on the thematic content and the thematic interest of the user, the level of detail of the underlying base map objects is changed. The boundaries of the thematic objects might act as topological constraint during the generation of the base map. The design of the generalization process is based on the AGENT model to realize the link between the thematic content and the base map object. The implementation of the generalization process has shown, that 1Spatial Clarity can be configured to apply this link sufficiently. The designed generalization process is applied to the generation of on-demand base maps for physical planning. The use case of physical planning is defined by the DURP ondergronden project. The formalization of the user requirements, which are the input for this generalization process, are part of research question 2.

Key publications: Foerster et al. [53, 61].

How can these on-demand base maps be disseminated on the web?

This research proposes that on-demand maps can be disseminated sufficiently through the generalization-enabled WMS. The generalization-enabled WMS is designed based on the WMS interface specification to consume user requirements for the on-demand base map, which are formalized as so-called user profiles. These user profiles are a complementary concept to the already existing concepts of SLD and WMC documents. In this research, user profiles are designed for the special case of base maps. The user profile consists of a generalization matrix, a topology-awareness list and a map generalization specification. These components allow the user to specify the desired level of detail of the base map objects. The user profile serves as an input for the generalization process incorporated inside the generalization-enabled WMS. The generalization-enabled WMS is thereby able to generate the on-demand base map accordingly. It generates the base maps using distributed geographic data provided by WPS. The generalization process is conceptually able to also apply Web Generalization Services to perform remote generalization functionality.

For this research the generalization-enabled WMS is implemented using GeoServer and 1Spatial Clarity. The so-called Clarity Datastore links GeoServer and 1Spatial Clarity to generate the on-demand base map based on the specific generalization process (previous research question). The generalization-enabled WMS is intended to be applied in RO-Online, the GII for physical planning in the Netherlands.

Key publications: Foerster et al. [53, 61].

How can processes for automated generalization be established on the web?

Automated generalization on the web is established by Web Generalization Services. In this research, the WPS interface specification has been selected to implement Web Generalization Services. The WPS interface specification is part of the OGC Web Services family and provides special facilities for distributed geoprocessing on the web. It has been implemented as part of this research to establish automated generalization on the web. The implementation is published as open source through the 52°North initiative [2]. Different client applications have been developed, which are also available as open source software. The implementation showed great flexibility regarding the data encodings, as it has been applied to automated generalization as well as to mass market applications such as Google Earth. Additionally, the WPS interface specification has been adopted by the ICA Commission on Generalisation and Multiple Representation as the common approach to publish Web Generalization Services. They propose an extension of the WPS specification to meet the specific requirements of generalization processing such as special data encodings and specific information on generalization operators.

Key publications: Foerster et al. [47], Foerster and Schäffer [50], Foerster et al. [51], Foerster and Stoter [54].

How can interoperability for web-based generalization processing be improved?

Web-based generalization is realized through Web Generalization Services. The question of interoperability is thereby closely linked to the problem of interoperability of Web Services. Web Service interoperability is difficult to achieve due to the loosely-coupled and stateless nature of Web Services. Interoperability can be established on a syntactic and a semantic level. Regarding the syntactic interoperability of Web Generalization Services, the WPS interface specification has been identified as a suitable candidate, as shown in the previous research question. As demonstrated by the implementation and the evaluation of the architecture and also described by various researchers, semantic interoperability remains unsolved. Still it is required to achieve meaningful generalization processing. For this research the semantic interoperability of Web Generalization Services is addressed in two ways. Firstly, defining a general classification of Content Transformation Services, which provides a higher-level concept to generalization. Secondly, by developing a classification of generalization operators. Generalization operators are one of the key aspects in generalization processing. This research proposes a classification of generalization operators based on the model of Grünreich and official standards for geographic data (General Feature Model and OGC GO-1 Application Objects). This classification of generalization operators is formalized in OCL. To describe specific generalization functionality based on the formalized operators, it requires a common interface of input and output parameters for the specific generalization operator. To illustrate this, the ratio-based simplification is designed as a common interface for the simplification operator. The interface of ratio-based simplification is implemented for meaningful generalization processing on the web on two different levels of complexity. In the simple scenario the interface of ratio-based simplification is specified by a WPS profile, which is referenced by specific generalization algorithms on the web, implementing this interface. In the extended scenario the WPS profile for ratio-based simplification is enhanced with additional metadata pointing to XMI document, which formalizes the interface in OCL. Based on the XMI document, the client can inspect the functionality of the process more thoroughly and does not need to know the meaning of the profile in advance. In the simple scenario the client can only determine if two provided processes are equal syntactically and semantically, but cannot apply any further reasoning based on the description. The different scenarios show that OCL and WPS profiles can be used in a complementary way.

Key publications: Foerster et al. [48, 49, 57].

6.2 Contribution

This thesis contributes to research on on-demand web mapping, automated generalization and web-based geoprocessing in general.

In particular, the proposed user profile as an established approach in mainstream IT can be considered as a new aspect in on-demand web mapping, since it defines the user requirements regarding the desired level of detail of a map, more particularly of a base map. The user profiles can be used in a complementary way to the existing concepts (WMC and SLD) for on-demand web mapping. The generalization-enabled WMS has been designed to consume the user profiles and to perform the generalization of the base map accordingly.

Regarding automated generalization, the designed process showed that the AGENT model is also applicable for designing and implementing specific links between the thematic content and the base map to perform the generalization according to the user requirements. Additionally, this research contributes to the theory of automated generalization by proposing a classification of generalization operators, which has been formalized in OCL.

Regarding the contribution to web-based geoprocessing, this classification is deployed through WPS profiles to ensure semantic interoperability of Geoprocessing Services and Web Generalization Services respectively. The combination of WPS profiles and OCL (encoded in XMI) is not only a new approach in the context of Web Generalization Services, but also a new approach for Geoprocessing Services in general.

All the developments regarding the WPS interface specification established in this research are available publicly through 52°North as open source software. The developed framework is currently maintained and extended by the Geoprocessing Community [1]. The implementation has already supported the development of several other applications and is the basis for further research in the context of real-time and distributed geoprocessing [19].

6.3 Discussion

This section discusses the limitations of the research regarding the two main issues addressed in this thesis, identifying topics for further research.

6.3.1 User Profiles and Generalization-enabled WMS

From a conceptual perspective, the user profiles in this research capture the user requirements regarding the level of generalization of a map. For this thesis, the user profile has been designed for the special case of on-demand base maps. The incorporated concepts (generalization matrix, map generalization specification and topology awareness list) are not based on thorough usability research, due to the technological focus of this research.

The execution of the automated generalization process has been incorporated in the generalization-enabled WMS, which is able to consume the user profiles through an extension of the WMS interface. Consequently, any WMS-client, which needs to access the generalization-enabled WMS to take advantage of the incorporated generalization functionality has to be extended. Extending specific service interfaces by so-called vendor-specific parameter is permitted by the specification. Still this vendor-specific parameter can be considered as a drawback, as it harms the interoperability of the architecture.

Further, the incorporated generalization process requires a lot of processing time considering the amount of objects. For 100 features the system took 30 seconds of processing time. Additionally, the generalization system did not scale well, when even more objects needed to be processed. This is caused by the applied optimization approach for cartographic generalization (e.g. agent-based generalization).

1Spatial Clarity was not able to connect to remote sources such as WFS, thus for the proof-of-concept, all the data had to be installed local. This is an obstacle in real world sencarios, when applying the presented approach to other data, which are mostly hosted remotely.

6.3.2 Semantic Interoperability of Web Generalization Services

The semantic interoperability for Web Generalization Services is ensured in the proposed architecture in two ways, firstly by classifying the generalization operators and secondly by deploying this classification as common interfaces through WPS Profiles and XMI-encoded OCL descriptions. The latter part can be accomplished in a simple scenario and a complex scenario. A proof-of-concept for demonstrating the semantic interoperability of Web Generalization Services based on the described approach is especially missing, as the OCL descriptions do not provide an unambiguous way of interpreting them. Thus, the kind of formalization provided by OCL is not all that complete. This thesis did not find a final proof in literature that OCL is suitable to formalize concepts strongly. However, this work claims, that OCL is still promising as it does not cause a model mismatch to UML, as it might be the case with other formalization mechanisms. Moreover, a strong formalization requires also a reasoning system which can make use of these formalizations. Such a reasoning system has been developed for instance by Lemmens [120]. But a reasoning system for automated generalization is still missing.

The concept of operators as applied in generalization research is not ready for formalization. As shown in this research, syntactic interoperability is required to provide semantic interoperability. However, the generalization operators do not specify any syntax. In this research this is solved by investigating common application of generalization operators and by extracting common interfaces. The applicability of this approach is shown for the example of ratio-based simplification. Overall, the concept of generalization operators has weaknesses regarding the formalization in a software system as it implies a high level of abstraction, which cannot be met by current modeling approaches.

6.4 Recommendations for Future Work

Based on the discussion in the previous section, some recommendations for future work can be given. The following items are identified as future work:

- Improve semantic interoperability of Web Generalization Services.
- Address the performance of the architecture
- Apply the presented architecture to RO-Online
- On-demand base maps for other types of thematic content
- Apply generalization also to the thematic content
- Standardize the user profile for web mapping
- Standardize the proposed classification of generalization operators.

These items are explained in the remainder of this section.

To improve the semantic interoperability of Web Generalization Services, further research has to be conducted to find more common interfaces for different generalization operators. Based on the results a generalization system might be enhanced to reason over the formalized descriptions of generalization functionality and finally to interact with these newly developed interfaces meaningfully. An agent-based system such as that incorporated in 1Spatial Clarity seems to be suitable as this already provides a mechanism to group generalization algorithms (in case of 1Spatial Clarity they are called actions). These actions could easily be attached to common interfaces and thereby be used by the agents inside the system. Finally, this would then enlargen the available generalization functionality and lead to more sophisticated results. The agent-based system would then only be used to plan and evaluate the generalization results, but the algorithms would be performed on the web in a distributed way.

Based on the experience of deploying the generalization-enabled WMS with real-world data, the performance of generating the map can be considered as an unsolved issue. In general, automated generalization is considered being a computational intensive task and performance considerations are outside the scope of this work. But further work in this direction is required especially looking at the application of on-demand web maps, at which different users might request several maps with different requirements at the same time. This study claims, that agent-based generalization is highly suitable to benefit from Grid Computing technology [98] looking at the requirements of Grid Computing and the capabilities of the agent-based system. Agent-based generalization provides a mechanism to divide and conquer the problem of generalization by the AGENT Model (i.e. meso and micro agents). This mechanism is the key requirement to apply Grid Computing technology successfully. As described in Foerster et al. [52], 1Spatial Clarity has to be enabled to execute the agents in parallel to take advantage of Grid Computing. In that case the generalization system becomes more scalable. Another solution might be to integrate other generalization approaches (batch-oriented or based on dynamic data structures), which are more scalable but which may lack of some cartographic quality.

To test the applicability of the proposed architecture and the proposed user requirements, an integration into the RO-Online architecture (Figure 2.11) is required. This integration is also a showcase to demonstrate how standards interoperate with each other and at which level. Again the lack of semantic interoperability will be encountered, especially on the data exchange level (i.e. what is a physical plan?). Thus, specific mechanisms, most likely implemented as ontologies [4], are required to ensure this semantic interoperability of data.

Regarding the DURP ondergronden project and the application of physical planning maps, the presented architecture might also be applied to other types of thematic maps. For instance soil maps might provide a similar notion as physical plans, as they also mostly consist of area objects. Further work needs to be carried out in terms of a technical study as well as in terms of usability research [40].

As the base map becomes an integral part of the map, especially in the context of multi-source cartography [143], the concept of fore and background of a map disappears. Therefore it might be also applicable to generalize the thematic content. This is especially interesting for physical plans, as the generalization of a large-scale plan could guide the planner to sketch plans on smaller scale. This would enable planners to create consistent plans over multiple scales and enhance the planning process. From the perspective on-demand web mapping, zooming (in or out) might also require the adjustment of the thematic content. Physical planning content might become unreadable on the map, thus automated generalization or explicit scale limits for specific plans may be incorporated in the architecture in the future.

To establish the user profile as a common approach to web mapping, standardizing this concept in a broader context with standard bodies such as OGC and ETSI is necessary. This would then give further proof of the approach presented in this thesis. Also the integration with mainstream-IT standards for mobile devices then becomes an issue. If the standardization of user profiles would mature, also the generalization-enabled WMS should be proposed as an extension of the existing WMS interface.

Finally, the proposed classification of generalization operators needs to be revised or extended to cover aspects, which have not been taken into consideration. This might be especially the case, if the data models of ISO and OGC used in this research are extended or applied to specific applications. Additionally, the classification only works, if there is a consensus by a large group coming from research and practice, which finally makes the classification a de-facto standard.

6.5 Synopsis of the Research

This thesis presents a web-based architecture for on-demand base maps. In particular, the research addresses the generation and dessimination of on-demand base maps on the web by the means of automated generalization and the meaningful integration of automated generalization functionality into the architecture. The on-demand base maps are generated for a fixed thematic content and depend on the user requirements, which are formulated by user profiles. These user-profiles are consumed by the generalization-enabled WMS, which performs the generation according to the user profile using its incorporated generalization system. The incorporated generalization system implements the agents-based approach and applies the AGENT model to generate the base map. The architecture is implemented using GeoServer and 1Spatial Clarity. The design and the implementation of the architecture are evaluated using the requirements identified in this research. Several limitations are identified, an important one being the lack of semantic interoperability to integrate meaningful generalization processing. Therefore, the thesis proposes a classification on the abstract level of Content Transformation Services and a more specific classification of generalization operators, which is formalized in OCL. The formalized generalization operators are then used to design common interfaces, which is demonstrated for the example ratio-based simplification. The designed interface is then implemented on the web meaningfully through WPS profiles.

Appendix A

Standards in Action

This section exemplifies the standards as applied in the thesis. The examples are meant to clarify the basic principles of the standards applied in this thesis. The examples stem from working implementations as applied in this research. The following standards are covered in this section: WPS, WMS, SLD and WMC.

Web Services communication is based on the Hypertext Transport Protocol (HTTP). The HTTP-message types used for Web Services are HTTP-GET and HTTP-POST. HTTP-GET contains the address of the server, a path and a query string. In case of HTTP-POST this query string is replaced by a message payload, which is encoded as XML.

A.1 WPS Walkthrough

The WPS interface specification is used in this research to realize Web Generalization Services (Section 2.3.1). The basic sequence of actions to discover and perform geoprocessing functionality hosted on WPS is GetCapabilities (HTTP-GET), DescribeProcess (HTTP-POST) and Execute (HTTP-POST) [136].

In this example a WPS instance is queried for a Douglas-Peucker algorithm [34] to simplify a set of road geometries as also applied in the mentioned risk management scenario in Section 2.3.1. The listed XML messages are in most cases generated by client applications, which guide the user during Web Services interaction by harvesting user input and performing the requests accordingly. Such a client application is presented in Section 2.3.1 and has been published as part of this research in [50, 166].

For reasons of simplicity it is assumed that the user knows the entry point of the WPS instance in advance. In real-world scenarios such entrypoints can be retrieved from catalog services. To get more information about the service the user queries the service metadata using a GetCapabilities request (Listing A.1). The response of the WPS instance is depicted in Listing A.2. From this response document the user can retrieve the service metadata, such as entrypoints for further communication (`OperationsMetadata`) or individual information about the provider (`ServiceProvider`). Also the provided processes are listed in the (`ProcessOfferings`). One of these processes listed is the Douglas-Peucker algorithm, which is briefly described with identifier, title and abstract. The identifier of the process can be used to retrieve further metadata.

Listing A.1: Example GetCapabilities request for WPS.

```
http://geoserver.itc.nl:8080/wps/WebProcessingService?REQUEST=GetCapabilities
    &Service=WPS
```

Listing A.2: Example GetCapabilities request for WPS.


```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-US"
    xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://geoserver.itc
    .nl:8080/wps/schemas/wps/1.0.0/wpsGetCapabilities_response.xsd"
    updateSequence="1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:wps="
    http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows
    /1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ows:ServiceIdentification>
    <ows:Title>My WPS</ows:Title>
    <ows:Abstract>Service based on the 52north implementation of WPS 1.0.0</
        ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>generalization</ows:Keyword>
      <ows:Keyword>geoprocessing</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    ...
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>52North</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://www.52north.org/"/>
    ...
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://geoserver.itc.nl:8080/wps/
              WebProcessingService"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="DescribeProcess">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://geoserver.itc.nl:8080/wps/
              WebProcessingService"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="Execute">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://geoserver.itc.nl:8080/wps/
              WebProcessingService"/>
          <ows:Post xlink:href="http://geoserver.itc.nl:8080/wps/
              WebProcessingService"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  </ows:OperationsMetadata>
  <wps:ProcessOfferings>
```

```
    <wps:Process wps:processVersion="2">
      <ows:Identifier>DouglasPeuckerAlgorithm</ows:Identifier>
      <ows:Title>douglasPeucker algorithm</ows:Title>
    </wps:Process>
    ...
  </wps:ProcessOfferings>
  ...
</wps:Capabilities>
```

DescribeProcess allows to access this metadata using the identifier of the designated process (e.g. `DouglasPeuckerAlgorithm`). The DescribeProcess request (Listing A.3) queries the WPS instance for further metadata on the specific process such as input and output parameters. This information is important to trigger the specific process appropriately. In the given example (Listing A.4), the Douglas-Peucker algorithm requires complex data for the geometries to be processed and literal data (of type double) to indicate the tolerance value the algorithm has to apply to the data.

Listing A.3: Example DescribeProcess request retrieving metadata about Douglas-Peucker algorithm.

```
http://geoserver.itc.nl:8080/wps/WebProcessingService?REQUEST=DescribeProcess
    &Service=WPS&Identifier=DouglasPeuckerAlgorithm
```

Listing A.4: Example DescribeProcess response describing the interface for the Douglas-Peucker algorithm.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ProcessDescriptions xmlns:ns="http://www.opengis.net/wps/1.0.0" xmlns:xsi
    ="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/
    wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="
    1.0.0"><ProcessDescription xmlns:wps="http://www.opengis.net/wps/1.0.0"
    xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org
    /1999/xlink" wps:processVersion="2" storeSupported="true" statusSupported
    ="false">
    <ows:Identifier>DouglasPeuckerAlgorithm</ows:Identifier>
    <ows:Title>douglasPeucker algorithm</ows:Title>
    <ows:Abstract>Uses JTS implementation. Does not support topological
        awareness</ows:Abstract>
    <ows:Metadata xlink:title="douglas_peucker"/>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>FEATURES</ows:Identifier>
        <ows:Title>input features</ows:Title>
        <ows:Abstract>Just features</ows:Abstract>
        <ComplexData>
        <Default>
          <Format>
            <MimeType>text/XML</MimeType>
            <Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</Schema>
          </Format>
        </Default>
        </ComplexData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>TOLERANCE</ows:Identifier>
        <ows:Title>Tolerance Value for DP Alg</ows:Title>
        <ows:Abstract/>
        <LiteralData>
        <ows:DataType ows:reference="xs:double"/>
        ...
        </LiteralData>
```

```
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>SIMPLIFIED_FEATURES</ows:Identifier>
        <ows:Title>smooth geometries</ows:Title>
        <ows:Abstract>GML stream describing the smooth feature.</ows:Abstract
            >
        <ComplexOutput>
        <Default>
        <Format>
          <MimeType>text/XML</MimeType>
          <Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</Schema>
        </Format>
        </Default>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</ns:ProcessDescriptions>
```

Based on these metadata, the client knows where (entry points in service metadata, Listing A.2) and how (process metadata, Listing A.4) to trigger the designated process. The client performs the Execute request (Listing A.5) with the designated parameters (geometries and tolerance value). The complex data (i.e. the geometries to be simplified) are included in the request as a reference to a WFS instance. The WPS has to retrieve the data from this location and process them accordingly. Finally, the WPS instance returns the simplified geometries (Listing A.5).

Listing A.5: Example Execute request for Douglas-Peucker algorithm.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/
     wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http:
     //www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
     instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://geoserver.itc.nl:8080/wps/schemas/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>DouglasPeuckerAlgorithm</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>FEATURES</ows:Identifier>
      <wps:Reference schema="http://schemas.opengis.net/gml/2.1.2/feature.xsd
          " xlink:href="http://geoserver.itc.nl:8080/geoserver/wfs?REQUEST=
          GetFeature&amp;typename=topp:states&amp;BBOX
          =-75.102613,40.212597,-72.361859,41.512517">
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>TOLERANCE</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>2</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
  <wps:ResponseDocument storeExecuteResponse="false">
    <wps:Output asReference="false">
      <ows:Identifier>SIMPLIFIED_FEATURES</ows:Identifier>
    </wps:Output>
  </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

Listing A.6: Example Execute response for Douglas-Peucker algorithm including process information and simplified geometries.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns:ExecuteResponse xmlns:ns="http://www.opengis.net/wps/1.0.0" xmlns:xsi="
    http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www
    .opengis.net/wps/1.0.0 http://geoserver.itc.nl:8080/wps/schemas/wps
    /1.0.0/wpsExecute_response.xsd" serviceInstance="http://localhost:8080/
    wps/WebProcessingService?SERVICE=GetCapabilities&amp;SERVICE=WPS"
    xml:lang="en-US" service="WPS" version="1.0.0">
  <ns:Process ns:processVersion="2">
    <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">
        DouglasPeuckerAlgorithm</ns1:Identifier>
  </ns:Process>
  <ns:Status creationTime="2009-11-16T17:24:14.809+01:00">
    <ns:ProcessSucceeded>The service succesfully processed the request.</
        ns:ProcessSucceeded>
  </ns:Status>
  <ns:ProcessOutputs>
    <ns:Output>
      <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">
          SIMPLIFIED_FEATURES</ns1:Identifier>
      <ows:Title xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="
          http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org
          /1999/xlink">smooth geometries</ows:Title>
      <ns:Data>
        <ns:ComplexData schema="http://schemas.opengis.net/gml/2.1.2/feature.
            xsd" mimeType="text/XML">
          <wfs:FeatureCollection xmlns="http://www.opengis.net/wfs" xmlns:gml
              ="http://www.opengis.net/gml" xmlns:states="http://www.
              openplans.org/topp" xmlns:wfs="http://www.opengis.net/wfs"
              xsi:schemaLocation="http://www.openplans.org/topp http://
              geoserver.itc.nl:8080/geoserver/wfs/DescribeFeatureType?
              typeName=topp:states http://www.opengis.net/wfs http://
              geoserver.itc.nl:8080/geoserver/schemas/wfs/1.0.0/WFS-basic.xsd
              ">
            <gml:boundedBy>
              <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326"
                  >
                <gml:coordinates cs="," decimal="." ts=" ">-80.5208,39.7195
                    -73.3451,45.0061</gml:coordinates>
              </gml:Box>
            </gml:boundedBy>
            <gml:featureMember>
              <states:states fid="states.39">
                <states:the_geom>
                  <gml:MultiPolygon srsName="http://www.opengis.net/gml/srs/
                      epsg.xml#4326">
                    <gml:polygonMember>
                      <gml:Polygon>
                        <gml:outerBoundaryIs>
                          <gml:LinearRing>
                            <gml:coordinates cs="," decimal="." ts=" ">
                                -79.7635,42.2673 -73.3451,45.0061
                                -74.0066,40.7039 -79.7635,42.2673</
                                gml:coordinates>
                          </gml:LinearRing>
                        </gml:outerBoundaryIs>
                      </gml:Polygon>
                    </gml:polygonMember>
                  </gml:MultiPolygon>
                </states:the_geom>
                ...
            </gml:featureMember>
          </wfs:FeatureCollection>
        </ns:ComplexData>
      </ns:Data>
    </ns:Output>
  </ns:ProcessOutputs>
```

```
</ns:ExecuteResponse>
```

This basic sequence of actions can be extended by requesting asynchronous processing or storing of process results on the server side.

The depicted request examples are based on 52°North WPS framework [1].

A.2 WMS Walkthrough

The WMS interface specification is used to enable on-demand maps on the web (i.e. generalization enabled WMS, Section 3.3 and Section 4.1). The basic sequence of actions to retrieve a map consists of performing the following operations: GetCapabilities and GetMap (both HTTP-GET) [132]. The given examples are based on the WMS interface specification version 1.1.1.

To get more information of the specific WMS instance, the user queries the service using the GetCapabilities operation (Listing A.7). The response of the WMS instance indicates the service metadata, including information about the provider and the layers, which are served by the specific instance (Listing A.8). A layer is available for a specific geographic extent and can be requested with specific styles (i.e. symbolization). The specific WMS GetCapabilities response describes that the WMS can serve Bestemmingsplan as a layer for the area of Enschede in the Netherlands.

Listing A.7: Example GetCapabilities request for WMS.

```
http://localhost:8080/geoserver/wms?REQUEST=GetCapabilities&SERVICE=WMS
```

Listing A.8: Example GetCapabilities response from WMS.


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://localhost:8080/geoserver/schemas
    /wms/1.1.1/WMS_MS_Capabilities.dtd">
<WMT_MS_Capabilities version="1.1.1" updateSequence="29">
  <Service>
    <Name>OGC:WMS</Name>
    <Title>My GeoServer with physical planning data</Title>
    <Abstract>
This is a description of your Web Map Server.
     </Abstract>
    <KeywordList>
      <Keyword>WFS</Keyword>
      <Keyword>WMS</Keyword>
      <Keyword>GEOSERVER</Keyword>
    </KeywordList>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="
        simple" xlink:href="http://localhost:8080/geoserver/wms"/>
     ...
  </Service>
  <Capability>
    <Request>
      <GetCapabilities>
        <Format>application/vnd.ogc.wms_xml</Format>
        <DCPType>
          <HTTP>
            <Get>
```

```
                <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                    xlink:type="simple" xlink:href="http://localhost:8080/
                    geoserver/wms?SERVICE=WMS&amp;"/>
              </Get>
              <Post>
                <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                    xlink:type="simple" xlink:href="http://localhost:8080/
                    geoserver/wms?SERVICE=WMS&amp;"/>
              </Post>
            </HTTP>
          </DCPType>
        </GetCapabilities>
        <GetMap>
        ...
          <DCPType>
            <HTTP>
              <Get>
                <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                    xlink:type="simple" xlink:href="http://localhost:8080/
                    geoserver/wms?SERVICE=WMS&amp;"/>
              </Get>
            </HTTP>
          </DCPType>
        </GetMap>
        ...
      </Request>
      <Exception>
        <Format>application/vnd.ogc.se_xml</Format>
      </Exception>
      <UserDefinedSymbolization SupportSLD="1" UserLayer="1" UserStyle="1"
          RemoteWFS="0"/>
      <Layer>
        <Title>My GeoServer WMS</Title>
        <Abstract>
This is a description of your Web Map Server.
       </Abstract>
        <SRS>EPSG:28992</SRS>
        <LatLonBoundingBox minx="5.77346637969378" miny="52.08228224617108"
            maxx="7.092432235268274" maxy="52.887379821562526"/>
        <Layer queryable="1">
          <Name>bestemmingsplan</Name>
          <Title>Bestemmingsplan</Title>
          ...
          <SRS>EPSG:28992</SRS>
          <LatLonBoundingBox minx="6.844986056380616" miny="52.18486021935283"
              maxx="6.913646304231541" maxy="52.20429497737166"/>
          <BoundingBox SRS="EPSG:28992" minx="254647.62847653677" miny="
              467305.09012648254" maxx="259385.97956194918" maxy="
              469563.68567804276"/>
          <Style>
            <Name>BP_polygon</Name>
            <Title>Default Styler</Title>
            <Abstract/>
            <LegendURL width="20" height="20">
              <Format>image/png</Format>
              <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                  xlink:type="simple" xlink:href="http://localhost:8080/
                  geoserver/wms/GetLegendGraphic?VERSION=1.0.0&amp;FORMAT=image
                  /png&amp;WIDTH=20&amp;HEIGHT=20&amp;LAYER=
                  topp:wesselerbrink_stripped2"/>
            </LegendURL>
          </Style>
        </Layer>
        ...
    </Capability>
</WMT_MS_Capabilities>
```

Figure A.1: Image of Bestemmingsplan returned from WMS instance based on the GetMap request of Listing A.9.

In the given case, the user is interested in physical planning at the municipal level (i.e. Bestemmingsplan), which is provided as one layer by this specific WMS instance. He/she thereby requests the specific map for the area of interest (indicated as bounding box for e.g. Enschede, the Netherlands) with a specific style (default) and in a specific image format (image/png) (Listing A.9). The result is a plain image (Figure A.1), which can be overlayed with different layers from different WMS instances, as depicted in the DURP ondergronden client (Section 4.1.3, Figure 4.6).

Listing A.9: Example GetCapabilities request for WMS.

```
http://localhost:8080/geoserver/wms?HEIGHT=535&SRS=EPSG%3A28992&WIDTH=1200&
    STYLES=&LAYERS=bestemmingsplan&FORMAT=image%2Fpng&SERVICE=WMS&VERSION
    =1.1.1&REQUEST=GetMap&BBOX
    =255146.21961858094,467611.71251788887,258200.23496660066,468974.5668669427
```

The examples of WMS listings are tested with the GeoServer implementation [70].

A.3 Example of SLD

SLD documents [137] are used to customize the symbolization and they are for this research incorporated in the user profile (Section 3.2). To define a symbolization in a standardized way based on the definitions of Geonovum [69] for physical plans in the Netherlands, SLD documents are one solution. Such an SLD document is depicted in Listing A.10. It defines the symbolization of Bestemmingsplannen, as served by the WMS in Appendix A.2.

Listing A.10: Example SLD document for defining the symbolization of Bestemmingsplannen in the Netherlands.

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld_StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer><Name>Default Styler</Name>
  <UserStyle>

    <Title>Default Styler</Title>
    <Abstract></Abstract>
    <FeatureTypeStyle>
      <!--FeatureTypeName>wesselerbrink_stripped</FeatureTypeName-->
      <Rule>
        <Name>Wonen</Name>
        <Title>Wonen</Title>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>NAAM</ogc:PropertyName>
            <ogc:Literal>Wonen</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">
              <ogc:Literal>#ffcd32</ogc:Literal>
            </CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">
              <ogc:Literal>#000000</ogc:Literal>
            </CssParameter>
          </Stroke>
        </PolygonSymbolizer>
      </Rule>
      ...
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

The example SLD document defines symbolization for area planning objects. In particular, it defines a filter for the attribute `NAAM` and the value `wonen` and applies for this filter a `PolygonSymbolizer` with a specific color of the filling (orange) and an outline (black). The result of the map visualized according to this SLD is depicted in Figure A.1. The given SLD document is tested with the GeoServer implementation [70].

An SLD document can be directly incorporated in a WMS request, to customize the symbolization of the layer provided by the WMS. The SLD document can be either incorporated inside the GetMap (HTTP-GET) request or as a link as illustrated in Listing A.11.

Listing A.11: Example GetMap request with referenced SLD document.

```
http://localhost:8080/geoserver/wms?Request=GetMap&SLD=http%3A//
    localhost:8080/data/mySLD.xml&....
```

A.4 Example of WMC

WMC documents [135] are used in this research to configure the final map and as a complementary concept to user profiles (Section 3.3.1). They allow users to configure the content of a map by linking remote WMS instances.

The given example (Listing A.12) is taken from the DURP ondergronden client (Section 4.1.3) and describes a map on municipal scale, as depicted in Figure 4.6. The given document defines the geographic extent displayed in the map, the size of the map and the layers incorporated in the map (on-demand base map, non-generalized map (hidden) and physical plan). Each layer is configured with an applicable style. Based on this document, the client application is able to retrieve the different layers via GetMap operation, as described in Appendix A.2.

Listing A.12: Example WMC document for defining the map content of the DURP ondergronden client.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<ViewContext version="1.0.0" id="bestemmingsplan" xmlns="http://www.opengis.
    net/context" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http:
    //www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.
    opengis.net/context http://schemas.opengis.net/context/1.0.0/context.xsd"
    >
  <General>
    <Window width="640" height="480"/>
    <BoundingBox SRS="EPSG:28992" minx="256301.6540364583" miny="
        468281.0183593753" maxx="256492.57070312498" maxy="468424.2058593753"
        />
    <Title xml:lang="en">Bestemmingsplan map</Title>
    <KeywordList>
      <Keyword>world</Keyword>
      <Keyword>atlas</Keyword>
    </KeywordList>
    <Abstract xml:lang="en">Bestemmingsplan</Abstract>
  </General>
  <LayerList>
    <Layer queryable="1" hidden="0">
      <Server service="OGC:WMS" version="1.0.0" title="OGC:WMS">
        <OnlineResource xlink:type="simple" xlink:href="http://localhost:8080
            /geoserver/wms"/>
      </Server>
      <Name>topp:TGT_GBKN_Building</Name>
      <Title>generalized GBKN Buildings</Title>
      <SRS>EPSG:28992</SRS>
      <FormatList>
        <Format current="1">image/gif</Format>
```

```
      </FormatList>
      <StyleList>
        <Style current="1">
          <Name>basemap_polygon</Name>
          <Title>polygons</Title>
        </Style>
      </StyleList>
    </Layer>
    <Layer queryable="1" hidden="1">
      <Server service="OGC:WMS" version="1.0.0" title="OGC:WMS">
        <OnlineResource xlink:type="simple" xlink:href="http://localhost:8080
            /geoserver/wms"/>
      </Server>
      <Name>topp:building</Name>
      <Title>non generalized GBKN Buildings outline</Title>
      <SRS>EPSG:28992</SRS>
      <FormatList>
        <Format current="1">image/gif</Format>
      </FormatList>
      <StyleList>
        <Style current="1">
          <Name>basemap_polygon_outline</Name>
          <Title>line</Title>
        </Style>
      </StyleList>
    </Layer>
    <Layer queryable="1" hidden="0">
      <Server service="OGC:WMS" version="1.0.0" title="OGC:WMS">
        <OnlineResource xlink:type="simple" xlink:href="http://localhost:8080
            /geoserver/wms"/>
      </Server>
      <Name>Bestemmingsplan</Name>
      <Title>Bestemmingsplan</Title>
      <SRS>EPSG:28992</SRS>
      <FormatList>
        <Format current="1">image/gif</Format>
      </FormatList>
      <StyleList>
        <Style current="1">
          <Name>BP_polygon</Name>
          <Title>BP_polygon</Title>
        </Style>
      </StyleList>
    </Layer>
  </LayerList>
</ViewContext>
```

The listed WMC document is tested with the WMS client application MapBuilder [125] and has been applied in the DURP ondergronden client (Section 4.1.3).

Appendix *B*

Examples of WPS Profiles

This appendix lists the WPS Profiles mentioned in this work. The WPS Profiles are based on the latest version of WPS 1.0.0 and the according schemas, which are available at `schemas.opengis.net/wps/1.0.0`.

B.1 Process Description Referencing the WPS Profile for Ratio-based Simplification

This process description is referencing the ratio-based simplification profile of Appendix B.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ProcessDescriptions xmlns:ns="http://www.opengis.net/wps/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="1.0.0">
  <ProcessDescription xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" wps:processVersion="2" storeSupported="true" statusSupported="false">
    <ows:Identifier>ratio-basedSimplification</ows:Identifier>
    <ows:Title>ratio-based Simplification</ows:Title>
    <wps:Profile>http://centralWPSrepository.com/ratio-basedSimplfication.xml</wps:Profile>
  </ProcessDescription>
</ns:ProcessDescriptions>
```

B.2 WPS Profile for Ratio-based Simplification

The ratio-based simplification profile specifies the process, as described in Section 5.3.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ProcessDescriptions xmlns:ns="http://www.opengis.net/wps/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/
```

```
    wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="
    1.0.0">
  <ProcessDescription xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows=
      "http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/
      xlink" wps:processVersion="2" storeSupported="true" statusSupported="
      false">
    <ows:Identifier>ratio-basedSimplification</ows:Identifier>
    <ows:Title>ratio-based Simplification</ows:Title>
    <ows:Abstract>WPS profile for ratio-based simplification</ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>FEATURES</ows:Identifier>
        <ows:Title>input features</ows:Title>
        <ows:Abstract>Just features</ows:Abstract>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/XML</MimeType>
              <Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</
                  Schema>
            </Format>
          </Default>
        </ComplexData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>ratio</ows:Identifier>
        <ows:Title>simplification ratio</ows:Title>
        <ows:Abstract/>
        <LiteralData>
          <ows:DataType ows:reference="xs:double"/>
          <ows:AllowedValues>
            <ows:Range>
              <ows:MinimumValue>0</ows:MinimumValue>
              <ows:MaximumValue>1</ows:MaximumValue>
            </ows:Range>
          </ows:AllowedValues>
        </LiteralData>
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>SIMPLIFIED_FEATURES</ows:Identifier>
        <ows:Title>simplified geometries</ows:Title>
        <ows:Abstract>GML stream describing the simplified feature.</
            ows:Abstract>
        <ComplexOutput>
          <Default>
            <Format>
              <MimeType>text/XML</MimeType>
              <Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</
                  Schema>
            </Format>
          </Default>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</ns:ProcessDescriptions>
```

B.3 WPS Profile for Ratio-based Simplification with Referenced XMI Metadata File

The extended version of the ratio-based simplification profile in Appendix B.2. It references a XMI file in the metadata element of the process description. The XMI file consists of the UML model for the features (General Feature Model) and the OCL description as specified in Listing 5.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ProcessDescriptions xmlns:ns="http://www.opengis.net/wps/1.0.0" xmlns:xsi
    ="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
    www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/
    wpsDescribeProcess_response.xsd" xml:lang="en-US" service="WPS" version="
    1.0.0">
  <ProcessDescription xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows=
      "http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/
      xlink" wps:processVersion="2" storeSupported="true" statusSupported="
      false">
    <ows:Identifier>ratio-basedSimplification</ows:Identifier>
    <ows:Title>ratio-based Simplification</ows:Title>
    <ows:Abstract>WPS profile for ratio-based simplification</ows:Abstract>
    <ows:Metadata xlink:href="http://centralWPSrepository.com/ratio-
        basedSimplificationModel.xmi"></ows:Metadata>
  ...
  </ProcessDescription>
</ns:ProcessDescriptions>
```

Appendix C

Survey Template

This section contains the template to perform the survey with the title *Current problems of automated generalization of topographic data at National Mapping Agencies Survey*, as described in Section 2.1.5. The complete analysis is extensively described in Foerster and Stoter [55], Foerster et al. [59]. The survey is twofold. The first part addresses the technical aspects of the generalization environment of the specific NMA. The second part focuses especially on the importance and the problems of generalization operators related to the production process at the scales, the specific NMA generates the topographic product. The results of the survey are depicted and discussed in Appendix D.

Part I

1) Do you apply separate processes for generalizing data (model generalization) and maps (cartographic generalization)?

☐ Yes

☐ No

1a) If yes (question 1): Why do you separate model and cartographic generalization?

☐ generalized data is part of our product line

☐ generalized data allows us to improve/ease the map production process

☐ generalized data is used internally for other tasks (e.g. military mapping)

☐ other reason:

2) Do you already take cartographic issues and symbolization issues into account within the model generalization process?

	Yes	No
Elimination		
Displacement		
Enlargement		
Typification		
Enhancement		
Others: [Fill in here]		

	Modeling	Execution	Evaluation
Model generalization			
Cartographic Generalization			

☐ Yes

☐ No

2a) If yes (question 2), which operators do you apply to maintain cartographic aspects already during model generalization?

3) What is the degree of automation of generalization (i.e. generalization without manual interaction) at your organization (from 0 - no automation to 100 - full automation)?

4) What are the most demanding issues to gain full automation concerning the different steps in generalization (0 - no demand, 1 - less demanding - 5 most demanding):

MODEL GENERALIZATION	Modeling	Execution	Evaluation
Algorithms for generalization			
Algorithms for pattern & conflict detection			
Appropriate mechanisms for orchestration these algorithms			
Others: [Fill in here]			

CARTOGRAPHIC GENERALIZATION	Modeling	Execution	Evaluation
Algorithms for generalization			
Algorithms for pattern & conflict detection			
Appropriate mechanisms for orchestration these algorithms			
Others: [Fill in here]			

5) Do you maintain references between the different scales in the database?

☐ Yes

☐ No

Part II

6) Please answer this question for each of the scales you apply generalization:

Source scale:

Target scale:

6a) What do you produce at this scale? (multiple answers possible)

Target

☐ Maps

☐ Data

If you only produce maps please continue with Question 6d.

6b) Are the data object-oriented? (multiple answers possible)

Source

☐ the source model is object-oriented

☐ the source data is stored in an object-oriented database

☐ the source model is implemented in a relational database

Target

- ☐ the target model is object-oriented
- ☐ the generalized data is stored in an object-oriented database
- ☐ the target model is implemented in a relational database

6c) Are the data topologically structured? (multiple answers possible)

Source

- ☐ None
- ☐ Graph structured topology (1-D) (i.e. network)
- ☐ Planar structured topology (2-D) (i.e. areas are seamless, no gaps & no overlaps)

Target

- ☐ None
- ☐ Graph structured topology (1-D) (i.e. network)
- ☐ Planar structured topology (2-D) (i.e. areas are seamless, no gaps & no overlaps)

6d) Which software product(s) do you use to generalize at this scale?

For model generalization:

For cartographic generalization:

7) Which are the most important feature types to appear in the dataset/map at this scale, and of which geometry type do they have to appear? (0 - no importance, 1 - less important, 5 - very important) Please indicate first the different scales (first row) you already introduced in Question 6. Then mark the importance of the different feature types regarding the scale and the geometry type.

Target scale			
	Point	Line	Polygon
Administrative			
Building			
Railway			
Road			
Relief			
Lake			
River			
Coastal feature			
Landcover			

Evaluating the importance of the operators (in the context of successful generalization processing)

8) Which could you consider as the most important operators for the supported scales regarding the feature type in the model generalization process? (0 no importance, 1 - less important, 5 - very important) Please indicate first the different scales (first row) you already introduced in Question 6. Then mark the importance of the different operators regarding the scale and the feature type.

Target scale							
	Collapse	Combine	Amalgamation	Reclassification	Class Selection	Simplification	Other
Administrative							
Building							
Railway							
Road							
Relief							
Lake							
River							
Coastal feature							
Landcover							

9) Which could you consider as the most important operators for the supported scales regarding the feature type in the cartographic generalization process? (0 no importance, 1 - less important, 5 - very important). Please indicate first the different scales (first row) you already introduced in Question 6. Then mark the importance of the different operators regarding the scale and the feature type.

Target scale							
	Enhancement	Enlargement	Displacement	Elimination	Typification	Amalgamation	Other
Administrative							
Building							
Railway							
Road							
Relief							
Lake							
River							
Coastal feature							
Landcover							

Evaluating current problems of operators (in the context of successful generalization processing)

10) Which could you consider as the most problematic operators at the supported scales regarding the feature type in the automated model generalization process? (0 no problem, 1 - less important problem, 5 - very important problem). Please indicate first the different scales (first row) you already introduced in Question 6. Then mark the importance of the different operators regarding the scale and the feature type.

Target scale							
	Collapse	Combine	Amalgamation	Reclassification	Class Selection	Simplification	Other
Administrative							
Building							
Railway							
Road							
Relief							
Lake							
River							
Coastal feature							
Landcover							

11) Which could you consider as the most problematic operators for the supported scales regarding the feature type in the automated cartographic generalization process? (0 no problem, 1 - less important problem, 5- very important problem). Please indicate first the different scales (first row) you already introduced in Question 6. Then mark the importance of the different operators regarding the scale and the feature type.

Target scale							
	Enhancement	Enlargement	Displacement	Elimination	Typification	Amalgamation	Other
Administrative							
Building							
Railway							
Road							
Relief							
Lake							
River							
Coastal feature							
Landcover							

Appendix *D*

Survey Results

The results listed here are taken from the article of Foerster et al. [59].

The survey distinguished between scale transitions as they are carried out at the NMAs. To conduct representative results, the analysis focused only on scale transitions that are applied by more than three participants (i.e. 1:10k-1:50k; 1:50k-1:100k; 1:50-1:250k), see Figure D.1. All results in the remainder of this section are analysed for these three scale transitions separately.

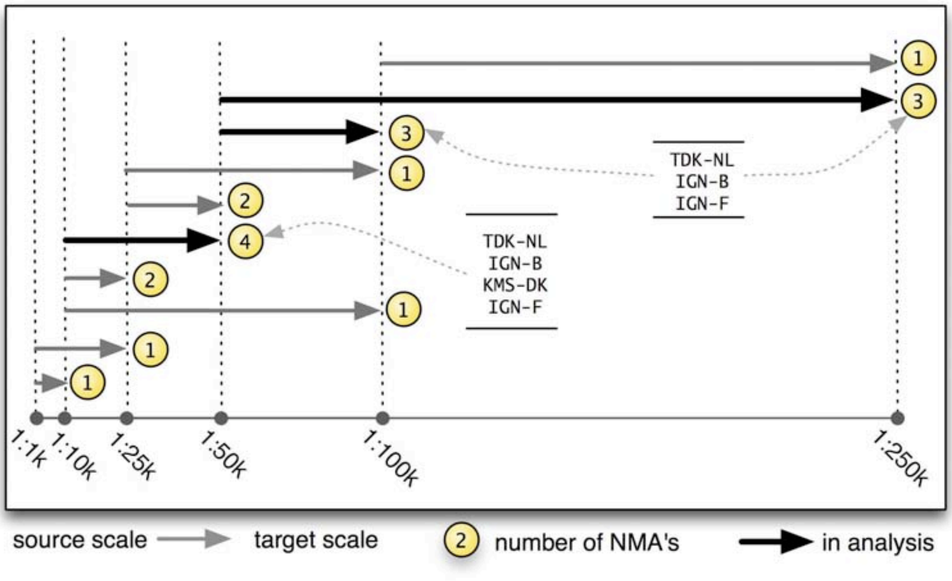


Figure D.1: Scale transitions collected in the survey - the highlighted ones are included in the analysis.

The list of operators incorporated in the survey is the same as used in this thesis for the classification of generalization operators (Section 5.2, Figure 5.3).

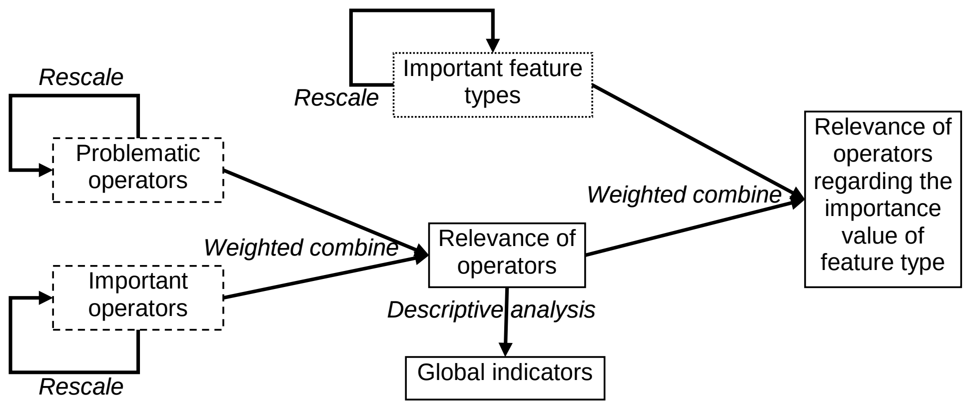


Figure D.2: Sequence of steps applied in the analysis of the survey.

To provide a comprehensive analysis of the performed survey, the following analysis steps have been designed:

1. Rescaling important and problematic operators
2. Calculating the relevance of operators
3. Weighting the relevance of the operators by the importance of feature types.

These sequence of analysis steps is depicted in Figure D.2 and is explained stepwise in the following.

Step 1. Rescaling important and problematic operators

The survey separated between "important" and "problematic" operators. Important means that an operator is often applied and plays a dominant role in the specific generalization process (applied on a specific scale transition and on a specific feature type). Whereas problematic means that a specific operator is lacking and it therefore exposes problems to the generalization process. Both measures address an important and specific aspect. The results of these two separate measures have been reported in Foerster and Stoter [55].

In this research the two measures are combined in an aggregated value. Therefore, the values (C) for the important and problematic generalization operators are rescaled to their local minimum and maximum using Equation D.1. Originally the participants were asked to rate the different variables using a value range from 0 (low) to 5 (high). After rescaling, all values are between 0 and 1 which allows us to compare and combine results of the different measures.

$$\forall c \epsilon C, c = \frac{c - min(C)}{max(C) - min(X)} \tag{D.1}$$

The resulting values are standardised on the local maximum ($max(C)$) and the local minimum ($min(C)$) of C. The rescaled values for the important and problematic operators are presented in Section D.1.

Step 2. Calculating the relevance of operators

To get a complete picture of the operators, an integrated measure is introduced, termed as the relevance of a specific generalization operator. The relevance measure combines the (rescaled) important and problematic values of operators using Equation D.2.

$$\forall g \epsilon G, \exists f \epsilon F, c = 0.5 \cdot g + 0.5 \cdot f \tag{D.2}$$

Equation D.2 weights the values of a set (g of G) by a corresponding measure (f) of another set (F) and applies a linear factor of 0.5, which weights both aspects equally.

The results of this analysis separated for model and cartographic generalization operators are presented in Section D.2. The relevance measure is further compiled to global indicators by descriptive statistics which are visualised using Box-Plot diagrams in Section D.2. The global indicators represent first quartile, third quartile, arithmetic mean and median for each of the scale transitions. The global indicators give additional information about the outcomes of the relevance measures for model and cartographic generalization operators at specific scales. Any variance indicator would also have been an interesting global indicator. However they have not been calculated as the number of collected survey answers per scale was too small.

Step 3. Weighting the relevance of the operators by the importance of feature types

In a next step the relevance of the operators are weighted by the rescaled importance values of the feature type. The results are presented in Section D.3. The relevance of operators already implicitly incorporates a certain degree of importance of the specific feature types. However, combining relevance with importance of feature types will both filter and exaggerate the relevant operators with respect to the most important feature types in the current products of NMAs. This new indicator better exposes the requirements for map production, since it provides not only insight into missing functionality, but also into which operators might be relevant in the future, i.e. how bad it is that they are missing?

The relevance of operators and the importance of feature type are weighted 0.5 and 0.5. Consequently, the importance and problematic characteristics of operators only influence this second measure by 0.25 each whereas the importance of the feature type is 0.5 of the complete measure. It may have been possible to weight the values by 1/3 each. However, in order to stress the role of the feature type within the generalization process and its importance regarding the operator, we equally weighted the relevance values of operators and the importance value of feature types.

D.1 Important and Problematic Operators for Model and Cartographic Generalization

This section introduces the rescaled values for important and problematic generalization operators. The original values were collected from 0 to 5 and can be found in Foerster and Stoter [55].

D.1.1 Important Generalization Operators

The rescaled values representing the importance of operators in relation to the different feature types are presented in Table D.1 for model generalization operators and in Table D.2 for cartographic generalization operators. The importance values of these two types of operators differ when considering the specific scale transition. The importance of model generalization is significantly higher at scale transition at smaller scales (1:50k - 1:250k). Whereas the importance of cartographic generalization operators is higher at larger scales (1:10k - 1:50k). NMAs consider simplification, amalgamation (model generalization) and displacement (cartographic generalization) as most important operators.

D.1.2 Problematic Generalization Operators

The lack of specific generalization operators in relation to a specific feature type and scale are depicted in Table D.3 (model generalization) and Table D.4 (cartographic generalization). Table D.3 shows that model generalization operators are not considered as problematic. Contrary, the cartographic generalization operators (Table D.4) are more problematic for current production lines. The most problematic operators are displacement and typification.

D.2 Relevant Generalization Operators

The results of the relevance measure, combining the importance and lacking characteristics of operators, are presented in Table D.5 (model generalization) and Table D.6 (cartographic generalization). All values are calculated based on the rescaled measures presented in Section D.1.

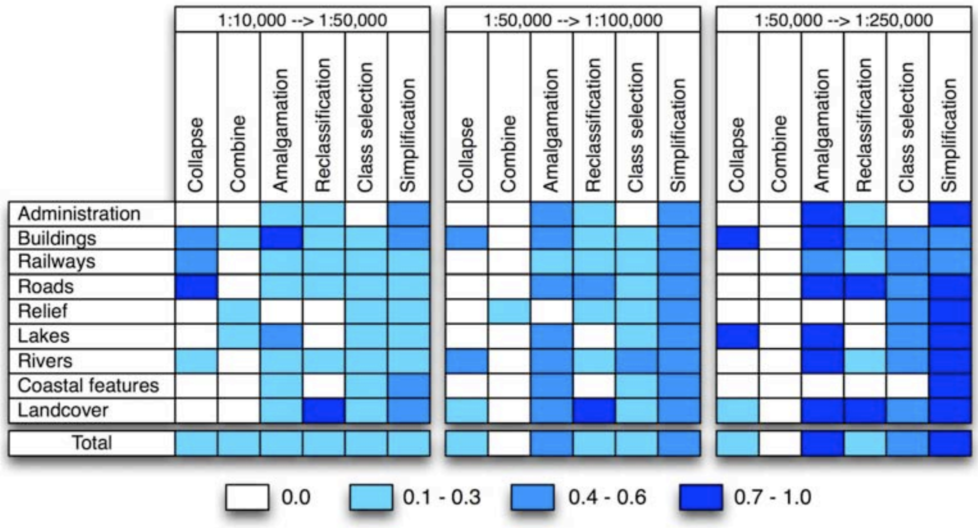


Table D.1: Importance of model generalization operators versus feature types related to scale.

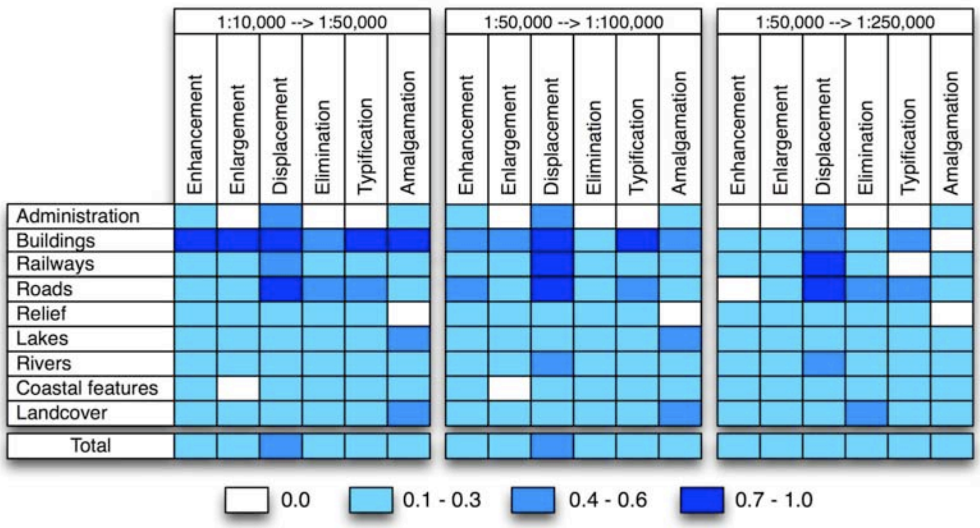


Table D.2: Importance of cartographic generalization operators versus feature types related to scale.

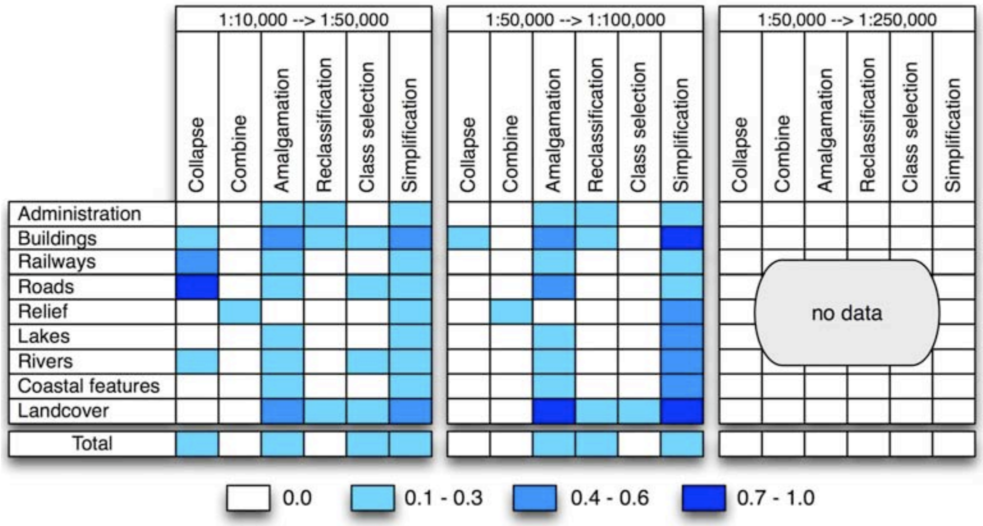


Table D.3: Problematic model generalization operators (no answers for 1:50k-250k available).

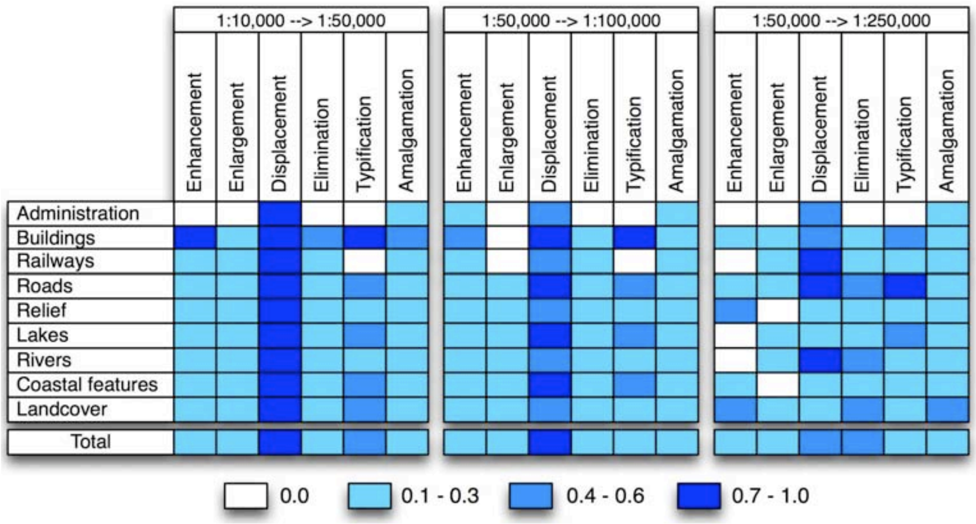


Table D.4: Problematic cartographic generalization operators.

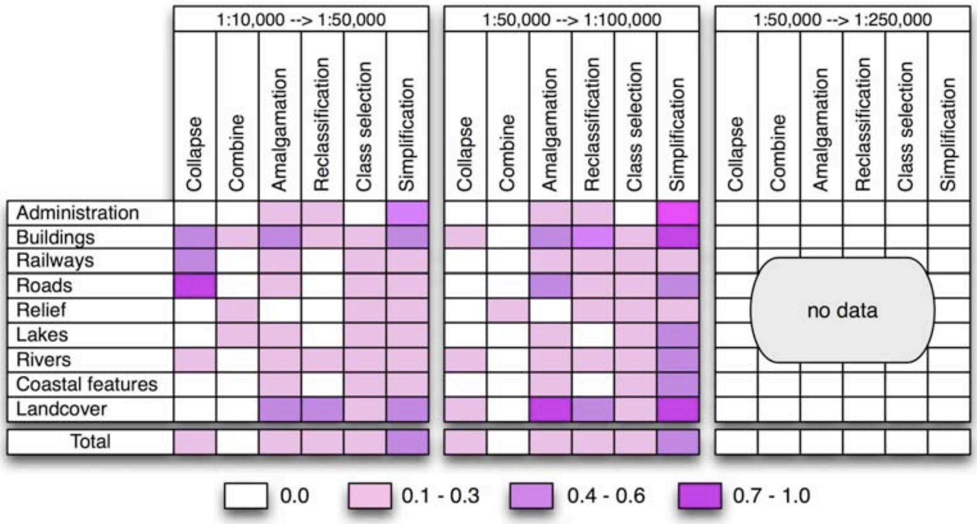


Table D.5: Calculated relevance values of model generalization operators separated for the major scale transitions.

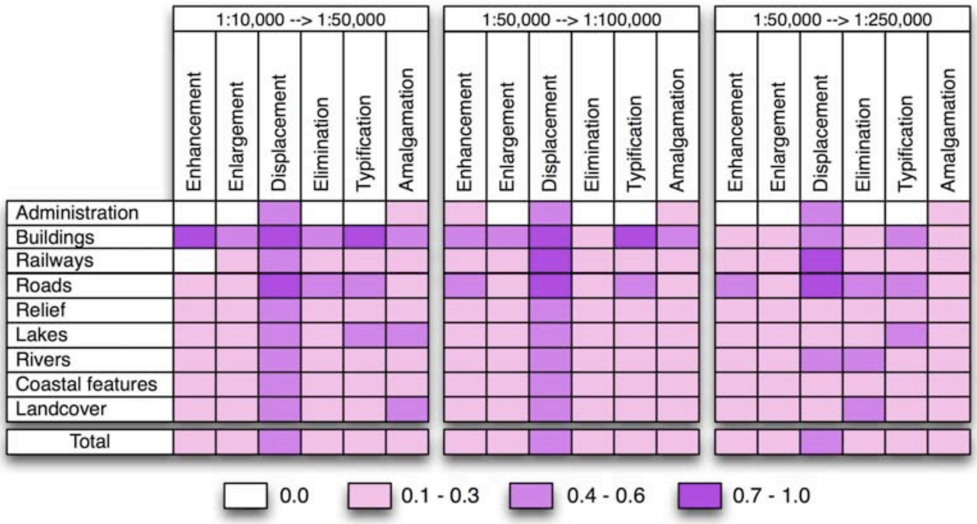


Table D.6: Calculated relevance values of cartographic generalization operators separated for the major scale transitions.

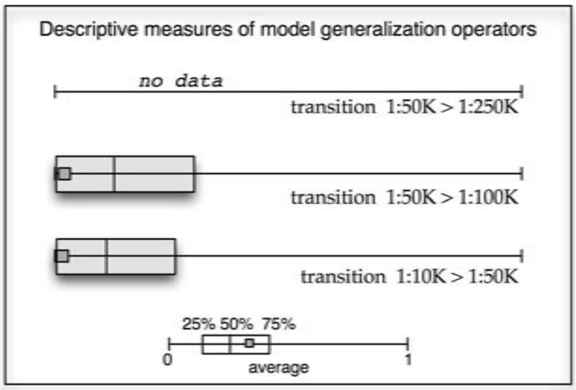


Figure D.3: Box-Plot diagram of the model generalisation operator measures (min=0, max=1) as presented in Table D.5.

Based on these tables, the following conclusions can be drawn. Simplification, collapse and amalgamation are the most relevant model generalization operators. Collapse is relevant at lower scale transitions (1:10k-1:50k), especially for roads, buildings and railways but not at the higher scale transition (1:50k-1:100k). This can be explained because already collapsed roads are reused at higher scales.

Table D.6 shows that the most relevant generalization operators for cartographic generalization are displacement and typification. Additionally, any operator applied to feature type buildings is highly relevant.

To compare the overall relevance of operators at certain scale transitions and between model generalization and cartographic generalization, Figure D.3 and Figure D.4 presents the results of the global indicators (Box-Plot diagram). The rescaled values are the basis for those diagrams. Thus, the value range is always between 0 and 1.

Several conclusions can be drawn from these global indicators. Firstly, the relevance of model generalization operators increases with decreasing scales (from 1:10k-1:50k to 1:50k-1:100k), whereas the relevance of cartographic operators decreases with decreasing scale. A second conclusion is that cartographic generalization operators are overall more relevant than model generalization operators. This is in line with the workshop conclusions that especially contextual operators (mostly cartographic generalization operators) are considered as problematic. In addition, the numbers support the initial findings of the survey reported in Foerster and Stoter [55]. Another observation from Figure D.3 and Figure D.4 is that the distribution of the values is different, as the median is above the average mean for model generalization operators. This can be explained by low relevance values for model generalization operators as shown in Table D.5. In the case of cartographic generalization it is slightly different. Some operators seem to be more relevant, as the mean is higher than the median, which is an indicator for statistical outliers.

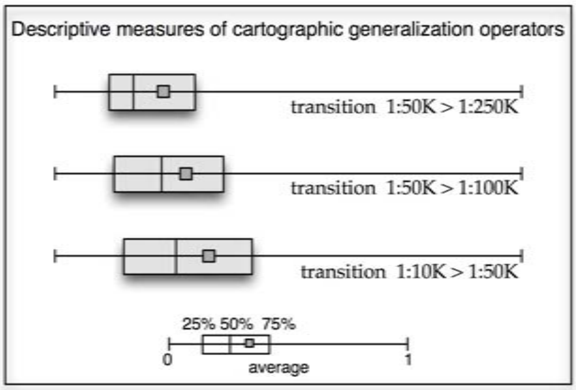


Figure D.4: Box-Plot diagram of the cartographic generalisation measures (min=0, max=1) as presented in Table D.6.

D.3 Relevance of Operators Weighted by Importance Value of Feature Types

Table D.7 shows the rescaled importance values of the different feature types regarding the specific scale transitions, which were originally collected from 0 (low) to 5 (high). The table shows that rivers and roads are the most dominant feature types for all scale transitions. Whereas, the building feature type becomes less important over decreasing scale. In addition, networks become more important at smaller scales.

In a second step the relevance of generalization operators (Section D.2) are weighted by the rescaled importance values of the feature types. This indicator combines the importance values of the feature type (Table D.7) according to Equation D.2 with the relevance values of the model generalization and cartographic generalization operators (Table D.5 and Table D.6). The results are depicted in Table D.8 and Table D.9 for respectively model generalization operators and cartographic generalization operators.

The following observations can be made from these tables. The generalization of buildings and roads appear to be the most relevant for model generalization (Table D.8). Especially, amalgamation of buildings seems to be highly relevant for map production at 1:10k-1:50k. In line with Table D.6, Table D.8 shows that amalgamation is of major concern at the investigated scales. In contrast to some of the extremes that disappeared compared to Table D.5. For example simplification turns out to be not that relevant overall for model generalization.

Also for cartographic generalization (Table D.9), weighting the relevance measures by importance values of feature types causes some extreme values to disappear. For instance displacement got a lower relevance, due to the lower importance values of the combined feature types. However, as rivers are highly relevant in map production,

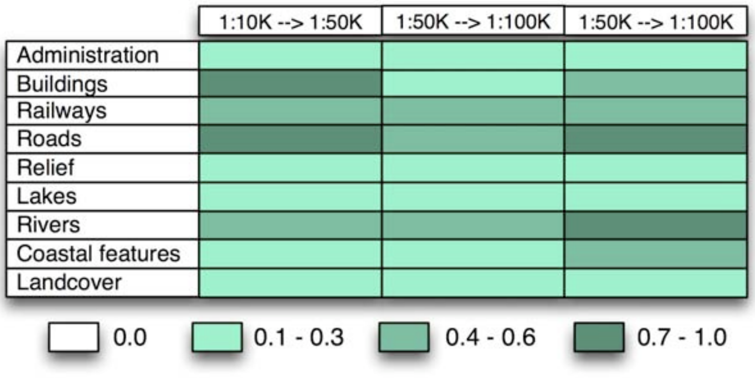


Table D.7: Importance values of feature types at certain scale transitions. The values are scaled regarding the local minimum and maximum.

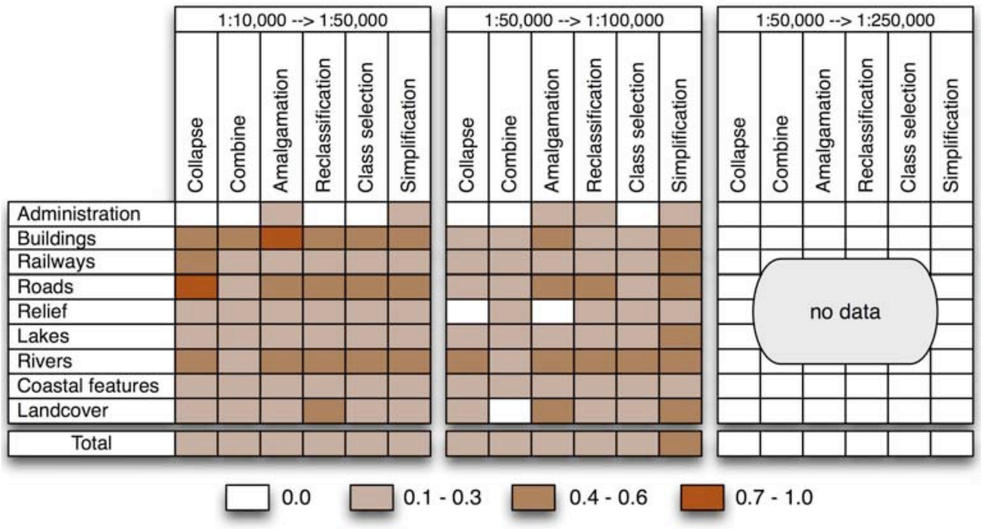


Table D.8: Relevance of model generalization operators weighted by the importance of feature types.

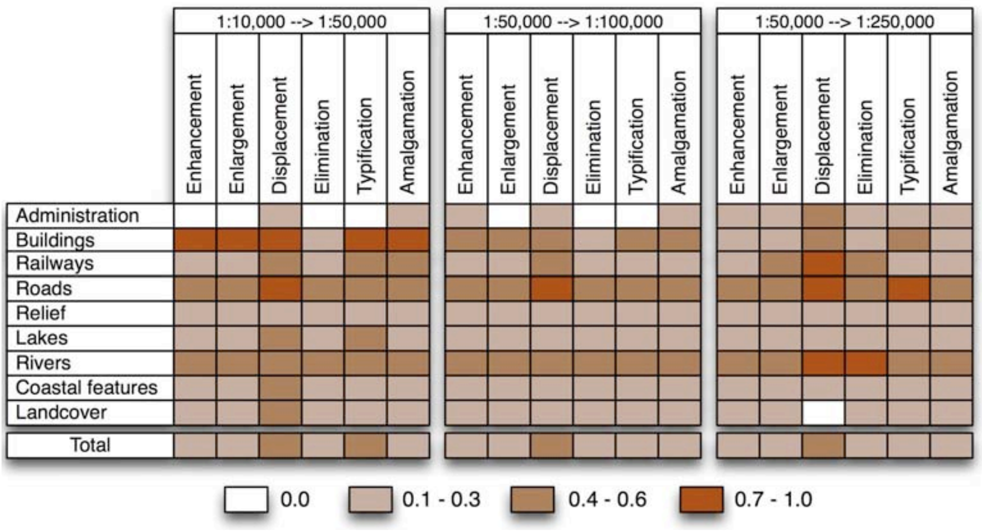


Table D.9: Relevance of cartographic generalization operators weighted by the importance of feature types.

all the related operators (i.e. enhancement and elimination of rivers) become more relevant. The same conclusion applies to roads (i.e. enlargement and elimination) and also to railways (i.e. Elimination and Enhancement).

D.4 Synopsis of the Survey

The analysis demonstrates the relevance of specific generalisation operators by combining the importance and problematic (i.e. lacking) aspects of operators. This shows that the relevance of model generalisation operators increases with decreasing scales, but never reaches the relevance level of cartographic generalisation operators. Weighting the relevance measures by importance values of feature types results in another valuable conclusion. Especially network-based feature types such as rivers, railways and roads are relevant for NMAs in combination with the operators enhancement, typification and elimination. Overall, contextual operators and operators that create generalised features that inherit a network-based structure are the main challenges for cartographic generalisation. This underlines the workshop findings.

Bibliography

[1] 52°North Geoprocessing Community (2009). Community website. `www.52north.org/wps`.

[2] 52°North Initiative (2009). 52°north website. `www.52north.org`.

[3] Ackoff, R. (1989). From data to wisdom. *Journal of Applied System Analysis*, 16:3–9.

[4] Agarwal, P. (2005). Ontological considerations in GIScience. *International Journal of Geographical Information Science*, 19(5):501–536.

[5] Alameh, N. (2003). Chaining geographic information web services. *IEEE Internet Computing*, 07(5):22–29.

[6] Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). *Web Services*. Springer Verlag, 1st edition edition.

[7] ArgoUML (2009). Project website. `argouml.tigris.org/`.

[8] Bader, M., Barrault, M., and Weibel, R. (2005). Building displacement over a ductile truss. *International Journal of Geographical Information Science*, 19(8-9):915–936.

[9] Bakker, N. and Kolk, B. (2003). A new object-oriented topographical database in GML. In *Proceedings of the 21st International Cartographic Conference (ICC)*, Durban, South Africa.

[10] Barrault, M., Regnauld, N., Duchene, C., Haire, K., Baejs, 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 *20th International Cartographic Conference - ICC*, pages 2110–2116, Bejing, China.

[11] Beard, K. M. (1991). Constraints on rule formation. In Buttenfield, B. and McMaster, R. B., editors, *Map Generalization: Making Rules for Knowledge Representation*, pages 121–135. Longman, London.

[12] Beckert, B., Hähnle, R., and Schmitt, P. H., editors (2007). *Verification of Object-Oriented Software: The KeY Approach*. LNCS 4334. Springer-Verlag.

[13] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):28–37.

[14] Bertolotto, M. and Egenhofer, M. J. (2001). Progressive transmission of vector map data over the world wide web. *Geoinformatica*, 5(4):345–373.

[15] Birth, K. (2003). ATKIS-Project modell- und kartographische generalisierung, und die entwicklung geht weiter. *Kartographische Nachrichten*, (3):119–126.

[16] Bishr, Y. (2006). Geospatial semantic web. In Rana, S. and Sharma, J., editors, *Frontiers in geographic information technology*, pages 139–154. Springer.

[17] Booch, G., Maksimchuk, R. A., Engel, M. W., Young, B. J., Conallen, J., and Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications.* Addison-Wesley Professional, 3 edition.

[18] Brassel, K. E. and Weibel, R. (1988). A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems*, 2(3):229–244.

[19] Brauner, J., Foerster, T., Schäffer, B., and Baranski, B. (2009). Towards a research agenda for geoprocessing services. In Haunert, J., Kieler, B., and Milde, J., editors, *12th AGILE International Conference on Geographic Information Science*, Hanover, Germany. IKG, Leibniz University of Hanover.

[20] Burghardt, D., Edwardes, A., and Mannes, J. (2004). An architecture for automatic generalisation of mobile maps. In Gartner, G., editor, *2nd Symposion on Location based service and telecartography*, Vienna, Austria.

[21] Burghardt, D. and Meier, S. (1997). Cartographic displacement using the snakes concept. In Förstner, W. and Plümer, L., editors, *Semantic Modeling for the Acquisition of Topografic Information from Images and Maps*, pages 59–71. Birkhäuser Verlag.

[22] Burghardt, D. and Neun, M. (2006). Automated sequencing of generalization services based on collaborative filtering. In Raubal, M., Miller, H., Frank, A., and Goodchild, M., editors, *Proceedings of Geographic Information Science, IfgiPrints*, volume 28 of *IfgiPrints*, pages 41–43, Münster, Germany.

[23] Burghardt, D., Neun, M., and Weibel, R. (2005). Generalization services on the web - classification and an initial prototype implementation. *Cartography and Geographic Information Science*, 32:257–268(12).

[24] Buttenfield, B. P. and McMaster, R. B., editors (1991). *Map Generalization: Making rules for knowledge representation.* Longman Scientific & Technical.

[25] Carver, S. and Peckham, R. (1999). Using GIS on the internet for planning. In Stillwell, J., Geertman, S., and Openshaw, S., editors, *Geographical Information and Planning*, Advances in Spatial Science, pages 371–390. Springer-Verlag.

[26] Cecconi, A. (2003). *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping*. PhD thesis, University of Zurich.

[27] Chaudhry, O. (2008). *Modelling Geographic Phenomena at Multiple Levels of Detail: A Model generalisation Approach based on Aggregation*. PhD thesis, University of Edinburgh.

[28] Chen, M., Ebert, D., Hagen, H., Laramee, R., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., and Silver, D. (2009). Data, information, and knowledge in visualization. *Computer Graphics and Applications, IEEE*, 29(1):12–19.

[29] Demuth, B. (2009). Dresden OCL toolkit. `dresden-ocl.sourceforge.net`.

[30] Deoliveira, J. (2008). GeoServer: uniting the GeoWeb and spatial data infrastrucutures. In *Proceedings of the 10th International Conference for Spatial Data Infrastructure*, St. Augustine, Trinidad.

[31] Di, L., Zhao, P., Yang, W., Yu, G., and Yue, P. (2005). Intelligent geospatial web services. In *Proceedings of Geoscience and Remote Sensing Symposium, 2005.*, volume 2 of *IEEE International*, pages 1229–1232.

[32] Diaz, L., Costa, S., Granell, C., and Gould, M. (2007). Migrating geoprocessing routines to web services for water resource management applications. In Wachowicz, M. and Bodum, L., editors, *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark.

[33] Dobing, B. and Parsons, J. (2005). Current practices in the use of UML. In *Perspectives in Conceptual Modeling*, volume 3770 of *Lecture Notes in Computer Science*, pages 2–11. Springer Verlag, Berlin.

[34] 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 caricature. *The Canadian Cartographer*, 10(2):112–122.

[35] Duchene, C. (2004). The CartACom model: a generalisation model for taking relational constraints into account. In *6th ICA Workshop on progress in automated map generalisation*, Leicester, UK.

[36] Duchene, C. and Gaffuri, J. (2008). Combining three multi-agent based generalization models: AGENT, CartACom and GAEL. In Ruas, A. and Gold, C., editors, *Headway in Spatial Data Handling*, Lecture Notes in Geoinformation and Cartography, pages 277–296. Springer.

[37] DURP ondergronden project (2009). Project website. www.durpondergronden.nl.

[38] Edwardes, A., Burghardt, D., Bobzien, M., Harrie, L., Reichenbacher, T., Sester, M., and Weibel, R. (2003). Map generalization technology: Addressing the need for a common research platform. In *Proceedings of the 21st International Cartographic Conference (ICC)*, Durban, South Africa.

[39] Edwardes, A., Burghardt, D., and Neun, M. (2005). Interoperability in map generalization research. In *International Symposium on Generalization of Information 2005*, Berlin, Germany.

[40] Elzakker, C. v. and Berg, W. v. (2009). Topografische ondergronden voor plankaarten: een gebruikersonderzoek. *Geo-Info: tijdschrift voor geo-informatie Nederland*, 6(3):64–69.

[41] ESRI (2006). ArcGIS server: ESRI's complete server GIS. ESRI whitepaper, ESRI Inc., Redlands, CA.

[42] ETSI (2005). Human factors (HF); user profile management. ETSI Guide EG 202 325 v1.1.1, ETSI.

[43] ETSI (2009a). Etsi website. `www.etsi.org`.

[44] ETSI (2009b). Human factors (HF): personalization and user profile management: Architectural framework. Technical specification TS 102 747 Draft, ETSI.

[45] ETSI (2009c). Human factors (HF); personalization and user profile management; user profile preferences and information. ETSI Standard draft ES 202 746, ETSI.

[46] Fink, J. and Kobsa, A. (2002). User modeling in personalized city tours. *Artificial Intelligence Review*, 18(1):33–74.

[47] Foerster, T., Burghardt, D., Neun, M., Regnauld, N., Swan, J., and Weibel, R. (2008a). Towards an interoperable web generalisation services framework - current work in progress. In *Proceedings of the 11th ICA workshop on generalization and Multiple representation*, Montpellier, France.

[48] Foerster, T., Lehto, L., Sarjakoski, L. T., Sarjakoski, T., and Stoter, J. E. (2009a). Map generalization and schema transformation of geospatial data combined in a web service. *Computers, Environment and Urban Systems*, in press.

[49] Foerster, T., Morales, J., and Stoter, J. E. (2008b). A classification of generalization operators formalised in OCL. In Pebesma, E., Bishr, M., and Bartoschek, T., editors, *Proceedings of the 6th Geographic Information Days*, volume 32 of *Ifgi Prints*, pages 141–156, Münster, Germany. Institute for Geoinformatics.

[50] Foerster, T. and Schäffer, B. (2007). A client for distributed geo-processing on the web. In Tayler, G. and Ware, M., editors, *W2GIS*, volume 4857 of *Lecture Notes in Computer Science*, pages 252–263. Springer.

[51] Foerster, T., Schäffer, B., Brauner, J., and Jirka, S. (2009b). Integrating OGC web processing services into geospatial Mass-Market applications. In *International Conference on Advanced Geographic Information Systems & Web Services, 2009.*, pages 98–103, Cancun, Mexiko. IEEE.

[52] Foerster, T., Stoter, J., and Morales, J. (2009c). Enhancing cartographic generalization processing with grid computing power and beyond. *GIS.Science*, 3:98–101.

[53] Foerster, T., Stoter, J., and Oosterom, P. v. (2009d). On-demand base maps on the web generalized according to user profiles. *International Journal of Geographical Information Science*, submitted.

[54] Foerster, T. and Stoter, J. E. (2006). Establishing an OGC web processing service for generalization processes. In *ICA workshop on Generalization and Multiple Representation*, Portland, Oregon, USA.

[55] Foerster, T. and Stoter, J. E. (2008). Generalization operators for practice - a survey at NMAs. In *Proceedings of the 11th ICA workshop on generalization and multiple representation*, Montpellier, France.

[56] Foerster, T. and Stoter, J. E. (2009). Providing customized base maps in a SDI for physical planning. In *GSDI 11*, Rotterdam, the Netherlands.

[57] Foerster, T., Stoter, J. E., and Köbben, B. (2007a). Towards a formal classification of generalization operators. In *Proceedings of the 23rd International Cartographic Conference (ICC)*, Moscow, Russia.

[58] Foerster, T., Stoter, J. E., Köbben, B., and Oosterom, P. v. (2007b). A generic approach to simplification of geodata for mobile applications. In Wachowicz, M. and Bodum, L., editors, *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark.

[59] Foerster, T., Stoter, J. E., and Kraak, M. (2009e). Challenges for automated generalisation at european mapping agencies - a qualitative and quantitative analysis. *Cartographic Journal*, in press.

[60] Foerster, T., Stoter, J. E., and Lemmens, R. (2007c). Towards automatic web-based generalization processing: a case study. In *10th ICA Workshop on Generalisation and Multi Representation*, Moscow, Russia.

[61] Foerster, T., Stoter, J. E., and Lemmens, R. (2008c). An interoperable web service architecture to provide base maps empowered by automated generalization. In Ruas, A. and Gold, C., editors, *Headway in Spatial Data Handling*, Lecture Notes in Geoinformation and Cartography, pages 255–76. Springer, Montpellier, France.

[62] Foster, I. and Kesselman, C., editors (2003). *The Grid 2: Blueprint for a new Computing Infrastrcture*. Morgan Kaufmann, San Francisco, California, USA, second edition edition.

[63] Gaffuri, J. (2008). *Automatic generalisation to take into account the field objects: the model called GAEL*. PhD thesis, Universite Paris-Est.

[64] Garcia, M. and Möller, R. (2007). Certification of transformations algorithms in Model-Driven software development. *Software Engineering 2007*, 105(ISBN 978-3-88579-199-7):107–118.

[65] GBKN, L. (2007). GBKN handboek. Technical Report Documentnummer: 07.05/052, Stichting Landelijk Samenwerkingsverband GBKN, Apeldoorn, the Netherlands.

[66] Geonovum (2007a). Framework van standaarden. Technical Report versie 2.0, Geonovum.

[67] Geonovum (2007b). Informatiemodel geografie (IMGeo). Technical Report version 1.0, Geonovum, Amersfoort, the Netherlands.

[68] Geonovum (2008a). Informatiemodel ruimtelijke ordening (IMRO) 2008. IMRO version 1.1, Geonovum.

[69] Geonovum (2008b). Standaard vergelijkbare BestemmingsPlannen. Technical Report SVBP2008, Geonovum.

[70] GeoServer (2009). Community website. `www.geoserver.org`.

[71] GeoTools (2009). Project website. `www.geotools.org`.

[72] Gottschalk, K., Graham, S., Kreger, H., and Snell, J. (2002). Introduction to web service architecture. *IBM Journal*, 41(2):170–177.

[73] GRASS GIS (2009). GRASS website. `http://grass.itc.it/`.

[74] Groot, R. and McLaughlin, J. (2000). *Geospatial data infrastructure: concepts, cases and good practice*. Spatial Information Systems and Geostatistics Series. Oxford University Press.

[75] Grünreich, D. (1992). ATKIS - a topographic information system as a basis for GIS and digital cartography in germany. *From digital map series to geo-information systems*, Geologisches Jahrbuch Reihe A(Heft 122):207 – 216.

[76] Haesen, R., Snoeck, M., Lemahieu, W., and Poelmans, S. (2008). On the definition of service granularity and its architectural impact. In *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, number 5047 in Lecture Notes in Computer Science, pages 375–389. Springer Verlag, Montpellier, France.

[77] Haire, K. (2001). Active object and agent based approaches to automated generalisation. In *Fourth Workshop on Progress in Automated Map Generalization*, Beijing, China.

[78] Hake, G., Grünreich, D., and Meng, L. (2002). *Kartographie*. de Gruyter, Berlin, 8 edition.

[79] Hardy, P., Hayles, M., and Revell, P. (2003). Clarity - a new environment for generalisation using agents, java, XML and topology. In *Fifth Workshop on Progress in Automated Map Generalization*, Paris, France.

[80] Harrie, L., Sarjakoski, L. T., and Lehto, L. (2002). A variable-scale map for small-display cartography. In *Proceedings of the Joint International Symposium on "GeoSpatial Theory, Processing and Applications" (ISPRS/Commission IV, SDH2002)*, page 6p., Ottawa, Canada.

[81] Harrie, L. and Sarjakoski, T. (2002). Simultaneous graphic generalization of vector data sets. *Geoinformatica*, 6(3):233–261.

[82] Harrower, M. and Bloch, M. (2006). MapShaper.org: a map generalization web service. *IEEE Computer Graphics and Applications*, pages 22–27.

[83] Haunert, J. (2008). *Aggregation in Map Generalization by Combinatorial Optimization*. PhD thesis, Leibniz Universität Hannover.

[84] Hespanha, J., van Bennekom-Minnema, J., Oosterom, P. v., and Lemmens, C. (2008). The model driven architecture approach applied to the land administration domain model version 1.1 - with focus on constraints specified in the object constraint language. In *FIG Working Week 2008 : Integrating generations and FIG/UN-HABITAT*, page 19, Stockholm, Sweden. FIG.

[85] Horak, J., Ardielli, J., and Horakova, B. (2009). Testing of web map services. *International Journal of Spatial Data Infrastructures Research*, (Special issue: GSDI 11):25.

[86] ICA (1973). *Multilingual dictionary of technical terms in cartography*. Steiner, Wiesbaden, Germany.

[87] ICA Commission on Generalisation and Multiple Representation (2009). ICA commission website. `ica.ign.fr`.

[88] Imhof, E. (1972). *Thematische Kartographie*. Walter de Gruyter.

[89] INSPIRE (2007). Directive 2007/2/EC of the european parliament and of the council of 14 march 2007 establishing an infrastructure for spatial information in the european community. *Official Journal of the European Union*, page 18.

[90] ISO (2008). Industrial automation systems and integration – product data representation and exchange – part 11: Description methods: The EXPRESS language reference manual. ISO standard ISO 10303-11:2004, ISO.

[91] ISO (2009). ISO website. `www.iso.org`.

[92] ISO/IEC (2005). Information technology - XML metadata interchange (XMI). Technical Report 19503:2005, ISO.

[93] ISO/TC 211 (2003). Geographic information - rules for application schema. ISO Standard 19109 ISO 19109, International Organization for Standardization.

[94] ISO/TC 211 (2004). Geographic information - simple feature access - part 1: Common architecture. ISO Standard ISO 19125-1, International Organization for Standardization.

[95] ISO/TC 211 (2005). Geographic information - services. ISO Standard 19119 ISO 19119, International Organization for Standardization.

[96] ISPRS working group on Multiple Representation of Image and Vector Data (2009). ISPRS working group website. http://www.ikg.uni-hannover.de/isprs/.

[97] Jabeur, N. (2006). *A Multi-Agent System for on-the-fly Web Map Generation and Spatial Conflcit Resolution*. PhD thesis, Laval University, Quebec.

[98] Jacob, B., Brown, M., Fukui, K., and Trivedi, N. (2005). *Introduction to Grid Computing*. IBM Redbooks. IBM.

[99] Jacobson, I., Booch, G., and Rumbaugh, J. (1998). *The Unified Software Development Process*. Addison-Wesley Longman, Amsterdam, the Netherlands.

[100] Jesus, J. d., Hiemstra, P., and Dubois, G. (2008). Web-based geostatistics using WPS. In *Proceedings of GI-days 2008*, Ifgi-prints. Institute for Geoinformatics.

[101] Jones, C. B. and Ware, J. M. (2005). Map generalization in the web age. *International Journal of Geographical Information Science*, 19(8-9):859–870.

[102] Kada, M. (2007). *Zur massstabsabhängigen Erzeugung von 3D-Stadtmodellen*. PhD thesis, University of Stuttgart, Institute for Photogrammetry.

[103] Keates, J. (1973). *Cartographic Design and Production*. Longman, London, UK.

[104] Kiehle, C., Heier, C., and Greve, K. (2007). Requirements for next generation spatial data Infrastructures-Standardized web based geoporcessing and web service orchestration. *Transactions in GIS*, 11(6):819–834.

[105] Kilpelaeinen, T. (1997). *Multiple representation and generalization of geo-databases for topographic maps*. PhD thesis, Finnish Geodetic Institute.

[106] Klien, E., Fitzner, D. I., and Maue, P. (2007). Baseline for registering and annotating geodata in a semantic web service framework. In Wachowicz, M. and Bodum, L., editors, *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, Aalborg University, Denmark.

[107] Kraak, M. (2001). Settings and needs for web cartography. In Kraak, M. and Brown, A., editors, *Web Cartography: developments and prospects*, pages 1–7. Taylor & Francis, New York, USA.

[108] Kraak, M. and Hootsmans, R. (1999). National mapping organisations and the world wide web, challenges and opportunities. In *19th International Cartographic Conference*, pages 619–628, Ottawa, Canada.

[109] Kraak, M. and Ormeling, F. (2003). *Cartography: Visualization of Geospatial Data*. Prentice Hall, Essex, UK, 2nd edition edition. Distinction of conceptual and graphic generalization.

[110] Kralidis, A. T. (2007). Geospatial web services: The evolution of geospatial data infrastructure. In Scharl, A. and Tochtermann, K., editors, *The Geospatial Web*, Advanced Infromation and Knowledge Processing Series, pages 223–228. Springer, London, UK.

[111] Kuipers, B. (1982). The "map in the head" metaphor. *Environment and Behavior*, pages 202–220.

[112] Lake, R. and Farley, J. (2007). Infrastructure for the geospatial web. In Scharl, A. and Tochtermann, K., editors, *The Geospatial Web*, Advanced Information and Knowledge Processing Series, pages 15–26. Springer, London, UK.

[113] Lamy, S., Ruas, A., Demazeu, Y., Jackson, M., Mackaness, W., and Weibel, R. (1999). The application of agents in automated map generalization. In *19th International Cartographic Conference*, Ottawa, Canada.

[114] Lawrence, V. (2004). The role of national mapping organizations. *The Cartographic Journal*, 41(2):117–122.

[115] Lecordix, F., Gallic, J. L., Gondol, L., and Braun, A. (2007). Development of a new generalization flowline for topographic maps. In *10th ICA workshop on Generalisation and Multiple Representation*, Moscow, Russia.

[116] Lecordix, F., Regnauld, N., Meyer, M., and Fechir, A. (2005). Magnet consortium. In *8th ICA Workshop on Generalization and Multiple Representation*, A Coruna, Spain.

[117] Lehto, L. (2007a). *Real-Time Content Transformations in a Web Service-Based Delivery Architecture for Geographic Information.* PhD thesis, Helsinki University of Technology - Department of Surveying.

[118] Lehto, L. (2007b). Schema translations in a web service based SDI. In Wachowicz, M. and Bodum, L., editors, *10th Agile International Conference on Geographic Information Science 2007*, Aalborg University, Denmark.

[119] Lehto, L. and Sarjakoski, L. T. (2005). Real-time generalization of XML-encoded spatial data for the web and mobile devices. *International Journal of Geographical Information Science*, 19(8-9):957–973.

[120] Lemmens, R. (2006). *Semantic interoperability of distributed geo-services.* PhD thesis, Delft University of Technology.

[121] Liu, Y., Molenaar, M., and Tinghua, A. (2001). Frameworks for generalization constraints and operations based on Object-Oriented data structure in database generalization. In Du, D., Li, H., Wang, J., Zhang, Q., Jiang, J., Liao, K., Wu, H., and Zheng, J., editors, *Proceedings of the 20th International Cartographic Conference*, volume 3, pages 2000–2012, Beijing.

[122] Lutz, M. (2007). Ontology-based descriptions for semantic discovery and composition of geoprocessing services. *GeoInformatica*, 11(1):1–36.

[123] Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors (2007a). *Generalisation of geographic information: cartographic modelling and applications*. Elsevier.

[124] Mackaness, W. A., Ruas, A., and Sarjakoski, L. T. (2007b). Observations and research challenges in map generalization and multiple representation. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalization of Geographic Information: Cartographic Modelling and Applications*, pages 315–323. Elsevier.

[125] MapBuilder (2009). Community website. `http://communitymapbuilder.org`.

[126] McMaster, R. B. and Shea, K. S. (1992). *Generalization in Digital Cartography*. American Association of Geographers.

[127] Nash, E. (2008). WPS application profiles for generic and specialised processes. In Pebesma, E., Bishr, M., and Bartoschek, T., editors, *Proceedings of the 6th Geographic Information Days*, volume 32 of *Ifgi Prints*, pages 69–79, Münster, Germany. Institute for Geoinformatics.

[128] NEN (2005). Basic scheme for geo-information - terms, definitions, relations and general rules for the interchange of information of spatial objects related to the earth's surface. Technical Report NEN 3610:2005, Nederlands Normalisatie-instituut, Delft, The Netherlands.

[129] Neun, M. (2007). *Data enrichment for adaptive Map Generalization using Web Services*. PhD thesis, University of Zurich.

[130] Neun, M. and Burghardt, D. (2005). Web services for an open generalisation research platform. In *8th ICA Workshop on Generalisation and Multiple Representation*, A Coruna, Spain.

[131] Nezhad, H. R. M., Benatallah, B., Casati, F., and Toumani, F. (2006). Web services interoperability specifications. *Computer*, 39(5):24–32.

[132] OGC (2004). OGC web map service interface. Implementation specification.

[133] OGC (2005a). GO-1 application objects. OGC implementation specification OGC 03-064r10, Open Geospatial Consortium, Inc.

[134] OGC (2005b). Web feature service implementation specification. Implementation specification OGC 04-094, Open Geospatial Consortium, Inc.

[135] OGC (2005c). Web map context documents. OGC implementation specification, Open Geospatial Consortium, Inc.

[136] OGC (2007a). OpenGIS web processing service. OGC implementation specification OGC 05-007r7, Open Geospatial Consortium.

[137] OGC (2007b). Styled layer descriptor profile of the web map service implementation specification. Technical Report as.

[138] OGC (2009). OGC website. `www.opengeospatial.org`.

[139] OGC and Whiteside, A. (2007). OGC web services common specification. OGC implementation specification 06-121r3, Open Geospatial Consortium.

[140] OMG (2009). OMG website. `www.omg.org`.

[141] Oosterom, P. v. (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., editors, *GIS and Generalization, Methodology and Practice*, pages 120–132. Taylor & Francis, London.

[142] Oosterom, P. v. (2005). Variable-scale topological data structures suitable for progressive data transfer: The GAP-face tree and GAP-edge forest. *Cartography and Geographic Information Science*, 32(4):331–346.

[143] Oosterom, P. v., Thijssen, T., Alkemade, I., and de Vries, M. (2001). Multi-source cartography in internet GIS. In Konecny, M., editor, *Proceedings of the 4th AGILE Conference*, pages 562–573, Brno, Czech Republic.

[144] OpenLayers (2009). Community website. `www.openlayers.org`.

[145] OpenStreetMap (2009). Community website. `http://www.openstreetmap.org/`.

[146] Ormsby, T., Napoleon, E., Burke, R., Groess, C., and Feaster, L. (2004). *Getting to Know ArcGIS desktop : basics of ArcView, ArcEditor and ArcInfo : updated for ArcGIS 9*. ESRI, Redlands, USA, 2nd edition edition.

[147] Ostländer, N. (2009). *Creating Specific Spatial Decision Support Systems in Spatial Data Infrastructures*. Phd, University of Münster.

[148] Ottens, H. (2004). An information model for strategic spatial policy documents. In *7th AGILE Conference on Geographic Information Science*, pages 605–611, Heraklion, Greece.

[149] Pinet, F., Duboisset, M., and Soulignac, V. (2007). Using UML and OCL to maintain the consistency of spatial data in environmental information systems. *Environmental Modelling & Software*, 22(8):1217–1220.

[150] Poppe, E. and Foerster, T. (2006). Automated application-driven generalization of base maps for DURP. In *Proceedings geo-innovatie 2006 : 4e symposium*, pages 85–87, Ede, the Netherlands. Ruimte voor Geo - Informatie (RGI).

[151] Refractions (2009). uDig website. `udig.refractions.net`.

[152] Regnauld, N. (2006). Improving efficiency for developing automatic generalization solutions. In *ISPRS Workshop: Multiple Representation and Interoperability of Spatial Data*, pages 1–5, Hannover, Germany.

[153] Regnauld, N. (2007). Evolving from automating existing map production systems to producing maps on demand automatically. In *10th ICA Workshop on Generalisation and Multiple Representation*, Moscow, Russia.

[154] Regnauld, N. and McMaster, R. B. (2007). A synoptic view of generalisation operators. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalization of Geographic Information: Cartographic Modelling and Applications*, pages 37–66. Elsevier.

[155] Regnauld, N. and Revell, P. (2007). Automatic amalgamation of buildings for producing ordnance survey 1:50 000 scale maps. *The Cartographic Journal*, 44(3):239–250.

[156] RGI (2009). Rgi website. `www.rgi.nl`.

[157] Roman, D. and Klien, E. (2007). SWING - a semantic framework for geospatial services. In Scharl, A. and Tochtermann, K., editors, *The Geospatial Web*, Advanced Information and Knowledge Processing Series, pages 229–234. Springer, London, UK.

[158] Ruas, A. (1998). OO-constraint modelling to automate urban generalization process. In *Spatial Data Handling 98 Conference Proceedings*, pages 225–235, Vancouver, Canada.

[159] Ruas, A. (2000). The roles of meso objects for generalization. In *Proceedings of the 9th International Symposium on Spatial Data Handling*, pages 3b50–3b63, Beijing, China.

[160] Ruas, A. (2001). Automatic generalisation project: Learning process from interactive generalisation. survey Official Publication No. 39, European Oragnization for Experimental Photogrammetric Research (OEEPE), Frankfurt am Main, Germany.

[161] Saalfeld, A. (1999). Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and Geographic Information Science*, 26(1):7–18.

[162] Sabo, M. N., Bedard, Y., Moulin, B., and Bernier, E. (2008). Toward Self-Generalizing objects and On-the-Fly map generalization. *Cartographica*, 43(3):155–173.

[163] Sankey, T. T. (2008). Scale, effects. In *Encyclopedia of GIS*, pages 1021–1026. Springer Verlag.

[164] Sarjakoski, T. and Sarjakoski, T. (2005). The GiMoDig public final report. Project report.

[165] Sarjakoski, T., Sester, M., Sarjakoski, L. T., Harrie, L., Hampe, M., Lehto, L., and Koivula, T. (2005). Web generalisation services in GiMoDig - towards a standardised service for real-time generalisation. In Toppen, F. and Painho, M., editors, *AGILE 2005*, pages 509–18, Estoril, Portugal.

[166] Schäffer, B. and Foerster, T. (2008). A client for distributed Geo-Processing and workflow design. *Journal for Location Based Services*, 2(3):194–210.

[167] Sester, M. (2005). Optimization approaches for generalization and data abstraction. *International Journal of Geographical Information Science*, 19(8-9):871–897.

[168] Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96–101.

[169] Sivasubramanian, S., Pierre, G., van Steen, M., and Alonso, G. (2007). Analysis of caching and replication strategies for web applications. *IEEE Internet Computing*, 11(1):60–66.

[170] Slocum, T. (1999). *Thematic Cartography and visualization*. Prentice Hall.

[171] Smaalen, J. W. N. v. (2003). *Automated Aggregation of Geographic Objects*. PhD thesis, Wageningen University.

[172] Steiniger, S. and Weibel, R. (2007). Relations among map objects in cartographic generalization. *Cartography and Geographic Information Science*, 34:175–197.

[173] Stollberg, B. and Zipf, A. (2007). OGC web processing service interface for web service orchestration - aggregating geo-processing services in a bomb threat scenario. In *Proceedings of Web and Wireless Geographical Information Systems*, number 4857 in LNCS, pages 239–251. Springer-Verlag, Heidelberg.

[174] Stoter, J. E. (2005a). Generalisation within NMAs in the 21st century. In *22nd International Cartographic Conference*, A Coruna, Spain.

[175] Stoter, J. E. (2005b). Generalization: The gap between research and practice. In *8th ICA Workshop on Generalization and Multiple Representation*, A Coruna, Spain.

[176] Stoter, J. E., Burghardt, D., Duchene, C., Baella, B., Bakker, N., Blok, C., Pla, M., Regnauld, N., and Touya, G. (2009a). Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems*, 33(5):311–324.

[177] Stoter, J. E., Kraak, M. J., and Knippers, R. A. (2004). Generalisation of framework data: a research agenda. In *ICA Workshop on Generalisation and Multiple Representation*, Leicester, UK.

[178] Stoter, J. E., Morales, J., Lemmens, R., Meijers, M., Oosterom, P. v., Quak, W., and Uitermark, H. (2008). A data model for Multi-Scale topographical data. In Ruas, A. and Gold, C., editors, *Headway in Spatial Data Handling*, Lecture Notes in Geoinformation and Cartography, pages 233–254. Springer.

[179] Stoter, J. E., Quak, W., and Hofman, A. (2009b). Harmonising and integrating two domain models topography. In van Loenen, B., Besemer, J., and Zevenbergen, J., editors, *SDI Convergence*, pages 89–106. Netherlands Geodetic Commission (NCG), Delft, The Netherlands.

[180] Sun (2009). Java programming language. `www.javasoft.com`.

[181] Systems, S. (2009). Company website. www.sparxsystems.com/.

[182] Team, N. S. D. (2008). Technical guidance to implement INSPIRE view services. Technical report, INSPIRE.

[183] Timm, J. T. E. and Gannod, G. C. (2007). Specifying semantic web service compositions using UML and OCL. In *IEEE International Conference on Web Services (ICWS 2007)*, pages 521–8.

[184] Töpfer, F. and Pillewizer, W. (1966). The principles of selection. *The Cartographic Journal*, 3(1):10–16.

[185] Turner, A. (2006). *Introduction to Neogeography*. Short Cut. O'Reilly.

[186] Turton, I. (2008). GeoTools. In Hall, G. B. and Leahy, M. G., editors, *Open source approaches in spatial data handlng*, Advances in geographic information, pages 153–167. Springer Verlag, Berlin, Germany.

[187] Vaughan-Nichols, S. (2002). Web services: Beyond the hype. *IEEE Computer*, 35(2):18–21.

[188] Vckovski, A. (1998). *Interoperable and Distributed Processing*. Research Monographs in GIS Series. Taylor & Francis, London, UK.

[189] Visvalingam, M. and Whyatt, J. D. (1993). Line generalization by repeated elimination of points. *Cartographic Journal*, 30(1):46–51.

[190] Vries, M. d. and Oosterom, P. v. (2008). Model generalization and methods for effective query processing and visualization in a web Service/Client architecture. In Belussi, A., Catania, B., Clementini, E., and Ferrari, E., editors, *Spatial data on the web*, pages 85–106. Springer.

[191] VROM (2007). The new spatial planning act gives space. Technical Report 8054, The Netherlands Ministry of Housing, Spatial Planning and the Environment, The Hague, the Netherlands.

[192] VROM (2009). RO-Online website. http://www.ruimtelijkeplannen.nl.

[193] Ware, M. J., Jones, C. B., and Thomas, N. (2003). Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8):743–769.

[194] Warmer, J. and Kleppe, A. (2003). *The Object Constraint Language*. Addison Wesley, second edition edition.

[195] Weibel, R. (1997). Generalization of spatial data: Principles and selected algorithms. In van Kreveld, M., Nievergelt, J., Roos, T., and Widmayer, P., editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340 of *Lecture Notes in Computer Science*, pages 99–152. Springer-Verlag, Berlin, Heidelberg, New York.

[196] Weibel, R. and Dutton, G. (1999). Generalising spatial data and dealing with multiple representations. In Longley, P., Goodchild, M., Maguire, D., and Rhind, D., editors, *Geographic Information Systems - Principles and Technical IssJurg Nievergelts*, volume 1, pages 125–155. John Wiley & Sons, 2nd edition edition.

[197] Weibel, R. and Jones, C. B. (1998). Computational perspectives on map generalization. *Geoinformatica*, 2(4):307–314.

[198] Werder, S. (2009). Formalization of spatial constraints. In Haunert, J., Kieler, B., and Milde, J., editors, *12th AGILE International Conference on Geographic Information Science*, page 13, Hanover, Germany. IKG, Leibniz University of Hanover.

[199] Williamson, I., Rajabifard, A., and Feeney, M. F. (2003). Future directions in SDI development. In *Developing Spatial Data Infrastructures: From concept to reality*, pages 301–312. Taylor & Francis.

[200] Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & sons, Liverpool, UK.

[201] Worboys, M. and Duckham, M. (2004). *A Computing Perspective*. CRC Press (Taylor & Francis), London, UK, 2nd edition edition.

[202] Yang, C. and Tao, C. V. (2006). Distributed geospatial information services. In Rana, S. and Sharma, J., editors, *Frontiers of geographic information technology*, pages 103–120. Springer.

ITC Dissertations

ITC Dissertations are listed online at:

http://www2.itc.nl/research/phd/phd_graduates.aspx