# GENERALIZATION AS AN OPENGIS SERVICE

## REQUIREMENTS AND A CONCEPTUAL MODEL FOR GENERALIZATION IN INTEROPERATING GIS

---

Masters thesis to acquire the postgraduate academic degree
Master of Science in Geographical Information Science and Systems,
Centre for Geoinformatics, University of Salzburg, Austria
submitted by

Dominik Käuferle
of Oberrohrdorf, Switzerland

supervisors:
Prof. Dr. Josef Strobl
Dr. Dirk Burghardt

Zürich 2005

This thesis was conducted following the directives for MSc master theses as part of the UNIGIS MSc course at the University of Salzburg. It was written single-handed by the author.

Diese Arbeit wurde gemäss den Richtlinien für MSc Abschlussarbeiten (Master Thesis) im Rahmen des Studiengangs UNIGIS MSc and der Universität Salzburg erstellt. Sie wurde vom Autor selbständig verfasst.

# Abstract

Key words: GIS, generalization service, generalization constraint, level of detail (LoD), conceptual model, service-orientation, OpenGIS™

Mapping geographic phenomena requires to generalize their representation. If geographic data is to be acquired only once, but shall be reused on the fly at multiple scales and in several applications, generalization must be conducted automatically according to the needs of the application and the user. Automatic generalization may not only be helpful in real-time mobile applications, a field which is often the prime example of current generalization research, but may help to significantly improve the reusability and effectiveness of geographic information in any spatially aware application.

Many approaches have been developed in recent years to implement automatic generalization in GIS, which unfortunately has proven to be considerably complex and difficult to automate. The results of these difficulties were isolated implementations of partial generalization functionality. While the focus was often to develop better generalization algorithms, little thought was spent on how to integrate the different approaches. An increasingly popular and successful approach for integration is known as the concept of *interoperability*, which comprises that interfaces are agreed on so that independently developed components *servicing* specific functions can work together. Such interoperable generalization services might greatly facilitate the development, reuse and integration of generalization in GIS and may finally help to solve the complex problem of automatic generalization. But the question must be answered first, which conceptual elements of generalization services need to be agreed on, in other words which conceptual requirements generalization services need to fulfill.

To contribute to such an answer, this thesis is concerned with identifying a *comprehensive* list of *essential* conceptual requirements for potential generalization services - essential in the sense that they should reflect fundamental objectives of generalization in an interoperable services framework, and comprehensive in the sense that the main requirements necessary to specify interoperable services should be included. The services framework specified by the Open Geospatial Consortium (OGC, 2004) is chosen as a model framework, because OGC specifications such as the Geographic Markup Language GML (OGC and ISO, 2004a) unambiguously define interoperability concepts as well as many important aspects of geographic information, such as data and geometry types. The OpenGIS Reference Model ORM (OGC, 2003a), which describes the concepts underlying the OGC framework, and three use cases are analyzed in order to derive essential conceptual requirements. The use-case specific requirements are crosschecked using a model of generalization constraints proposed by Weibel and Dutton (1999) and a model of graphical variables proposed by Bertin (1973). Based on the identified requirements, the author proposes a simple constraint-based model for generalization services, which should allow to build modular and flexible generalization services for the OGC framework. The list of over 30 essential requirements and the conceptual model proposed in this thesis may be useful as a conceptual basis for the development of generalization specifications and services, or they may be used as a reference to re-consider the design of existing services.

# Zusammenfassung

Schlüsselbegriffe: GIS, Generalisierungsdienst, Contraint, level of detail (LoD), konzeptuelles Modell, Dienste-Orientierung, OpenGIS$^{TM}$

Um geographische Information kartografisch zu erfassen, muss diese generalisiert, d.h. vereinfacht werden. Wenn geographische Daten nur einmal erhoben, aber in Echtzeit für mehrere Massstäbe und in diversen Anwendungen wiederverwendet werden sollen, muss die Generalisierung automatisch erfolgen, gemäss den Bedürfnissen der Anwendungen und Benutzer. Automatische Generalisierung könnte nicht nur für Echtzeit-Anwendungen auf Mobilgeräten von Nutzen sein – ein in aktuellen wissenschaftlichen Projekten zur Generalisierung oft verwendetes Beispiel – sondern könnte die Wiederverwendbarkeit und Effektivität geographischer Information in jeglicher Art von Software mit räumlichen Funktionen wesentlich verbessern.

In den letzten Jahren wurden verschiedenste Ansätze entwickelt, um den Generalisierungsprozess zu automatisieren, welcher sich leider als äusserst komplex und schwierig zu automatisieren erwiesen hat. Das Resultat dieser Schwierigkeiten waren oft isolierte Implementierungen partieller Generalisierungsfunktionen. Während viele Anstrengungen unternommen wurden, um bessere Generalisierungsalgorithmen zu entwickeln, wurde der Integration der verschiedenen Ansätze wenig Beachtung geschenkt. Ein zunehmend erfolgreich angewendeter Integrationsansatz ist unter dem Begriff *Interoperabilität* bekannt und beinhaltet eine Einigung über Schnittstellen, sodass unabhängig voneinander entwickelte *Dienste* unterschiedlicher Funktionalität zusammen verwendet werden können. Interoperable Generalisierungsdienste könnten die Entwicklung, Wiederverwendung und Integration der Generalisierung in GIS stark begünstigen und dazu beitragen, das komplexe Problem der automatischen Generalisierung zu lösen. Jedoch müsste zunächst die Frage beantwortet werden, auf welche konzeptionellen Elemente von Generalisierungsdiensten man sich einigen müsste, mit anderen Worten, welche konzeptionellen Anforderungen Generalisierungsdienste erfüllen müssten.

Als Beitrag zu einer solchen Antwort beschäftigt sich die vorliegende Masters-Arbeit mit der Identifikation einer *umfassenden* Liste *essentieller* konzeptioneller Anforderungen an Generalisierungsdienste – essentiell im Sinn, dass die Anforderungen grundlegende Ziele der Generalisierung in einem interoperablen Dienste-Framework wiederspiegeln sollten, und umfassend in dem Sinn, dass diejenigen Anforderungen eingeschlossen sind, die für die Spezifikation interoperabler Dienste wichtig oder notwendig sind. Das Dienste-Framework des Open Geospatial Consortium (OGC, 2004) dient in dieser Arbeit als Modell-Framework, da OGC-Spezifikationen wie z.B. die Geography Markup Language GML (OGC and ISO, 2004a) Konzepte zur Interoperabilität sowie viele weitere Aspekte geographischer Information, z.B. Daten- und Geometrietypen, eindeutig definieren. Um essentielle Anforderungen abzuleiten analysiert die vorliegende Arbeit das OpenGIS Reference Model ORM (OGC, 2003a), welches die konzeptuelle Basis des OGC-Frameworks beschreibt, sowie drei vom Autor vorgeschlagene Anwendungsfälle. Die Plausibilität und Wichtigkeit der anwendungsfall-spezifischen Anforderungen wird anhand eines Constraint-basierten Generalisierungsansatzes von Weibel and Dutton (1999) und dem bekannten Modell graphischer Variablen von Bertin (1973) gegengeprüft. Aufbauend auf den Anforderungen schlägt der Autor ein einfaches Constraint-basiertes Generalisierungsmodell vor, welches den Aufbau modularer und flexibler Generalisierungsdienste erlauben sollte. Die Liste der über 30 essentiellen Anforderungen könnte als konzeptionelle Basis für die Entwicklung von Generalisierungsspezifikationen und -diensten dienen, oder sie könnten verwendet werden, um den Aufbau bestehender Dienste zu überdenken.

# Acknowledgements

Many thanks to Dani Peter who inspired me to this thesis, when he told me the story of a former practical student who had to redigitize the borders of Kanton Luzern at 1:25’000 because they were ‘only’ available digitally at 1:10’000. Many thanks to my supervisor Prof. Strobl for providing the excellent UniGIS course and to my supervisor Dr. Dirk Burghardt, who spared many hours of his time for valuable discussions and comments. Many thanks to everybody who critically reviewed the text: Alistair Edwardes, Prof. Micheal Goodchild, Moritz Neun, Fredy Städler, Denise Ulrich, Stefan A. Voser and Prof. Robert Weibel. Many thanks to my employer Geoinformation und Vermessung, Kanton Luzern for providing data and technical resources. Many thanks to my wife Christine for every day since we met. Many thanks in advance to any reader who takes an interest in this thesis.

# Table of Contents

# List of Tables

# List of Figures

# 1 Introduction

## 1.1 Problem statement

This thesis is about map generalization in the context of interoperating Geographical Information Systems (GIS). It focuses on the question how generalization could be provided as a service in a standardized service-oriented architecture. The OpenGIS services framework of the Open Geospatial Consortium (OGC, 2004) will serve as a model for such an architecture. The basic promise of generalization services is that high-resolution geographic data could be used dynamically at many scales by many applications over the Internet. The aim of this work is to provide a list of requirements for potential OpenGIS generalization services and a conceptual model describing their essential capabilities and operations.

Electronic information systems are powerful because they separate information content from its myriad representations. This generality allows information systems to use the same information content in a large number of contexts. What we ultimately call content is the data we save in computers - an abstraction comprising particular aspects of reality. What we call representation is usually further abstracting reality in that we select and accentuate certain aspects of interest within the data for display in a text, spreadsheet, graphic or map.

In GIS, information comprises spatial and thematic characteristics of phenomena which have a location on the surface of Earth (e.g. Walford, 2002). The spatial characteristics of phenomena, such as their physical measures, location, size, shape or volume, are abstracted to coordinate-based geometries (one of the great inventions in mapping). In modern electronic GIS, symbology is applied directly to these geometries to represent the underlying phenomena on a map. This approach to create maps is simple and powerful. However, it faces a basic dilemma regarding the resolution of such geometries, forcing the user to trade off the level of detail of the content against the simplicity of the representation, both being equally important objectives in GIS. Whilst a high level of detail, resulting in many objects with high-resolution geometries or many locations in the data, is beneficial in most situations to accurately calculate distances, areas, perimeters etc., a high level of detail may be obstructive when one needs to represent all that detail on a single map, simply because map space is limited. The traditional solution in GIS has been to model, acquire and store several sets of the same data, one per level of detail, often being associated with a duplication of effort and redundancies in the data.

The intention of presenting new concepts for a stricter separation of content and representation and a more flexible handling of scale in GIS may be better explained by evaluating the traditional mapping process and its shortcomings as opposed to a potential electronic mapping process which would overcome these shortcomings. The following two figures highlight some of their differences. Figure 1.1 illustrates the traditional mapping process in GIS concerned with the production of paper maps. Most abstraction, also called generalization, occurs during the production of the data, which is prepared at different resolutions for the use at specific scales. One major shortcoming of the traditional approach is the disconnected and redundant existence of the same objects in several datasets, which makes it time-consuming or impossible to analyze specific phenomena at different scales. Another shortcoming is the temporal and economic effort needed to produce and maintain the different resolution datasets, with associated problems of consistency and data quality.

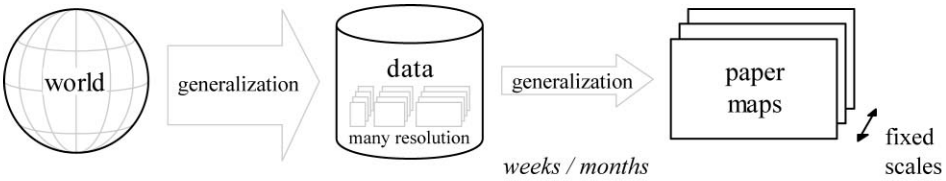


Figure 1.1 The traditional mapping process

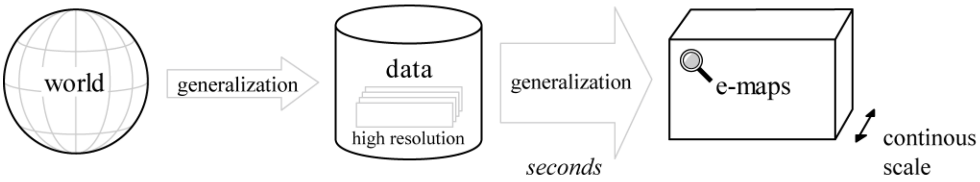


Figure 1.2 A potential electronic mapping process

A future electronic mapping process (Figure 1.2) might forgo different resolutions and store less datasets, but at high resolution. The resolution of the data is down-sampled to a specific scale the moment a user queries an electronic map. All the generalization necessary to adapt the data to the queried map is accomplished automatically in this moment. The main advantage of such a scenario would be the multiple use of data in multiple applications at multiple scales with associated advantages of reduced redundancies and maintenance costs, shorter update cycles as well as increased speed and flexibility for viewing and analyzing geographic information. Systems implementing generalization in such a scenario could be based on distributed and reusable software components, so-called services, with the additional benefit that anybody could provide and use such services. To make generalization services generally available and to allow them to interoperate with a wide range software products over the Internet would require improvements in three basic areas:

a) *Algorithms*, which allow for efficient generalization of high-resolution datasets in real-time.
b) *Models*, which allow the definition of scale-specific characteristics of phenomena and the derivation of scale-specific dataset resolutions.
c) *Interfaces*, which define how generalized data is provided, queried and exchanged between the services in interoperable systems.

The work presented here focuses on conceptual aspects of the last area, interfaces, with associated considerations of conceptual modeling. The main two questions posed in this work are the following:

*What are the requirements for generalization services in interoperating GIS?*

*Which capabilities should generalization services provide?*

Thus, the thesis seeks to contribute to the general issue proposed by Müller et al. (1995): To "Identify the objectives of generalization in the digital context" and to develop "a clear vision of what generalization should be able to accomplish in the digital context."

## 1.2 Overview

The approach presented here is targeted to architects of geographic information systems or spatial data infrastructures as well as developers of generalization components. Generalization is treated equivalent to 'abstraction' of geographical information in this thesis, extending the representational view with issues of data modeling and processing. It is the aim of this thesis to explore questions concerning the integration of generalization operators in a widely accepted framework of interoperating components. The OGC framework has been chosen as a model framework because it is one of the most far-reaching attempts of international cooperation in the GIS industry thus far. The model proposed in this thesis builds on concepts described in specific versions of OGC documents (Table 1.1), some of which have been adopted as ISO standards within the ISO 19100 series.

Table 1.1 OGC document versions

| **title** | **version** | **type[1]-year-ID** | **related to or same as ISO** | **reference** |
|---|---|---|---|---|
| Abstract Specification Topic 0 - Overview | 4 | AS-99-100r1 | - | OGC, 1999a |
| Abstract Specification Topic 1 - Spatial Schema | 5 | AS-01-101 | ISO 19107 | OGC and ISO, 2001 |
| Abstract Specification Topic 12- Service Architecture | 4.3 | AS-02-112 | ISO 19119 | OGC and ISO, 2002 |
| Geography Markup Language (GML) | 3.1.0 | RP-03-105r1 | ISO 19136 | OGC and ISO, 2004a |
| Metadata | 5 | AS-01-111 | ISO 19115 | OGC and ISO, 2000 |
| OpenGIS Reference Model (ORM) | 0.1.2 | ATB-03-040 | - | OGC, 2003a |
| Styled Layer Descriptor (SLD) | 1.0.0 | IS-02-070 | - | OGC, 2002c |
| Web Coverage Service (WCS) | 1.0 | IS-03-065r6 | - | OGC, 2003b |
| Web Feature Service (WFS) | 1.0 | IS-02-058 | - | OGC, 2002a |
| Web Map Service (WMS) | 1.3 | IS-04-024 | ISO 19128 | OGC and ISO, 2004b |

Versions of selected OGC documents, ordered by name. [1] OGC document types: AS = Abstract Specification, IS = OpenGIS Implementation Specification. RP = Recommendation Paper, ATB = Approved Technical Baseline.

The Geography Markup Language (GML) is used for data modeling and exchange in the OGC framework and thus plays a central role for potential OGC generalization services. This thesis refers to GML 3.1.0, the latest specified version at the time of writing. Compared to the two prior main releases, GML 3 has a much richer vocabulary and grammar to describe spatial data. It is acknowledged that at the time of this writing (almost certainly) no Web Feature Services (WFS), Web Map Services (WMS) or Java tools exist that can handle the full complexity of GML 3. Because GML 3 is backward compatible with GML 1 and 2, and applications are not required or expected to use all of the GML schemas, references to GML 1 and 2 are only included in this thesis as an exception.

Following to this introduction, chapter 2 introduces generalization as a multi-faceted topic relating to many aspects of geographic information such as scale, resolution, abstraction, accuracy and data modeling. An overview of interoperability, the scope of the OGC, its relationship to ISO and the basic concepts of the Geography Markup Language GML is also provided.

The largest part of the thesis is devoted to the detection of 'essential' requirements for generalization services with the underlying idea to provide a solid basis for a conceptual model. Requirements should be 'essential' in the sense that they should reflect fundamental objectives of generalization in an interoperable services framework. It is assumed that two types of requirements are essential. One type is of a general nature and deals with principle theories and concepts concerning information or technology rather than with the actual service itself. Chapter 3 analyzes OGC's fundamental models and architecture by examining the OpenGIS Reference Model ORM (OGC, 2003a) one chapter at a time. A second type of requirements deals with the actual generalization functionality, and depends upon user demands. To formalize such requirements, it is common to define use cases. Three use cases for generalization services are proposed in chapter 4. To methodically (and comprehensively) deduct requirements for generalization services, generalization in the use cases is considered from two sides simultaneously, once regarding basic categories of generalization constraints proposed by Weibel and Dutton (1998), and once regarding basic categories of graphical variables proposed by Bertin (1973).

In chapter 0, an outline of a conceptual model for generalization services is proposed based on the requirements resulting from the two earlier chapters. The model formally follows the recommendations for an Essential Model in the OGC's Abstract Specification Overview (OGC, 1999a). A brief discussion of the requirements and the conceptual model as well as some basic proposals for future work are finally given in chapter 6. As it is common in many OGC and ISO specifications, the glossary of key terms are provided as part of the introduction.

## 1.3 Key terms

**client**

Software component that can invoke an operation from a server (OGC and ISO, 2004b)

**cartographic generalization**

Cartographic generalization represents the process of deriving a graphic product or visualization from a source database. (Weibel and Jones, 1998)

**CGI**

"The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers." (W3C, 2004h)

**coverage**

"feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain" (ISO/DIS 19123)

**dynamic generalization**

Generalization mechanisms that are fully automatic and are able to dynamically react on input parameters / Map generalization in the context of interoperable geographic information systems.

**FGS**

Feature Generalization Service: "Service that reduces spatial variation in a feature collection to increase the effectiveness of communication by counteracting the undesirable effects of data reduction." (ISO 19119: OGC and ISO, 2002)

**generalization**

An abstraction process that consists of the reduction, simplification and accentuation of the information contained in a dataset (e.g. Ruas, 2002a)

**geographic data**

"Given facts relating to features which are spatially referenced to the Earth's surface" (Walford, 2002).

**GIS**

Geographic Information Systems

**interface**

"A named set of operations that characterize the behavior of an entity" (OGC, 2002d)

**interoperability**

"A bottom-up integration of existing systems and applications that were not designed to be integrated when they were built" (UCGIS, 1996)

**ISO**

International Standardisation Organisation (ISO, 2004)

**ISO TC/211**

Technical Committee 211 of ISO is concerned with the work on ISO 19100 standard series – Geographical Information. (ISO/TC211, 2004)

**LoD**

Level of Detail

**model generalization**

Generalization due to changes in the conceptual model (Harrie and Sarjakoski, 2002, Ai and van Oosterom, 2001)

**ODP**

Open Distributed Processing (OGC and ISO, 2002).

**OGC**

Open Geospatial Consortium, formerly known as OpenGIS Consortium (OGC, 2004)

**ORM**

"The OpenGIS Reference Model (ORM) provides a framework for the OGC Technical Baseline. The OGC Technical Baseline consists of the currently approved OpenGIS Specifications as well as for a number of candidate specifications that are currently in progress" (OGC, 2004 and 2003a)

**operation**

"A specification of a transformation or query that an object may be called to execute. Each operation has a name and a list of parameters." (OGC and ISO, 2002)

**request**

"Invocation of an operation by a client" (OGC and ISO, 2004b)

**resolution**

a) Refers to the level of spatial or temporal detail within a dataset (OGC and ISO, 2004a) and has an intrinsic similarity to the meaning of scale. In this thesis the term resolution is preferred in the context of data. Scale is preferred in the context of maps.
b) The mapping of names to URIs in connection with namespaces (OGC, 1999c)

**response**

"Result of an operation returned from a server to a client" (OGC and ISO, 2004b)

**RM-ODP**

Reference Model of Open Distributed Processing, defined in ISO/IEC 10746 (ISO, 1996)

**scale**

Four different scale connotations after Bian (1997):

a) Map scale: a larger scale map provides more detail (traditional cartographic meaning)
b) Geographic scale: spatial extent of a study area, "...also called extent or domain in ecology literature (Turner et al., 1989; Stoms, 1994)"
c) Resolution: size of the smallest distinguishable part of a spatial data set
d) Operational scale: referring to the scale at which a phenomenon operates

**scaling**

Process of changing the scale

**scaling behavior**

The way that feature properties change when gradually changing scale

**server**

Actual implementation of a service (OGC, 2002b)

**service**

"... distinct part of the functionality that is provided by an entity through interfaces" (OGC, 2002d). In other words, a well defined set of actions.

**SOA**

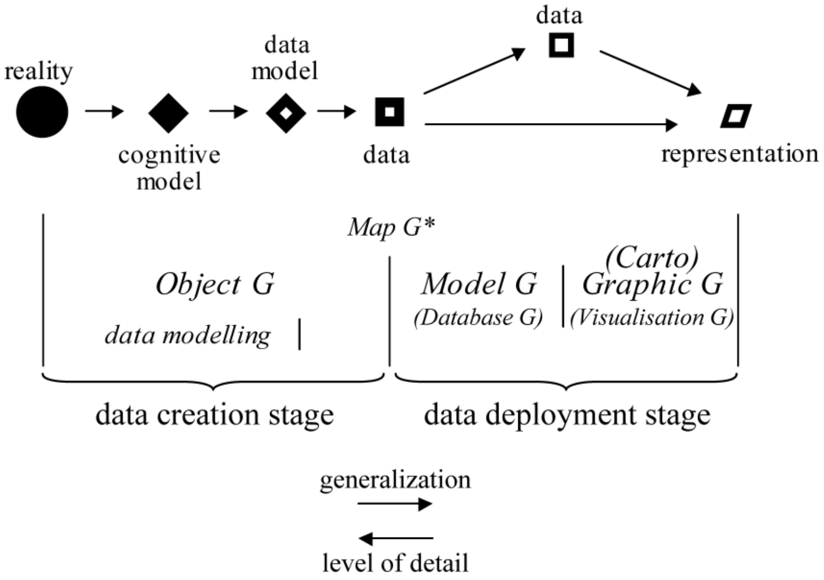
Service-oriented architecture. "A service-oriented architecture is essentially a collection of services." (Service-architecture.com, 2004)

# 2 Fundamentals

## 2.1 Terms and Concepts concerning Generalization

“The questions what the map generalization process is and how to describe the process are basic issues in the research field of generalization conceptual modeling” (Ai and van Oosterom, 2001). The purpose of the following paragraph is to introduce into generalization as a multi-faceted conceptual topic together with some of the related terminology.

The basic working material of geographic information systems is spatial data. From a data viewpoint, two principle stages can be distinguished. One stage occurs up to the initial creation of the data, the other occurs in processes applied to the data from then on (e.g. analysis, visualization, fusion, ...). Thus, the two stages could be called *data creation* and *data deployment stage*. No matter how spatial data is created in the data creation stage, there is always generalization in the sense of an abstraction process involved (Figure 2.1). The fundamental reason is that the world contains an approximately infinite amount of information at every given location (e.g. Goodchild, 2001, Weibel and Dutton, 1999), but that for humans or computers it is only possible to store and process a finite amount of information.



All data modeling and acquisition activities up to the initial creation of data can be seen as part of a *data creation stage* regarding generalization, and all activities using or transforming the data afterwards as a *data deployment stage*.
* G = generalization.

Figure 2.1 Generalization terminology and two stages of generalization

“Generalization ... is intrinsically related to the term ‘abstraction’ … (latin, *abstrahere*)” (Brassel and Weibel, 1988). Abstraction in the *data creation stage* is a transformation process, typically comprising the steps of developing cognitive models, then data models and finally digital data. The formalization of this abstraction is known as *data modeling*, and its results are *data models* (schemas) that serve as well defined blueprints for the actual data. The term *object generalization* has occasionally been used in generalization literature to refer to generalization performed during the initial data model creation (e.g. Grünreich, 1995, Weibel, 1996, Bobzien and Morgenstern, 2003). Object generalization aims at abstracting geographical phenomena to an extent to which their important characteristics are preserved regarding a specific level of detail and map purpose.

The *data deployment stage* (Figure 2.1) can be subsumed as a transformation of spatial data from a high level of detail to a lower level of detail. It's focus is the analysis or the communication of geographical concepts. Because it is usual to communicate geographical concepts in the form of maps, the term *map generalization* is widely used as an umbrella term for stage one and two generalization (e.g. McMaster and Shea, 1992, Buttenfield, 1995, Müller et al., 1995, Ware et al., 2003). A trend can be observed in generalization literature to use the term map generalization more specifically for the data deployment stage and to subdivide that stage into model generalization and (carto-)graphic generalization (e.g. Grünreich, 1995, Sester, 2001, Bobzien and Morgenstern, 2003). "Map generalization should never be equated merely with simplification and scale reduction. On the contrary, it represents a process of informed extraction and emphasis of the essential while suppressing the unimportant, maintaining logical and unambiguous relations between map objects, maintaining legibility of the map image, and preserving accuracy as far as possible." (Weibel and Jones, 1998).

*Model generalization* has also been referred to as database generalization (Zhou et al., 2000) or statistical generalization (Brassel and Weibel, 1988), though the meaning of the terms might differ slightly. Model generalization aims at the controlled data reduction in the spatial, thematic, and/or temporal domain without considerations to the visual representation (Ai and van Oosterom, 2001). The result is new data that has simplified values, e.g. simplified geometries, and/or a simplified schema, e.g. less attributes or simplified classifications. *Cartographic generalization* in contrast deals with the abstraction of graphical symbology considering limited map space (Weibel and Dutton, 1999). It has to resolve spatial conflicts that occur between graphical objects by applying operations such as elimination, displacement, collapse or aggregation (e.g. Doihara et al., 2002). Cartographic Generalization has also been called visualization generalization (Ai and van Oosterom, 2001) or view generalization (Burghardt et al., 2004).

To extend the view on the data deployment stage, it should be noted that many important data deployment activities aim at the opposite of model generalization, namely to increase the information content and complexity of geographic data. These activities can be classified into three groups:

- Editing: adding data to existing geographic data
- Joining: combining geographic data with other data based on unique identifiers
- Overlay: combining sets of geographic data based on their common spatial-temporal properties (an idea probably introduced by I. McHarg, 1969, and contributing significantly to the power of GIS)

Several authors in the generalization research field have attempted to comprehensively describe the generalization process and proposed conceptual models for the automation of generalization (e.g. Morrison, 1974, Brassel and Weibel, 1988, Nickerson and Freeman, 1986). Among the most general approaches is the model of McMaster and Shea (1992), who partition the problem of generalization in the three basic areas *philosophical objectives* (why to generalize), *cartometric evaluation* (when to generalize) and *spatial & attribute transformations* (how to generalize), and provide lists of elements associated with these areas. For instance, they decompose the philosophical objectives into three types of elements: theoretical, application-specific, and computational (Table 2.1).

Table 2.1 Why to generalize (by McMaster and Shea, 1992)

| **theoretical elements** | **application-specific elements** | **computational elements** |
|---|---|---|
| 1. reducing complexity<br>2. maintaining spatial accuracy<br>3. maintaining attribute accuracy<br>4. maintaining aesthetic quality<br>5. maintaining a logical hierarchy<br>6. consistently applying generalization rules | 1. map purpose and intended audience<br>2. appropriateness of scale<br>3. retention of clarity | 1. cost effective algorithms<br>2. maximum data reduction<br>3. minimum memory / disk requirements |

The complete model can be found in McMaster and Shea (1992) or Weibel and Dutton (1999).

Why, when and how to generalize might be extended with a fourth area, which is underlying all of these: *what* to generalize? The subjects of generalization, i.e. the geographical data generated in natural and social sciences, are bearers of many problems concerning the representation of complex interrelationships and processes between real-world phenomena at different scales (e.g. UCGIS, 1998, Goodchild and Quattrochi, 1996, Renschler, 2002). Apart from the simple geometries proposed by ancient Greek philosopher Euclid, science has not provided many comprehensive concepts to systematically deal with the spatial complexity of our World up to date. Two exceptions to this rule are worthy of mention: fractal dimensions (Mandelbrot, 1977) and the 'first law of geography' (Tobler, 1970). Fractal dimensions are one of the very rare comprehensive mathematical concepts that extend the Euclidian space to describe complex natural forms[1]. Tobler's law of geography refers to the phenomenon of spatial auto-correlation known from spatial statistics. It simply states that all things are related, but nearby things are more related than distant things[2]. Ecological and environmental sciences are strongly exposed to problems of complexity and scale and have provided some valuable (mainly non-mathematical) concepts, e.g. Allen and Starr (1982), whose merit is an in-depth discussion of the coherences between the difficult topics of complexity, hierarchy, scale and detail in nature, or Wiens and Milne (1989), who emphasize the non-objectivity of scale[3]. A conceptual

---

[1] Complex fractal forms are described with a 'fractal dimension' that is a real number between the Euclidian dimensions (0, 1, 2 and 3). For example, 1.9345 for a linear structure that almost fills a plane.

[2] The validity of this principle can easily be tested by trying to imagine a world where it was not true. "Such a world would be impossible to describe or inhabit, since the full range of variation could be encountered over vanishingly small distances" (Goodchild, 2003).

Milne (1989), who emphasize the non-objectivity of scale[3]. A conceptual model for generalization in interoperable systems is not necessarily directly concerned with the underlying problems of scale and complexity in the data. This task can be delegated to the data modelers. But a framework for generalization must be flexible and allow to account for the various problems of scale in a way that does not violate the concepts defined in the data models.

## 2.2 Generalization as a Technology

Major efforts have been made in recent years in academia and industry to develop algorithms and methods to perform automatic generalization in GIS. Despite these efforts, the complexity of most generalization tasks still leaves many open questions to be explored. “What makes generalization so difficult is that there is no unique solution, but numerous constraints have to be taken into account…” (Zhou et al., 2000). Many generalization methods and algorithms have been developed to address these problems, some of which are shortly explained in the following incomplete list:

- The *Douglas-Peucker-Ramer* algorithm (DPR) is an effective polyline generalization method that was proposed independently by Ramer (1972) and Douglas and Peucker (1973). DPR calculates the perpendicular distance between a vertex and the connecting line of neighboring vertices, and compares the distance with a predefined threshold. DPR is widely used and several improvements have been proposed, e.g. topological checks to avoid (self-)overlaps (McKeown et al., 1999).
- *Voronoi/Delauney Methods*: Voronoi diagrams and Delauney triangulations are mathematical methods to describe tessellations of Euclidean spaces. There is a wide field of applications concerning Voronoi diagrams (e.g. Okabe et al., 2000). Voronoi and Delauney algorithms can be deployed in generalization when deleting and reinserting vertices in linear geometries during the simplification process (e.g. Gold and Thibault, 2001, Mostafawi et al., 2003).
- *Optimization methods* treat the resolution of spatial conflicts as an optimization problem, which can be solved iteratively. Polynomial splines, so-called snakes (Burghardt and Meier, 1995) or gradient descent and simulated annealing (Ware and Jones, 1998, Ware et al., 2003) are examples of line displacement and smoothing algorithms using optimization. Harrie and Sarjakoski (2002) proposed least-squares adjustment and conjugate-gradient optimization methods to simultaneously resolve graphical conflicts between multiple map objects.
- *Agent-based methods* regard map objects as agents (Duchêne, 2003). Agents act autonomously and are capable to intelligently interact with other agents (Wooldridge, 2002). In map generalization, agents negotiate their precedence to resolve spatial conflicts (AGENT, 1999). Galanda and Weibel (2002) and Galanda (2003) have demonstrated the effectiveness of this approach for categorical polygon data.

---

[3] Their argumentation is that the same process (e.g. landscape change) can have very different effects on different organisms (e.g. a beetle, a bird or a human) and is perceived very differently by these organisms.

- *Lattice-based methods* impose a lattice (grid) on the map space with a cell size of the target resolution. The grid cells are scanned in one step (an analogy to charged-coupled device CCD technology in digital cameras). Functions performing collapse, aggregation, separation or displacement can be initiated on cell-contained points or vertices based on the scan results. An implementation of this concept is the adaptive lattice model (ALM) proposed by Doihara et al. (2002) and Wang et al. (2002).
- *Imagery methods*: Opposed to the methods above which are mostly concerned with vector data, there is a whole field of research concerned with the generalization of raster data (e.g. remote sensing). Raster analysis and transformation is often focused on the recognition of shapes or on the description of variability within the data. Important tools in this respect are fractals and multifractals (e.g. Goodchild and Mark, 1987, Lam and Quattrochi, 1992, Pecknold et al., 1997, Frohn, 1998), variograms or box-counting (Xia and Clarke, 1996) or block vs. focal methods (e.g. Van Paddenburg and Wachowicz, 2001).

To use generalization methods on large data volumes in real-time applications over the Internet, the speed of data transmission is a central problem. Recently, Bertoloto and Egenhofer (2001) have proposed a mechanism to progressively transmit geographical vector data, which allows to start operations on large sets of data before the whole dataset is loaded. The W3C is currently evaluating the possibilities for a binary version of XML (W3C, 2004d) to allow faster XML access and transmission, which might be a great help for directly processing GML over the Internet.

One branch of activities in the generalization research community in recent years has focused on multi-representation databases (Buttenfield, 1993, Egenhofer et al., 1994, Jones et al., 1996, Timpf and Devogele, 1997, Devogele et al., 1998, Vangenot, 2001, Vangenot et al. 2002, Ruas, 2002b, Kreiter, 2002, Cecconi, 2003). Their concept is to store multiple interconnected LoDs (levels of detail) in one database, and to perform automatic generalization only between those LoDs. A key advantage opposed to file-based storage is that relationships can be modeled and managed between the different representations within the database. Problems of the multi-representation approach are its high storage, modeling and maintenance overhead and the fact that different applications may require mainly LoDs that are not directly stored in the database (Zhou et al., 2002). However, the multi-representation database and the real-time generalization approaches are not mutually exclusive.

To conclude, map generalization is a complex issue for which no simple or isolated solutions exist (Weibel and Dutton, 1999), and it seems to be part of a basic dilemma. Though we wish to have access to geographical information as detailed and well-described as possible (to compute accurate results), in the moment we do use it, we also demand the freedom to only use those parts that matter to us. Additionally, we desire to do so in a user-friendly, quick and visually effective way, without unnecessary detail that distracts our focus. Some possible solutions to this dilemma may be offered by deploying the concept of interoperability, which is the foundation of the OpenGIS framework and of the generalization services considered here.

## 2.3 Interoperability, OGC and ISO

Interoperability can for example be defined as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" (IEEE, 1990). Interoperability is undoubtedly beneficial for end users. The following well-known real-world examples of successful interoperable systems may illustrate this:

- CDs / MP3 files can be played with any CD / MP3 player available worldwide.
- HTML, the hypertext markup language, can display web sites with any HTML conformant web browser around the world.

In the case of GIS, interoperability means that users with different GIS systems on different hardware platforms can use, share, display, process and exchange the same geographic information (e.g. Sondheim et al., 1999). The local hard disk is not necessarily the best place to store and maintain data and programs. Distributed services can be used to access and process remote information over the Internet or other networking technologies such as the Grid (see chapter 3.1.1 for some notes on grid computing and GIS). To achieve interoperability, application designers must specify how their systems exchange and use information. This collaborative effort is called *standardization* and is usually conducted by industry and university consortiums or national and international standardization organizations (Table 2.2).

Table 2.2 Some organizations involved in the standardization of information services

| | | |
|---|---|---|
| W3C | World Wide Web Consortium | http://www.w3.org/ |
| OASIS | Organization for the Advancement of Structured Information Standards | http://www.oasis-open.org/ |
| GGF | Global Grid Forum | http://www.ggf.org/ |
| IETF | Internet Engineering Task Force | http://www.ieft.org/ |
| OMG | Object Management Group | http://www.omg.org/ |
| OGC | Open Geospatial Consortium | http://www.opengeospatial.org/ |
| ISO | International Standardization Organization | http://www.iso.org/ |

"The developers of a specification … have to ensure that what they specify can be implemented by all the different implementers: it should not depend on a certain platform, it should not be open to different interpretations, and, if possible, it should be testable" (Bos, 2004). Although interoperability may be beneficial for many consumers and software companies, the standardization required to reach interoperability faces at least two principle difficulties, which may expose any approved standard to the risk of non-acceptance in the markets:

- Complexity. It is relatively easy for two parties to agree on the solution of a very particular problem. If complex demands and technologies are involved and many parties must agree on a solution, standardization is difficult and disagreement is likely.
- Speed of change. By their nature, standardization efforts take time and at the time of their approval, they may not reflect market needs anymore. Standards may themselves

be subject to change and may make it difficult for software developers and customers to decide on adjusting or redesigning their current systems. At any stage, the benefits of the new standards must outweigh the cost of redesigning or exchanging systems.

Despite its problems, standardization is a common procedure in the Information Technologies community and the markets have quickly accepted many Internet standards such as IP, DNS, HTTP and XML.

One of the most active international organization involved in standardizing geographic information and services in recent years has been the Open Geospatial Consortium (see OGC, 2004 or Kottman, 1999). The OGC describes it's mission and activities as follows: "OGC is a not-for-profit trade association dedicated to promoting new technical and commercial approaches to interoperable geoprocessing. The OGC was founded in response to widespread recognition of the problem of non-interoperability in the geospatial industry and the many negative ramifications for industry, government and academia. ... the vision of OGC is that of a national and global infrastructure in which geospatial or location referenced data and geoprocessing resources move freely, fully integrated with the latest distributed computing technologies, accessible to everyone. ... The core mission of OGC is to deliver spatial interface and encoding specifications that are openly and publicly available for global use. This mission is achieved through organizing interoperability projects, working toward consensus, formalizing OGC specifications, developing strategic business opportunities and standards partnerships, and promoting demand for interoperable products" (OGC, 2003a). This statement makes clear that the OGC's activities are not only about web mapping, which is one of the common misconceptions about the OGC, but rather about the integration of 'geospatial' into all main-stream information technologies. Among the most prominent and known OGC specifications are:

- A comprehensive conceptual framework for geographic information, including the feature model.
- A comprehensive language to describe geographic information, the Geography Markup Language (GML).
- A comprehensive language to perform queries on geographic databases, called Feature Encoding (FE).
- A comprehensive language to symbolize and graphically represent geographic information, called Styled Layer Descriptor (SLD).
- Specifications for a framework of web-based services including: Web Map Services (WMS), Web Feature Services (WFS), Web Coverage Services (WCS), Coordinate Transformation Services (CTS), Catalog Services and Gazetteer Services.

Most of the encodings that the OGC has published for services and geographic data are based on the Extensible Markup Language XML (W3C, 2004f) and modeled using the Unified Modeling Language™ UML (UML, 2004). Many of the underlying ideas of OGC encodings have been borrowed from XML and related specifications, such as the Resource Description Framework (W3C, 2004b), in order to comply with the developments of the Internet as defined by the World Wide Web Consortium.

An important standards partnership of the OGC in the last years has been the cooperation with the International Standardisation Organization ISO, a de-jure standards organization for all issues of standardization. “A technical committee of the ISO, TC/211, has been working on a unified set of standards for geographic information since 1994. TC/211 has published a collection of standards in the 19100 series, most of which provide an abstract framework for the description of geographic objects, including their relationships and coordinate reference systems.” (Lake et al., 2004). Some ISO standards of the 19100 series are strongly influenced by the OGC’s work or are approximately the same as OGC specifications. Conversely, some of the more recent OGC specifications (e.g. GML 3.0 and 3.1) have been based on ISO/TC 211 specifications.

## 2.4 Geography Markup Language (GML)

The Geography Markup Language is a key component of the OpenGIS interoperability framework. GML consequently plays a central role in the development of potential generalization services within the framework. This section shall give a very short introduction into GML.

“Geography Markup Language is an XML grammar written in XML Schema for the modeling, transport, and storage of geographic information. A geographic **feature** is ‘an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth’” (OGC and ISO, 2004a, see also Greenwood and Hart, 2003). Features may thus be abstractions of phenomena such as landscape elements, weather phenomena, waterways, plants, animals, human constructions, air pollution concentrations and so on. “The number of **properties** a feature may have, together with their names and types, are determined by its type definition. ... The state of a feature is defined by a set of properties, where each property can be thought of as a {name, type, value} triple” (OGC and ISO, 2004a).

**Geometries** of features are properties with a geometry as a value. GML defines the types of geometry that can be used for geometric property values. While GML version 1 and 2 (OGC, 2000 and 2002d) predefined only few geometry types, the latest published version, GML 3.1 (OGC and ISO, 2004a) offers more geometry types including curves and surfaces and the possibility to user-define geometries. A feature may have several geometric properties, e.g. a point property as an indication of its location, a polygon property as a representation of its shape and another polygon as a generalized representation of its shape. A feature may contain one or several other features, in which case it is called a **feature collection**.

To encode specific features in GML, such as the Kappel bridge in Luzern (Switzerland), the GML user creates **feature instances**, which must have unique identifiers within a GML document. To create the instance, the user must have a schema (blueprint) for bridge instances, that defines which properties and structure a bridge may consist of. Such schemas are called **application schemas** in GML and they can be created by anyone. Application schemas contain structural definitions of concrete phenomena such as bridges, rivers, vegetation cover, temperature distribution a.s.o., but not the descriptions of the phenomena themselves. To be GML application schemas, they must be based on the so called GML **core schemas**, which are maintained by the OGC. The 25 core schemas in GML 3.1 (OGC and ISO, 2004a) define the basic elements, so-called classes, to describe features, geometries, links, coverages (e.g. raster), property units, topology, styles, temporal types and many other GML objects. Technically, application and core schemas are **XML schema** documents (W3C, 2004i), which can make use of other schemas by importing them at the beginning of the XML document. Figure 2.2 depicts the interrelationships between GML schemas and instances, and who (usually) creates and maintains them.

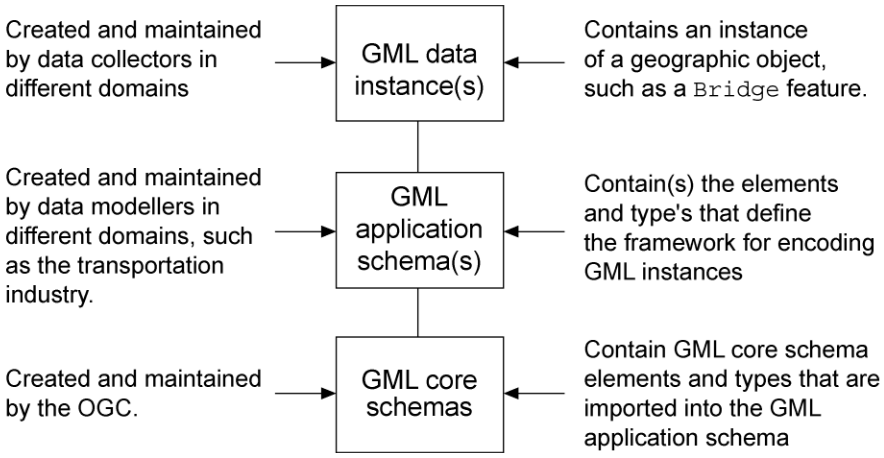


Figure 2.2 GML schemas and instances (Lake et al., 2004)

Several key concepts distinguish GML from today's de-facto geographic encoding standards such as the shapefile (ESRI, 1998):

- A feature may have more than one geometric property.
- A feature may consist of other features (feature collection).
- Feature relationships, topologies and temporal properties can be described in GML.
- The current version 3.1 (OGC and ISO, 2004) provides numerous geometric primitives to build geometries in 0 to 3 dimensions. Various types of complex geometric constructs can be assembled from the geometric primitives.
- Raster data representing discrete or continuous geographic phenomena can be modeled in GML. "OGC uses the term *coverage* to refer to any data representation that assigns values directly to spatial position" (OGC, 2003a). Examples of coverages include raster imagery, observations and measurement data.
- Features, properties and schemas can be remote. They can be included in a GML document from a remote GML document over the Internet by referring to that source with XLinks (W3C, 2004g).
- XML namespaces are used in GML to ensure that names within application and core schemas are unique.
- GML 3.1 is based on international standards defined by the ISO. (Annex D of the GML 3.1 Recommendation Paper [OGC and ISO, 2004a] describes the relationships to corresponding ISO 19100 standards)

- Anyone can use the GML core schemas to build his/her own application schemas.
- GML is an object-oriented modeling approach. It is not bound to the relational structure of tables to describe feature properties. Mapping between relational models and GML and vice versa is possible (Lake et al., 2004), but not always trivial.
- „Like XML, GML is not simply a collection of elements and attributes used to encode geographic documents. It is also a language that provides mechanisms for structuring and defining complex models of geographic information that can be applied in a wide range of geospatial applications" (Lake et al., 2004)

GML is not the only XML-based language for encoding geographic information. For example, the Data Promotion Center DPC in Japan, a non-profit organization funded by the Japanese Ministry of Economy, has been developing G-XML since 1999 (G-XML, 2004). OGC and DPC agreed to "incorporate several key concepts from G-XML into GML 3.0" (Lake et al., 2004) and with the release of G-XML 3.1 in April 2004, G-XML has been specified as a well constructed application schema of GML 3.1. In Switzerland, KOGIS (a governmental coordination group for GIS) has developed Interlis since 1991. Interlis 1 has become a national standard for Swiss cadastral surveying and the XML-based Interlis 2 has been published as a Swiss standard in 2003 (Interlis, 2004). However, GML has received far more attention because of its international scope and funding and is likely to have a stronger and more sustainable impact on the GIS industry than any national initiative. With 'ISO 19137 - Generally used profiles of the spatial schema and of similar important other schemas' work is on the way that shall allow to "integrate existing standards and de-facto standards such as GeoVRML, ALKIS (Germany), Interlis (Switzerland), and DIGEST (NATO) into the world of ISO 19100 standards" (Kresse and Fadaie, 2004).

# 3 Generalization Service Requirements based on the OpenGIS Reference Model

One of the first questions to answer when designing new software such as generalization services is what the software should be able to do. The answers to this question can be expressed as optional and mandatory requirements. Requirements derive from a context, such as customer needs, the state of technology, time-to-market considerations or legal obligations. There are many different levels of granularity for requirements. While some requirements relate to the wider conceptual framework that the software or service should comply to, e.g. defined by modeling, encoding or architecture frameworks, other requirements relate to specific user needs or the type of information used by the service. Examples of use-case-specific requirements will be covered later on in chapter 4. In the following chapter, requirements for web-based generalization services are proposed that relate to the foundational concepts of the OGC framework, which the OGC describes in the *OpenGIS Reference Model* ORM (OGC, 2003a). The methodology in this chapter is to follow the sections of the ORM in order to derive conceptual requirements for generalization services from their contents.

The ORM is organized in five viewpoints, a scheme which is inherited from the Reference Model for Open Distributed Processing RM-ODP (ISO/IEC 10746-1 to 4), an ISO standard for architecting open distributed processing systems (Figure 3.1).

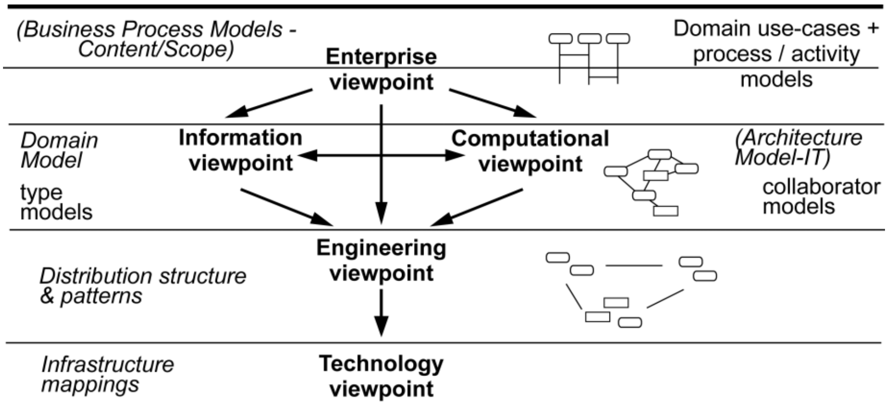


Figure 3.1 RM-ODP viewpoints (OGC, 2003a)

In the ORM, the RM-ODP viewpoints are used to describe the characteristics of the OpenGIS framework. The five viewpoints in the ORM (OGC, 2003a) are:

- Enterprise viewpoint - describes the OpenGIS framework in terms of its purposes and business perspectives.
- Information viewpoint - describes the OpenGIS framework in terms of its content, focusing on the semantics of information and information processing.

- Computational viewpoint - describes the OpenGIS framework in terms of its functions and contains definitions of what services, interfaces, operations and service metadata are, how services should be classified and typed and how they may interact.
- Engineering viewpoint - describes the OpenGIS framework by relating the theoretical concepts to specific components or ‘tiers’, along a network.
- Technology viewpoint - describes the OpenGIS framework by capturing how information is encoded for runtime use.

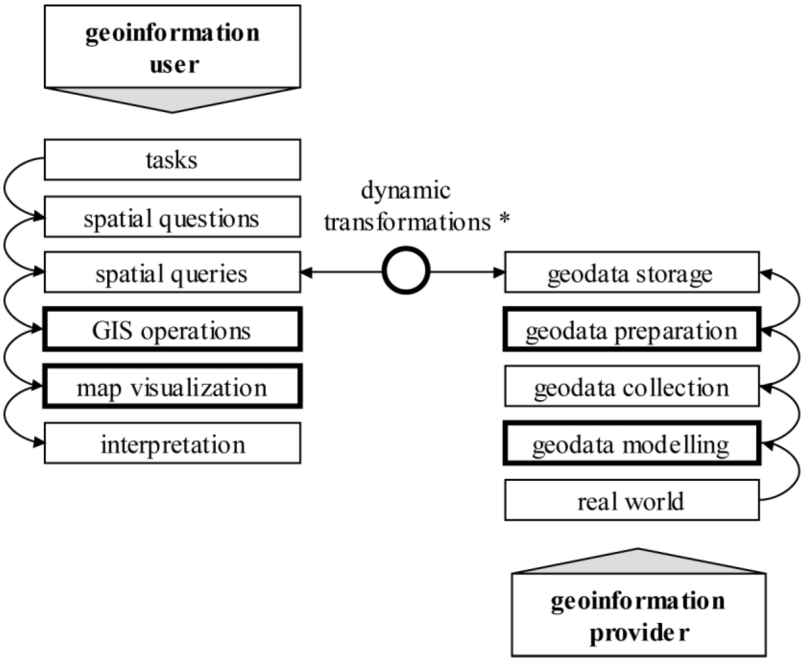
In the sections below, the concepts in each of the five ORM viewpoints will be shortly described or partly cited and then commented. The length of the citations and comments may vary considerably and should reflect the complexity of the corresponding issue or its importance concerning generalization and conceptual modeling. The formulation of the requirements is necessarily subjective. The reader may test the validity of the requirements by asking inverse questions such as: If a generalization service would *not* fulfill the requirement, would it still comply with the ORM? The requirements are flagged with numbers in square brackets, e.g. [3], for easier reference. Table 3.3 and Table 3.4 at the end of the chapter provide recapitulating lists divided into mandatory and optional requirements.

## 3.1 ORM Enterprise Viewpoint

### 3.1.1 Generalization Value Chains

The enterprise viewpoint briefly describes the OGC’s “business perspective, purpose, scope and policies” (OGC, 2003a), and introduces *geospatial location* as being integral to all aspects of the ORM. The argumentation is that geospatial location, as a ubiquitous information ingredient, is a foundational property for modeling the world and that location and time can be used as unifying identifiers to relate previously unrelated information. The enterprise viewpoint further illustrates an informative value chain to express the stages in which the value of geospatial information is increased from raw data to finished geoinformation products. However, the ORM enterprise viewpoint does not mention generalization. It shall therefore be tried to sketch out a business perspective for users and providers of generalization in the OGC framework. Figure 3.2 depicts such a perspective in the form of a simple user-provider model and their task chains, i.e. the value of generalization for users and providers is evaluated following their task chains.

The task chain of the geoinformation user starts with any sort of real-world problem. To effectively solve the problem users defines tasks (Timpf, 2003) and questions that relate to the problem. If the questions contain spatial question words (where, how far, within, ...) or a location, users need to find and access geographic information and define queries against existing geographic data stores. With potentially helpful data at hand, the user often has to apply a number of specialized GIS operations, e.g. to convert, analyze or generalize the data and he or she may use generalization and appropriate symbolization for visual display on a map. The interpretation of the customized data or its visual representation should contribute to answer the questions and to solve the initial problem.



Task chains for the two basic roles of geoinformation users and providers. The bold outlines indicate where model or cartographic generalization may take place. *Dynamic transformations during data query may consist of coordinate transformations, filtering operations, dynamic generalization or other transformations.

Figure 3.2 Generalization in a user-provider model

The focus of the user in this process is usually to solve the initial problem, and a central factor for a successful solution is the availability of quality information. The level of detail (LoD) in the data is an important facet of data quality for at least two reasons:

- the LoD should be appropriate for analysis and visualization
- it may be useful to represent the same information at different scales

In case the data at hand is by magnitudes too detailed (e.g. if retrieved from a high-resolution data store), both reasons may require to generalize the data. It could be a major advantage for users if the generalization process could be delegated to fully automatic generalization services plugged in the data query process (circle in the center of Figure 3.2), especially if this generalization was able to enhance the result in respect to the task at hand. The benefits would be gained flexibility and speed in accessing and deploying geographic information. From the user task chain, the following requirement can be deducted:

[1] Generalization services should be able to account for the users task at hand, e.g. by treating information that is more important for a task specifically.

The task chain of providers reveals quite different tasks. Data is produced in several stages including modeling, collection, preparation and storage, with potential generalization steps prevailing in data modeling and preparation. While the conceptual abstractions in the data modeling process must include the work of humans (object generalization), data preparation (sometimes including model generalization) may comprise deterministic production steps and could be done automatically. The difficulty for information providers is to provide quality data for multiple uses of data, because the data quality required by the user depends on the user's task at hand. The provider can neither envision all probable uses before data acquisition nor can he pre-produce tailored datasets or finished products for all these uses. Where this is tried, a problem of updating the information in all these datasets arises quickly. At the same time, data providers often coincide with being the producers of high quality mapping products (topographic maps, street maps, cadastre plans etc.) at different scales and they may wish to accelerate the production of these products using one single high-resolution data source. The benefits of generalization services for providers may be a facilitation and promotion for the multiple use of high-resolution data at different scales and levels of detail and thus a wider customer base, as well as reduced production costs for high quality mapping products. From the providers task chain, the following requirement for generalization services can be deducted:

[2] Generalization services should provide the possibility for application providers to adapt the service to their specific needs.

### 3.1.2 Requirements on OGC Technologies

The enterprise viewpoint proceeds with a list of high-level requirements which OGC technologies must fulfill to support the geospatial information value chain. Some aspects of this list include (please refer to OGC, 2003a for the complete list):

- openness (support of standard interfaces / component architectures / independently developed implementations of services; adaptability to changing business and operational requirements)
- accommodation of authentication, security, privacy features
- platform independence
- vendor neutrality
- data content format independency

Consequently these requirements are also valid for generalization services:

[3] Generalization services must provide standard query interfaces, be platform and data content format independent as well as vendor neutral.

## 3.2 ORM Information Viewpoint

### 3.2.1 Geographic Features

"The starting point for modeling of geographic information is the geographic feature. A feature is an abstraction of a real world phenomenon. A geographic feature is a feature associated with a location relative to the Earth. A digital representation of the real world can be thought of as a set of features. ... Geographic features occur at two levels: feature instances and feature types .... At the instance level, a geographic feature is represented as a discrete phenomenon that is associated with its geographic and temporal coordinates. These individual feature instances are grouped into classes with common characteristics — feature types" (OGC, 2003a). Because a feature is defined as an abstraction of a phenomenon, the generalization of a feature is a further abstraction of that phenomenon. Consequently, features are the main subjects of generalization and the main input to generalization services.

[4] Generalization services (must) use features as an input.

"A feature is not defined in terms of a single geometry, but rather as a conceptually meaningful object within a particular domain of discourse, one or more of whose properties may be geometric" (OGC, 2003a). The possibility to define multiple geometric properties allows application designers and data modelers to define multiple geometries for the representation at multiple scales, in which case they need to indicate their intention in the data. Though this approach may be practically useful a) to define multiple representations in GML or b) to map existing multi-representation databases to GML, it is conceptually not untainted, because it mingles content and representation in GML rather than to separate them, and it introduces redundancies in the data. However, because the possibility of defining multiple scale-dependent geometries exists in GML, generalization services must be able to deal with them:

[5] Generalization services must be able to interpret multiple scale-dependent geometries of a specific feature, if present.

The ORM specifies 15 basic characteristics of features, four of which will be first listed and then discussed below concerning their impact on generalization:

- "Within an information community or enterprise, there should be only one Feature per real-world entity. The granularity is user-determined." (OGC, 2003a)
- "Features are not classed, but they have Product View, i.e., application-oriented views that are classed. ... This means that somewhere (possibly distributed) there is a set of Product Views of multiple classes associated with this Feature, each with named attributes for what it means to be in their respective classes of application objects." (OGC, 2003a)
- "For complex features represented by Collections ..., the components of one Feature may be Features in their own right. Features that are Collections will always have one or more child Features. ... A collection is a special category of feature that represents a collection of features that have common metadata and formal relationships. Collections possess all the characteristics of a feature, i.e., they are complex features." (OGC, 2003a)
- "A Feature must always record the most accurate or most detailed value of each attribute (root value), and is responsible (albeit indirectly) for deriving the application-specific versions of each attribute for each of its Product Views[4]. ... " (OGC, 2003a)

These characteristics have implications on the level of abstraction that is inherent to features, and consequently some conclusions can be drawn for generalization. First, there is neither only one feature per real-world entity nor is the number of features indefinite, but ideally there is one feature per real-world entity per information community. The starting point for generali-

---

[4] The term product view follows a similar concept as database views and is not related to the RM-ODP viewpoints.

zation are therefore possibly multiple definitions of the same real-world entities (the features). Second there is a separation between features and product views. While features have root values which are as detailed as reasonable, product views provide application-oriented derivates of the root values. Product views may contain more or less detailed classifications of the root values. Generalizations of features can therefore be regarded as product views of these features.

Thirdly, a feature can be composed of other features, in which case it is a *feature collection*. For example, an airport can be modeled as a feature which comprises hangar and runway features and is near a population center called city. Feature collections are a (non-geometric) way to *group* features and arrange them in *hierarchies*. Because they are part of the inherent meaning of the data, generalization services must not manipulate the composition of such groups and hierarchies. Hierarchies are an essential object-oriented modeling construct, because they allow to define the semantic level of detail in the representation of real-world entities. Semantic means that the level of detail is defined using linguistic (non-mathematical) hierarchies such as ‘body / hand / finger’ or ‘house / room / kitchen’. Semantic hierarchies are very often related to the size of the objects and are therefore often scale-dependent. Among others, David Mark has pointed out this issue for generalization: “(...) features such as spits, or fjords, or drumlins, occur only in particular sizes or size ranges. Conceptually, most of these features can be considered to be involved in part-whole relations: each is composed of parts that may themselves be landforms of particular named kinds, and each may form a part of a landscape assemblage of a particular type. In many cases, successful cartographic generalization will require that compound features be ‘recognized’, be ‘parsed’ into their component features, and be generalized in context-dependent and phenomenon-dependent ways.” (Mark, 1989). An alternative way (to feature collections) of modeling part-whole relations is by the classification of attributes, an approach which is very common in today’s geographic data models[5].

[6] Generalization services must be able to interpret the semantic level of detail in geographic information (as defined in feature collections, attribute classifications or other modeling constructs).

[7] Generalization services must respect groups and hierarchies defined by feature collections

### 3.2.2 Spatial-temporal Geometry and Topology

“Geometry provides the means for the quantitative description, by means of coordinates and mathematical functions, of the spatial characteristics of features, including dimension, position, size, shape, and orientation. ... A geometric object is a combination of a coordinate geometry and a coordinate reference system. In general, a geometric object is a set of geometric points, represented by direct positions. A direct position holds the coordinates for a position within some coordinate reference system. ... A conceptual temporal schema defines the concepts needed to describe the temporal characteristics of geographic information as they are abstracted from the real world. Temporal characteristics of geographic information include feature attrib-

---

[5] Attribute classifications are very common, because they are simple and very effective in relational (table-based) and raster models. Object-oriented hierarchies allow more accurate and complex descriptions, but are computationally more demanding. Data modelers are free to combine the two approaches in GML.

utes, feature operations, feature associations, and metadata elements that take a value in the temporal domain." (OGC, 2003a)

One question that arises here is if scaling and generalization should be restricted to spatial-temporal attributes, or if all other attributes (the thematic ones) should be scalable, too. GIS software development, and thus generalization implementations, traditionally has focused on the spatial attributes for two reasons. First, geometry is usually the most complex and storage intensive attribute of features and it is a supposition for all GIS functions to efficiently handle geometry. Secondly, geometries are defined by the means of coordinates and mathematical functions in contrast to most other attributes which are modeled with the means of names or observation values. This makes the geometry attribute more unambiguous and computationally accessible than other attributes. The same will be true for temporal attributes as international standards for temporal primitives and reference systems will be further developed. But these reasons are rather of technical than conceptual nature, and they may become less critical in the future if better technologies and tools to handle semantics become available, such as ontologies in general (e.g. Frank, 2001 and 2003, Kuhn, 2001) or the Web Ontology Language OWL specifically, recently drafted as a standard by the World Wide Web Consortium (W3C, 2004b). Kuhn (2003) has proposed the conceptual idea that spatial and temporal attributes can be regarded as semantic special cases of thematic attributes and that semantic reference and projection systems could be defined to map between different semantic spaces. Such concepts may provide foundations for the generalization of thematic feature properties.

[8] Because all properties of geographic information may have a scale, generalization services should be able to reduce the resolution of all these properties when scale is changed, regardless if these properties are of geometric, temporal or thematic type.

"Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically and continuously – for example, when geographic data is transformed from one coordinate system to another. ... The most productive use of topology is to accelerate computational geometry. Geometric calculations such as containment (point-in-polygon), adjacency, boundary, and network tracking are computationally intensive. For this reason, combinatorial structures known as topological complexes are constructed to convert computational geometry algorithms into combinatorial algorithms. Another purpose is … to relate feature instances independently of their geometry. ... Query operators are a mechanism for characterizing topological relations between different features. ... Typical names for these query operators include 'contains', 'intersects' and 'equals' operations" (OGC, 2003a). Following the ORM definition of topology above, generalization (as well as coordinate transformation) can be regarded as a transformation under which some of the characteristics of geometric figures remain *invariant*. For example, if the city hall was north of main street before generalization, it should also be north of main street after generalization. Or if bridge B spans river R before generalization, it should still span R after generalization. Not to preserve topology during generalization can result in changes in the semantics of the data. It is consequently mandatory to preserve topology during generalization. Weibel (1996) referred to this requirement as topological constraints.

[9] Generalization services must be able to preserve the geometric characteristics defined by topology

### 3.2.3 Spatial Referencing and Location Organizer Folders

“Spatial referencing can be accomplished by aggregating ... information items that share a common location in space and time. OGC has defined the Location Organizer Folder (LOF) as a general, multi-source information container model for handling sets of interrelated spatial-temporal information” (OGC, 2003a). Generalization services should provide output that can be saved in a LOF. This requirement is fulfilled easily (and will therefore not be listed separately), because LOFs can contain just about any kind of spatially referenced data.

### 3.2.4 Coverages including Imagery

“A coverage is a feature that associates positions within a bounded space (its spatial-temporal domain) to feature attribute values (its range).... Examples include a raster image, a polygon overlay, or a digital elevation matrix. A coverage can represent one feature or a collection of features to model spatial relationships between, and the spatial distribution of, earth phenomena” (OGC, 2003a). Handling coverage data for generalization includes sophisticated methods such as automatic structure recognition, automatic categorizations or image enhancement, which is a complex field of its own. For the conceptual model it shall suffice to note that generalization can not be restricted to vector data, but must account for all the types of modeling in GML.

[10] Generalization services should be able to process all feature types contained in GML, vector data as well as coverage data (e.g. imagery).

An important property of GML 3 is that it allows to mix different geometric models in feature definitions. This will allow data modelers in the future to implement ‘hybrid’ data models, such as the ‘fields of objects’ approach proposed by Cova and Goodchild (2002).

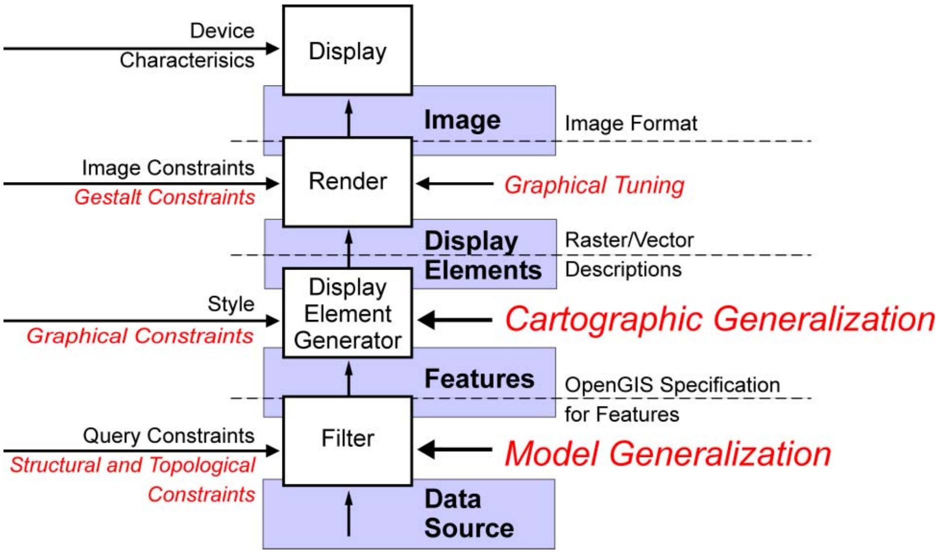
### 3.2.5 Portrayal and Human Interface

“Portrayal is the presentation of information to humans, e.g., a map. A map is a two-dimensional visual portrayal of geospatial data; a map is not the data itself. Two or more maps with the same geographic extent and coordinate reference system can be accurately layered to produce a composite map. Information types associated with geospatial data visualization are shown in the context of the portrayal process ....

1. Image or picture of the data, e.g., a map to be displayed.
2. Display elements, e.g., lexical description of graphics to be drawn onto the target display.

... The creation of display elements requires two inputs; features and style .... Style for portrayal requires symbology, a methodology for describing symbols and mapping of the schema to an application schema.” (OGC, 2003a)

The OGC illustrates the portrayal process in the Portrayal Model (OGC, 2003a, Cuthbert, 1998) as a sequence of independent processing layers: filter, display element generator, render and display. Figure 3.3 illustrates how different aspects of generalization can be associated with the layers in the Portrayal Model. Generalization is a *filtering process* regarding the decision which features shall be represented at a specific scale (filtering of features) and which geometric detail is required (filtering of the nodes or vertices, also called control points in GML terminology). Because this filtering type of generalization may alter geometric characteristics of features, it may be associated with the term ‘model generalization’ as discussed in section *2.1*. Generalization is a *display element generation* process regarding the decision how scale-dependant graphical conflicts must be resolved, and consequently may be associated with ‘cartographic generalization’ in that case. Finally the relative size and distribution of symbols on a map (resulting in the overall visual pattern and contrasts) is scale-dependent and may call for controlled graphical tuning, e.g. to keep the overall impression of symbol colors or the overall contrasts of the map constant. All aspects of generalization require instructions on how to process their input when the map scale is changed, i.e. special generalization or graphical constraints. The types and number of constraints to be fulfilled are selected according to application-specific needs. The layers in the portrayal model imply that functionalities for each layer can be implemented by components which are independent from components in other layers.



The red/italic terms refer to generalization and have been added to the model by the author. The typology of generalization constraints follows Weibel and Dutton (1998), though the associations of these constraint types with the layers of the portrayal model are ‘loose’, i.e. some constraints may be used in different layers.

Figure 3.3 The Portrayal Model (OGC, 2003a) and generalization (constraints)

Following the portrayal model in Figure 3.3, *model generalization* would take features (data source) and generalization constraints as an input and provide generalized features as an output. The user may experience model generalization as part of a data query interface. Whenever a user queries a catalogue, gazetteer or geocoding services for geographic data, she or he may wish to specify the target resolution or scale at which the data will be used in order to receive features, and especially geometries, that are useful at such scale. *Cartographic generalization* would take features, style and generalization constraints as an input and provide generalized display elements as an output. The dominant operation in electronic mapping to perform cartographic generalization is the ‘zoom’, which could be experienced by a user as an intelligent adaptation of the data to the required level of detail (Frank and Timpf, 1994, Vangenot, 2001, Renschler, 2002). Such zooming has therefore been called adaptive zooming (e.g. Brühlmeier, 2000) and some implementations with similar functionality are already available (e.g. Map24, 2004, NASA, 2004). *Graphical tuning* (if required by an application) could be part of the rendering process.

[11] Generalization services should foresee capabilities for model generalization, cartographic generalization and graphical tuning.

### 3.2.6 Data Set and Service Metadata

“Metadata is data about data. ... Metadata elements and schema are used by data producers to characterize their geographic data” (OGC, 2003a) The OGC metadata standard is specified in the OGC Abstract Specification, Topic 11: OpenGIS Metadata (OGC and ISO, 2000), which is the same as ISO 19115. “Metadata is applicable to independent datasets, aggregations of datasets, individual geographic features, and the various classes of objects that compose a feature. ... The conceptual schema for dataset metadata defines an extensive set of metadata elements; typically only a subset of the full number of elements is used. A subset of the elements known as the core metadata elements required to identify a dataset is defined, and typically used for catalogue purposes” (OGC, 2003a). Metadata is a potential information and configuration source for services, and metadata containing information about the resolution or scale of the data set may be of particular interest for generalization services. Among the core metadata elements in ISO 19115, an element named `spatialResolution` exists that allows data producers to specify the spatial resolution of their data. The `spatialResolution` can be specified either with a `equivalentScale` element (“level of detail expressed as the scale of a comparable hardcopy map or chart”) or a distance element (“ground sample distance”) (OGC and ISO, 2000). The `spatialResolution` element is the only metadata element in ISO 19115 referring to the spatial resolution of data and it’s use is optional. The following schematic hierarchy depicts the `spatialResolution` metadata element:

(`MD_Metadata` >
`MD_DataIdentification.`**`spatialResolution`** >
`MD_Resolution.equivalentScale` or
`MD_Resolution.distance`)

Services involved in generalization might use the `spatialResolution` information to verify for a requested target resolution if there is a need to generalize the input dataset. Future versions of the metadata specification may provide more sophisticated models to describe the overall resolution of a dataset.

[12] Generalization services should be able to interpret the existing metadata elements indicating spatial resolution or scale

The core metadata elements are not the only metadata elements available. Application and service designers can extend the ISO 19115 model with their own service-specific metadata elements, classes or packages, allowing the creation and exchange of sophisticated and powerful configuration schemes. The drawback of this approach is that a service can only use data for which the data producer actually created the content of these extension elements, which is a problematic assumption in a distributed environment.

A different type of metadata is used to describe geographic services. “The most basic operation all OGC services must provide is the ability to describe themselves. This ‘Get Capabilities’ operation is common to all OWS1 services (comment of the author: OWS stands for OGC Web Services Initiative). The result of invoking this operation on a service is a message containing a ‘capabilities document’ describing the service” (OGC, 2003a). ISO 19119 provides a metadata model for service instances (OGC and ISO, 2002).

[13] A generalization service must be able to describe its capabilities. In an OGC environment, the capabilities must be described according to ISO 19119.

## 3.3 ORM Computational Viewpoint

### 3.3.1 Service Architecture

Although not explicitly mentioned in the ORM, the OpenGIS services architecture obviously builds on the so-called *service-oriented architecture* SOA, a software development approach which originated in the early nineties and has recently received attention through the advent of web services (e.g. Wiehler, 2004). In order to assess the role of generalization services in the OGC services framework, it is necessary to understand the principles and the context of SOA. For the OGC framework, the service-oriented approach is defined in the Abstract Specification, Topic 12 - OpenGIS Service Architecture, which is the same as ISO 19119 – Geographic Information Services (OGC and ISO, 2002). Only some of the principles of SOA shall be mentioned here (refer to Di, 2004a for a good overview of those principles regarding ISO 19119):

- Services are well-defined sets of actions, describing the messages exchanged by the service and the consumer of the service. The state of a service does usually not depend on the state of other services, i.e. they are *stateless*. The action of a service finishes, when the service has returned the results of the service invocation.
- The three types of actors in SOA are the *providers* who provide specific services over the Internet, the *requesters* (users) who request information services, and the *brokers* who help the requestors to find the right services. Basic operations of services include *publish*, *find*, *bind*, and *chain* (Figure 3.4). The idea behind this scheme is that “everyone on the Internet can set up a web service to provide service to anyone who wants” (Di, 2004a) and thus that many services will be available.

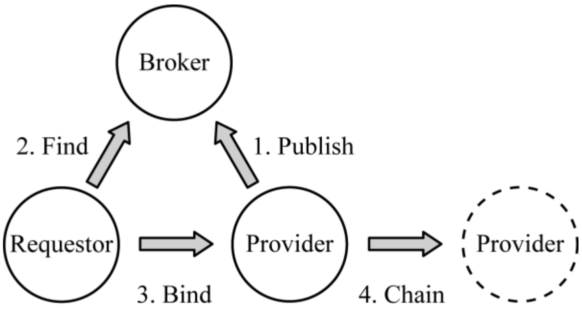


Figure 3.4 The basic service operations (after Di, 2004a)

- To solve complex tasks, individual services can be chained (*service chaining*).
- Services are implementation and platform independent, which means they can be written in any suitable programming language and be implemented on any suitable hardware platform and even at many different network environments (see requirement [3]). They must, however, adhere to interface standards such as ISO 19119 or the ones shown in Figure 3.5. Currently, the major two network environments include the Web and the Grid. The Grid (or grid computing) is a middleware-based networking technology for sharing high-end computing resources (Foster et al., 2002 and 2001, Foster and Kesselman, 1998 and 1999). Compared to the Web, the Grid provides advanced features such as *stateful* interactions and service lifetime management. With the Web Service Resource Framework (WSRF), work is currently on the way to converge the technologies for the Web and the Grid (GGF, 2004).

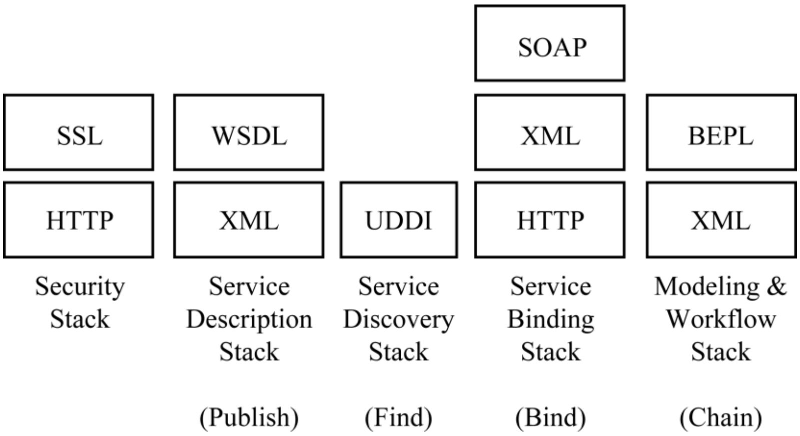
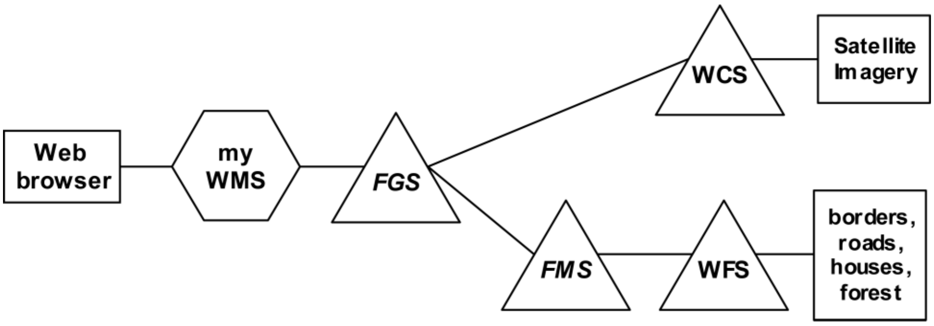


Figure 3.5 The major web service standards (Di, 2004a)

### 3.3.2 Service Chaining

Service chains combine services in a dependent series to achieve large tasks. Service chaining can be performed dynamically and provide just in-time integration of the required functionality. From a services point of view, task-oriented generalization as discussed in section 3.1.1 is a complex dynamic service chain. Figure 3.6 shows a simple generalization example service chain.



WMS = Web Map Service, FGS = Feature Generalization Service, WCS = Web Coverage Service, FMS = Feature Manipulation Service, WFS = Web Feature Service. No OGC specifications exist for FGS and FMS.

Figure 3.6 An example for a hypothetical generalization service chain.

ISO 19119 (OGC and ISO, 2002) defines three architectural design patterns for service chaining:

- User defined (transparent) chaining: the Human user manages the workflow.
- Workflow-managed (translucent) chaining: in which the Human user invokes a Workflow Management service that controls the chain and the user is aware of the individual services.
- Aggregate service (opaque): in which the user invokes a service that carries out the chain, with the user having no awareness of the individual services.

ISO 19119 further proposes architectural considerations that should be applied to all services supporting service chaining:

- Operations should be modeled as messages. A message operation shall consist of a request and response.
- A client should have the option of receiving just the status of an operation and the data should be accessible through a separate operation.
- “For simplicity it is desired that a service be stateless, i.e., that a service invocation be composed of a single request-response pair with no dependence on past or future interactions.”

Because these considerations are valid for all OGC services, they can be adopted as essential requirements for generalization services:

[14] Generalization Service operations should consist of request and response messages.

[15] Generalization Services should separate the control of the service from the access to the data resulting from the service.

[16] Generalization Services should be stateless. (As mentioned above, the statelessness applies to web services rather than grid services)

### 3.3.3 Service Types

The OGC calls its services framework the OWS Service Framework (OSF), where OWS stands for OpenGIS Web Services. The taxonomy for OFS is described in more detail in ISO 19119 (OGC and ISO, 2002). The taxonomy defines service categories and lists example services for each category, containing two examples of generalization services as shown in Table 3.1.

Table 3.1 Service categories according to ISO 19119 – Geographic Information Services

| OGC Service Taxonomy Categories (ISO 19119) | Generalization Service examples |
|---|---|
| Geographic human interaction services | Feature generalization editor |
| Geographic model/information management services | |
| Geographic workflow/task management services | |
| Geographic processing services<br>Geographic processing services – spatial<br>Geographic processing services – thematic<br>Geographic processing services – temporal<br>Geographic processing services – metadata | Feature generalization service |
| Geographic communication services | |
| Geographic system management services | |

ISO 19119 has been harmonized with earlier versions of Topic 12 of the OGC Abstract Specification and is now the same as Topic 12 (OGC and ISO, 2002).

Specifying (service) types is fundamental to software design. Typing makes it possible to build software clients that know and check the validity of types and prevents services or other entities from providing overlapping functionalities. “An entity [e.g. a service] is of a particular type if its properties satisfy that type. Types must match before some action can occur.” (OGC, 2003a) If generalization services are to be used in a OGC context, they have to be functionally decomposed and typed according to the ISO 19119 or OSF services taxonomy. Two examples of generalization services exist in ISO 19119:

- The feature generalization editor is described as a “Client service that allows a user to modify the cartographic characteristics of a feature or feature collection by simplifying its visualization, while maintaining its salient elements – the spatial equivalent of simplification.” (OGC and ISO, 2002) While a generalization editor may represent a useful product for electronic cartography, static editing it is not a focus of this work.
- The feature generalization service is described as “Service that generalizes feature types in a feature collection to increase the effectiveness of communication by counteracting the undesirable effects of data reduction.” The definition for feature generalization service is broad, but it provides a potential container for dynamic generalization

operators as discussed in sections *3.1 ORM Enterprise* Viewpoint and *3.2 ORM Information* Viewpoint.

[17] Generalization Services should be unambiguously typed following the services taxonomy in ISO 19119. The processing functionalities for dynamic generalization should be grouped under the service type 'feature generalization service'.

Annex A of the ORM 'Mapping ORM requirements to OGC services' lists existing or potential OGC service types. Table 3.2 shows some service types from ORM Annex A that may be valuable candidates for dynamic generalization service chaining. As can be easily seen in the list, there are currently not many service types that have published specifications.

Table 3.2 Examples of service types potentially useful in generalization

| **Topic 12 Service type** | **Topic 12 High Level** | **OGC Specifications** | **ORM Requirements** |
|---|---|---|---|
| Chain definition | Task Management | Work in OWS is highly relevant. This includes the OGC General Services | Model Collaboration |
| Workflow enactment | Task Management | None | Collaboration |
| Feature Manipulation | Processing - Spatial | None | Analysis |
| Feature Generalization | Processing - Spatial | None | Analysis |
| Thematic Classification | Processing – Thematic | None currently but could be built based on capabilities in WFS and WCS. | Analysis |
| Feature Generalization | Processing – Thematic | None | Analysis |
| Spatial Counting | Processing – Thematic | None currently but could be based on WFS, Filter, and GML | Analysis |
| Reduced Resolution Generation | Processing – Thematic | Imagery Handling | Analysis |
| Image manipulation | Processing – Thematic | Imagery Handling | Analysis |
| Statistical Calculation | Processing – Metadata | None | Analysis |
| Geographic Annotation | Processing – Metadata | XIMA for annotations. Registries. | Visualization, publishing |

The table is an excerpt from ORM Annex A 'Mapping ORM requirements to OGC services' and lists types potentially being useful for dynamic feature generalization. *Topic 12* refers to 'Abstract specification – Topic 12, OpenGIS Service Architecture' (OGC/ISO, 2002d), which is the same as ISO 19119.

Services that are used for a specific task can be logically grouped to a Service Organizer Folder (SOF), a structure defined in ISO 19119 (OGC and ISO, 2002). "A service chain combines services to produce results that the individual services could not produce alone. ... The human user that constructed a new chain or invoked an existing chain of services should determine semantic validity of the results. ... Some factors to consider in the semantic evaluation of a chain result are listed below:

- Appropriateness of starting data: are the based datasets suited to the subsequent processing? For example, accuracy and resolution of the data, thematic values are relevant.

- Effect of services on data: how do the individual services effect the data, e.g., error sources and propagation.
- Sequence of the services: how does the order of the chain affect the results? For example, should a spatial operation, e.g., orthorectification, be performed before or after a thematic operation, e.g., resampling the attribute values?

The evaluations depend upon understanding the services, e.g., through review of the service metadata, but also rely upon the users understanding of the combinations of the services." (OGC and ISO, 2002) Service chains are particularly interesting for dynamic generalization if they can be constructed by some automatic mechanism. This pre-proposes that the mechanism finds enough information in the metadata to decide about the validity and appropriateness of the input data.

[18] Generalization Services should be able to validate the appropriateness of the input data for the type of generalization at hand.

## 3.4 ORM Engineering Viewpoint

### 3.4.1 Clients and Application Services

"The engineering viewpoint helps to articulate a key distinction among distributed systems:

- Thin clients rely on invoking the services of other components (servers, middleware) ...
- Thick clients handle much of the necessary computation and data/metadata management themselves. ...

Web Mapping is one of the key areas in which OGC has explored and discussed thin and thick clients. ... The Essential Model (author's note: of WMS) suggests that in a World Wide Web environment, a thin client may be an unadorned Web browser with no need for Java applets or plug-ins. ... Thick clients (usually applets, plug-ins, or standalone applications) move some or all of the feature-rendering functionality into the client side, and may allow more complex user input. The Web Map Server Interfaces Implementation Specification, v1.0 discusses Web mapping architectures based on thin, thick, and medium clients. ... Experience has shown that most Web mapping architectures based on thin clients and/or the picture case rely on a server-side Viewer Client Generator to process client requests, maintain or transfer state between requests, and return responses as HTML pages. ...

OpenGIS Services are accessible from Application Services operating on user terminals (e.g., desktop, notebook, handset, etc.) or servers that have network connectivity and that utilize OpenGIS service interfaces and encoding specifications. ... Application Services should be able to execute not only on the user's desktop (or handset), but also on a server on the network. Examples of server-side Application Services include compute-intensive (and/or I/O-intensive), server-based applications like those required for Image Processing or Route Determination." (OGC, 2003a)

Following the experience and recommendations of the OGC, computationally intense services, such as generalization services, would likely be implemented as server-side application service components. But this is not a conceptual requirement, because application service architecture can be adapted to application needs and the technology available.

### 3.4.2 Distribution Transparencies

To describe "how a system can hide complexities associated with system distribution from applications" (OGC, 2003a), the ORM defines eight transparencies (access, failure, location, migration, relocation, replication, persistence, transaction). These transparencies are not a direct concern for the conceptual model as a single OGC service, and they therefore do not provide additional requirements for the conceptual model of generalization services. But they are important basic conditions for the programmatic implementation of an application service and should be considered when implementing a generalization service.

## 3.5 ORM Technology Viewpoint

The OpenGIS Information Framework is "comprised of two basic sets of information modeling constructs: Descriptive Components and Runtime Components" (Figure 3.7)(OGC, 2003a).

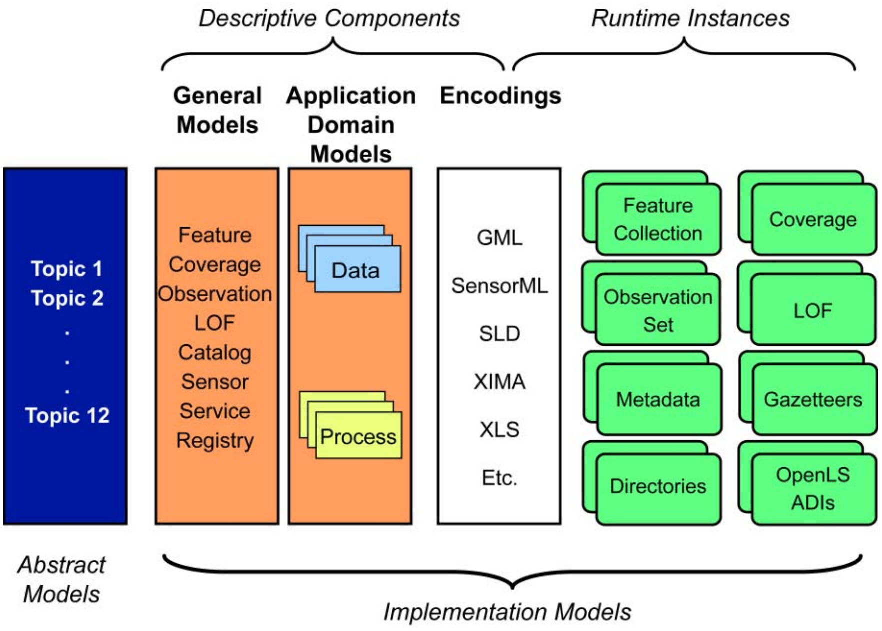


Figure 3.7 OpenGIS information framework (OGC, 2003a)

"Multiple platform-specific specifications are necessary because of the variety of DCP's and the differences in the way in which they support the functional requirements. It is assumed that one conceptual specification will be the basis for multiple platform-specific implementation specifications." (OGC, 2003a) Specifications are only considered complete if they have a conceptual model and at least one implementation. The OGC does not provide a general model for generalization and consequently there are no base classes to describe the generalization behavior of data. By not providing a general model, the task of describing generalization processes or the generalization behavior of data is delegated to application designers to include such descriptions in domain specific data and process models. If application designers find it necessary to construct new model classes or encodings, they need to thoroughly investigate the existing model classes and encodings in order to reuse them to the furthest extent possible.

[19] Application domain data and process models for generalization must reuse existing OGC model components and encodings to the furthest extent possible.

Up to now, the following generalization related elements exist in the OGC framework and should be known to generalization application designers:

- *MD_DataIdentification.spatialResolution* element in ISO 19115 metadata (OGC, 1999a)
- *gml:spatialResolution* element in GML (OGC and ISO, 2004)
- *MinScaleDenominator* and *MaxScaleDenominator* elements in SLD (OGC, 2002c) and WMS (OGC and ISO, 2004b)

## 3.6 Conclusions

The OpenGIS framework depicted in the OpenGIS Reference Model ORM (OGC, 2003a) provides some simple elements to specify resolution as well as the definition of a generalization service type in the services taxonomy. There are several conceptual requirements for generalization services that can be derived from a strict interpretation of the ORM. These requirements may be useful for application designers who wish to develop generalization services that interoperate with other OGC services. By and large, these requirements reflect the rules after which the OGC framework is built and after which new models and services must be built. Requirements are regarded as *mandatory* (Table 3.3) if the ORM or corresponding implementation specifications explicitly demand a certain behavior, concept or procedure or if they are regarded to be essential for interoperability. A conceptual model should at least account for these mandatory requirements. Table 3.4 subsumes optional requirements for generalization services that were derived from the ORM in chapter 3. They are regarded as optional in the sense that a conceptual model may leave it to the application designer to implement a corresponding functionality or not. The requirements presented here comprise only a selection of the possible requirements that refer to the OGC framework. More requirements exist if the OGC and ISO implementation specifications are accounted for in detail.

Table 3.3 Mandatory generalization service requirements based on the ORM viewpoints

| **Generalization Services ...** | **justification** | **ORM viewpoint** | **no.** |
|---|---|---|---|
| must provide standard query interfaces, be platform and data content format independent as well as vendor neutral | These high-level requirements are explicitly listed in the ORM enterprise viewpoint. Please refer to the ORM (OGC, 2003a) for the full list and exact formulation of these requirements. | enterprise | [3] |
| (must) use features as input | The OGC framework uses GML for data exchange. GML is based on the concept of features. A conceptual model for generalization services must determine how services deal with the various types of features* and feature geometries**. | information | [4] |
| must be able to interpret the semantic level of detail in geographic information | The approach presented here acts on the assumption that the semantics of geographic information (their inherent meanings and concepts) and their level of detail are primarily defined in the data model(s). Use-case-specific semantics can additionally be defined using constraints. Generalization services must be able to interpret the various modeling constructs that contain semantics in GML***. | information | [6] |
| must respect groups and hierarchies defined by feature collections | Groups and hierarchies defined within the data model should be respected as a part of the information semantics and must not be broken up by generalization (see also ***). | information | [7] |
| must be able to preserve topology | Topology defines the geometric characteristics that are invariant under a transformation. Consequently, generalization must not change such characteristics (see also ***). | information | [9] |
| must be able to interpret multiple scale-dependent geometries of a specific feature, if present | GML provides the possibility to define multiple scale-dependent geometries. Generalization services must be able to deal with them ****. | information | [5] |
| must be able to describe its capabilities | This requirement applies to all OGC services (see OGC and ISO, 2002). | computational | [13] |
| application domain data and process models must reuse existing OGC model components to the furthest extent possible | This requirement applies to all OGC services (OGC, 2003a). | technology | [19] |

The numbers in square brackets refer to the order of occurrence in this chapter.

* Features include vector and/or raster data (object and field based modeling). Feature types include feature collections, coverages, observations, temporal features etc.

** see chapter 4.3.3 and Table 4.1 further below for an overview of GML geometries.

*** Some of the constructs that allow a data modeler to give a meaning to features and elements are typing (the definition of named classes), structuring (the order and position of elements within a GML document), grouping (by using feature collections or attribute classifications/categorizations), as well as strong relationships and topologies.

**** Though practically useful, modeling multiple geometries for multi-scale representation introduces redundancies in the data and should conceptually be avoided, if possible.

Table 3.4 Optional generalization service requirements based on the ORM viewpoints

| **Generalization Services ...** | **justification** | **ORM viewpoint** | **no.** |
|---|---|---|---|
| should be able to account for the users task at hand | This requirement is important when certain aspects of information shall be accentuated, which is common in cartographic generalization. | enterprise | [1] |
| should provide the possibility for application providers to adapt the service to their specific needs | The possibility for custom configuration may be a requirement in data preparation for high quality data products. | enterprise | [2] |
| should be able to reduce the resolution of all feature properties | Many variables (interval, ratio or circular) that are defined as properties in GIS have a scale, or have an inherent relation to spatial scale*. | information | [8] |
| should be able to process all feature types contained in GML (incl. coverages, observations, measurements etc.) | GML contains various classes to model vector data as well as raster data (coverages). A conceptual model must consider all these types, one service may only implement functionality for one or two classes. | information | [10] |
| should provide capabilities for model generalization, cartographic generalization and graphical tuning | This requirement bases on the assumption that aspects of generalization may occur at different stages of the portrayal process. | information | [11] |
| should be able to interpret the existing metadata elements indicating spatial resolution or scale | Currently, the optional core element MD_DataIdentification > spatialResolution can be used to describe the overall resolution of a dataset. Generalization services should consequently be able to use these elements. | information | [12] |
| operations should consist of request and response messages | This requirement applies to all OGC services. | computational | [14] |
| should separate the control of the service from the access to the data resulting from the service | This requirement applies to all OGC services. | computational | [15] |
| should be stateless | Stateless operations are a recommendation of ISO 19119 to keep web service architectures simple. Other networking technologies, such as grid services, may support stateful operations | computational | [16] |
| should be known service types | All service instances are of specific service types and the client knows the type prior to runtime. An unambiguous service taxonomy is defined by ISO 19119. | computational | [17] |
| should be able to validate the appropriateness of the input data for the type of generalization at hand | This is a recommendation for all OGC services. | computational | [18] |

* This is rather a long-term issue and may only be helpful in scientific applications. However, it should conceptually be accounted for.

# 4 Generalization Service Requirements based on Use Cases

This chapter will focus on generalization service requirements as they might derive from use cases of GML-aware applications. A hypothetical cadastral information service (CIS) of the Swiss Kanton[6] Luzern and a hypothetical data ‘harmonization’ service will be introduced to construct three simple use cases. The cadastral data serves as an example of vector data with a particularly high level of detail (LoD), the aspect that it is ‘cadastral’ is of minor importance here. The use cases will be availed to discuss potential requirements for the generalization of GML based data and maps. Use case A and B describe two distinct functions of the CIS where generalization is a background process and use case C provides an example where a user actively invokes generalization in order to harmonize the geometric level of detail in two or several data sets. The three use cases in section 4.1 include a textual description of the case and a short discussion of general aspects regarding generalization in the use case. Sections 4.2 - 4.4 discuss generalization-specific requirements based on the use cases and section 4.5 provides a summarizing list of the requirements. The numbering of requirements in square brackets is continued from chapter 3 for easier reference. Use cases are essentially a textual or graphical description of a system’s behavior from a stakeholders point of view. They are thought to stimulate discussion within a team about an upcoming system (Cockburn, 2001), most often with the aim of developing precise system requirements. The use cases presented in this chapter are so-called ‘black-box use cases’: they describe *what* the system should do rather than *how* it should be implemented.

## 4.1 Use Cases

### 4.1.1 Cadastral Information Service (CIS)

A hypothetical CIS of Kanton Luzern in Switzerland shall provide the basis for the two following use cases A and B. The CIS shall build on an OGC web mapping service (WMS) providing cadastral information from a high-resolution cadastral database over the Internet. The general set-up shall be as follows:

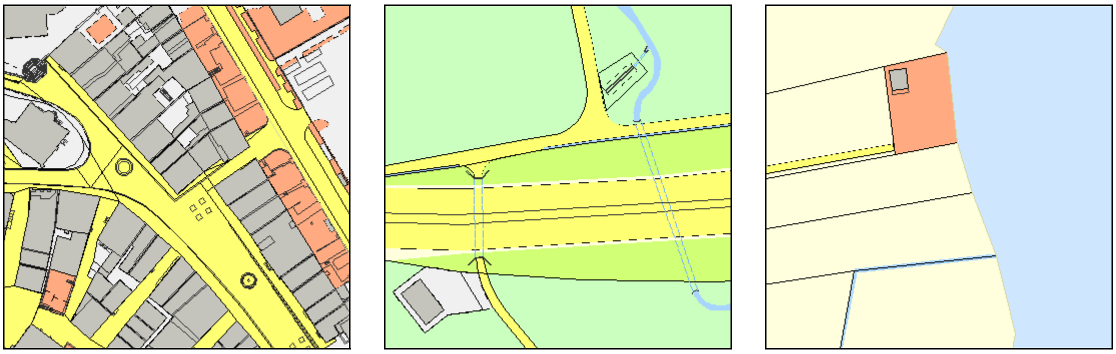
- The user interface is accessible over the Internet. The map is 400 x 300 pixels in size, showing features in two dimensions (2D).
- The cadastral database serving as data source contains high-resolution data sets with positional accuracies ranging from 1 cm (parcels in cities) to 5 m (ground cover in mountain areas). Every feature exists once in the database and is updated on a irregular basis.

---

[6] A Kanton is an administrative unit in Switzerland, comparable to states in the US (but much smaller in size). There are 26 Kantone in Switzerland. (admin.ch, 2004)

- The map shows cartographically appealing and consistent views of the cadastral information at all zoom levels (Use case A). These views are generated dynamically using automatic generalization.
- The map items can be clicked on to retrieve property information of the features they represent (Use case B). At scales where the map is generalized and thus shows less than the full detail, map items may represent groups or even complex aggregations of features. Information retrieval by clicking on items provides access to the property information of all members of a group or structure.
- The CIS provides the most actual data available. Information is stamped by date. The service does not give access to historical data or versions of the information. Thus the temporal domain of the data is not a central focus.
- The CIS provides planning states for cadastral objects, e.g. parcels, buildings or roads, as a separate data layer including the planned extent (geometries) of these objects.
- The user interface offers standard tools such as zooming, panning, an info button, key word and thematic search, printing (e.g. in ISO A4, A3) and choosing display layers.

Figure 4.1 a) – c) show scenes from the hypothetical CIS based on existing cadastral data.



a) Town center with buildings (grey), streets (yellow) and gardens (red)

b) Highway and some small roads (yellow) in the forest (green), and a creek (blue)

c) Lakeside with lake (blue), pasture (light yellow) and a garden (red)

Reproduced with permission of Geoinformation und Vermessung Kt. Luzern, 2004.

Figure 4.1 Scenes from a hypothetical cadastral information service, town of Sursee, Switzerland. 1:3’000.

The CIS would include the following main feature types, which would be defined in a CIS application schema in GML (Currently, Swiss cadastral information is described in Interlis):

- Cadastral information: parcels, many types of ground cover (including buildings, roads, waters, woods: polygon geometries), many types of single objects (e.g. subterrain structures, power lines etc.), location names
- Administrative areas and borders (in three levels *Kanton*, *Amt* and *Gemeinde*)

- Building addresses
- Digital elevation model (DEM) and shaded relief

The positional accuracies of the cadastral layers vary over the area of the Kanton according to legal requirements. Five classes of positional accuracies exist. Accuracies around 1 centimeter are common settlement areas (or where economic interest in the parcels is high). The DEM and shaded relief is supposed to have a resolution of 0.5 meter.

### 4.1.2 Use Case A: Zoom (in CIS)

The scope of use case A is using the zoom function of the Cadastral Information Service CIS outlined above. The CIS is WMS-based and provides cadastral information of the Swiss Kanton Luzern from a high-resolution GML database[7]. The zoom shall be scale-less and adaptive, where scale-less refers to not-fixed zooming steps and adaptive refers to adoption of the level of detail to the available map space (generalization). The zoom shall further be enabled from the regional level (whole Kanton) down to full detail at parcel level. Zooming is blocked below parcel level, when no more detail can be represented. Zooming in from the Kanton level shall *gradually* reveal more detail. The WMS is chained with a *Generalization Service*, which is chained with a Web Feature Service WFS (OGC, 2002a) and a Web Coverage Service WCS (OGC, 2003b). The WFS and WCS perform the retrieval of vector and raster data from a geographic database. The generalization service may consist of one or several services that take on distinct generalization tasks.

***Use Case A**: Zoom in a WMS Cadastral Information Service*

**Primary Actor**: CIS user

**Stakeholders and Interests**:

CIS user – wants a graphical and flexible way of navigating through the cadastral information and a meaningful cartographic representation of the cadastral information at all zoom levels

Survey authority – wants to provide multi-scale cadastral information access from a single high-resolution database (reduction of update cost) and to facilitate the use of cadastral information for different groups of interest

**Precondition**: CIS is connected to the Internet, has logged in to the CIS, and activated the zoom tool (click on the ‘zoom’ tool button).

**Minimal Guarantee** (case of failure): The server(s) log activity and send data ungeneralized, if generalization fails. An error message is provided in case of failure.

**Success Guarantee** (case of success): The CIS user can gradually zoom in and out the map while the map content adapts to the available map space.

**Main Success Scenario** (sequence of actions in case of success):

1. CIS user zooms, which invokes a GetMap request to the WMS
2. The WMS forwards the request to the *Generalization Service*

---

[7] Meant is a database which stores cadastral data as GML or which can generate GML on the fly upon query. No such database exists in the Kanton Luzern as of today.

3. The *Generalization Service* performs the following tasks:
    3.1. Request the indicated cadastral data from the WFS and WCS
    3.2. Calculate the required *target resolution* from the GetMap request parameters
    3.3. Model Generalization:
        a. Reduce the geometric level of detail in the data to a level appropriate regarding the *target resolution* (by thinning or masking out geometries)
    3.4. Cartographic Generalization:
        a. Resolve symbology conflicts (e.g. by local displacement or resizing of overlapping symbols)
        b. Resolve labeling conflicts (by masking out, displacement, resizing, changing style...)
    3.5. Return generalized data to the WMS as GML containing:
        a. Generalized geometries of features or feature collections
        b. Links to the identifiers of all features contained in a feature collection (where the generalized geometry of a feature collection serves as a representation of a group of features)
        c. Properties that are used for labeling
    3.6. Return adapted symbology to the WMS as SLD
4. WMS renders the map and returns it to CIS client
5. The CIS client displays the map

Two aspects apparent in use case A may be characteristic for generalization when used as a dynamic service:

- Generalization is initiated as part of another action (zooming) and the user may be unaware of generalization taking place. It acts as a background process that ensures the map at the new scale being useful and visually appealing. Generalization is not actively requested by the user in this use case.
- The total response time must be short to not disturb the user’s zooming experience. Ideally, zooming is a fluidly streaming motion for which extent and speed are controllable with the mouse.

An immediate response time for interactive use may be one of the hardest requirements imposed on the engineering of generalization services and presupposes highly performing components, algorithms and processing models.

### 4.1.3 Use Case B: Information Retrieval (in CIS)

Use case B builds on the same CIS as use case A. The user can query feature properties by clicking on map items in the CIS. In order to allow the selection of map items by clicking on them, the WMS must send a descriptive graphical format to the client, such as Scalable Vector Graphics SVG (W3C, 2004e). Clicking on map items shall be possible at all scales and shall show property information of the features underlying the selected map items in a property window or frame. The information returned may include the community or district names, parcel numbers, addresses, landowner names, date of last survey activity, coordinates of parcel centers etc. The amount and types of information depend on the scale and may consist of a) The actual detailed property information and/or b) aggregations of the underlying property information, e.g. counts, total area, average values. Feature property information is presented in a way suitable to display hierarchies, such as collapsible trees or diagrams, to provide quick and flexible access to the information. The form of this presentation shall remain consistent across zoom levels. In this use case, the WMS is directly chained with a WFS (OGC, 2002a) and WCS (OGC, 2003b).

***Use Case B**: Information Retrieval in a WMS Cadastral Information Service*

**Primary Actor**: CIS User (same as use case A)

**Stakeholders and Interests**:

CIS user – wants access to information about items in the map at all zoom levels.

Survey authority – wants to provide multi-scale cadastral information products from a single high-resolution database (reduction of update cost) and to facilitate the use of cadastral information for different groups of interest

**Precondition**: The CIS user has the CIS with a map window open and is connected to the Internet. The map either shows a scene at the lowest zoom level, in which case it shows the full detail of the original data, or on a higher zoom level, in which case it shows a generalized representation of the original data.

**Minimal Guarantee**: The CIS server, WFS and WCS log activity and throw an error if information retrieval or processing fail.

**Success Guarantee**: The CIS user chooses the layer(s) of interest, e.g. by selecting layers from a list. By click on a map item, the CIS user receives the property information of the related features.

**Main Success Scenario**:

1. The CIS user submits a query by clicking on a map item (e.g. a parcel, community, road, etc.)
2. The CIS client queries the WFS and/or WCS for property information.
3. The WMS queries the WFS and WCS and returns the property information to the CIS client
4. The CIS client displays the property information in the form of text, lists, trees and/or tables.

**Variations**:

1. The CIS interface allows multiple selections of map items

The first notion about use case B is that generalization is not part of the actual query process in use case B. It rather happens beforehand, e.g. as part of the zooming described in use case A. Generalization delivers the prerequisites for information retrieval, namely the map items (comparable to *display elements* in the Portrayal Model, chapter 3.2.5) and the information that allows links back to the underlying features. The second notion is that information selection at a low level of detail (small scale) can be seen as a pre-selection or group-selection process, which may just be an exploratory step in the user's task chain. The idea to dynamically link large-scale map views with the original high-resolution data for exploratory purposes is a slightly less sophisticated variation of the similar idea to dynamically link several views of the same data, which is an important principle of tools used for Exploratory Spatial Data Analysis ESDA (e.g. Anselin, 1999). ESDA tools, such as SAGE (Haining et.al, 1998) or GeoDa (GeoDa, 2004) offer dynamically linked windows with a variety of maps and statistical graphs and allow the user to interactively manipulate the analysis environment, while changes in one window are active instantaneously in all other windows. Like ESDA tools are helpful interfaces for the visual exploration of the statistical characteristics of spatial data, the linking of data and maps on different zooming levels as suggested by the use cases above may be the basis for innovative interfaces which improve the visual exploration of geographic data at multiple scales.

### 4.1.4 Use Case C: Dataset Harmonization

The scope of use case C is a data 'harmonization' application, which shall allow a user to adapt the geometric LoD in two or several GML data sets to a common level, which must be equal or lower than the one of the lowest-detailed data set[8]. The use case is restricted to data with the same projection and approximately the same extent. The output GML shall include links to all features in the original data (as in use case B), and thus pertain access to the full information contained in the original data. The data harmonization application is chained with a *Generalization Service* that performs the corresponding processing for vector and raster data. As in use case A, the generalization service may consist of one or several services that take on distinct generalization tasks[9].

*Use Case C: Data harmonization*

**Primary Actor**: GML-User

**Stakeholders and Interests**:

GML-User – wants to adjust the geometric level of detail in two geographic data sets as a preprocessing step for mapping or analysis.

**Precondition**: The GML-User has the Data Fusion Application open and is connected to the Internet. The GML-User has the original GML data sets on disk. As a variation, the Data Fusion Application may allow to search Catalog Services and the data needs not to be downloaded before generalization.

**Minimal Guarantee**: The Data Fusion Application and *Generalization Service* log activity and throw an error if information processing (or retrieval) fails.

**Success Guarantee**: The GML-User chooses the data sets of interest and submits them to data fusion. The GML-User receives one or several generalized GML data sets to be saved to disk.

**Main Success Scenario**:

---

[8] An every-day-real-world-example may be an environmental planning or engineering company, which is contractor of Kanton Luzern and needs to produce a 1:35'000 map of a certain area as part of a planning task. The company could use the Kanton's cadastral data (e.g. the ground cover information), but needs to generalize it for integration in the 1:35'000 map.

[9] In some cases, data harmonization may extend to conflation, i.e. the adaptation of information content in different datasets (e.g. aligning roads in one dataset to houses in another),where more complex constraints must be satisfied.

1. The GML-User selects one or several input GML data sets from disk or from a catalogue service.
2. The GML-User specifies:
    2.1 the location(s) on disk to save the output to
    2.2 one of three options concerning the geometric resolution of the output (see the discussion later on in this section on how the geometric detail could be specified)
        - Manual entry of the desired geometric resolution for the output GML (default)
        - Automatic analysis of the geometric detail of all data sets, determination of the lowest-detailed data set and automatic submission to generalization
        - Analysis and display of the geometric detail of all data sets before submission to generalization
    2.3 if the input datasets are treated separately or if they are combined (in GML, features with different geometry types can be combined in one dataset)
    2.4 if links in the output GML are relative (only available for data residing on disk) or absolute
    2.5 if 'advanced' generalization to resolve geometric conflicts shall be performed
3. The GML-User submits a processing request to the *Generalization Service*
4. The *Generalization Service* performs the following tasks
    4.1. Reduce the geometric level of detail according to the options set in steps 2.2 and 2.3
    4.2. Resolve geometric conflicts (e.g. by local displacement of overlapping lines or polygons), if requested by the user in step 2.5
    4.3 return the generalized output GML to the Data Fusion client
5. The Data Fusion client saves the output GML as specified in steps 2.1 and 2.4, displays a success message and/or offers to display the data

Use case C describes a situation where a user actively requests generalization. The tasks that the generalization service must perform in this case are approximately the same as in step 3.2 of use case A, which may be seen as an implementation of model generalization (e.g. Grünreich, 1992) with the main purpose of controlled data reduction (Weibel and Dutton, 1999).

## 4.2 Input Requirements

### 4.2.1 Total Response Time

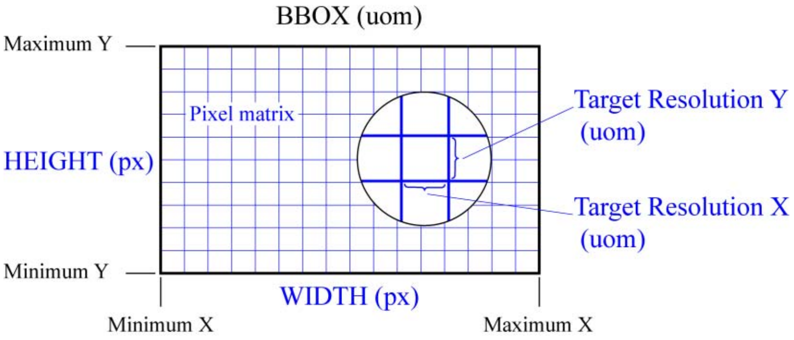
The most obvious requirement for generalization being part of a graphical zoom function (**use case A**) is that generalization must be processed very efficiently, because users are accustomed to very short response times for zooming. Performance is an issue that spans the whole discussion of automated or dynamic generalization in the context of electronic mapping. It is one of the requirements which is hard to satisfy in real-time web applications using today's technologies on high-resolution data. Though performance is certainly a relevant requirement in dynamic use cases (**use case A**), users may accept longer response times in situations where they actively invoke a generalization service (**use case C**). As a trade-off for waiting time they may expect functionalities giving them precise control over the output (compare requirement [2]).

[20] Generalization services should have short response times if used in real-time applications

### 4.2.2 Target Resolution

An essential input to control the generalization process is the *target resolution* (**use cases A-C**). It tells the generalization service to what level of detail the generalization process has to proceed and may simply be a distance on ground denoting the size of the smallest representable feature. For example, a resolution of 2 meters would mean that only objects with a horizontal extension greater than 2 meters or variations in form greater than 2 meters could be represented. The problem with resolution in terms of a distance is twofold. One problem is that distances on a map are given in pixels or centimeters, whereas distances in the data are given in units of measure over ground (e.g. meters) defined by the projection of the data set. A second problem is that distances may vary across the map, which depends on the geodetic datum and the map projection employed. One pixel in the map center is often not exactly the same distance over ground as one pixel in a corner of the map.

To solve the first type of problem, a conversion from pixels to the data set's units of measure must be accomplished. Three parameters that are mandatory in a WMS GetMap request are designed to enable such conversions: `BBOX`, `WIDTH` and `HEIGHT`. `BBOX`, `WIDTH` and `HEIGHT` define a pixel matrix in which each pixel covers an area on ground (OGC and ISO, 2004b) (Figure 4.2).



The figure is based on figure 5 in the WMS Implementation Specification, version 1.1.1 (2002b). „Each pixel covers an area on ground" (OGC, 2002b). Target resolution in X and Y could be specified as the distance over ground covered by the edges of a pixel in the center of the map. The center of each pixel may be used to define the allowed coordinate values for geometry storage. uom = Units of Measure, px = pixel.

Figure 4.2 Target resolution defined by BBOX, HEIGHT, WIDTH

The extent of the data in coordinate measures over ground (defined by `BBOX`) is associated with the extent of the map in pixels simply by fitting the over-the-ground extent into the map extent (defined by `WIDTH` and `HEIGHT`). Because the designers of WMS introduced this kind of association and the parameters are available in every `GetMap` request, it is almost selfevident to use these parameters to calculate the target resolution. Though pixels are square by definition (OGC and ISO, 2004b), `WIDTH` and `HEIGHT` can represent different distances over ground, and a vertical and a horizontal resolution can therefore be specified separately. Other mechanisms to calculate the target resolution may be developed by application designers, but the capability to calculate the target resolution from the `GetMap` parameters `BBOX`, `WIDTH` and `HEIGHT` should be mandatory.

In regard to the second problem (of distorted map space), using a constant target resolution as proposed above introduces positional inaccuracies in the output data, because features in the middle of the map would be generalized differently than features on the edge of the map. The question is if these inaccuracies can be accepted in the case of generalization. At least two pragmatic considerations regarding electronic maps suggest that the answer is yes. First, generalization always introduces positional inaccuracies by changing the form and position of features. Secondly, by definition of the use cases above, the user cannot request the positions of generalized features, but must query the original data to know the exact position of a feature (use case B). Both arguments suggest that it may not be necessary to define overly precise target resolutions.

Unlike in **use case A**, where the target resolution can be calculated from the `GetMap` request parameters, the generalization service in **use case C** has no WMS parameters available. **Use case C** specifies that there should be the options of manual entry of a target resolution or automatic determination by the analysis of the contained geometries. In case of manually specifying the target resolution, the user interface should provide the possibility of using different connotations of scale (e.g. those identified by Bian, 1997), such as a map scale (1:25'000, 1:375'000) or a distance on ground (5 meters, 400 yards, 0.5 nautical miles). To disburden the generalization service, it could be defined that apart from the WMS parameters `BBOX`, `HEIGHT` and `WIDTH`, the generalization service must accept another parameter denoting a distance over ground, e.g. in meters. This parameter could be called `TARGETRESOLUTION`. To offer other measures of a target resolution in the user interface as well as to calculate `TARGETRESOLUTION` from these measures would be the responsibility of the calling application, e.g. the data fusion application in **use case C**,

[21] Generalization services must be able to a) accept a predefined parameter for the target resolution or b) to calculate the target resolution from WMS request parameters

A second case is the automatic determination of the target resolution. One possibility would be to let the calling application perform this task, in which case the application would need to load the data from disk or download it from a WFS or WCS before the generalization request. However, it is not unreasonable to let the generalization service perform this task and to define the disclosure of a GML dataset's resolution in the form of a distance on ground as an essential capability of generalization services. The application would send the data, if the data resided on disk, or would provide WFS/WCS requests and the generalization service would retrieve the data. As a measure for `TARGETRESOLUTION`, the generalization service might search for the smallest distance between any coordinate pair within each dataset and use the largest of these smallest distances for `TARGETRESOLUTION` (the resolution of the lowest resolution dataset). For coverages, the information of the resolution (cell size) can be deducted from the `domainSet` and `lonLatEnvelope` elements a in a GML coverage (OGC and ISO, 2004) or from a `DescribeCoverage` request against a WCS (OGC, 2003b).

[22] Generalization services should be able to determine a default resolution from the geometries contained in a GML dataset

## 4.3 Process Requirements

### 4.3.1 A Concept for Constraint-Based Processing

Once the generalization service in **use cases A and C** has received the input data and parameters, it must perform the generalization process. The generalization process can be understood as a specific kind of transformation performed on GML, much like coordinate transformation or image processing, but with another purpose and work flow. According to the use case descriptions, the input to generalization consists of complex GML and optional SLD elements, and the output consists of less complex or otherwise changed GML and SLD elements. Because GML and SLD elements are encoded in XML, the generalization process is essentially an XML transformation and may comprise the following steps:

- Parse the XML tree
- Construct an execution plan containing transformation instructions
- Perform transformations on XML elements
- Build the new XML tree structure

Parsing the XML tree includes to locate and interpret all elements defining features and their geometries. Constructing an execution plan may be controlled by anything from simple predefined rules to complex and dynamic artificial intelligence based instructions. Performing the transformations means to change GML and SLD elements according to the execution plan.

As a guideline for the transformation process, someone must define the conditions that must be met in order to declare the process as successfully completed. Such conditions are called *constraints* (e.g. OMG, 2003b, see also the Portrayal Model in OGC, 2003a), and they must be resolved at the end of a process, though any number of actions can be applied to resolve them (Beard, 1991). Weibel and Dutton (1998) define a constraint as “a limitation that reduces the number of acceptable solutions for a problem.” Examples of generalization operators using constraints have been implemented by Harrie (1999 and 2004), Burghardt and Meier (1997), McKeown et al. (1999) or Højholt (2000). From a conceptual point of view, it may be more important to define constraints or a language to describe them than the actions to resolve them (because actions are engineered by implementers). A set of generalization constraints can be associated with each kind of geo-spatial object (Ruas, 1998a). In GML, this corresponds to associating sets of generalization constraints with types of features or types of properties, especially geometric properties. Types of features, properties and geometries are defined in GML schemas, and to remind of *Figure 2.2 GML Schemas and Instances*, there are two distinct levels: core and application schemas, the first being defined by the OGC and the second by any application designer or data modeler. Thus, there may exist two principle types of generalization constraints in GML. One type of constraint would be defined by the OGC and concern core schema elements, and consequently would be applicable to any GML document containing these elements. The other type of constraint would cover all other situations, may be defined by application schema designers, and may only be applicable in combination with a specific application schema. For the matter of clarification, constraints that would be predefined by the OGC and generalization functions resolving such constraints shall be named *default* here and all other types of constraints and generalization functions shall be called *advanced*. A set of actions resolving constraints shall be a *behavior*. They shall be defined as follows:

- A *default generalization constraint* is a condition that must be met for the generalization of a particular GML core schema element, e.g. `gml:FeatureCollection` or `gml:Polygon`. Default constraints would be defined by the OGC and have unambiguous names (identifiers).
- The *default behavior* of a generalization service is a set of functions which allows to resolve default generalization constraints[10].
- An *advanced generalization constraint* is a constraint other than a default constraint and may concern the generalization of a particular GML application schema element, e.g. named feature types such as rivers, houses, parcels or user-defined geometries.
- An *advanced generalization behavior* is a set of actions which allows to resolve one or more advanced generalization constraints[11].

The role of default constraints and behaviors is to provide a default way to accomplish generalization on any GML document containing the corresponding GML core elements. For instance, default behaviors would provide a default way to handle line strings, polygons, feature collections (hierarchies in the data) and so on. The main characteristics of these defaults are that

- they are predictable
- they are defined by consensus
- they can be tested
- they can be reused.

Following the concept of core schemas, default constraints should be unambiguously defined and named in an international standard to ensure interoperability. It is common that such standard specifications are published in textual form and that it is up to the implementer to interpret the specification correctly. Any generalization service should provide a default behavior that would cover a minimal set default constraints or GML core schema elements respectively, because then, such services would work with a wide range of GML data. In order to establish generalization as a basic functionality in GIS, this requirement should be mandatory.

---

[10] 'Default' does not necessarily mean simple. Default constraints and behaviors may be concerned with the generalization of very complex core geometries, such as aggregated 3-dimensional solids or composite splines, and may be very difficult to implement.

[11] 'Advanced' does not necessarily mean complex. Advanced refers rather to the possibility to improve the default constraints and behaviors, if an implementer wishes to do so for his application.

[23] Generalization services must provide a default generalization behavior which covers a minimal set of default constraints or GML core schema elements respectively

A conceptual model for generalization should further allow application designers to define advanced constraints and behaviors in order to solve specific generalization problems. The role of advanced constraints and behaviors is to implement better generalization solutions than provided by the defaults. Those who define new advanced constraints must also provide (or initiate the provision of) an advanced behavior capable of resolving the constraint. Although the concept of default and advanced constraints is derived from the idea of core and application schemas in GML, it is not restricted to the application with GML. Instead, it incorporates the simple idea to provide basic classes which can be reused, a concept which can be applied to any data modeling or programming environment.

If the generalization process is guided by constraints, the question must be answered how to conceptually define such constraints in a comprehensive manner allowing the universal application to arbitrary use cases (e.g. **use cases A - C**). The problem shall be viewed from two sides simultaneously. Once by considering the broad categories of constraints proposed by Weibel and Dutton (1998):

- *graphical* – specify minimum size and proximity properties
- *topological* – ensure that existing relationships of connectivity, adjacency and containment between features are maintained or modified correctly[12].
- *structural* – define criteria that describe both spatial and semantic structure
- *Gestalt* – relate to aesthetics and visual balance
- *process* – reflect how operators are selected and sequenced[13],

and once by taking a users perspective, who usually experience generalization on the basis of the graphical output it provides (**use case A**). In his classic book ‘Sémiologie Graphique’ (Bertin, 1973), cartographer Jacques Bertin described how the cognition of a graphical system is controlled by the following graphical variables: *location*, *form* (usually referred to as ‘shape’ in GIS), *size*, *direction*, *brightness*, *pattern* and *color* (Figure 4.3). These variables are valid for the 2-dimensional case and can roughly be adopted for electronic maps such as in **use case A**. According to Bertin (1973), the number of variables in a graphical system must be limited and the other components of the system must be invariant, in order to achieve a consistent graphical representation. Translated to map generalization, this means that some graphical variables must be more invariant during the generalization process than others. In the next three sections, the two different views will be used to illustrate and discuss some requirements concerning default and advanced generalization constraints.

---

[12] Topological constraints ware already covered in 3.2.2, see requirement [9]

[13] This constraint category is concerned with the sequence of tasks between services (service chains) or within a service, and as a matter of implementation will not be covered here.

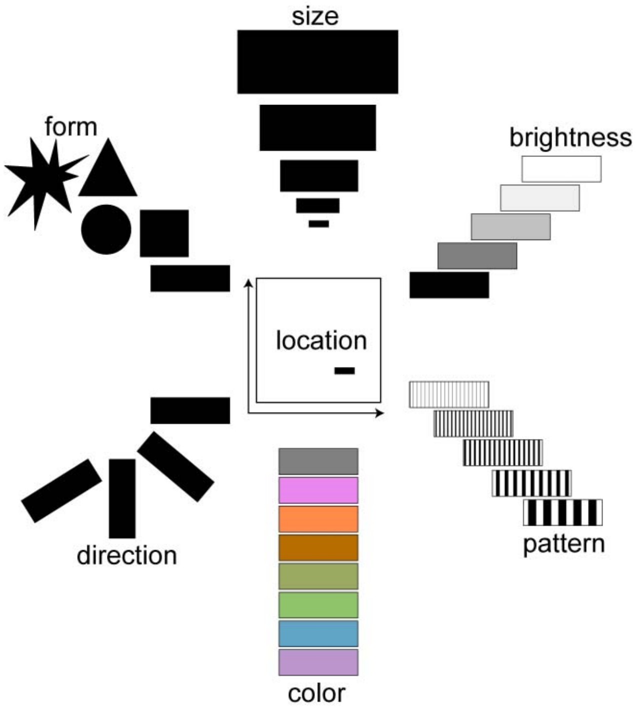


Figure 4.3 Seven variables of a graphical system (Bertin, 1973)

### 4.3.2 Graphical Constraints / the Variables Size and Location

*Graphical constraints* are concerned with the restrictions imposed by the graphical display system and may refer to minimal areas, width or length of features (symbols, markers) or to minimal separations between them (Weibel and Dutton, 1998). *Size* is consequently the most important graphical variable regarding the satisfaction of graphical constraints. It may also be one of the first to consider, since one of the first objectives in the generalization process to strive for is to omit features with geometries too small to be represented at the target resolution. Such features require no further generalization processing. This process has also been referred to as part of *selection* (e.g. Beard, 1991, McMaster, 1991, Ruas, 2002a, Edwardes et. al, 2004). The omission of features may base upon a simple test for each feature geometry: *Is the size of a geometry larger than the target resolution?* Nickerson (1991) has called omission upon this test the 'minimum size rule'. The size of features can be indicated with the 'envelope' in GML (`gml:boundedBy` property) and it may not even be necessary to calculate it. The minimum size rule may be applied to different geometry types (lines, polygons) or symbology types (line markers or picture symbols) and may present one of the most essential default generalization constraints.

[24] Generalization services must implement minimum size rules as part of a default behavior

Preserving a minimal distance between features may require minor displacements and may therefore affect the graphical variable *location*. Of course, the location of features (called 'positions' in GML) should only change to an extent that helps to improve the legibility of the map.

Examples of potential default graphical constraints might comprise:

- the size of polygons (instances of `gml:Polygon`) should not change significantly relative to the size of other polygons
- the position of features or symbols should not change during generalization, unless by the influence of features or symbols with a higher precedence in competition for map space
- if displacement is necessary between linear features (instances of `gml:LineString`), a default minimum separation distance shall be guaranteed

Examples of potential advanced graphical constraints that an application designer might define:

- if [x] and [y], aggregate neighboring buildings (instances of `my:building`[14])
- avoid self-intersection and preserve a minimum separation distance between contour lines (instances of `my:contourLine`)
- roads (instances of `my:road`) have precedence over rivers (instances of `my:river`) if conflicts of map space occur

### 4.3.3 Structural Constraints / the Variables Form and Direction

The graphical variable *form* may often be the main focus in the generalization of individual features (e.g. parcels in **use cases A and C**). For groups of features, additional *structural* characteristics must be accounted for, such as parallel arrangements, clusters, feature density or directional arrangements (Weibel and Dutton, 1998). *Direction* is usually among the characteristics that are more invariant than variable, since it shows the orientation of features in space. However, it may be more important in some situations to show the orientation relative to other features, e.g. if features are oriented parallel to a line. The variable *direction* may be relevant in road or river networks (direction of water or traffic flows). Structural constraints may not only be concerned with form and direction, but at least with two more structures that are available in a GML: topology and feature collections. As already discussed in sections *3.2.1 Geographic Features* and *3.2.2 Spatial-temporal Geometry and Topology*, generalization services are obliged to retain the grouping and hierarchy defined by feature collections (requirement [6]) and the geometric associations defined by topologies (requirement [9]). There may be more structural characteristics that one might consider, such as categorizations of properties or preserving logical context, “e.g. a building that falls into a lake is usually out of context” (Weibel and Dutton, 1998).

Different GML geometry types have different characteristics concerning the form (shape) that a geometry of such a type may have. Thus, different structural constraints may be required for different GML geometry types. Before listing some examples of potential structural constraints, a short overview over GML geometries shall be given and the implications on the variable *form* shall be discussed.

---

[14] “my:” stands for a namespace defined in a user-defined application schema

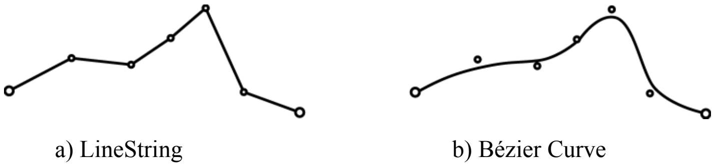
The GML 3 geometry schemas are an implementation of a subset of ISO/DIS 19107 (Lake et al., 2004). The geometry types and elements of GML 3.1 are grouped in five schemas (Table 4.1).

Table 4.1 GML 3.1 geometry types and schemas

| schema | contents |
|---|---|
| `geometryBasic0d1d.xsd` | • `Point` (0D)<br>• `LineString` (1D, linear interpolation) |
| `geometryBasic2d.xsd` | • `Polygon` (2D, linear interpolation) |
| `geometryPrimitives.xsd` | • Curves (1D, composed of segments)<br>  • `LineStringSegment`<br>  • `Arc, ArcByBulge, ArcByCenterPoint, ArcString, ArcStringByBulge`<br>  • `Circle, CircleByCenterpoint`<br>  • `CubicSpline, BSpline, Bezier`<br>  • `Clothoid, GeodesicString`<br>• Surfaces (2D, composed of one or more surface patches that are connected one to another)<br>  • `PolygonPatch`, `Triangle`, `Rectangle`<br>  • `Ring`, `LinearRing` for boundaries<br>  • `PointGrid`<br>  • Gridded Surfaces: `Cone`, `Cylinder`, `Sphere`<br>  • Polyhydral Surfaces: `polygonPatches,TrianglePatches`<br>  • Triangulated Surfaces: `Tin`<br>• Solids (3D. The extent defined by the boundary surfaces → shells) |
| `geometryAggregates.xsd` | • Aggregates (geometry collections with no specific constraints)<br>  • `MultiGeometry`, `MultiPoint`, `MultiCurve`, `MultiSurface`, `MultiSolid` |
| `geometryComplex.xsd` | • Composites (contiguous collections of geometries)<br>  • `CompositeCurve`, `CompositeSurface`, `CompositeSolid`<br>• Complexes (consist of different composites)<br>  • `GeometricComplex` |

GML 3.1 (OGC and ISO, 2004a) defines over 50 distinct geometric types (based on 178 different GML classes) that can be used in application schemas for the construction of points, curves, surfaces and solids. The terms Primitive, Complex, Composite and Aggregate derive from ISO 19107 – Spatial Schema (ISO, 2000), which is the basis for GML 3.1.

`GeometryBasic0d1d.xsd` contains elements useful to define points and lines, `geometryBasic2d.xsd` contains elements to define polygons. Many or even most applications will not require more than these two schemas. Together with `geometryAggregates.xsd`, they also contain elements required for backwards compatibility with GML 2.1.2 (OGC, 2002d). The two schemas `geometryPrimitives.xsd` and `geometryComplex.xsd` contain entirely new GML 3.0/3.1 geometry elements (Lake et al., 2004). **Primitives** are geometric objects that can not be decomposed into other objects (OGC and ISO, 2004a). All geometry instances in actual GML documents derive directly or indirectly from such primitives. `Point`, `LineString`, `Polygon`, `Arc`, `Circle`, `CubicSpline`, `Bezier`, `PolygonPatch`, `Triangle`, `Rectangle` or `PointGrid` are examples of GML 3.1 geometric primitives. The backbone of geometries are their **control points**. Control points correspond to tuples of coordinates and are relative to some Coordinate Reference System CRS. Control points can be encoded with the `pos`, `posList` or `pointRep` elements (OGC and ISO, 2004a), the `coordinates` element is available for backwards compatibility with GML 2. The sequence of coordinate tuples in these elements defines the start-to-end direction of the primitive. The shape of a primitive between the control points is defined by an **interpolation function**, which in the simplest case is linear. Figure 4.4 depicts a `LineString` and a `Bezier` curve with 7 control points each as examples of GML primitives with different interpolation functions.


a) LineString

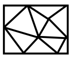
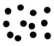
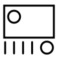
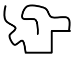
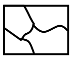
b) Bézier Curve

Figure 4.4 Geometric primitive examples with different interpolation functions

In the following, the most obvious implications of the shape of GML 3.1 geometries on the process of simplifying individual geometries shall be discussed. Note that when looking at GML data, we look at instances of geometries, and only indirectly at the primitives (schemas) they derive from.

The simplest of all geometries is the Point. In terms of graphical variables, it has no shape or size, only a location. It cannot be simplified. Some other geometries also have a regular shape with very few control points and a predefined interpolation function, such as straight lines, circles or triangles (Table 4.2a). Their shape cannot be simplified either. All geometries, for which the number of control points is low and where the relative positions of their control points and their interpolation function cannot vary, fall in this category. The default behavior of generalization can therefore only change the size of these geometries or reduce the shape to its location or to a point. Grids are a type of geometry that also have a regular shape, but consist of many control points (Table 4.2b). If the grid pattern is equally spaced and orthogonal, such grids can be simplified by aggregating 4, 16, 32 and so forth neighboring points or cells. The shape of the new grid will have the same visual appearance, but will consist of less control points. Changes of the grid in size are not without effect on the properties of the grid points or cells, because the values of grid points or cells are directly associated to a location. Grids (e.g. raster imagery) are such a common data source in GIS that they certainly deserve special attention, e.g. by using specialized processing services for generalization (see also OGC and ISO, 2002).

Table 4.2 GML geometries (0, 1, 2 D) and possibilities to simplify their shape

| **type of geometry** | **examples** | **simplification of shape** |
|---|---|---|
| a) few control points / regular shape | • point<br>— straight line<br>⌒ arc<br>○ circle<br>△ triangle<br>▭ rectangle | none |
| b) many control points / regular shape | ⠿ point grid<br>▦ rectangular grid | aggregation of 4/16/32… |
| c) many control points / irregular shape |  line string<br> polygon, surface<br> curve<br> irregular surface, e.g. TIN | elimination of selected control points |
| d) many control points / complex geometry |  aggregated points<br> aggregated geometry<br> composite curve<br> composite surface | elimination of selected control points + aggregation / simplification / elimination of member geometries |

a) The regular shape of these geometries cannot be simplified (generalized), because their interpolation function and the number and relative positions of their control points cannot vary.

b) The shape of grids can be simplified by aggregating 4, 16, 32 and so on points or cells.

c) Irregular geometries are common in GIS, especially line strings and polygons. Their shape can be simplified by eliminating selected control points.

d) The overall shape of complex geometries (composites, aggregates, complexes) is determined by their boundaries. The shape can be simplified by simplifying the boundary and by simplifying, aggregating or eliminating the individual member geometries.

A third category comprises geometries with an irregular shape and many control points (Table 4.2c), characteristics which apply to many geometries stored in GIS, especially lines and polygons. `LineString` geometries may contain any number of control points greater than one, and `Polygon` geometries any number greater than two. For those geometries that need to be simplified, the generalization process must identify which control points can be eliminated without violating constraints defined on the shape. A corresponding default behavior can be based on three types of information concerning the geometry:

- the number of control points
- the type of interpolation function between control points
- the direction and length of vectors that define the interpolation function between control points (for spline functions, these vectors must be defined in GML, for lineStrings, vectors between control points can be constructed from the sequence and relative positions of the control points)

Different constraints may not only be applicable to different geometry types, but even to different types of interpolation functions (e.g. `linear`, `elliptical`, `tin` etc., see the `primitives.xsd` schema for predefined interpolation types in GML 3.1).

A fourth category (Table 4.2d) pools geometries being assembled from other geometries. The shape of complex geometries is given by its outermost boundary, which can be defined explicitly for some complex geometry types. Generalization might view complex geometries as boundaries with an 'inner structure". Because the boundaries and inner structures are composed of various member geometries, which are themselves composed of geometric primitives, the same default generalization constraints may be valid for the complex geometries as for the primitives. From a graphical point of view, complex geometries may be very similar to feature collections, differences between the two may in fact be marginal. Modelers have always the choice to model complex entities as feature collections or as a single feature with a complex geometry (Lake et al., 2004). However, that modeling choice should not significantly effect the result of generalization.

From the considerations on constraints concerning the geometric shape of GML 3 geometries, the following application specific requirements for generalization can be deducted:

[25] Generalization services should be able to retain structural characteristics of features and feature groups (e.g. complex polygonal structure should not be oversimplified to a triangle, a curve with strong bends should retain strong bends, a principally rectangular or a round shape should remain principally rectangular or round, a group of linearly aligned features should stay aligned etc.)

[26] Generalization services should provide different specialized generalization algorithms for the different types of geometries and interpolation functions.

[27] Generalization services should be able to process complex geometries and their boundaries.

[28] Generalization services should account for style elements in the resolution of graphical conflicts, if those style elements influence the size or shape of feature representations

A common problem when aggregating features is how to aggregate their thematic properties. This issue is usually referred to as 'semantic generalization'. In map generalization, it is of particular importance for those properties that are used for labeling and conditional styling (color sets). What should be done? Should the property values be recalculated, should they be reclassified (and how) or should they be dropped? The answer is that it depends – e.g. on the type of variable. Different rules apply for nominal, ordinal, ratio or interval types of properties. Following the concept outlined above, *constraints* could be used to define such rules. While ratio or interval variables allow to recalculate averages or sums (e.g. for temperatures, prices, weights etc.), nominal and most ordinal variables can only be reclassified. A constraint for property aggregation must specify a) the type of variable and b) the type of conversion (calculate average, calculate sum, reclassify to …). Because GML does not foresee the possibility to define the variable type of feature properties, property aggregation must be described with advanced constraints. Since property aggregation is common in generalization, it may not make sense that each and every user define their own constraints, which might offer interesting opportunities for the vendors of GIS systems and services to offer *their* advanced constraints and behaviors..

Examples of potential default structural constraints might comprise:

- The shape of a feature geometry may be subject to considerable change during generalization as long as the visual characteristics of that shape are preserved (e.g. right angles, convexity/concavity of areas).
- Members of feature collections can only be aggregated with other members of the same collection.

Examples of potential advanced structural constraints that an application designer might define:

- Buildings or cities (`my:building`, `my:city`) can not overlap with lakes (`my:lake`)
- Rows of buildings parallel with a road must remain aligned to that road
- When the digital elevation model which defines the watershed is generalized, tThe direction and location of a river must be preserved correctly (water must flow downhill along the line of least resistance)
- The price of houses (a property of `my:house` features) must be averaged when houses are aggregated.

### 4.3.4 Gestalt Constraints / the Variables Pattern, Brightness and Color

Gestalt constraints relate to aesthetics and visual balance (Weibel and Dutton, 1998), which comprises many subjective aspects of the overall map impression, such as smoothness, color balance, contrast(s), focus or orderliness. Bertin's (1973) graphical variables, pattern, brightness and color may be associated with Gestalt constraints to some extent. Pattern, if applied as symbology to laminar features, should adapt according to the relative change in size of the corresponding features. Though brightness and color are usually kept invariant across scales (red remains red), undesired changes in the overall map impression may result when changing scale. This may for example be due to the changes in relative size of the 'color patches', i.e. the symbols, on the map (See Itten, 1974, for an in-depth discussion on contrasts and on the effects of the relative distribution of colors). Actually, such effects are most apparent when a map is not generalized at all, the result usually being an overcrowded illegible map with a grayish impression. Application designers may want to correct such negative effects by slightly correcting colors, brightness or patterns. The benefit of such corrections may be less obvious for 2-D maps than for 3-D visualizations, where it is common to manipulate the granularity and color contrast of the fore- versus the background in post-processing rendering (e.g. Häberling, 2003).

Symbology, such as color, brightness and pattern is defined in Styled Layer Descriptor *SLD* documents (OGC, 2002c) or with SLD elements within GML (OGC and ISO, 2004a). Rendering a map based on GML data and SLD styles is the role of Web Map Services *WMS* (OGC and ISO, 2004b), such as in **use case A**. The map generation approach of WMS is to render a sequence of feature types as named layers, e.g. `Rivers`, `Roads`, `Houses` etc. applying a sequence of named styles, e.g. `BlueLine`, `CenterLine` or `Filled`. The order of layers is given by the sequence of layers and styles in the SLD document or the WMS request.

Layers defined first are drawn first, an approach called the 'painters model'. `FeatureTypeStyles` and `Rules` can be used for scale-dependent styling (OGC, 2002c). SLD may influence the size of objects on the map by applying thick line styles (see graphical constraints above), but it does not influence the position or shape of geometries (see structural constraints). Generalization services should be able to account for the model deployed in WMS and SLD, specifically:

[29] Generalization services must account for scale-dependent and conditional styling

[30] Generalization services should account for the sequence and content of (WMS) layers (painters model) when resolving graphical conflicts.

Examples of potential Gestalt constraints might comprise:

- The over-all contrast of a map shall be kept constant when changing scale
- The information density in the map shall be kept constant when changing scale (principle of 'constant information density' by Frank and Timpf, 1994)

### 4.3.5 Notes on Default and Advanced Constraints

Default constraints focus on GML core elements and could be used with any GML dataset containing these elements, but would allow for no sophistication in the generalization process. In many cases, e.g. when providing thematic point data together with a background raster map (e.g. Burghardt et al., 2004), application requirements may be simple and the default generalization behaviors may suffice to satisfy those requirements. Additionally, the role of maps is changing due to electronic mapping: "… typical maps generated in GIS are no longer complex multi-purpose maps with a multitude of feature classes involved, but rather single-purpose maps consisting of a small number of layers" (Weibel and Dutton, 1999). However, many applications (such as the CIS in **use case A**) will have specific generalization needs, e.g. regarding specific feature types or specific graphical conflicts. Advanced generalization constraints would allow application designers to address such needs and would focus on elements defined in application schemas or metadata profiles. In fact, one can imagine many of the 'classical' generalization operations to be guided by advanced constraints, because many of those require some foreknowledge about a specific feature type or at least must be able to identify that feature type, e.g. the in the generalization of buildings (e.g. Ruas, 1998b, Sester, 2001, Lal and Meng, 2004, Forberg, 2004), road networks (e.g. Van Kreveld and Peschier, 1998, Wang and Doihara, 2004), topography (e.g. Weibel and Heller, 1991) or contour maps (Li and Sui, 2000). Apart from the use in a specific thematic domain, advanced constraints might help to solve specific geometric or graphic problems, such as the smoothing or displacement of lines (Burghardt and Meier, 1997), e.g. by replacing linear interpolation by spline interpolation. Yet another set of problems that might fall in the domain of advanced constraints and behaviors may be associated with specific strategies of data modeling, e.g. polygon or raster classification schemes (e.g. Galanda, 2003). This particularly abundant type of data modeling (used e.g. for land cover, vegetation cover, land use classes) is a simple alternative to modeling parts-of-whole relations with feature collections and contains useful aggregation information for generalization.

Contrary to default constraints, advanced constraints must be technically exchangeable and machine-readable, to enable a service to check the constraints. For this purpose, constraints must be unambiguous and typed. It would certainly be efficient and flexible, i.e. interoperable, if application designers could revert on a standardized schema for the description of advanced constraints. Such a schema might for example be based on the Object Constraint Language (OMG, 2003b), a recent specification which is part of the Unified Modeling Language (UML, 2004). However, in today's OGC framework, a standard for the description of generalization constraints is lacking and would require the development of:

- A General Model for generalization and generalization constraints to explain the theoretical foundations of generalization constraints (Section 6.2.1.1 of the ORM, OGC, 2003a)
- an XML encoding for generalization constraints (Section 6.3 of the ORM, OGC, 2003a), defined within a) an additional GML schema, b) an ISO 19115 metadata profile or c) a separate encoding specification, which should be based on a formal language.

## 4.4 Output Requirements

The resulting output of the generalization process in **use cases A and C** is geographic data encoded as GML and style encoded as SLD (**use case A** only). Following the argumentation that generalizing GML is an XML transformation, a new XML tree or several trees must be constructed for output, after the transformations on individual XML elements have been completed. Because generalization is usually an intermediate process between the query and the display or analysis of geographic data, generalization services should principally provide output in the same language or encoding as they received the input. Furthermore, which is intuitive, generalization should not change the number of datasets. If four datasets of GML data are given as an input, a generalization service should provide four datasets as an output.

[31] Generalization services should provide the same number of datasets in the output in the same language/encoding (GML, SLD, …) as they received the input

**Use case B** demands that the user can query original information directly from a generalized map. To support such functionality, the map symbols must include information that allows to link back to the original features[15]. One way of referencing original features is to use the generalized shape or map symbol boundary and perform a zonal query. In other words, location can be used to relate generalized representations of features and feature groups with the original features. However, location-based queries would only be approximate, since generalized shapes do not necessarily coincide with the boundaries of the contained original features. Different implementations would thus yield different query results. A more deterministic approach to

---

[15] Remember that service operations are usually stateless - the client can only know what the server provides.

reference original features would be the use of feature identifiers (IDs). Generalized map objects are related to the original features in one-to-one or one-to-many or many-to-many relationships. The definition of ID-based links would allow to maintain these relationships in the output. The concept to provide feature identifier links with the generalized representations of features pointing to the original features shall be termed 'connected'[16] generalization output here. A problem that must be solved with connected output is the potentially large number of links. If all feature links would be encoded in the output, the number of links would be equal to the number of original features. This is obviously neither practical nor desirable, since this number may be excessive and contradict the objective of generalization to provide simplified output. No solution is provided in this thesis how to implement connected output. It shall suffice conceptually to note the potential need (**use case B**) to encode connected output, and then to delegate the question of how to do this to further research and implementers. However, the concept of connected output is an elegant way to by-pass the problem of semantic generalization (see 4.3.3), because original property values could be accessed ex post. Some brief notes on this topic are added to the discussion in chapter 6.

[32] Generalization services should be able to provide GML output that encodes features (for which generalization simplifies, dissolves or hides its geometric properties) as links to these features

## 4.5 Conclusions

Chapter 4 introduced three different use cases for the generalization of GML data representing cartographic generalization (**case A**), information retrieval from generalized data (**case B**) and model generalization (**case C**). It was argued that the generalization of GML can be regarded as an XML transformation process. It was further argued that two types of generalization constraints should be defined, default and advanced, to comply with the delegation of responsibilities in GML, which is a key concept of the GML specifications. Default constraints would have to be defined by the OGC and would comprise conditions concerning core GML elements and basic SLD elements. Their role would be to provide a default way of generalizing any GML data. Advanced constraints would cover all other generalization situations, and would for example comprise conditions concerning application schema elements. The definition of advanced constraints would be the responsibility of application schema designers. Introducing advanced generalization constraints in the OGC framework would presuppose the prior development of a General Model for generalization (according to the ORM: OGC, 2003a) and an XML-based encoding, which might for example be based on the Object Constraint Language (OMG, 2003b). Various examples of default and advanced constraints were given in textual form. Assuming an adaptation of the corresponding standards, constraints might be a part of GML application schema, metadata or SLD documents. Table 4.3 and 0 list the requirements that were educed in this chapter. Table 4.3 contains mandatory requirements based on the use cases and associated considerations concerning GML. They are regarded to be mandatory in the sense that

---

[16] The idea of this terminology is to highlight the principle ability to *connect* from the generalized data back to the source data (which is analogous to the general concept of hyperlinks).

a conceptual model must necessarily provide mechanisms to fulfill them. 0 lists requirements which are optional in this respect.

GML 3 provides an extensive framework of modeling classes for zero to three-dimensional geographic information based on ISO 19107 – Geographic Schema (ISO, 2000). This diversity offers opportunities for application development and might promote the interoperability and integration of a wide range of GIS, planning and engineering applications. On the other hand, introducing many modeling classes also introduces complexity for functions that must be able to process any GML dataset, such as a default generalization behavior. To establish generalization as a basic functionality in the OGC framework, default generalization constraints would need to be defined by the OGC, and the GML, SLD and/or metadata specifications would need to be extended with the possibility to define advanced constraints. An implementation specification for generalization services should demand that any generalization service should be able to cover a minimal set of default generalization constraints.

Table 4.3 Mandatory generalization service requirements based on three use cases

| **Generalization Services ...** | **justification** | **Use Case*** | **no.** |
|---|---|---|---|
| must be able to a) accept a predefined parameter for the target resolution or b) to calculate the target resolution from WMS request parameters | The target resolution is necessary to indicate the target state of generalization. It must be provided or calculated before generalization. | A | [21] |
| must provide a default behavior which can resolve a minimal set of default generalization constraints | Fulfilling this requirement makes sure that a generalization service can process any GML document containing specific types of GML core schema elements. | A, C | [23] |
| must implement minimum size rules as part of a default behavior | Defining minimum sizes is essential for almost any kind of generalization and should therefore be a default. | A, C | [24] |
| must account for scale-dependent and conditional styling | SLD supports scale-dependent and conditional styling. Hence, it must be accounted for in generalization. | A | [29] |

* The three use cases A, B and C are described in chapter 4.1.

Table 4.4 Optional generalization service requirements based on three use cases

| Generalization Services ... | justification | Use Case* | no. |
|---|---|---|---|
| should have short response times if used in real-time applications | Users expect short total response times in real-time applications. | A | [20] |
| should be able to determine a default resolution from the geometries contained in a GML dataset | This requirement is important in cases where the user wants to adjust the geometric level of detail of one or several datasets. | C | [22] |
| should be able to retain structural characteristics of features and feature groups | This requirement refers to the resolution of structural constraints. | A, C | [25] |
| should provide specialized generalization algorithms for different types of geometries and interpolation functions | Different types of geometries and interpolation functions are associated to different generalization problems. | A, C | [26] |
| should be able to process complex geometries and their boundaries | Complex features are part of GML 3.1 and should consequently be accounted for. | A, C | [27] |
| should account for style elements in the resolution of graphical conflicts | Style elements, e.g. line-width, symbol or text label size, may influence the size or shape of feature representations. To account for style is a basic requirement for any cartographic generalization. | A | [28] |
| should account for the sequence and content of (WMS) layers when resolving graphical conflicts | The sequence of layers may be used as a simple order of precedence between layers in the resolution of graphical conflicts. | A | [30] |
| should provide the same number of datasets in the output in the same language/encoding (GML, SLD, …) as they received the input | Generalization is usually an intermediate processing service, e.g. between a WMS and a WFS. | A, B, C | [31] |
| should be able to provide GML output that encodes features as links to these features | This requirement is essential if generalized features should allow to link back to exactly the contained original features. | A, B, C | [32] |

* The three use cases A, B and C are described in chapter 4.1.

# 5 A Conceptual Model for Feature Generalization Services

This section presents a possible conceptual model for *Feature Generalization Services*, based on the requirements discussed in prior chapters (see tables 3.3, 3.4, 4.3 and 4.4). The model is presented in a form similar to an Essential Model according to the Open Geospatial Consortium's (OGC) Abstract Specification Overview (OGC, 1999a). An Essential Model "should explain in real world terms the objects, interfaces, behaviors, and parameters that are the subject of the Topic". Feature Generalization Service (FGS) is a feature processing service type that has been identified in section 8.3.5.2 of ISO 19119, Geographic Information Services (OGC and ISO, 2002), but for which no further draft or implementation specifications have been published up to date. This chapter proposes some considerations, parameters and capabilities for feature generalization services as a potential basis for such specifications. References to requirements from chapters 3 and 4 are given in square brackets (e.g. [15]) to indicate a relatedness with the corresponding topic.

## 5.1 Context

Geographic Information Communities need a technology that allows them to use geographic data in a way that enables users to dynamically explore and analyze that data across a wide range of map scales. Using geographic data across scales requires that the level of detail (LoD) in the data is adapted to a level of detail that can be represented at a requested scale. The LoD in a specific geographic dataset (e.g. produced for the use at a scale of 1:25'000) can only be decreased for the use at smaller scales, e.g. 1:200'000, but it cannot be increased. The dynamic reduction of the LoD in geographic data is the task of Feature Generalization Services (FGS).

Generalization is essentially an abstraction process ([1], [20]). The aim of generalization is to highlight the important information and to hide the unimportant information, in order to optimize the readability of the map at a specific scale or to minimize the volume of the data. Generalization is thus a complex process including the simplification of detail in the data, most often geometric detail or textual information (labels), as well as the exaggeration of detail that is particularly important in a specific application.

In electronic map applications, generalization is ideally initiated as an integrated function of the zoom, for which purpose generalization must be fully automatic. Employing generalization as a service in the OGC web services framework ([3], [14] - [18]) might facilitate the use of high-resolution geographic data at many scales in many applications, e.g. for exploratory visualizations. The OGC web services framework allows the construction of service chains in order to solve complex problems. FGS could be used as an intermediate component in such service chains, for example between Web Map Services, which carry out the map display (OGC and ISO, 2004b) and Web Feature Services (OGC, 2002a) and Web Coverage Services (OGC, 2003b), which provide the vector and raster data. Different types of specialized generalization services may exist, e.g. for coverages, tins or non-linear geometries ([10], [11], [26]).

High-resolution data, which is increasingly available in the GIS community, provides the best basis for dynamic generalization, because it contains most detail. Using high-resolution data in combination with generalization services may be beneficial for Geographic Information Communities by reducing redundancy problems (instead of many datasets for multiple scales, only few high-resolution datasets need to be maintained), by giving users real-time access to high-resolution data and by expanding the fields of application for high-resolution data.

## 5.2 Essential Functions

The goal of generalization functions is to satisfy *generalization constraints* ([5] - [9]). The way how the functions attain this goal is not predefined. Generalization constraints are equality or inequality relations concerning the values of arbitrary map, symbology or data properties. Constraint satisfaction attempts to assign values to these properties so that the constraints are true. Generalization constraints can generally be used as conditions representing the relative importance of visual feature or map symbol characteristics, with the underlying idea that important characteristics should be more invariant in the generalization process than unimportant characteristics. This essential model proposes to define generalization constraints as follows:

- Two basic categories of generalization constraints exist: default and advanced.
    - **Default constraints** relate to GML, SLD or metadata core schema elements[17]. They are specified by the Open Geospatial Consortium (OGC) and published in one or several of their standards. ([23])
    - **Advanced constraints** cover all other cases. For example, they may refer to GML application schema elements or may be used to override default constraints. They can be defined by anyone. ([2], [19])
- Generalization constraints are formalized descriptions which may inherit other generalization constraints.
- Generalization constraints can be defined within three types of documents[18]: Geographic Markup Language GML application schema (OGC and ISO, 2004a) ([4]), Styled Layer Descriptor SLD (OGC, 2002c) ([29], [30]) and ISO 19115 Metadata (OGC and ISO, 2000) ([12]).[19]
- A set of functions to resolve a set of generalization constraints is called a generalization *behavior*. Corresponding to the two types of constraints, there are two types of behaviors: default and advanced. A Feature Generalization Service FGS may provide alternative behaviors for the same set of constraints.

Figure 5.1 illustrates the proposed relationships between the behaviors of a generalization service and the generalization constraints using UML elements: A default behavior may resolve

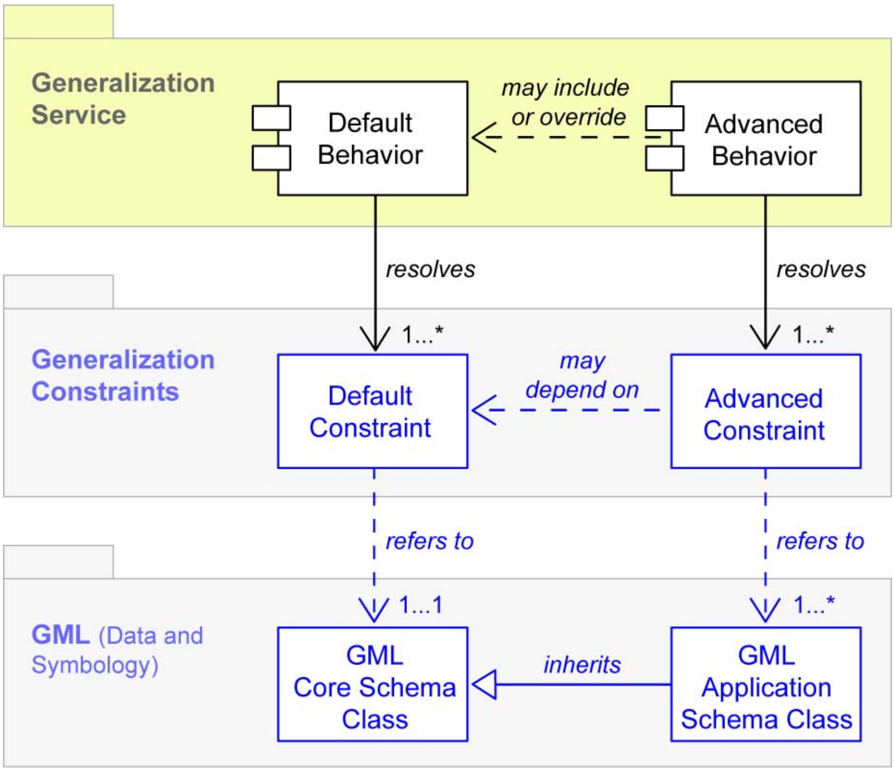
---

[17] Strictly speaking, *core schema* elements only exist for GML and ISO 19115 Metadata, not for SLD. The term *core* shall also be used for SLD elements with predefined names in this context, e.g. `LabelPlacement`, `PointSymbolizer` or `Stroke` (see OGC, 2002c).

[18] In the GML context, a document is an XML file containing structured descriptions (e.g. data or a model) based on a specification. A document is not the specification itself.

[19] The role of these types of documents for the description of generalization constraints should be clarified in an implementation specification for FGS.

one or several default constraints which refer to one GML (SLD or metadata) core element each, an advanced behavior may resolve one or several advanced constraints which may refer to one or several GML application schema elements.



This diagram shows UML elements and combines the implementation view (behaviors) with the information view (constraints and GML classes). The term ‘default’ refers to the capability to perform a default way of generalization on any GML document, ‘advanced’ refers to the possibility for application designers to extend the default behavior. The constraints define the generalization result, the behaviors generate that result.

Figure 5.1 Relationships between behaviors, constraints and GML classes

### 5.2.1 Default Behaviors

The role of default constraints and behaviors is to provide a default way of generalization on any arbitrary GML dataset. Because the default constraints are to be defined in international standards, the result of different implementations of default generalization behaviors should vary only marginally across implementations – and should thus yield predicable results for the user. The default behaviors of a FGS allow the satisfaction of a set of default generalization constraints. Each default constraint is associated with one particular GML core element or SLD element ([23] - [29]). The feature generalization service implementing default behaviors must indicate the core elements or schemas for which it supports generalization, i.e. it will provide behaviors to resolve all default constraints associated with those core elements. A client requesting generalization can thus initiate default behaviors on any GML document containing the supported core schema elements. Besides the data, the only required input is a target resolution (or the map and data extents, which allow to calculate a target resolution). In many situations, a simple default generalization may suffice or may at least be better than no automatic generalization at all.

*Capabilities and Request Parameters*

- Capabilities Document – Like any OGC service, a FGS must be able to describe its capabilities and provide a capabilities document upon request of a client ([13]).
- Version support – A FGS must indicate the GML schemas or elements it supports. It can partially or fully support one or several GML versions. The request to a FGS must contain a statement indicating the GML elements, schemas or versions supported by the FGS. ([23])

  Implementation might define that the value of a GMLVERSION CGI parameter in the FGS request may be a GML version number, e.g. `3.1.0` which would mean full support of that GML version, or a list of individual GML schema names, given as tuples of schema and GML version e.g. `geometryBasic0d1d.xsd,3.1.0`. A schema name would indicate that all elements in this schema as well as all elements in the schemas included within that schema are supported. Additionally, an optional inclusion or exclusion list of GML core elements, e.g. `clothoid`, `pointGrid`, `Tin`, and so on, could be indicated. From GML 3 on, it could be recommended that the `feature.xsd` and `topology.xsd` schemas are supported, which include `geometryBasic0d1d.xsd`, `geometryBasic2d.xsd`, `temporal.xsd`, `units.xsd`, `gmlBase.xsd`, `measures.xsd`, and `basicTypes.xsd`.

- Model and Cartographic generalization – A FGS must indicate if it is capable to account for symbology in the generalization process (cartographic generalization). This capability is additional to the generalization of features and their geometries (model generalization), which is the minimum capability. Accounting for symbology may influence the generalization process, because the size and shape of symbols may differ significantly from the size and shape of feature geometries. The FGS request should allow to globally specify if symbolizations shall be accounted for in the generalization process or not. ([28], [29])

  Implementation could foresee a USESTYLE CGI parameter for the FGS request with values ‘yes’ or ‘no’.

- Output – Generalization services should provide the same number of datasets in the output in the same language/encoding (GML, SLD, …) as they received the input ([31]). Every generalization service should be capable to provide ‘connected’ generalization output ([32], chapter 4.4).

  Implementation might define an `OUTPUT` CGI parameter for the FGS request with values ‘connected’ or ‘disconnected’. The concept of ‘connected’ output is not further investigated here, but some brief notes are provided in chapter 4.4 and the discussion in chapter 6.

- Target resolution - Generalization is regarded as a mapping of a set of features or their symbology from a source resolution (resolution of the source dataset) to a target resolution (resolution of the target dataset). A target resolution, given as a distance over ground, or parameters suitable to calculate a target resolution must therefore be provided in an FGS request. Every generalization service must be capable to calculate a target resolution from the `BBOX`, `WIDTH` and `HEIGHT` parameters of a WMS request. ([21], [22])

Implementation might define a `TARGETRESOLUTION` CGI parameter. The value of `TARGETRESOLUTION` might be a comma separated list of one to three distances over ground in meters, denoting the resolution in x, y and z directions. Giving only one value would indicate equal resolution on all axes, two values different resolutions on the x and y axis for 2-dimensional data and three values the resolutions for 3-dimensional data.

- WMS requests – Every FGS should be able to interpret and/or forward complete Web Map Service, Web Feature Service and Web Coverage Service requests. This facilitates the inclusion of generalization services in service chains. ([14], [17])

*Default Generalization Behaviors*

- Geometric Simplification – Default behaviors should be able to simplify the shape of geometries and reduce their size. Geometries that are smaller than the target resolution shall either be collapsed to a point or they shall be aggregated with one or more other geometries at the same location to a new feature geometry ([5], [7], [10], [25], [26]).
- Cartographic Generalization - If the option of cartographic generalization is supported and a WMS request is provided as an input, default behaviors should at least account for:
  - The size of point and picture symbols (`Graphic` or `Mark`). ([28])
  - The width of line markers (`Stroke`). ([28])
  - Classifications of properties defined for conditional rendering. ([29])
  - Graphical conflicts caused by the overlay of data in layers. ([30])

The SLD specification provides a mechanism to define different rendering styles for different scale ranges (within the `FeatureTypeStyle` and `Rule` elements, which are "… used to group rendering instructions by feature-property conditions and map scales." OGC, 2002c). The difference between using scale-specific SLD rules and using default constraints with a FGS is that the FGS a) works continuously across scales, b) is capable to modify the shape and location of feature geometries and c) works on a wide range of GML documents (without the need of prior work by the application designer).

### 5.2.2 Advanced Behaviors

Advanced behaviors of a FGS are sets of generalization functions supplementary to the default behaviors. The goal of advanced behaviors is to satisfy advanced constraints. Advanced constraints would be described with a formal language, which is yet to be defined (a corresponding XML encoding could for example be based on the Object Constraint Language OCL, OMG, 2003b.). Data modelers can use advanced constraints to resolve specific generalization problems associated with GML application schema elements or SLD elements or with a certain combination of feature types, or they may use advanced constraints to override particular default constraints. Among the potential fields of application for advanced constraints and behaviors are:

- Model Generalization - Advanced constraints and behaviors may be useful for a variety of structural generalization tasks: Simplifying and aggregating categorized polygon data, generalizing contour lines in consideration of topography, generalizing road networks or landscape mosaics, or recognizing ridges, valleys, dams or bridges in TIN surface models, (See section 4.3.5). ([25])

  Implementation might define that an advanced constraint given the name of an existing default constraint will override the corresponding default constraint.

- Cartographic Generalization – Advanced constraints and behaviors may be useful for a variety of cartographic demands, such as gradually changing the size of symbols for point data within certain scale ranges, retaining group characteristics (e.g. the alignment of buildings parallel to a road), dynamically changing the width of line markers, smoothing some feature types and others not, specifying non-default generalization preferences on WMS map layers or the displacement of competing map symbols (e.g. a railway track running parallel to a road). ([28] - [30])

- Context recognition – Though many applications may use a particular choice of high-resolution datasets, there may be differences between applications in regard of which features are important at different scales ([1], [2]). Small lakes may not be important in road maintenance applications and may be hidden at small scales, but the same lakes may be very important in a fire fighter application, where each source of water may be crucial. Advanced constraints could be used to designate such features and the way they may be exaggerated relative to their natural size.

## 5.3 Use Cases

Please see chapter 4.1 for three sample use cases.

## 5.4 Package Dependencies

Figure 5.2, a simple UML package diagram, is intended to support the model outlined above. It shows how generalization constraints relate to other information types of the OGC framework. The information type of generalization constraints relies on or refers to classes from GML, Style (SLD) and Metadata information types. The generalization constraints package contains the default and advanced constraint sub-packages. Potential relationships between default and advanced constraints are not shown and would depend on the definition of a general model (OGC, 2003a) for generalization.

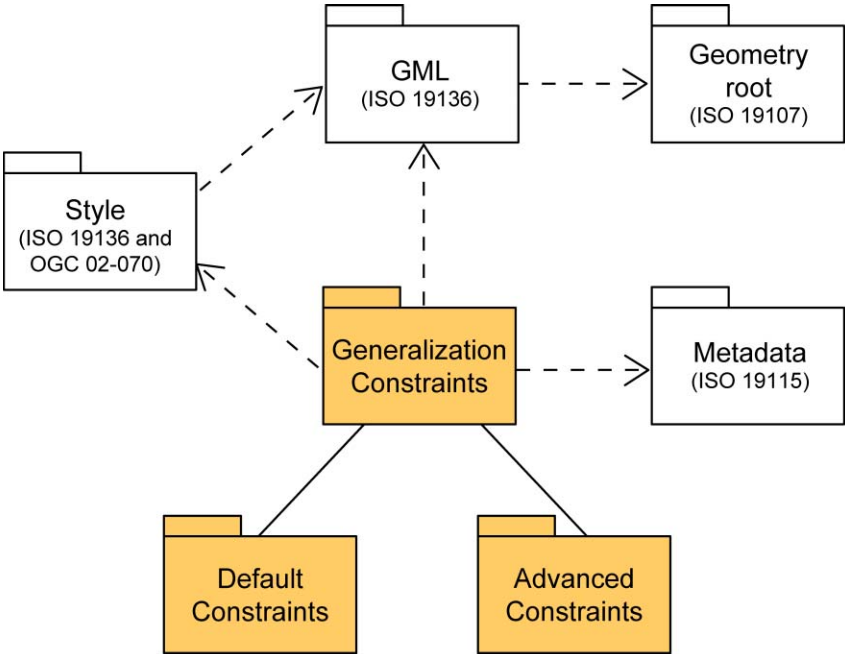


Figure 5.2 Dependencies between generalization constraints and other information types

# 6 Discussion

One of the two main questions of this thesis was “What would be the conceptual requirements for generalization services in interoperating GIS?” The work presented here has identified over thirty essential requirements for potential OpenGIS generalization services, about two-thirds being derived from the OGC Reference Model and one-third from three use cases that were proposed for dynamic generalization. Essential means the requirements should reflect fundamental objectives of generalization in an interoperable services framework. Some of these requirements were regarded to be mandatory in the sense that a conceptual model would necessarily have to account for them. Mandatory requirements include platform and data content format independency, the provision of standard interfaces, and vendor neutrality [3] or the ability to provide a description of service capabilities [13]. Such requirements apply to any new OGC service. Other mandatory requirements relate to the generalization process and to constructs in the OGC’s basic information model as implemented in GML 3.1 (OGC and ISO, 2004a), which is based on ISO 19107 (ISO, 2000). Among those are the obligation to use features as an input [4] (as defined in the GML specifications), the ability to respect groups and hierarchies [7] and interpret the semantic level of detail [6] in the data, to preserve (GML 3.1) topology [9], and to recognize and process multiple scale-dependent feature geometries, if present [5]. Mandatory requirements that were identified in the course of analyzing the use cases are the definition of the target resolution as an essential input parameter [21], the implementation of so-called ‘minimum-size-rules’ [24], as well as the ability to account for scale-dependent and conditional styling [29].

A second class of requirements were regarded as optional in respect of how or if they should be defined in an implementation specification, e.g. those derived from the computational viewpoint of the ORM ([14] - [18]), which reflects the basic concepts of service operations. Other requirements take a user-centered view: the ability to account for the user’s task-at-hand [1], the need for short response times in interactive real-time applications [20], the capability to adapt a service to specific generalization needs [2] and the ability to perform model as well as cartographic generalization [11]. The remaining requirements relate to the transformation of spatial information in the generalization process, i.e. to characteristics of metadata [12], SLD ([28], [30]) and GML ([8], [10], [22], [26], [27]) elements. Since requirements are formalized objectives, a list of essential requirements may be useful as a conceptual basis for the development of generalization services. Or they may be used as a reference to re-consider the design of existing services.

The second main question posed by this thesis was, “Which capabilities should generalization services provide?” To describe such capabilities, a conceptual model for generalization services in the OGC framework was proposed based on the list of essential requirements. The basic goal of generalization in the model is to satisfy generalization constraints. The key concept in the model is the differentiation between two basic generalization constraint types, default and advanced, and two types of associated generalization behaviors (Figure 5.1). The term ‘default’ refers to the capability to perform a default generalization on any GML document, while ‘advanced' refers to the possibility for application designers to extend the default constraints and behaviors. The concept of default and advanced constraints derives from the concept of core and application schemas in GML and should allow achievement of the same objectives: to efficiently delegate responsibilities in a distributed environment, while remaining modular and favoring the reusability of modeling classes and software components. One necessary capability of a generalization service is to indicate to client applications which GML elements and which version of GML it supports (it provides a default behavior). Furthermore, generalization services should be capable of calculating the target resolution from WMS requests, which is regarded as one of the most basic input parameters. Another suggested capability is that a generalization service should be able to receive and process WMS, WFS or WCS requests to facilitate the construction of service chains with those services.

An interesting issue of its own is the proposed capability to provide 'connected' output with links back to the original features [32] based on their feature identifiers. Generalized map objects are related to the original features in one-to-one, one-to-many or many-to-many relationships. Maintaining these relationships in generalized output would make it possible to use GML as a distributed multi-representation or multi-LoD database and to use dynamic generalization services as a key component to access and browse these databases. Such an approach would be more flexible and powerful than the construction of centralized multi-representation databases. XLink (W3C, 2004g) is usually the technology of choice to implement relationships between XML elements. Unfortunately, it is conceptually and technically not trivial to efficiently encode such links, since the number of links in the output is potentially huge, if the map extent is large and the data is high-resolution. No solution is provided in this thesis, but further investigation is warranted, leading over to the question which other issues regarding generalization services might require further investigation.

The research community provides first examples of generalization services processing GML in an OGC services environment. One example is a prototype for the generalization of point data on mobile devices which was developed in the project WebPark (Burghard et al., 2004) and which is based on the open-source Java framework of the Deegree project (lat/lon, 2004). Another example is the ongoing project GiMoDig on real-time mobile GIS (GiMoDig, 2004, Harrie, 2004) which reports successful implementations of a generalization service communicating on OGC interfaces and is technically based on XSLT (W3C, 2004j) and Java Topology Suite JTS (Vivid Solutions, 2004).

However, establishing generalization services as commodity components in interoperating GIS is much more ambitious than developing successful prototype implementations, and will require additional work in several domains. On the specification side, a formal language for generalization constraints must be developed (and preferably specified by OGC or ISO) and default constraints for GML core elements must be identified and formalized (and preferably specified by OGC or ISO). A framework of modeling classes to extend generalization capabilities with advanced constraints must be provided. On the technical side, existing tools such as JTS (which supports OGC 'Simple Features for SQL' standard) must be extended to support the full complexity of GML including complex 3-D geometries, topologies and temporal properties. New XML transformation technologies may be necessary regarding the interpretation and processing of large numbers of features in GML datasets. Furthermore, many research questions on the reliability, quality and limitations of constraint based generalization in a service-oriented environment need to be answered. Apart from a clear conceptual framework, it is likely that combined efforts in all these domains are required to ensure over the long term that distributed geographic data can be reused by many distributed applications at scales and levels of detail of their choice.

# References

ADMIN.CH (2004): The Swiss Cantons Online website. http://www.admin.ch/ch/e/schweiz/kantone/index.html (04-10-03)

AI, T. and VAN OOSTEROM, P. (2001): A map generalization model based on algebra mapping transformation. In Walid G. Aref, editor, Proceedings of the Ninth ACM International Symposium on Advances in Geographic Information Systems, 21-27.

AGENT (1999): Selection of Basic Algorithms, Report for the AGENT project. http://agent.ign.fr/deliverable/DD2.pdf (04-12-01).

ALLEN, T. and STARR, T. (1982): Hierarchy. Perspectives for ecological complexity. The University of Chicago Press. Chicago. 289p.

ANSELIN, L. (1999): Interactive technique and exploratory spatial data analysis. In Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. (Eds.): Geographical Information Systems, Volume1: Principles and Technical Issues, Second Edition. John Wiley & Sons, New York. 253-266.

AXPAND: http://www.axes-systems.com/ (04-12-01)

BEARD, M.K. (1991): Constraints on Rule Formation. In: BUTTENFIELD, B.P and MCMASTER, R.B. (Eds.): Map generalization: Making rules for knowlegde representation. Longman Scientific & Technical, London.

BERTIN, J. (1973): Sémiologie graphique. Gauthier-Villars, Paris. 421p.

BERTOLOTO, M. and EGENHOFER, M. (2001): Progressive Transmission of Vector Map Data over the World Wide Web. GeoInformatica **5**(4): 345-373.

Bian, L. (1997): Multiscale Nature of Spatial Data in Scaling Up Environmental Models. In: QUATTROCHI D.A. and GOODCHILD, M.F. (Eds.): Scale in Remote Sensing and GIS. Lewis Publishers, Boca Raton, 13-26.

BOBZIEN, M. and MORGENSTERN, D. (2003): Abstracting and Formalizing Model Generalization. Fifth ICA Workshop on Progress in Automated Map Generalization, Paris.

BOS, B. (2004): Interoperability (An essay on W3C's design principles). http://www.w3.org/People/Bos/DesignGuide/interoperability.html (04-12-01).

BRASSEL, K.E. and WEIBEL, R. (1988): A Review and Framework of Automated Map Generalization. International Journal of Geographical Information Systems, **2**(3): 229-244.

BRÜHLMEIER, T. (2000): Interaktive Karten – adaptives Zoomen mit Scalable Vector Graphics. Diplomarbeit, Geographisches Institut, Universität Zürich.

BURGHARDT, D. and MEIER, S. (1997): Cartographic Displacement Using the Snakes Concept. In: FÖRSTNER, W. and PLÜMER, L. (1997): Semantic Modeling for the Acquisition of Topographic Information from Images and Maps. Birkhäuser, Basel, 69-71.

BURGHARDT, D., EDWARDES, A. and MANNES, J. (2004): An Architecture for Automatic Generalization of Mobile Maps. 2nd Symposium on Location Based Services and TeleCartography, Vienna, January, 2004.

BUTTENFIELD, B. (1993): Multiple Representations. NCGIA Research Initiative 3, Closing Report. http://www.ncgia.ucsb.edu/Publications/Closing_Reports/CR-3.pdf (04-11-26)

CECCONI, A. (2003): Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping. Ph.D. thesis, Department of Geography, University of Zurich.

COCKBURN, A. (2001): Writing Effective Use Cases. Addison-Wesley, Boston. 258p.

COVA, T.J. and GOODCHILD, M.F. (2002): Extending geographical representation to include fields of spatial objects. International Journal of Geographical Information Science, **16**: 509-532.

CUTHBERT, A. (1998): User Interaction with Geospatial Data. OpenGIS® Project Document 98-060.

DEVOGELE, T., PARENT, C. and SPACCAPIETRA, S. (1998): On Spatial Database Integration. International Journal of Geographic Information Systems, **12**(4): 335-352.

DI, L. (2004a): Distributed Geospatial Information Services-Architectures, Standards and Research Issues. XXth ISPRS Congress, Commission 2. Istanbul, Turkey. 12-23 July 2004. http://www.isprs.org/istanbul2004/comm2/papers/121.pdf (04-12-01)

DOIHARA, T., WANG, P. and LU, W. (2002): An adaptive lattice model and its Application to Map Simplification. Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002. www.isprs.org/commission4/proceedings/pdfpapers/301.pdf (04-09-10)

DOUGLAS, D. H. and PEUKER, 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.

DUCHENE, C. (2003): Coordination multi-agents pour la généralisation automatique. Bulletin d'information de l'IGN. **74**(2003/3): 87-95.

EDWARDES, A. BURGHARDT, D. and WEIBEL, R. (2004): Portrayal and Generalization of Point Maps for Mobile Information Services, Congress Proceedings.

EGENHOFER, M., CLEMENTINI, E. and DI FELICE, P. (1994): Evaluating Inconsistencies Among Multiple Representations. Sixth international Symposium on Spatial Data Handling, Scotland, 901-920.

ESRI (1998): ESRI Shapefile Technical Description. An ESRI White Paper – July 1998. Available over the internet at http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf (04-12-01)

FORBERG, A. (2004): Generalization of 3D Building Data on a Scale-Space Approach. In Proceedings of the XXth ISPRS congress in Istanbul, July 2004, Commission IV papers, Vol. XXXV, part B4. http://www.isprs.org/istanbul2004/comm4/papers/341.pdf (04-10-10)

FOSTER I., C. KESSELMAN, J.M. NICK and S. TUECKE (2002): The Physiology of the Grid: An open Grid services architecture for distributed systems integration. Open Grid Service Infrastructure WG, Global Grid Forum. http://www.globus.org/research/papers/ogsa.pdf (04-12-01)

FOSTER I., C. KESSELMAN and S. TUECKE (2001): The Anatomy of the Grid – Enabling Scalable Virtual Organizations. Intl. J. of High Performance Computing Applications, **15**(3): 200-222.

FOSTER, I. and C. KESSELMAN, editors (1999): The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers.

FOSTER I. and C. KESSELMAN 1998. The Globus Project: A Status Report. Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop, pp. 4-18. ftp://ftp.globus.org/pub/globus/papers/globus-hcw98.pdf (04-12-01)

FRANK, A. (2003): Ontology for spatio-temporal databases. In: KOUBARAKIS, M.E.A. (Ed.), Spatiotemporal Databases: The Chorochronos Approach, Lecture Notes in Computer Science, Berlin, Springer-Verlag, 9-78.

FRANK, A. (2001): Tiers of ontology and consistency constraints in geographical information systems. International Journal of Geographical Information Science, **15**(7): 667-678.

FRANK, A. and TIMPF, S. (1994): Multiple Representations for cartographic objects in a multi-scale tree - an intelligent graphical zoom. Computers and Graphics, **18** (6): 823-829.

FROHN, R. (1998): Remote Sensing for Landscape Ecology. New metric indicators for monitoring, modelling, and assessment of ecosystems. – Boca Raton. 98p.

FURNAS, G., and BEDERSON, B. (1995): Space–scale diagrams: Understanding multiscale interfaces. In: Proceedings of ACM Conf Human Factors in Computing Systems (CHI 95), May 1995, Denver, Colorado. 234-241.

GALANDA, M. (2003): Automated Polygon Generalization in a Multi Agent System. Ph.D. thesis, Department of Geography, University of Zurich.

GALANDA, M. and WEIBEL, R. (2002): An Agent-Based Framework for Polygonal Subdivision Generalisation. Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002. http://www.isprs.org/commission4/proceedings/pdfpapers/041.pdf (04-11-26)

GEODA (2004): GeoDa, Geodata Analysis Software homepage. http://www.csiss.org/clearinghouse/GeoDa/ (04-09-27)

GGF (2004): The WS-Resource Framework. http://www.globus.org/wsrf/ (04-12-01).

GIMODIG (2004): Real-time generalisation and multiple representation in the GiMoDig mobile service. Project report. http://gimodig.fgi.fi/pub_deliverables/GiMoDigD711_721_731_generalisation.pdf (04-11-17)

GLOBUS (2004): Globus homepage, http://www.globus.org/ (04-12-01).

GOLD, C.M. and THIBAULT, D. (2001): Map generalization by skeleton retraction. Proceedings of the 20th International Cartographic Conference (ICC 2001), Int. Cartographic Association, 2072-2081. http://www.voronoi.com/pdfs/2000-2003/Map_Generalization_by_Skeleton_Retraction.pdf (04-12-01)

GOODCHILD, M.F. (2003): The Nature and Value of Geographic Information Science. In: DUCKHAM, M., GOODCHILD, M.F. and WORBOYS, M.F. (Eds.): Foundations of Geographic Information Science. Taylor & Francis, London, 19-32.

GOODCHILD, M.F. (2001): Models of Scale and Scales of Modelling. In: TATE, N.J. and ATKINSON P. M. (Eds.): Modelling scale in geographic information science. Wiley, Chichester, 3-10.

GOODCHILD, M.F. and MARK, D.M. (1987): The fractal nature of geographical phenomena. Annals of the Association of American Geographers, **77**(2): 265-278.

GOODCHILD, M.F. and QUATTROCHI, D.A. (1997): Scale, Multiscaling, Remote Sensing, and GIS. In: QUATTROCHI D.A. and GOODCHILD, M.F. (Eds.): Scale in Remote Sensing and GIS. Lewis Publishers, Boca Raton, 1-11.

GREENWOOD, J. and HART, G. (2003): Sharing Feature Based Geographic Information - A Data Model Perspective. Proceedings of the 7th International Conference on GeoComputation, Southampton, September 2003. http://www.geocomputation.org/2003/Papers/Greenwood_Paper.pdf (04-12-01)

GRUBER, T.R. (1993): A translation approach to portable ontology specifications. International Journal of Knowledge Acquisition for Knowledge-Based Systems **5**(2):199-220.

GRÜNREICH, D. (1992): ATIKS – a topographic information System as a basis for GIS and digital cartography in Germany. In Vinken R. (Ed.): From digital map series in geosciences to geo-information systems. Geologisches Jahrbuch Reihe A, Heft 122. hannover, Federal Institute of Geosciences and Recources: 207-216.

GRÜNREICH, D. (1995): Development of computer-assisted generalization on the basis of cartographic model theory. In: MÜLLER, J.C., LAGRANGE, J.P., WEIBEL, R. (Eds.): GIS and Generalization – Methodology and Practice. Taylor & Francis, London. 47-55. [MindMap]

GUARINO, N., (1998): Formal Ontology and Information Systems. In proceedings of 1st International Conference on Formal Ontology in Information Systems, edited by N. Guarino (Trento, Italy: IOS Press), 3-15.

G-XML (2004): G-XML Project website. http://gisclh.dpc.or.jp/gxml/contents-e/ (04-12-01)

HÄBERLING, C. (2003): Topographische 3D-Karten – Thesen für kartographische Gestaltungsgrundsätze. PhD thesis no. 15379, Institute of Cartography, ETH Zürich. 155p.

HAINING R.P., WISE S.M. and MA J. (1998): Exploratory Spatial Data Analysis. Journal of the Royal Statistical Society (Series D): The Statistican, **47**(3), 457-469.

HARRIE, L.E. (2004): Using Simultaneous Graphic Generalisation in a System for Real-Time Maps. ICA Workshop on 'Generalisation and Multiple representation' 20-21 August 2004, Leicester.

HARRIE, L. AND SARJAKOSKI, T. (2002): Simultaneous Graphic Generalization. The Cartographic Generalizaion of Vector Data Sets. GeoInformatica, **6**(3): 233-261.

HAKIMPOUR, F. (2003): Using Ontologies to Resolve Semantic Heterogeneity for Intgrating Spatial Database Schemata. Ph.D. thesis, Department of Geography, University of Zurich.

HØJHOLT, P. (2002): Solving space conflicts in map generalization: Using a finite element method. Cartography and GIS, **27**: 65-73.

IEEE (1990): Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York.

INTERLIS (2004): Interlis – the GeoLanguage, homepage. http://www.interlis.ch/ (04-12-01)

ISO (2004): International Standardisation Organization, homepage. http://www.iso.org/ (04-12-01)

ISO (2000): ISO Technical Committee 211 Geographic Information/Geomatics. ISO DIS 19107, Geographic Information – Spatial Schema.

ISO (1996): International Standardisation Organization, Reference Model of Open Distributed Processing RM-ODP [ISO/IEC 10746].

ISO/TC211 (2004): Technical Committee 211 of the International Standardisation Organization, homepage. http://www.isotc211.org/ (04-12-01)

ITTEN, J. (1974): The Art of Color. John Wiley & Sons. 116p.

JONES, C.B., HARITH, A. and TUDHOPE, D. (2003): Geographical Terminology Servers – Closing the Semantic Divide. In: DUCKHAM, M., GOODCHILD, M.F. and WORBOYS, M.F. (Eds.): Foundations of Geographic Information Science. Taylor & Francis, London, 205-222.

JONES C.B., KIDNER, D.B., LUO, L.Q., BUNDY, G.LL. and WARE, J.M. (1996): Database Design for a Multiscale Spatial Information System. International Journal of Geographical Information Systems, **10**(8), 901-920.

KELLY, R.E.J., DRAKE, N.A. and BARR, S.L. (2004): Spatial Modelling of the Terrestrial Environment. John Wiley& Sons, Chichester.

KOTTMAN, C.A. (1999): The Open GIS Consortium and progress toward interoperability in GIS. In: GOODCHILD, M.F., EGENHOFER, M., FEGEAS, R. and KOTTMAN, C. (Eds.): Interoperating Geographic Information Systems. Kluwer, Norwell, USA. 427-442.

KREITER, N. (2002): Multirepräsentationsdatenbank als Basis von topographischen Landeskarten. Diplomarbeit ETHZ, Zürich.

KRESSE, W. (2004): Standardization of Geographic Information. In Proceedings of the XXth ISPRS congress in Istanbul, July 2004, Commission II papers, Vol. XXXV, part B2. Available at http://www.isprs.org/istanbul2004/comm2/papers/132.pdf (04-10-10)

KRESSE, W. and FADAIE, K. (2004): ISO Standards for Geographic Information. Springer, Berlin. 216p.

VAN KREVELD, M. and PESCHIER, J. (1998): On the Automated Generalization of Road Network Maps. Proceedings of the 3rd International Conference on GeoComputation, University of Bristol, United Kingdom, September 1998.

KUHN, W. (2001): Ontologies in support of activities in geographical space. International Journal of Geographical Information Science, **15**(7): 613-631.

KUHN, W. (2003): Semantic reference systems. International Journal of Geographical Information Science, **17**(5): 405-409.

LAKE, R., BURGGRAF, D.S.,TRNINIĆ, M. and RAE, L. (2004). GML – Geography Mark-up Language. Foundation for the Geo-Web. John Wiley & Sons, Chichester. 323p.

LAM, N., and D. A. QUATTROCHI (1992). On the issues of scale, resolution, and fractal analysis in the mapping sciences. The Professional Geographer, **44**:88-98.

LAL, J. and MENG, L. (2004): 3D building recognition using artificial neural network. ICA Workshop on Generalization and Multiple representation. August 2004, Leicester.

LAT/LON (2004): Company homepage. http://www.latlon.de/ (04-11-26)

LI, Z. and SUI, H.G. (2000). An Integrated Technique for Automated Generalisation of Contour Maps. The Cartographic Journal, 37(1): 29-37.

LIU, Y., MOLENAAR, M., AI, T. and LIU, Y (2003): Categorical Database Generalization in GIS aided by Data Model. Proceedings of the 21st International Cartographic Conference (ICC) Durban, South Africa, August 2003.

MANDELBROT, B.B. (1977): Fractals: Form, Chance and Dimension. Freeman, San Francisco. 365p.

MARK, D.M. (1989): Multiple Views of Multiple Representations. In: BUTTENFIELD, B.P. and DELOTTO, J.: Multiple Representations: Report on the Specialist Meeting. National Center for Geographic Information and Analysis, Santa Barbara, CA. Report 89-3.

MAP24 (2004): http://www.mapsolute.com/, (04-12-01)

MC HARG, I.L. (1969): Design with Nature. Natural History Press, New York. 198p.

MCKEOWN ,D. , MCMAHILL, J. and CALDWELL, D. (1999): The use of spatial context in linear feature simplification. GeoComputation. http://www.geovista.psu.edu/sites/geocomp99/gc99/020/gc_020.htm (04-10-05)

MCMASTER, R.B., and SHEA, K.S. (1992): Generalization in Digital Cartography. (Resource Publications in Geography). Washington, D.C.: Association of American Geographers.

MCMASTER, R.B. (1991): Conceptual frameworks for geographical knowlegde. In: BUTTENFIELD, B.P and MCMASTER, R.B. (Eds.): Map generalization: Making rules for knowlegde representation. Longman Scientific & Technical, London.

MORRISON, J.L. (1974): A Theoretical Framework for Cartographic Generalization with Emphasis on the Process of Symbolization. International Yearbook of Cartography, **14**: 115-127.

MOSTAFAVI, M.A., GOLD, C. and DAKOWICZ, M. (2003): Delete and insert operations in Voronoi/Delaunay methods and applications. Computers & Geosciences, **29**, 523-530.

MÜLLER, J.C., WEIBEL, R., LAGRANGE, J.P. and SALGÉ, F. (1995): Generalization: state of the art and issues. In: MÜLLER, J.C., LAGRANGE, J.P., WEIBEL, R. (Eds.): GIS and Generalization – Methodology and Practice. Taylor & Francis, London. 3-16.

NASA (2004): World Wind (free terrain viewer) homepage. http://learn.arc.nasa.gov/worldwind/ (04-12-01)

NICKERSON, B.G. (1991): Knowledge engineering for generalization. In: BUTTENFIELD, B.P and MCMASTER, R.B. (Eds.): Map generalization: Making rules for knowlegde representation. Longman Scientific & Technical, London.

NICKERSON, B.G. and FREEMAN, H.R. (1986): Development of a Rule-based System for Automatic Map Generalization. Proceedings, Second International Symposium on Spatial Data Handling, Seattle, 537-556.

OGC (2004): Open Geospatial Consortium homepage. http://www.opengeospatial.org/ (04-12-01).

OGC (2003a): OpenGIS™ Reference Model, Version 0.1.2. Abstract Specification OGC 03-040.

OGC (2003b): Web Coverage Service (WCS) Implementation Specification, Version: 1.0.0. OGC 03-065r6.

OGC (2002a): Web Feature Service (WFS) Implementation Specification, Version: 1.0.0. OGC 02-058.

OGC (2002b): Web Map Service (WMS) Implementation Specification, Version: 1.1.1. OGC 02-068r2.

OGC (2002c): Styled Layer Descriptor (SLD) Implementation Specification, Version: 1.0.0. OGC 02-070.

OGC (2002d): OpenGIS™ Geography Markup Language (GML) Implementation Specification, Version: 2.1.2. OGC 02-069. GML 2.1.2 has been deprecated.

OGC (2001): Filter Encoding (FE) Implementation Specification, Version: 1.0.0. OGC 02-059.

OGC (2000): OpenGIS™ Geography Markup Language (GML) Implementation Specification, Version: 1.0. OGC 00-029. GML 1.0 has been deprecated.

OGC (1999a): The OpenGIS™ Abstract Specification, Topic 0: Abstract Specification Overview, Version 4. Abstract Specification OGC 99-100r1.

OGC (1999b): The OpenGIS™ Abstract Specification, Topic 4: Feature Collections, Version 4. Abstract Specification OGC 99-110.

OGC (1999c): The OpenGIS™ Abstract Specification, Topic 5: Features, Version 4. Abstract Specification OGC 99-105r2.

OGC (1999d): OpenGIS™ Simple Features Specification For SQL, Revision 1.1. OGC 99-049.

OGC and ISO (2004a): OpenGIS™ Geography Markup Language (GML) Implementation Specification, Version: 3.1.0, Recommendation Paper OGC 03-105r1. Same as ISO/CD 19136, Geographic Information – Geography Markup Language (GML).

OGC and ISO (2004b): Web Map Service Implementation Specification, Version 1.3. OGC 04-024. Same as ISO/DIS 19128, Geographic Information – Web map server interface.

OGC and ISO (2002): The OpenGIS™ Abstract Specification, Topic 12: OpenGIS Service Architecture, Version 4.3. Abstract Specification OGC 02-112. Same as ISO/DIS 19119 – Geographic Information Services.

OGC and ISO (2001): The OpenGIS™ Abstract Specification, Topic 1: Feature Geometry, Version 5. Abstract Specification OGC 01-101. Same as ISO/TC 211 DIS 19107, Geographic Information – Spatial Schema.

OGC and ISO (2000): The OpenGIS™ Abstract Specification, Topic 11: OpenGIS(tm) Metadata, Version 5. Abstract Specification OGC 01-111. Same as ISO/TC 211 DIS 19115, Geographic Information – Metadata.

OMG (2003a): UML Notation Guide v1.5. http://www.omg.org/docs/formal/03-03-10.pdf (04-07-15)

OMG (2003b): Object Constraint Language 2.0. Object Management Group (OMG) Final Adopted Specification, ptc/03-10-14.

OKABE, A., BOOTS, B., SUGIHARA, K. and NOK CHIU, S. (2000): Spatial Tesselations – Concepts and Applications of Voronoi Diagrams. Second Edition. John Wiley & Sons, Chichester. 671p.

PECKNOLD, S., LOVEJOY, S., SCHERTZER, D. and HOOGE, C. (1997): Multifractals and Resolution Dependence of Remotely Sensed Data: GSI to GIS. In: QUATTROCHI D.A. and GOODCHILD, M.F. (Eds.): Scale in Remote Sensing and GIS. Lewis Publishers, Boca Raton, 361-394.

RAMER, U. (1972): An iterative procedure for the polygonal approximation of plane Curves. Computer Graphics and Image Processing, **1**: 244-256.

RAVEGEO (2004): http://www.idevio.com/ravegeoinfo.htm (04-12-01)

RENSCHLER, C. (2003): Designing geo-spatial interfaces to scale process models: the GeoWEPP approach. Hydrological Processes, 17, 1005-1007.

RUAS, A. (2002a): Les problématiques de l’automatisation de la généralisation. In: Ruas, A. (Ed.): Généralisation et représentation multiple. Hermès Science Publications, Paris, 74-89.

RUAS, A. (2002b): Echelle et niveau de détail. In RUAS, A. (Editor): Généralisation et représentation multiple. Hermès Science Publications, Paris, 25-44.

RUAS, A. (1998a): OO-constraint modeling to automate urban generalisation process. 8th International Symposium on Spatial Data Handling (SDH'98), Vancouver, p225-235.

RUAS, A. (1998b): Method for building displacement in automated map generalisation. International Journal of Geographic Information Science, **12**(8): 789-803.

SERVICE-ARCHITECTURE.COM (2004): Web Services and Service-Oriented Architectures, homepage. http:// www.service-architecture.com/ (04-12-01)

SESTER, M. (2001): Massstabsabhängige Darstellungen in digitalen räumlichen Datenbeständen. Habitilationsschrift, Deutsche Geodätische Kommission, München, 114p.

SONDHEIM, M., GARDELS, K. and BUEHLER, K. (1999): GIS Interoperability. In Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. (Eds.): Geographical Information Systems, Volume1: Principles and Technical Issues, Second Edition. John Wiley & Sons, New York. 125-155.

STOMS, D. (1994): Effects of habitat map generalization on biodiversity assessment. Photogr. Eng. Remote Sensing, **58**, 1571-1578.

TIMPF, S. and DEVOGELE, T. (1997): New tools for multiple representations. ICC'97, Stockholm. http://e-collection.ethbib.ethz.ch/show?type=inkonf&nr=152 (04-10-01)

TIMPF, S. (2003): Geographic Activity Models. In: DUCKHAM, M., GOODCHILD, M.F. and WORBOYS, M.F. (Eds.): Foundations of Geographic Information Science. Taylor & Francis, London, 241-254.

TOBLER, W.R. (1970): A computer movie simulation urban growth in the Detroit Region. Economic Geography, **46**: 234-240.

TURNER, M.G., O'NEILL, R.V., GARDNER, R.H. and MILNE, B.T. (1989): Effects of changing spatial scale on the analysis of landscape pattern. Landscape Ecology, **3**, 153-162.

UCGIS (via R. MCMASTER, 1996): University Consortium for Geographic Information Science. Research priorities for geographic information science, Cartography and Geographic Information Systems, **23**(3): 115-127.

UCGIS (1998): University Consortium for Geographic Information Science. Scale, White Paper no. 6. http://www.ucgis.org/priorities/research/research_white/1998%20Papers/scale.html (04-12-01)

UDDI (2004): Universal Discovery Description and Integration homepage. http://uddi.org/ (04-12-01)

UML (2004): Unified Modeling Language homepage. http://www.uml.org/ (04-12-01)

VAN PADDENBURG, A. and WACHOWICZ, M. (2001): The Effect Of Spatial Generalisation On Filtering Noise For Spatio-Temporal Analyses. Proceedings of the 6th International Conference on GeoComputation, Brisbane, Australia. September 2001.

VANGENOT, C., PARENT, C. and SPACCAPIETRA, S. (2002): Modélisation et manipulation de données spatiales avec multireprésentation dans le modèle MADS. In RUAS, A. (Editor): Généralisation et représentation multiple. Hermès Science Publications, Paris, 93-111.

VANGENOT, C. (2001): Multi-représentation dans les bases de données géographiques. Ph.D. thesis nr 2430, École Polytechnique Fédérale, Lausanne. 166p.

VIVID SOLUTIONS (2004): JTS Topology Suite, homepage. http://www.vividsolutions.com/jts/JTSHome.htm (04-11-18).

W3C (2004a): Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-concepts/ (04-12-01)

W3C (2004b): OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-features/ (04-07-10)

W3C (2004c): OWL Web Ontology Language: Use Cases and Requirements. W3C Recommendation 10 February 2004. http://www.w3.org/TR/webont-req/ (04-12-01)

W3C (2004d): XML Binary Characterization Properties: W3C Working Draft 05 October 2004. http://www.w3.org/TR/xbc-properties/ (04-11-26)

W3C (2004e): Scalable Vector Graphics (SVG) homepage. http://www.w3.org/Graphics/SVG/ (04-09-25)

W3C (2004f): Extensible Markup Language homepage. http://www.w3.org/XML/ (04-09-01)

W3C (2004g): W3C XML Pointer, XML Base and XML Linking homepage. http://www.w3.org/XML/Linking (04-09-01)

W3C (2004h): Common Gateway Interface homepage. http://www.w3.org/CGI/ (04-09-01)

W3C (2004i): XML Schema, homepage. http://www.w3.org/XML/Schema/ (04-09-01)

W3C (2004j): The Extensible Stylesheet Language Family (XSL), homepage. http://www.w3.org/Style/XSL/ (04-10-01)

WALFORD, N. (2002): Geographical Data – Characteristics and Sources. John Wiley & Sons, Chichester.

WANG, P., DOIHARA, T. and LU, W. (2002): Spatial Generalization: An Adaptive Lattice Model based on Spatial Resolution. Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002. http://www.isprs.org/commission4/proceedings/pdfpapers/291.pdf (04-11-26)

WANG, P. and DOIHARA, T. (2004): Generalization of Roads and Buildings. In Proceedings of the XXth ISPRS congress in Istanbul, July 2004, Commission IV papers, Vol. XXXV, part B4. http://www.isprs.org/istanbul2004/comm4/papers/352.pdf (04-10-10)

WARE, J.M. and JONES, C.B. (1998): Conflict Reduction in Map Generalization Using Iterative Improvement. GeoInformatica **2**(4): 383-407.

WARE, J.M., JONES, C.B. and THOMAS, N. (2003): Automated map generalization with multiple operators: a simulated annealing approach. International Journal of Geographical Information Science, **17**(8): 743-769.

WEIBEL, R. (1997): Generalization of Spatial Data: Principles and Selected Algothithms. In: VAN KREVELD M., NIEVERGELT J., ROOS, T. and WIDMAYER P. (Eds.): Algorithmic Foundation of Geographic Information Systems. Springer, Berlin, 98-152.

WEIBEL, R. (1996): Generalization of Spatial Data. In CISM Advanced School on Algorithmic Foundations of Geographical Information Systems. 346-367.

WEIBEL, R. (1995): Three essential building blocks for automated generalization. In: MÜLLER, J.C., LAGRANGE, J.P. and WEIBEL, R. (Eds.): GIS and Generalization – Methodology and Practice. Taylor & Francis, London. 56-68.

WEIBEL, R. and DUTTON G. (1999): Generalising spatial data and dealing with multiple representations. In Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W. (Eds.): Geographical Information Systems, Volume1: Principles and Technical Issues, Second Edition. John Wiley & Sons, New York. 125-155.

WEIBEL R. and DUTTON G. (1998): Constraint-based Automated Map Generalization. In: Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver. 214–224.

WEIBEL R. and JONES, C.B. (1998): Computational perspectives on map generalization. Guest editorial for special issue on Map Generalization, GeoInformatica, **2**(4): 307-314.

WEIBEL R. and HELLER M., (1991): Digital terrain modelling. In: D. Maguire, M. Goodchild, D. Rhind (Eds.): Geographical Information Systems: Principles and Applications. Wiley & Sons, Inc., pp. 269-297.

WIEHLER, G. (2004): Mobility, security and Web Services : technologies and service-oriented architectures for a new era of IT-Solutions. Publicis Corporate Publishing, Erlangen. 219p.

WIENS, J.A. and MILNE, B. (1989): Scaling of ‚landscapes' in landscape ecology, or, landscape ecology from a beetle's perspective. In: Landscape Ecology, **4**(2): 87-96.

WOOLDRIDGE, M. (2002): An introduction to multi-agent systems. Wiley, Chichester. 348p.

XIA, Z. and CLARKE, K.C. (1997): Approaches to Scaling of Geo-Spatial Data. In: QUATTROCHI D.A. and GOODCHILD, M.F. (Eds.): Scale in Remote Sensing and GIS. Lewis Publishers, Boca Raton, 309-360.

ZHOU, X, PRASHER, S. and KITSUREGAWA, M. (2002): Database Support for Spatial Generalisation for WWW and Mobile Applications. Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE'2002), pages 239-246. December 2002, Singapore.

ZHOU, X., ZHANG, Y., LU, S. and CHEN, G. (2000): On Spatial Information Retrieval and Database Generalization. Proceedings of the 2000 International Conference on Digital Library: Research and Practice. November 2000, Kyoto, Japan.