

IKG Institute of Cartography and Geoinformation

EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH Institute of Cartography and Geoinformation

MASTER THESIS

Service-oriented Architecture for Thematic Cartography on the Web

Author: Andreas KRIMBACHER andreas.krimbacher@gmail.com Supervisor: Prof. Dr. Lorenz Hurni

Advisors: Dr. Hans-Rudolf BÄR Dr. Ionut IOSIFESCU ENESCU Florian STRAUB

January 30, 2014

Geomatic Engineering and Planning MSc Department of Civil, Environmental and Geomatic Engineering

I

Preface

This project is motivated by my fascination for the opportunities for map production in the field of Web cartography, arising form the fast development of Web technologies. The large amount of data and services available on the Web, which can be utilized for the creation of thematic maps and the possibilities for decentralized and automated map creation, inspired me to work on this project for the last four months.

I would like to thank the Department of Cartography and Geoinformation at ETH Zürich for giving me the opportunity to deepen my knowledge in the field of Web cartography with this project.

Furthermore, I would like to express my special appreciation and thanks to my advisors, Dr. Hans-Rudolf Bär, Dr. Ionut Iosifescu Enescu, and Florian Straub, who gave me constructive comments and warm encouragement.

Also, I would like to thank Axel and Alice for correcting the language of this thesis.

Zürich, January 27, 2014 Krimbacher Andreas

Abstract

This thesis develops a conceptual model for the production of thematic maps which utilizes the Web infrastructure. The derivation of the model is performed in two steps. In the first, we develop a workflow model, which covers the steps from data import to publishing and is based on the creation process of standard thematic map representations. For the modeling we focus on two-dimensional map representations and processing techniques for vector data. Based on the workflow, the second step derives a conceptual model which is designed as a service-oriented architecture and can be implemented with decentralized Web services. An examination of the model with respect to important concepts of thematic Web cartography and a prototypical application show that service-oriented architectures are suitable for thematic map production. The main benefits are thereby the direct integration in the Web infrastructure and the possibilities for automated map creation. The main limitations with respect to realization are the lack of standards and specifications for a comprehensive description of the map symbology and the map creation process. For the advancement of service-driven map creation on the Web, this work contributes an elementary model, which assists the developments in automated mapping and can increase the networking between cartographers with respect to map production and the development of mapping techniques.

Table of Contents

List of Abbreviations VI					
Lis	List of Figures VII				
Lis	List of Tables VII				
1.	Intro	oduction	1		
	1.1.	Research Area	2		
	1.2.	Statement of Problem	2		
	1.3.	Research Questions and Objectives	3		
	1.4.	Motivation	3		
	1.5.	Structure of the Report	4		
2.	Met	hodology	5		
	2.1.	A Generic Workflow for Thematic Cartography on the Web	5		
	2.2.	A Service-oriented Architecture for Thematic Cartography	6		
	2.3.	Proof-of-Concept and Evaluation	6		
	2.4.	Criticism	6		
3.	Fund	damentals	8		
	3.1.	Thematic Cartography and Geovisualization	8		
		3.1.1. Definition and Cartographic Map Production	8		
		3.1.2. Standard Thematic Map Representations	11		
	3.2.	The Internet and Web Cartography	12		
		3.2.1. The Modern Web and the GeoWeb	12		
		3.2.2. Service-oriented Architectures and Web Services	13		
		3.2.3. Web Cartography with SOA	15		
	3.3.	Concepts for Thematic Web Cartography	17		
		3.3.1. Interactivity	17		
		3.3.2. Collaboration and Neogeography	19		
		3.3.3. Linking of Documents, Data and Services	21		
		3.3.4. Searchability and Findability	22		
4.	Мос	leling	23		
	4.1.	Requirements for the Conceptual Model	23		
	4.2.	Derivation of the Generic Workflow	24		
		4.2.1. Definition of the Scope of Thematic Cartography	24		
		4.2.2. A Generic Workflow for Thematic Cartography on the Web	25		

	4.3.	The Conceptual Model	
		4.3.1. A Service-oriented Architecture for Thematic Cartography on the Web	28
		4.3.2. The Map Description	36
		4.3.3. Different Implementations of the Conceptual Model	38
		4.3.4. OGC Web Standards and Specifications for the Conceptual Model $\ .$.	40
5.	Imp	lementation	42
	5.1.	Application Design	42
		5.1.1. Client-side Implementation	42
		5.1.2. Server-side Implementation	46
	5.2.	Map Creation	46
		5.2.1. Choropleth Maps with Standardized Data	46
		5.2.2. Proportional Symbol Maps and Diagrams	48
		5.2.3. The Cartographic Procedure Service and Dot Maps	49
6.	Eval	luation and Discussion	51
	6.1.	Standard Thematic Map Representations and the Generic Workflow	51
	6.2.	Implementation of the Concepts for Thematic Web Cartography	53
	6.3.	Discussion of Research Findings	55
7.	7. Conclusions and Outlook		58
	7.1.	Conclusions	58
	7.2.	Outlook	59
Bi	bliog	raphy	60
Α.	Арр	pendices	64
	A.1.	Third-party Libraries and Software	64
	A.2.	Server-side API	66
	A.3.	Service Chain Objects and Map Description	67
	A.4.	Symbology Descriptions for Example Maps	73
В.	Dec	laration of Originality	77

List of Abbreviations

AJAX	Asynchronous JavaScript and XML
AMS	Analysis and Manipulation Services
BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
CPS	Cartographic Procedure Service
DCS	Data Conversion Service
DTM	Desktop Mapping System
FGDC	Federal Geographic Data Committee
GML	Geography Markup Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
RDF	Resource Description Framework
REST	Representational State Transfer
SLD/SE	Styled Layer Descriptor/Symbology Encoding
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SR	Symbology Repository
TMS	Thematic Map Service
URI	Uniform Resource Identifier
WADL	Web Application Description Language
WFS	Web Feature Service
WMS	Web Map Service
WSDL	Web Services Description Language
XML	Extensible Markup Language

List of Figures

3.1.	Communication Model
3.2.	Cartography Cube
3.3.	GIS-VIS Approach for Map Production 10
3.4.	Technologies for Service-oriented Architectures
3.5.	Architecture of the CartoService Concept
3.6.	Stages of Action for Map Interaction
3.7.	RDF Triple 21
4.1.	Generic Workflow for Thematic Map Production on the Web
4.2.	Data Conversion Service Interface 29
4.3.	Interfaces of Analysis and Manipulation Services
4.4.	Cartographic Procedure Service Interface
4.5.	Thematic Map Service Interface 33
4.6.	Symbology Repository Interface
4.7.	Service-oriented Architecture for Thematic Map Production on the Web 35
4.8.	Elements of Map Description
4.9.	Database-Centered Implementation of the SOA
4.10.	Map-Description-Centered Implementation of the SOA
5.1.	AngularJS Modules for the Client-side Application
5.2.	Editor in the Client-side Application
5.3.	Sequence Diagram Choropleth Map
5.4.	Example Choropleth Map
5.5.	Sequence Diagram Proportional Symbol Map
5.6.	Example Proportional Symbol Map
5.7.	Sequence Diagram Dot Map
5.8.	Example Dot Map 50

List of Tables

3.1.	Taxonomy of Interaction Types	19
4.1.	Requirements for Conceptual Model	24
$6.1. \\ 6.2.$	Evaluation of Standard Mapping Techniques	$\frac{51}{53}$
A.1. A.2. A.3.	Software	$\begin{array}{c} 64 \\ 64 \\ 65 \end{array}$
A.4.	REST Requests	66

1. Introduction

The evolution of Web mapping and the rise of the GeoWeb provide increasing possibilities to produce and distribute geographic products. Furthermore, 95% of all data contain geographical references and can be visualized with a map (Perkins, 2010). This thesis focuses on thematic maps for the Web which are used to communicate information about a specific theme and geographical reference.

The quality of thematic maps on the Web is based primarily on the representation quality and the capabilities of interaction. Despite the well-known design principles for thematic maps, which have evolved from the research on analog and digital map production, a great number of thematic maps have inadequate cartographic quality (Engemaier and Asche, 2011). Besides, most of the maps do hardly utilize the possibilities of interaction that the digital output devices provide.

For the production of thematic maps for the Web offline and online tools as well as software packages are available. The offline production of thematic maps is often based on different software packages such as ArcGIS, MapPublisher, Adobe Illustrator, and Adobe Photoshop which are used for different purposes. In addition, plug-ins such as the Illustrator Plug-ins for Cartography developed at ETH Zürich are used to extend the functionalities (Hurni and Hutzler, 2008). For online production of thematic maps, closed and centralized systems such as CartoDB¹ or Indiemapper²change, which that often lack sufficient functionality to analyse and visualize data can be used. Another approach involves using the ArcGIS Server³ or addthe QGIS Server⁴ which provide selected functionalities of the desktop version through a server interface.

The emergence and success of technologies under the paradigm of *software as a service* opens up new ways of providing functionality and raises new questions about how to produce thematic maps in the future (Spence et al., 2009).

The thesis takes its motivation from the current lack of efficient production systems that utilize the benefits of service-oriented architectures (SOA) and state-of-the-art Web technologies for the production of thematic maps for the Web.

¹https://cartodb.com/

²http://indiemapper.com

³http://www.esri.com/software/arcgis/arcgisserver

⁴http://live.osgeo.org/de/overview/qgis_mapserver_overview.html

1.1. Research Area

For the production and supply of thematic maps on the Web a long tradition of research on thematic mapping from analog to digital production techniques can be utilized. The techniques for thematic cartography for this thesis are mainly based on the textbooks from Slocum et al. (2008) and Dent et al. (2008).

Displaying thematic maps on a digital medium also is about interaction. Research was mainly done in the area of geovisualization. A broad perspective on interaction theories can be found in Roth (2012), where a basic taxonomy for cartographic interaction is provided by reviewing the major contributions in the field.

The final major research area for thematic cartography on the Web covers the infrastructure implications of production and distribution . In this thesis, the focus is thereby on service-oriented architectures. SOA for the distribution of thematic maps on the Web is mainly based on OGC Web services. Additions to the OGC standards were proposed in Iosifescu-Enescu (2011) and Rautenbach et al. (2013). Former developed an extended WMS to provide additional capabilities for cartographic diagrams and legends. The work of Rautenbach et al. (2013) proposes a framework for orchestrating OGC services to produce thematic maps. A first approach to develop a concept for modelling and visualization of quality maps with a service-oriented approach can be found in Asche and Engemaier (2012).

1.2. Statement of Problem

To provide and create thematic maps for the Internet, we are restricted to closed and centralized systems. The extension of the Internet and the emerging field of technologies under the paradigm of software as a service leads to a re-evaluation of the questions: How do we produce thematic maps? How do we implement production functionalities? How do we distribute thematic maps?

Most studies on thematic cartography for the Web are based on OGC Web services and focus on the distribution of thematic maps. To enhance the quality of thematic maps on the Web, we also need to consider the production process with respect to the Web infrastructure.

The research in this thesis will focus on the production process of thematic maps on the Web with respect to a service-oriented architecture. The production process will be modelled by analyzing and adopting standard work flows and processing techniques for thematic Web cartography.

1.3. Research Questions and Objectives

The structure of the thesis is given be three research questions (Q) and two objectives (O):

- **Q1:** Can the elements and the sequences of thematic map production for the Web be modeled in one workflow?
- **Q2:** Can the thematic map production workflow be modeled with a service-oriented architecture?
- **Q3:** Are the existing standards and technologies sufficient for the implementation of the service-oriented architecture?
- **O1:** To develop a conceptual service-oriented architecture for thematic map production on the Web which considers important concepts for Web cartography.
- **O2:** To deliver a proof-of-concept by implementing selected functionalities of the service-oriented architecture.

The first two questions concern the development of a conceptual model for map production on the Web. Therefore, the thesis develops a generic workflow, that later on is transformed into a service-oriented architecture. The third question looks at standards and technologies to implement this conceptual model, which will be used in the proof-of-concept.

1.4. Motivation

The motivation for this thesis arises from the benefits of a service-oriented production process. The main advantages are in the field of Web cartography, for neogeographers and for rich Internet applications.

For the field of Web cartography a service-oriented production process provides a better integration of different information, enhanced capabilities for collaborative map creation and a better integration of interactivity and searchability. The benefit in data integration is especially the better combination of data from WebGIS with other spatial information on the Web. For collaborative mapping, a SOA allows to share and combine not only the end product, but also intermediate products, production techniques and the cartographic process. Dynamic linking of different layers of thematic maps, as well as the input data, can provide real-time capabilities and enhance collaborative mapping. An additional benefit, which could increase the quality of the created maps, is the combination of creating and presenting maps utilizing only one architecture.

With a browser-based user-interface the functionalities of the services can be made accessible to neogeographers. The benefits are: easier combination of different sources of information, the creation of thematic maps without the need of specialized software, and enhanced capabilities for collaborative map production and sharing cartographic products. The benefits of the decentralized approach for map production, can also be utilized for rich Internet applications that implement functionalities of thematic cartography. Benefits are especially the scalability and flexibility of the functionalities provided as services. With respect to platform-independence the SOA allows the use of various input and output devices with different characteristics such as screen size or the type of interaction.

1.5. Structure of the Report

The introductory chapter 1 provides an overview of the research area, states the research questions and objectives, and describes the motivation for this thesis. Chapter 2 describes the methodology of the approach. Chapter 3 summarizes the fundamentals for the field of thematic Web cartography and the Web infrastructure. In addition, common standards and technologies for thematic Web cartography are introduced. In chapter 4 the development of the conceptual model is presented. In a first step, a generic workflow is proposed and evaluated with standard techniques for thematic cartography. The second step transforms the generic workflow into a service-oriented architecture and discusses different ways to implement the architecture. In addition, the service-oriented model is compared to OGC Web services and other specifications. After the modeling chapter, the implementation of the functionalities to create the example maps is covered in chapter 5. The evaluation and discussion of the work is presented in chapter 6 by verifying the capabilities of the serviceoriented model with respect to important concepts for thematic Web cartography and the Web infrastructure. Furthermore, the research questions are reviewed with respect to the developed workflow and service-oriented architecture as well as the availability of standards and technologies. The thesis concludes with chapter 7, which also gives an outlook to further work and opportunities.

2. Methodology

The aim of this thesis is to develop a conceptual model based on a service-oriented architecture for thematic map creation on the Web. The derivation of the conceptual model is therefore performed in a sequential manner in three major steps. Preceding to the modeling process, the literature review identifies important concepts with respect to Web cartography and the Web infrastructure. Subsequently, the first step of the modeling process, derives the requirements from the discussed concepts and formulates a generic workflow for the production of thematic maps on the Web. In the second step, a service-oriented architecture is derived by specifying services and additional components to perform the steps of the generic workflow. In the last step, selected functionalities for the creation of three example maps are implemented to demonstrate the operability of the conceptual model. In addition, the model is evaluated by discussing it with respect to the requirements.

The different steps to derive the conceptual model and to evaluate the result are outlined blow. The last section will discuss restrictions and limitations of the approach.

2.1. A Generic Workflow for Thematic Cartography on the Web

The modeling of the workflow for thematic Web cartography is divided in three steps: The definition of requirements, the restriction of the scope of thematic cartography, and the modeling process.

The first step, the specification of the requirements, is based on the literature review, which identifies important concepts for thematic Web cartography. For the specification of the requirements, the functionalities for the implementation of the different concepts are identified.

In addition to the requirements, the second step restricts the field of thematic cartography to attain a practicable scope for the limited time frame of this thesis. The restrictions are thereby to two-dimensional map representations based on vector features for the map creation process.

For the modeling of the generic workflow for thematic map production, the steps and sequences to create the map representation have to be identified. The derivation of this elements is based on the production process of standard map representations for two-dimensional thematic cartography, which are presented in the well-known textbooks of Slocum et al. (2008) and Dent et al. (2008).

2.2. A Service-oriented Architecture for Thematic Cartography

Service-oriented architectures (SOA) are an established design pattern on the Web, which enable a decentralized implementation of functionalities. The aim in this thesis is to utilize the benefits of SOA for thematic map production. Therefore, the second step of the modeling process, derives a conceptual model from the generic workflow developed in the preceding step. The conceptual model develops specific Web services for the functionalities of the workflow and includes additional components to perform the map creation process.

In addition to the modeling, the concept of service chaining and the description of the map representation is discussed to demonstrate the capabilities for an automated creation process. Furthermore, we will validate different implementations of the conceptual model to obtain a basis for the proof-of-concept and the evaluation.

2.3. Proof-of-Concept and Evaluation

The derivation of the conceptual model is based on standard map representations of thematic cartography and important concepts for Web cartography. To verify the results of the modeling process we will test the operability of the model in a proof-of-concept and evaluate the model with respect to the requirements.

The proof-of-concept will implement selected functionalities for the creation of three example maps. The aim of the implementation is thereby to test the practicability of the conceptual model and to verify the applicability of the SOA for automated thematic map creation. In addition, the proof-of-concept should investigate the capabilities of current standards and technologies for the implementation of the conceptual model.

In addition to the implementation, the conceptual model is evaluated with an argumentative approach. Doing so, the model is discussed with respect to the support of the functionalities specified in the requirements.

2.4. Criticism

The purpose of this research is to develop and examine a concept for thematic map production on the Web. To interpret the results of this thesis for the field of Web cartography, we have to consider the restrictions and limitations of the approach presented in the previous sections.

The main restrictions in the modeling process are with respect to the scope of thematic cartography and the architecture of the conceptual model. The derivation of the generic workflow is thereby restricted to two-dimensional map representations and to the analysis of the production process of standard map representations. For the specification of the conceptual model the approach is restricted to a SOA. The limitations of the approach are mainly with respect to the verification of the derived model. Hence, the scope of this thesis does not allow to demonstrate the operability of all components in the conceptual model. In addition, an quantitative evaluation is not possible by the reason of the conceptual nature of the derived model. The evaluation is therefore limited to an argumentative approach. Another limitation concerns the derivation of the requirements. These are based on the concepts for Web cartography derived in the literature review and therefore depend on a selective approach.

3. Fundamentals

The following chapter will discuss the fundamentals of thematic cartography with respect to the Web infrastructure to provide a basis for the modeling task. In the first section, a short overview of thematic cartography and map production is given. The following section, show the development of the Web and cartography on the Web with focus on service-oriented architectures (SOA). In addition to SOA, section three discusses important concepts for thematic Web cartography.

3.1. Thematic Cartography and Geovisualization

This section on thematic cartography and geovisualization is divided into two subsections. The first gives and overview of thematic cartography and map production. The second describes standard thematic map representations.

3.1.1. Definition and Cartographic Map Production

The definition in Slocum et al. (2008, P. 1) describes thematic maps, in contrast to generalreference maps, as maps which are "used to display the spatial pattern of a particular theme or attribute". A similar definition is given in Dent et al. (2008, P. 2), which divides maps into general purpose (reference) maps and thematic maps. The later is thereby defined as a map "which show the spatial distribution of a particular geographical phenomena".

For the description of the production process of thematic maps the communication model for geographical information is used. A general model is thereby introduced in Hake et al. (2002, P. 22). The authors identify a primary, a secondary, and a tertiary model. The primary model is created by collecting data from the real world. In a second stage the secondary model, the cartographic representation, is generated from the data of the primary model. The tertiary model is formed in the last stage when the user interprets the map.

Another map communication model (Figure 3.1) is given in Slocum et al. (2008), which identifies five basic steps for thematic map production. In the first step, the map maker has to determine the purpose of the map. In the second and third step, the user collects appropriate data and defines the map parameters. In the fourth step the map is constructed from the collected data. In the last step, the result is evaluated and published according to the map media. If the map has been regarded unsatisfactory in the evaluation step, the map creation process starts again with step three, by redefining the map parameters.





Figure 3.1.: Communication Model (Slocum et al., 2008, P. 11)

Figure 3.2.: Cartography Cube (MacEachren and Taylor, 1994, P. 6)

When it comes to cartographic communication we also have to think about the term visualization, which is tightly connected with digital production and publishing systems as well as GIS systems. The relation between communication and visualization is best described with the *cartography cube* presented in Figure 3.2 (MacEachren and Taylor, 1994). Communication is thereby a public task which presents knowns and is characterized by low interactivity. Visualization, on the opposite site of the cube, is a private task which involves high interactivity and reveals unknowns.

The differentiation between communication and visualization can also be used to show two general approaches of map production. Traditional maps with focus on communication are produced on Desktop Mapping (DTM) systems, which generates map representations from a cartographic model by using cartographic representation methods. For the creation of interactive maps with focus on visualization, map makers often make use of GIS, which provides a variety of analysis functionalities. The map representation is thereby created by a data-to-vector-geometry transformation, which often lack cartographic representation principles. (Asche et al., 2013)

To overcome the limitations of DTM systems and GIS for map production, different concepts proposed to combine the two approaches. Hardy and Kressmann (2005) propose an extension of GIS with cartographic representations. According to the authors, the approach enables high-quality cartography within a database-centered environment by providing a symbolization pipeline.

Another concept, with focus on atlas systems, is proposed in Bär and Sieber (1999). The workflow for the *GIS and Multimedia Cartography* approach consists of a GIS followed by a graphics application and a multimedia authoring system. The graphics application thereby transforms the GIS objects into cartographic objects while preserving GIS attributes. The

publishing of the map, based on the representations from the graphics application, is performed in the authoring system. The maintained GIS attributes enables the provision of analysis functions in the authoring system and therefore highly interactive maps.

A similar approach to the previous concept is developed in Asche et al. (2013). The authors distinguish the approaches of GIS and VIS (visualization systems) for map production. It must be noted that VIS refers to graphic-oriented software and should not be confused with the approach of visualization mentioned before. With VIS the study instead identifies the previous mentioned Desktop Mapping systems. The proposed approach enables the production of map representations out of geodatabases by coupling the GIS and VIS approach as shown in Figure 3.3. Through the coupling, geo data as well as cartographic models are combined in one concept for the creation of quality maps.



Figure 3.3.: GIS-VIS Approach for Map Production (Asche et al., 2013, P. 226)

The section reveals different approaches to thematic cartography. The definite solution comes neither from a map communication nor a visualization approach. The right position in the *cartography cube* has to be determined for each application individually. In addition, the production environment has to be chosen according to the requirements of the map representation. The discussed studies show that the combination of GIS and DTM systems can enhance cartographic quality and interactivity.

3.1.2. Standard Thematic Map Representations

This section gives an overview of standard techniques for two-dimensional thematic mapping. The review is thereby based on the well-known textbooks of Slocum et al. (2008) and Dent et al. (2008) including the following techniques: Choropleth Map, Dasymetric Map, Isarithmic Map, Cartogram, Proportional Symbol Map, Dot Map, and Flow map.

For the *choropleth map*, derived data (ratio or rate with respect to the area) of enumeration units, such as countries or states, is mapped by coloring or shading the corresponding areas. Choropleth representations can be divided into classed and unclassed maps. For the former the data has to be classified according to the characteristics of the data and the use case. In unclassed maps data values are directly mapped to unique symbols and are often used for data exploration tasks.

The *dasymetric map* is similar to the choropleth map, whereby the mapped areas are calculated from the enumeration units and additional information, such as data for ancillary zones or raster data for land use or other attributes. The benefit of dasymetric mapping compared to choropleth mapping is that density variations within enumeration units can be mapped.

An *isarithmic map* displays three-dimensional geographical data. The visualization can be performed with isolines and coloring or shading areas between isolines, with continuous-tone, and with wire-frame representations. To derive the data for the mapping an interpolation of the initial data has to be performed. Thereby, isometric mapping, with data occurring at points, and isoplethic mapping, with data occurring over geographical areas, is distinguished.

Another area based mapping technique are *cartograms*. The mapping of the data is therefore performed through scaling the areas of the enumeration units proportional to the corresponding attribute values. In contiguous cartograms boundaries between enumeration units are preserved in contrast to non-contiguous cartograms, where the mapped areas float in the mapped space. Specialized forms of contiguous cartograms are circle and rectangular cartograms, where the mapped areas are generalized to circles and rectangular figures.

In the *proportional symbol map* point data is mapped with geometric (circles, squares, spheres, cubes ...) or pictorial symbols. The point location can thereby be measured directly (true point data) or can be derived from other representations (conceptual point data), for example by calculating the centroid of an enumeration unit. The size of the symbols is calculated from the corresponding attribute values, whereby different scaling and classification approaches are used. In addition to the location and size, the map maker has to consider the overlap of symbols to provide an efficient map representation.

Dot maps are used to communicate variations in spatial density by placing dots inside enumeration units, whereby the amount of dots depends on the unit value (the count represented by each dot) and the corresponding attribute value of the area. To determine the size and unit value for a dot map a graphical device called nomograph can be used. The placement of the dots is either performed manually or based on algorithms, whereby additional information, such as data for ancillary zones, raster images, or digital elevation models can be utilized.

The *flow map* show the movement between (geographic) locations and can be divided into five sub-types. Distributive flow maps show movements between points, whereby true point and conceptual data can be used. Network flow maps reveal movements in networks such as a road systems. The mapping of the data is thereby commonly performed by varying line width or line color. Another form are radial flow maps, which instead of mapping actual routes only visualizes the direction of the flow with respect to a reference point. Continuous flow maps are used to map phenomenas, such as wind speed, which can be described with a vector field. The last type are telecommunication flow maps, which map Web traffic and other telecommunication data and often have no geographical space.

3.2. The Internet and Web Cartography

The Web and cartography on the Web are under rapid development. To give an overview of the current state, the first subsections of this section cover recent developments of the Web and the GeoWeb. The following two sections describe SOA on the Web and approaches to service-oriented cartography.

3.2.1. The Modern Web and the GeoWeb

The expansion of the Internet connects more and more people around the world. From 2005 to 2011 the number of individuals who are using the Internet increased from 1.024 billion to 2.273 billion. In addition, the access to the Internet is shifting to mobile subscriptions. In the year 2011, active mobile-broadband subscriptions reached an amount of 1.155 billion. (ITU, 2013)

With the rise of the Internet the keywords Web 2.0 and Semantic Web are closely connected. In contrary to the static Web of the early years of th Internet, the Web 2.0 is characterized by community based websites. This pages allow to collaboratively generate information as well as sharing the information. Another concept of the Web 2.0 are Mashups, which combine different information sources to generate new values. On the technological site, AJAX enhance the creation of Web pages for the Web 2.0 by enabling the creation of highly interactive user interfaces. The main goal of the Semantic Web is to extend the data description to create machine readable information. This goal is achieved by complex and comprehensive modeling with standards such as the Web Ontology Language (OWL) or the Resource Description Framework (RDF). The complexity of the modeling process often implies heavy-weight tools and sophisticated reasoning. (Berners-Lee et al., 2001),(Ankolekar et al., 2008)

In contrary to see the Semantic Web and the Web 2.0 as competing visions, there are more and more opinions that the two ideas complement each other. The Semantic Web can benefit from an collaborative approach to solve the complexity of the semantic modeling process. In addition, Web 2.0 applications can benefit from the Semantic Web through benefits for the exchange and reuse of the created information. (Ankolekar et al., 2008) The developments of Web cartography are best described by the change in map distribution. The first static Web mapping site, which provided raster images as GIF format, dates to 1993 and was the Xerox PARC Map Viewer. The first major distributor of Web maps was MapQuest, which started the service in 1996 and provide user-defined street maps. The rapid dissemination of Web maps at this time is best shown by the downloads from MapQuest on a daily basis, which increased from 700,000 in 1997 to 20,000,000 in 2001. (Peterson, 2003)

The next important year for Web mapping is 2005 in which Google Earth and Google Maps are introduced. The increased awareness of geospatial products and data combined with technologies of the Web 2.0 brought a new generation of mapping products and applications. The distribution of maps changed thereby from the provision of simple raster images to more interactive representations utilizing vector data and relying on AJAX for better user experience. (Crampton, 2009),(Haklay et al., 2008)

The rise of geographic information and products on the Web is identified with the term geospatial Web or GeoWeb. The most important factors of the GeoWeb in the recent years are the development of Mapping APIs, which provide geo data and services, and the availability of standards. Especially, the OGC Web Service specifications are important since the release of the Web Map Service (WMS) standard in the year 2000. Additional advantages of the GeoWeb are open source tools and crowd-sourced data, most notably therefore is the OpenStreetMap project. (Haklay et al., 2008),(Crampton, 2009)

Summarizing, Web cartography shifted from static maps to interactive and dynamic maps. From raster images to a wider use of vector graphic formats. The distribution is changing to distributed and service-oriented approaches and serves more and more individual and single purpose maps. Another development is just at the beginning with the expansion of the mobile Web. New applications will therefore have to consider different screen sizes and input methods. In addition, more and more devices with GPS capabilities enhance the development of location based applications. (Peterson, 2008),(Schnabel and Hurni, 2009)

3.2.2. Service-oriented Architectures and Web Services

A service-oriented architecture (SOA) is a paradigm to arrange and utilize different functionalities and resources that may be distributed over different domains. The term *Web service* refers to the realization of the SOA paradigm in the Web infrastructure. The Web service interface which is exposed to the Web is called application programming interface (API). To provide interoperability, SOA and Web services are standardized for the Web by the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Object Management Group (OMG) and The Open Group. (Kohlborn and Rosa, 2012)

The main standards for Web Services, categorized by their abstraction level, are shown in Figure 3.4. In the following, we will discuss the main specifications with respect to messaging and description of services and service orchestration to provide the basics for this thesis.



Figure 3.4.: Technologies for Service-oriented Architectures (Kohlborn and Rosa, 2012, P. 113)

On the messaging and description level two architectural styles can be distinguished, namely WS-* and Representational State Transfer (REST). Services defined with the standards of WS-* can use different messaging protocols, where the most common is SOAP. REST services are identified by URI and use HTTP actions (i.e., HTTP PUT, GET, POST and DELETE) to manipulate the state of the resource. RESTful Web services are simple and lightweight, but lack of methods to describe service contracts and miss other specifications such as WS-Security to handle security issues. To describe Web services, WSDL (Web Services Description Language) can be used with the architectural style of WS-* specifications. For the description of RESTful Web services, WADL (Web Application Description Language) or WSDL 2.0 is used. (Kohlborn and Rosa, 2012)

For the orchestration and choreography of Web services two major standards are available. WS-BPEL (Web Services Business Process Execution Language), or short BPEL, is a XMLbased language for the definition of orchestration models, which represent the order of service operations. BPEL descriptions can be automatically executed by a BPEL engine. The connection between BPEL and Web services is managed by three communication activities (Invoke, Receive and Reply), which can be directly mapped to Web service operations defined in a WSDL document. Similar to BPEL, standardized by OASIS, the Object Management Group (OMG) specified the Business Process Model and Notation (BPMN) language, which is also directed to a non-technical audience by providing a standardized graphical notation. (Kohlborn and Rosa, 2012) Standardization with focus to geospatial data are mainly introduced by the Open Geospatial Consortium (OGC). The OGC is an international industry consortium of 473 companies, government agencies and universities. The most common implementation to provide maps over the Web is given by the specification of the Web Map Service (WMS). Similar to the WMS, the Web Map Tile Service (WMTS) specification defines a standard to distribute maps with an approach based on map tiles. In addition to WMS and WMTS, which commonly provide maps as raster images, the Web Feature Service (WFS) specification is used to provide vector data. Another important specification for geospatial Web Services is WPS (Web Processing Service), which defines a generic interface to implement functionalities for processing geospatial data.

3.2.3. Web Cartography with SOA

In the literature different approaches to Web cartography with service-oriented architectures are presented. In the following, the major contributions of the recent years to service-oriented Web cartography beyond the scope of OGC Web services are described. In the end of this section a short overview of OGC Web service orchestration is given.

In the dissertation of Iosifescu-Enescu (2011) a Web service to automatically generation maps from GIS data is developed. The definition of the symbolization is based on the Styled Layer Descriptor (SLD) and the Symbology Encoding (SE) specifications, which are standardized by the OGC and are used with WMS implementations. To achieve an appropriate symbolization for topographic and thematic maps SLD and SE are extended for cartographic needs.

To provide cartographic maps with a Web service, Iosifescu-Enescu (2011) defines the Map and Diagram Service Interface (MDSI), which can be seen as an enhanced Web Map Service. Additional functionalities to the WMS interface are functionalities to generate diagrams for given tabular data, to request a legend graphic for a specific layer, to get the style description for a specific layer, and to get a layer description containing the schema information of a layer.

A service-oriented approach to create choropleth and proportional symbol maps is proposed in Rautenbach et al. (2013). The input of data is thereby implemented with a Web Feature Service. To statistically analyze the data and prepare a SLD style sheet for the rendering process the WPS interface is used. Rendering and distribution of the map is implemented with a WMS, which renders the data according to the SLD style sheet. The study proposes that a automated thematic map production is possible, but involves the risk of producing maps that are misinterpreted by the user.

Asche and Engemaier (2012) developed the concept CartoService (Figure 3.5), which proposes a service-oriented architecture for the creation of cartographic maps. The model distinguishes the application layer for the user interface, the processing layer for the combination of different services, and the data layer for data storage and management. To identify different internal and external services a service repository is used. For data preparation the concept suggests the following services: data import, analysis, filter, harmonization, and generalization. To visualize the data a symbolization service and a rendering service are defined.



Figure 3.5.: Architecture of the CartoService Concept (Asche and Engemaier, 2012, P. 418)

Additional to the definition of the components of the service-oriented architecture, also the orchestration of different services is important to enable an automated or semi-automated map creation process. Web service orchestration is mostly described with the Business Process Execution Language (BPEL), which utilize SOAP and WSDL. To enable WPS implementations for the use with BPEL, Sancho-Jiménez et al. (2008) proposes a proxy to provide the SOAP and WSDL interface for any WPS implementation. An extension of this concept proposes Fleuren and Müller (2008) by suggesting an intermediary SOAP proxy also for other OGC Web services such as WMS and WFS. In addition the study demonstrates the possibilities for a stateful Web Map Service by using a PostGIS database which is directly connected to the proxy.

Beside orchestration with BPEL, Supavetch and Chunithipaisan (2011) proposes an interface independent orchestration language, which is executed by an execution engine implemented with a WPS. The results of the study show that the approach allows a flexible combination of different service types, but it also emphasizes that more complex components has to be included in the orchestration of services. Examples therefore are the process status, process monitoring, asynchronous orchestration, and notification model.

Another difficulty, with regard to automated service matching, is mentioned in Kiehle et al. (2006), which states that a spontaneous linkage of services is only possible if the execution engine has sufficient semantic information about the Web services.

3.3. Concepts for Thematic Web Cartography

In addition to service-oriented architectures discussed in the previous sections, this section will focus on four important concepts for thematic cartography on the Web. The first is thereby interactivity which is an important aspect of digital map products. The second concept, collaboration, is an element of the Web 2.0 and the GeoWeb as for example the OpenStreetMap project shows. The last two concepts cover linking of information and services as well as searchability and findability. These concepts are inherent to the Web infrastructure since the beginning and are also important for the semantic Web.

3.3.1. Interactivity

For the conception and production of digital cartographic products interaction is very important. Digital technologies not only gives us the possibility to display different cartographic representations for low cost, but also gives us the possibility to provide interactive tasks with the map and the underlying data.

The main benefits which can be achieved with vector graphic formats for interactive mapping are mentioned in Lienert et al. (2012, P. 24) as following: (a) [vector graphics] are scalable without loss of information or graphical artifacts; (b) the symbolization is adjustable on-the-fly (e.g., line width, transparency, fill color); (c) the geometry and symbolization can be animated; (d) map features can be shown and hidden without regenerating and reloading the entire map; (e) attributes can be attached to each individual map feature; (f) map features, such as diagrams, can be generated on-the-fly; and (g) the geometry can be changed, allowing for lossless projection to other coordinate systems.

When it comes to interaction we also have to think about user-centered design, which can be seen in the approaches of high-quality digital map products, such as the Atlas of Switzerland (Sieber et al., 2009) and the Atlas of Canada (Kramers, 2008).

For a comprehensive model of a user-centered design for digital cartographic representations we will use the approach of Roth (2012), which is based on the stages of action model proposed in Norman (2002). The adoption for cartographic interaction of Norman's interaction model is shown in Figure 3.6 and can be divided into an execution and an evaluation phase.

In the execution phase, Roth (2012, P. 381) classifies interaction taxonomies into objectivebased, operator-based, and operand-based approaches, which are named the *three O's* of cartographic interaction. The objective-based approach describes interaction at the stage of forming an intention to interact. This stage emphasizes the task the user may wish to complete. In the operator-based approach, taxonomies are covered that deal with specifying the action for interacting with the computing device. In the last group, the operand-based taxonomies, approaches are covered which specify the implementation at the computing device.

For this thesis the implementation of interaction at the computing device is important. Therefore, we will focus on the operand-based approaches of interaction. The literature review in Roth (2012) reveals two types of interaction taxonomies for operand-based interactions: type-



Figure 3.6.: Stages of Action for Map Interaction (Roth, 2012, P. 379)

centric and state-centric. Approaches that discriminate primitives according to the characteristics of the presented information are type-centric taxonomies. State-centric taxonomies are oriented to the production process of cartographic maps from raw data to the presentation of the map. In this thesis we want to model the production of thematic maps on the Web. Therefore, we will focus on state-centric interaction theories from the field of cartography and GIS, which are presented in Persson et al. (2006) and Crampton (2002).

Persson et al. (2006) develops a comprehensive typology of around 70 interactivity functions, which are organized in 8 different interaction types. The typology is based on a literature review of existing divisions and categorizations of interactions. The first three types of interactivity functions concern the map creation process. Thereby functions are identified with respect to the model of the cartographic communication process. The identified types are the interaction with the cartographic representation (T1), the interaction with the algorithms for the creation of the representation (T2), and the interaction with the primary model (T3). The fourth and fifth interactivity types covers interaction with multiple views. Furthermore, the study distinguishes between the arranging of multiple views (T4) and the dynamic linking of different display types (T5, Brushing). The sixth and seventh type of interactive functions include functions related to the third and fourth dimensions, which are the change of the temporal dimension (T6) and the interaction (T8) such as panning or scrolling.

The state-centric approach in Crampton (2002) distinguishes between contextualizing interaction, interaction with the representation, interaction with the data and interaction with the temporal dimension. A comparison with the previous mentioned interaction types (Table 3.1) show that the taxonomy in Crampton (2002) can be seen as a generalization of the interactivity types proposed in Persson et al. (2006).

Table 3.1.: Taxonomy of Interaction Types	
Interaction with the Data	Interaction with the Data Representation
• Interaction with the Primary Model/ Database Query (T3)	• Interaction with the Representation Model (T1)
• Dynamic Linking with further Display Types (T5, Brushing)	• Interaction with the Algorithms for the Creation of a Representation (T2)
	• Interaction with the (Pseudo-) 3D Vi- sualization (T7)
	• System Interaction (T8)
Interacting with the Temporal Dimension	Contextualizing Interaction
• Interaction with the Temporal Dimension (T6)	• Arranging Many Simultaneous Views (T4)

For the scope of this thesis, the four interaction types proposed in Crampton (2002) are sufficient to consider interactive functions in the production process of thematic maps. With respect to the stages of the action model for user-centered design the work therefore focuses on operand-based interactivity. Nevertheless, it should be noted that all stages of the model should be considered when the findings of this thesis are implemented in a productive mapping application.

3.3.2. Collaboration and Neogeography

The main advantages of the Geospatial Web or GeoWeb are identified in Crampton (2009) as crowd-sourced data, open source tools and services, and participation and syndication. Tightly coupled with the developments of the GeoWeb in the recent years are the change of the Web infrastructure to a collaborative and interactive platform, known as the Web 2.0. With respect to the GeoWeb the recent developments can be summarized with the term Web Mapping 2.0.

The key Web Mapping 2.0 technologies mentioned in a study by Haklay et al. (2008) are the availability of affordable GPS devices for positioning and data collection, the Web 2.0 technology AJAX for interactive applications, and the development of mapping APIs for the provision of mapping functionalities and data. Beside the technologies, the authors discuss different concepts for collaboration, which are inherent to Web Mapping 2.0. For the scope of this thesis the concepts of knowledge collectives and peer production networks are important.

The concept of knowledge collectives describes the sharing and managing of information and common resources. An example for a knowledge collective is the Open Street Map (OSM) project, which provides crow-sourced geodata. In addition, the OSM project is an example for a peer production network, which enables people to work together for solving problems and producing goods. With respect to OSM, this means distributed data collection and community based development of mapping tools. (Haklay et al., 2008)

Another concept of Web Mapping 2.0 are mash-ups, which are used since 2000 by utilizing the OGC standards (Haklay et al., 2008). Map mash-ups are the combination of datasets from different resources and can be easily created by non-expert users on platforms like MapTube¹. Another example for a mash-up is the London Profiler² project, which is mentioned in Hudson-Smith et al. (2009) as a low cost and easy to use solution for the communication of spatial information between stakeholders and the government.

With the rise of the GeoWeb and Web Mapping 2.0 another important development is connected: Neogeography. The essence of neogeography is best described by the definition in Turner (2006, P. 2-3):

Neogeography means "new geography" and consists of a set of techniques and tools that fall outside the realm of traditional GIS, Geographic Information Systems. Where historically a professional cartographer might use ArcGIS, talk of Mercator versus Mollweide projections, and resolve land area disputes, a neogeographer uses a mapping API like Google Maps, talks about GPX versus KML, and geotags his photos to make a map of his summer vacation.

Essentially, Neogeography is about people using and creating their own maps, on their own terms and by combining elements of an existing toolset. Neogeography is about sharing location information with friends and visitors, helping shape context, and conveying understanding through knowledge of place.

The definition shows a rather unconventional approach to cartography and GI science, with respect to professional techniques. In the literature this issue is discussed under the headings "deprofessionalization or reprofessionalization?" (Crampton, 2009) or "cult of the amateur or mass collaboration?" (Haklay et al., 2008). The answers to the questions mostly remains open or ideologically biased. For the scope of this thesis we will use the pragmatic approach of Haklay et al. (2008, P. 2034), who proposes: "It is not either one or the other, and there is clearly a space for both, so a synergistic approach is required."

Web Mapping 2.0 also involves a change in the provision of data and how we evaluate data. Haklay et al. (2008) proclaims a change from a linear, publishing "push" model, with central data collection and distribution, to an inter-networked, participatory model where information

¹http://www.maptube.org/

²http://www.londonprofiler.org/

is collected collaboratively and can be accessed through many channels. With respect to data quality, solutions have to be found to ensure accuracy (including metadata), coverage and resolution as well as privacy and copyright (Hudson-Smith et al., 2009).

The discussed concepts shows collaborative approaches for cartography and GIS on the Web, but also in other areas of cartography an adoption of collaborative concepts is taking place. An example is the open atlas platform concept proposed in Sieber et al. (2011). Thereby, capabilities are provided to extend the core atlas application with third party modules. In addition, the concept enhances collaboration by enabling the contribution of statistical and graphical geodata.

3.3.3. Linking of Documents, Data and Services

The principle of linking documents, the concept of Hyperlinks, is inherent to the Web since the beginning. With the development of the Semantic Web and the adoption of linked data principles, like RDF, new ways for publishing and searching data on the Web are emerged (Bizer et al., 2009). In the following we will shortly discuss the linkage of documents and data in the Web infrastructure. For the linking of services see section 3.2.2 about service-oriented architectures and Web services.

With respect to linkage of information on the Web two concepts can be distinguished: Linking of documents and linking of data. The first concept is the basic structure of the Web. URLs, a subset of URIs, are used to identify different HTML documents, which are transmitted over the HTTP (or HTTPs) protocol. With the concept of Hyperlinks Web pages can reference to other HTML documents and can therefore establish a one way link. The concept of linking data on the Web is an approach of the Semantic Web. Thereby, URIs and HTTP are supplemented by RDF, which links data objects and includes a description of the connection. The link is thereby defined by a *subject, predicate, object* triple, whereby the subject and object are both data resources identified with URIs. The predicate describes the relationship of the subject and the object and is as well represented by a URI. (Bizer et al., 2009)

An example for a RDF triple is given in Figure 3.7, which states that the resource http:// www.w3.org/People/Berners-Lee/card#i is a member of the resource http://dig.csail. mit.edu/data#DIG. The URI http://xmlns.com/foaf/0.1/member defines the relationship of the type *member* in RDF.

> Subject: ://dig.csail.mit.edu/data#DIG Predicate: ://xmlns.com/foaf/0.1/member Object: ://www.w3.org/People/Berners-Lee/card#i

> > Figure 3.7.: RDF Triple (Bizer et al., 2009, P. 4)

3.3.4. Searchability and Findability

Since the beginning of the Web, approaches for information retrieval are important to deal with the large amount of data. The terms searchability and findability are best described with the taxonomy in Broder (2002), who distinguishes informational and navigational queries. Navigational (searchability) queries deal with the purpose of searching information which is assumed to be available on the Web. In contrast, navigational (findability) queries, or *known item* searches, are used to find a concrete piece of information.

A comprehensive discussion of the field of information retrieval on the Web with respect to cartographic applications would be beyond the scope of this thesis. In the following the discussion is limited to two important aspects of mapping applications on the Web with respect to searchability and findability.

The first aspect is the way in which the data is published on the Web. To enable the access to data for other participants, especially search engines such as Google, four concepts can be distinguished (Ceri et al., 2013). The first concept identifies all approaches where information in a database can only be accessed by query forms. The information which is hidden in the databases is thereby known as the *Deep Web*. The access of the data by query forms can not be utilized by search engines. Therefore, it is the least accessible method for data provision. The second concept to provide data is over Web application program interfaces (API). Web APIs provide better access to data compared to query forms, but mostly are not accessible for search engines. The third concept are microfromats, which used to add metadata directly into HTML pages. This additional descriptions can be automatically extracted by search engines and should therefore be preferred to the previous concepts. The last and most preferable concept is linked data, which was already discussed in the previous section. A specific way for publishing RDF triples in Web pages is RDFa, thereby the subject, predicate and object can be directly defined in the HTML Web page.

The discussion show different approaches for publishing data on the Web ordered by the accessibility for search engines. It should be noted that for the development of an application also other aspects such as development costs and data security have to be considered.

Beside the aspect of how the data is published, it has also be considered which data is published to enhance searchability and findability. Thereby, an important aspect is the data about the data: metadata. To describe geographical data the most important standards are the ISO standard 19115 and the FGDC Content Standard for Digital Geospatial Metadata. In addition to metadata standards, the concepts of tagging and ontologies are important. With respect to tagging the approach of social tagging, where metadata is created in a crowd-sourced manner, is more and more utilized with the development of the Web 2.0 (Csaba, 2013). Ontologies can provide a more flexible access to the information as shown in Farazi et al. (2013) with the Semantic Geo-Catalogue project.

In conclusion, the aspects of how we publish data and how we describe the data are important to provide searchability and findability. Publishing information in a machine readable way will enable functionalities for searching and finding information inside the application and improve the accessibility of the application for search engines.

4. Modeling

In this chapter a conceptual model based on a service-oriented architecture for the production of thematic map representations on the Web is derived. Therefore, the first section formulates the requirements for the modeling process. The following two sections present the modeling process in two steps. In the first, a generic workflow is presented, which is based on the production of standard cartographic map representations. In the second step, the tasks of the generic workflow are modeled with a service-oriented architecture by defining specific Web services and additional components for the map creation process.

4.1. Requirements for the Conceptual Model

The requirements for the conceptual model are derived from the aspects of Web cartography and the modern Web infrastructure as stated in the previous chapter. The specifications, presented in Table 4.1 are thereby structured into five groups: One group with requirements from the field of digital map production and fours groups of requirements which are based on the concepts for thematic Web cartography.

The first group contains requirements which arise from the production process. The literature shows, that an efficient concept for cartographic map production considers GIS functionalities and DTM (Desktop Mapping) functionalities. With respect to thematic map production the model should support the standard thematic map representations discussed in section 3.1.2.

The second group considers the implementation and provision of interaction functionalities. The requirements are thereby based on the study of Crampton (2002), which distinguishes between interaction with the data, the data representation, the temporal dimension, and contextualizing interaction.

The third group aggregates specifications which consider collaborative aspects. The discussion on collaboration and neogeography shows that the concepts of peer production and knowledge collectives are important for Web cartography. Therefore, the conceptual model should enable collaborative map production and the sharing of intermediate and end products as well as functionalities.

The fourth group is based on the concept of linking documents, data and services. In the requirements the linking of documents is supplemented by the linking of maps as a special form of a document. With respect to linking data the model should consider approaches of the semantic Web such as RDF triples.

Table 4.1.: Requirements for Conceptual Model	
Map Production	► Support GIS functionalities
_	► Support DTM functionalities
	► Support of standard thematic map representations
Interaction	► Enable interaction with the data
	▶ Enable interaction with the data representation
	▶ Enable interaction with the temporal dimension
	► Enable contextualizing interaction
Collaboration	► Enable collaborative map production
Condocration	 Enable sharing of intermediate and end products
	► Enable sharing of functionalities
Linking	► Enable the linking of documents and maps
	\blacktriangleright Enable the linking of data
	► Enable the linking of services
Searchability/Findability	► Enable different publishing concepts
	► Enable the generation of metadata
	0

In the last group requirements which enhance searchability and findability are regarded. To optimize maps for search engines, the conceptual model should allow for different publishing concepts. In addition, the generation of metadata has to be considered to support searching and finding of information.

4.2. Derivation of the Generic Workflow

After the direction of the modeling process was set in the previous section with the requirements, a generic workflow for thematic cartography will be developed in this section. A workflow for the whole realm of thematic cartography would be an endless task. Therefore, in the first section the field of thematic cartography is restricted to a scope which is applicable for Web cartography. In the second section, the generic workflow is presented, which is based on the production process of standard thematic map representations.

4.2.1. Definition of the Scope of Thematic Cartography

To limit the scope for the modeling process in this thesis, the realm of thematic cartography is restricted with respect to the map representation and the functionalities for data processing. The modeling process will consider the production of two-dimensional map representations. The mapping of the data is thereby restricted to coordinate reference systems based on cartographic projections, which allows to maintain the coordinate reference of all input data. The second restriction concerns the processing of data. The processing of raster data is in comparison to point, line and area data difficult to automate and will therefore not be considered in the workflow. It should be noted, that this restriction does not exclude the usage of raster data. Raster images can be implemented in a map as a separate layer, such as a base map. In addition, the rendering of the vector data to publish the map as a raster image is possible.

4.2.2. A Generic Workflow for Thematic Cartography on the Web

In this section a generic workflow for thematic cartography is presented. The modeling is based on the production of standard thematic map representations introduced in section 3.1.2 and focuses on the essential steps to produce thematic maps on the Web.

The purpose of the workflow in this section is to provide a basis for the implementation of the service-oriented architecture. Therefore, the workflow is restricted to tasks which involve software functionalities. To clarify the scope of the workflow, we can compare it to the communication model in Slocum et al. (2008), which was introduced in section 3.1.1. The approach in this thesis considers the constructive tasks of selecting data, constructing the map, and publishing the map. The conceptual tasks of defining the purpose of the map, defining the parameters of the map, and evaluating the map are independent of the software and therefore not considered in the workflow.

The tasks of the generic workflow model, which is shown in Figure 4.1, can be generalized into four stages, which will be discussed in the following.

Data Preparation

The data preparation includes the selection, the preparation and the integration of the data for the map. For the data selection we assume that the map creator selects a subset of data from a data source on the Internet or a local storage. The workflow thereby distinguishes between spatial and attributive data. After the selection step, the data is analyzed and manipulated to get the data basis for the further process. This step includes tasks such as verifying the data quality, checking the topology of spatial data sets or combining different data sets. In the data integration task, the attributive data is combined with the spatial data to get a uniform vector data set for the further mapping process.

Data Generation

The data generation is used to derive specialized mapping data from the data basis which was prepared in the preceding step. Examples for this stage of the workflow are the creation of point data for a dot map, the calculation of areas for a cartogram, or the interpolation of data for an isarithmic map.

Mapping

The mapping includes an analysis and manipulation step and a symbolization step. The mapping is thereby performed in an iterative manner, called the *mapping loop*. In the first step the vector data set is adjusted for symbolization. This for example implies the classification

of the data set, the calculation of the size for proportional symbols, or the adjustment of point features for symbol placement. In the second step the map representation is created by combining the vector data with point, line and area symbolizers. In addition to the rendering of the map symbology, the placement of labels has to be performed in the *mapping loop*.

Presentation

The presentation stage contains the map composition and the map publishing. The former includes the combination of different map layers, the creation of a legend for the map data and the placement of additional map elements such as map title, scale bar, and imprint. The last step of the workflow considers the publishing of the map on the Web.



Figure 4.1.: Generic Workflow for Thematic Map Production on the Web

In addition to the iterative procedure in the mapping stage, the workflow is repeated at the data generation step or from the beginning, if the result from the map generation loop is not sufficient.

The presented workflow follows a strict layer-based approach for map creation, whereby one layer is created by performing the tasks from selecting the data to symbolizing the data. The workflow implies that one layer represents a two-dimensional data table corresponding to a specific standard map representation and to one scale. Multivariate maps are thereby constructed with separate layers for each map representation. In addition, multi-dimensional data sets, for example, with values for different time periods or different scales, are mapped by generating multiple layers.

Beside the representation of the data by point, line and area symbols, labels can be added with two approaches. In the first approach the anchor point of a label is defined by the mapped point, line, and area data. The second approach is a separate label layer which defines the anchor point of a label combined with the label text.

The combination of different layers to construct the final map representation is performed in the map composition step. Thereby, the map creator has to select the appropriate layers for the map and has to define the layer properties, such as transparency or masking of labels. It has to be noted, that the combination of layers has to be already considered in the mapping stage to provide a well designed map representation. For example, other layers have to be considered in the placement of labels to provide good readability of the map.

To produce effective thematic maps the knowledge of visual variables, knowledge about the symbology, is important. In the workflow this is implied in the mapping stage and the map composition. For the mapping, the map creator has to derive parameters for the symbolization. This for example implies the calculation of classes, which has to consider the symbology as well as the characteristics of the used data. In the map composition, knowledge about the symbology is needed to generate the legend corresponding to the theme of the map.

In addition to the steps for map creation, the workflow in Figure 4.1 is extended with the interaction tasks mentioned in Crampton (2002), which where discussed in section 3.3.1.

For the interaction with the data, functionalities have to be provided which allow to query and analyze the underlying data of the map representation. With respect to the workflow, these functionalities can be provided in the steps ranging from the data selection to data analysis and manipulation in the mapping stage.

Interaction with the data representation involves the data and parameters for the symbolization, the symbology, and the map representation. The data for the mapping is derived in the data generation step. An interactive task can be the changing of the dot placement algorithm for a dot map. Parameters for the symbolization are derived in the mapping stage with the analysis and manipulation step. An example for interaction is the reclassification of data for a choropleth map. Interaction with the symbolization, such as changing the color scheme, involves tasks affecting the selected point, line, or area symbolizer. With respect to the map representation, interactive functions are affecting the map view, for instance panning or zooming, and have to be provided in the publishing step.
Contextualizing interactions involve the combination of different layers and the arranging of multiple views, which can be performed with the functionalities included in the map composition.

The last interactive task summarizes interactions with the temporal dimension, which considers dynamic mapping techniques such as fly-bys and fly-throughs or navigation. The implementation of these tasks is performed in the publishing step.

4.3. The Conceptual Model

In this section we will present and discuss a conceptual model for thematic cartography on the Web. The conceptual model is thereby modeled as a service-oriented architecture (SOA) and is based on the generic workflow from the previous section.

The content of this section is structured into four subsections. In the first the components of the SOA are introduced and an implementation of the generic workflow is presented to demonstrate the fundamental functioning of the model. The second subsection discusses the description of the map creation process with respect to service orchestration. In the third, different implementations of the conceptual model are presented to demonstrate the possibilities of the approach. The last subsection compares the components of the conceptual model with OGC Web services and additional specifications to reveal similarities to the current GeoWeb infrastructure.

4.3.1. A Service-oriented Architecture for Thematic Cartography on the Web

The description of the SOA in this section is divided into two parts. First, the components of the SOA are presented and identified with the steps in the generic workflow. In the second step, the orchestration of the specified components to perform the steps of the generic workflow is demonstrated.

Before providing details on the implementation of the different components, an overview of the derived Web services is given in the following.

The first step in the generic workflow, which involves functionalities for selecting and importing the data, is realized in the *Data Conversion Service*. To provide basic functionalities for analyzing and manipulating the data, five services are introduced, which are summarized as *Analysis and Manipulation Services* (AMS). In addition to the AMS the *Cartographic Procedure Service* is defined to provide specialized processing techniques for thematic mapping. The last service, called *Thematic Map Service*, provides functionalities for rendering and publishing the map.

An important design aspect of the SOA is the separation of data processing and symbolization. Therefore, the services for data import, data processing, and map publishing are supplemented by the *Symbology Repository* for the description of map symbologies.

Data Conversion Service

For the production of thematic maps, data has to be selected and imported into the map production system. In the SOA, this tasks are provided by the Data Conversion Service (DCS), which supports the selection of data from online as well as local databases and storages. Furthermore, the DCS converts the data from the source data type into a data type appropriate for the layer-based workflow. This, for example, implies the conversion of multidimensional data sets, such as statistical data with a location, time, and attribute dimension, into two-dimensional data sets.

To handle different sources, the functionalities of the DCS are organized into so-called *conversions*. One conversion thereby provides the functionalities to select and convert the data from a specific data source. To control the transformation process the user can set parameters, which are individual to each conversion.

An example for a conversion is the selection of data from the GADM¹ (Global Administrative Areas) Web page. The DCS therefore has to download data from the GADM Web page, according to the user-defined selection parameter, and convert the data into a user-defined data format. Another example would be the import of statistical data from the OECD Open Data API².

The service interface of the DCS is presented in Figure 4.2 and provides the operations getCapabilities, getConversionDescription, and getData.



Figure 4.2.: Data Conversion Service Interface

The implementation of the getCapabilities operation returns service metadata and follows the implementation standard for OGC Web services.

The getConversionDescription operation provides metadata for a specific conversion. The response is thereby similar to the getCapabilities operation a XML-based document, which

¹http://www.gadm.org/

²http://stats.oecd.org/OpenDataAPI/Index.htm

contains information about available datasets, output formats, and parameters for the conversion process.

With the *getData* operation a specific conversion is performed. The request therefore has to specify the conversion name, the required conversion parameters, and optionally the output data type. To import local data the *getData* request allows to upload data, which can be processed by a *conversion*.

Analysis and Manipulation Services

In the generic workflow functionalities to analyze and manipulate the data are needed in the preparation and the mapping stage. In the SOA, the functionalities of both stages are classified according to similar functions and are implemented with five services.

The interfaces of the analysis and manipulation services are shown in Figure 4.3. The functionalities of each service, with exception of the Reprojection Service, are structured into operations and subordinated methods. To get an overview of the available operations and methods a *getCapabilities* request is used, which provides service metadata similar to the DCS. Specific information for a processing method can be requested with the *getMethodDescription* operation. The return message thereby includes a method description, a specification of available processing parameters, and possible input and output formats.

In addition to the uniform structure of all processing services, the request and response message is uniform over all services to enhance interoperability and service chaining. The request message contains the method name, required method parameters, and the data of one or multiple layer. Optionally the input and output data type can be specified if it is supported by the processing method. With respect to the response message, processing methods are restricted to return two-dimensional data sets which can be processed in the layer-based workflow.

In the following we will discuss the functionalities of the processing services. It should be noted that the proposed operations provide the basic functionalities for thematic map production. The derivation of a complete list of processing functionalities would be beyond the scope of this thesis. Furthermore, the intention in this section is to show the functionalities of the processing services and to present a structure for a first implementation.

The Reprojection Service provides transformation operations for vector and raster data. The specification of the source and target reference system has thereby be provided as a method parameter in the request message. The second processing service, the Classification Service, provides methods for data classification with the *classifyData* operation. In addition, the service provides functionalities to cluster data (*clusterData*) and to calculate scaling parameters (*scaleData*). Methods for placing labels and symbols are implemented in the Placement Service. The last two services group functionalities with respect to spatial statistics and spatial analysis. The Spatial Statistics Service involves methods to calculate statistics, analyze patterns, analyze distributions, and interpolate data. The implementation of the Spatial Analysis Service supports methods for measuring, analyzing geometries, generalizing data, and calculating topology properties.

«serviceInterface» Reprojection Service

- + getCapabilities(GetCapabilitiesRequest) :GetCapabilitiesResponse
- + reprojectVectorData(ProcessingServiceRequest) :ProcessingServiceResponse
- + reprojectRasterData(ProcessingServiceRequest) :AMServiceResponse

«serviceInterf ace» Classification Service

- + getCapabilities(GetCapabilitiesRequest) :GetCapabilitiesResponse
- + getMethodDescription(MethodDescriptionRequest) :MethodDescriptionResponse
- + classify Data(ProcessingServ iceRequest) :ProcessingServ iceResponse
- + clusterData(ProcessingServiceRequest) :ProcessingServiceResponse
- + scaleData(ProcessingServiceRequest) :ProcessingServiceResponse

«serviceInterface» Placement Service

- + getCapabilities(GetCapabilitiesRequest) :GetCapabilitiesResponse
- + getMethodDescription(MethodDescriptionRequest) :MethodDescriptionResponse
- + placeLabel(ProcessingServ iceRequest) : ProcessingServ iceResponse
- + placeSymbol(ProcessingServiceRequest) :ProcessingServiceResponse

«serviceInterf ace» Spatial Statistics Service

- + getCapabilities(GetCapabilitiesRequest) :GetCapabilitiesResponse
- + getMethodDescription(MethodDescriptionRequest) :MethodDescriptionResponse
- + calculateStatistic(ProcessingServiceRequest) :ProcessingServiceResponse
- + analyzePattern(ProcessingServ iceRequest) :ProcessingServ iceResponse
- + analyzeDistribution(ProcessingServiceRequest) : ProcessingServiceResponse
- + interpolate(ProcessingServiceRequest) :ProcessingServiceResponse

«serviceInterf ace» Spatial Analysis Service

- + getCapabilities(GetCapabilitiesRequest) :GetCapabilitiesResponse
- + getMethodDescription(MethodDescriptionRequest) :MethodDescriptionResponse
- + measure(ProcessingServiceRequest) : ProcessingServiceResponse
- + analyzeGeometry (ProcessingServiceRequest) : ProcessingServiceResponse
- + generalizeData(ProcessingServiceRequest) :ProcessingServiceResponse
- + calculateTopology (ProcessingServ iceRequest) :ProcessingServ iceResponse

	«ty pe» Method DescriptionRequest	«type» MethodDescriptionResponse	
++++	OperationName :string MethodName :string	+ Return :string	
«ty pe» Processi ngServiceRequest		«type» ProcessingServiceResponse	
	ProcessingServiceRequest	ProcessingServiceResponse	
+	ProcessingServiceRequest MethodName :string	ProcessingServiceResponse + Return :DataLayerType [1*]	
+++	ProcessingServiceRequest MethodName :string MethodParameter :ParamType [0*]	ProcessingServiceResponse + Return :DataLayerType [1*]	
+++++++	ProcessingServiceRequest MethodName :string MethodParameter :ParamType [0*] DataLayerType :string [01]	ProcessingServiceResponse + Return :DataLayerType [1*]	
+ + + +	ProcessingServiceRequest MethodName :string MethodParameter :ParamType [0*] DataLayerType :string [01] DataLayer :LayerData [1*]	ProcessingServiceResponse + Return :DataLayerType [1*]	

Figure 4.3.: Interfaces of Analysis and Manipulation Services

Cartographic Procedure Service

In comparison to the AMS, which provide basic and reusable methods, the Cartographic Procedure Service (CPS) is designed for the implementation of specific mapping procedures analog to the data generation step in the generic workflow.

The interface of the CPS, which follows the same conception as the previous services, is presented in Figure 4.4. To get service metadata and information about available methods a *getCapabilities* request has to be performed. Further information for a specific method can be requested with the operation *getMethodDescription*. The third operation allows to execute a method and uses the same specification for request and response messages as the AMS.



Figure 4.4.: Cartographic Procedure Service Interface

Thematic Map Service

The last service specification in the SOA, the Thematic Map Service (TMS), provides methods for the symbolization and map publishing step. The TMS is designed as an extended WMS with additional functionalities for service chaining, symbol rendering, and legend creation.

In the SOA we can define a map by describing the map creation process, which will be discussed in detail in the next section. For the understanding of the TMS, it is important to know that the map description includes the mapping data for each map layer, the service chain to derive the data, and a symbology and legend description. To create the map representation, the TMS renders the vector data of each layer with the specified symbology. In addition to the usage of the layer data in the map description, the TMS can execute the service chain specified in the map description and derive the layer data on-the-fly.

The interface of the TMS is shown in Figure 4.5. The first three operations represent the standard WMS implementation with the getMap operation for providing the rendered map layers as raster images. To perform the map creation process on the user side, the map description can be requested with the getMapDescription operation. The request message thereby allows to specify distinct layers. In addition, four boolean flags can be used to include or exclude the layer data, the service chain information, the symbology, and the legend description. Furthermore, the *RefreshLayerData* flag can be set, which will update the layer data according to the service chain.

Beside the rendering of the map representation, the TMS provides functionalities to render symbols and create legends. To render a symbol of a map on the server the request message has to include the layer name and the id of the feature connected to the symbol. To render the symbols for local map data, the user can include a map description in the request message. Furthermore, the request message can contain an alternative symbology and a specification of the graphic format. The request message to create a legend is similar to the previous one and allows the specification of distinct layers, the map description, an alternative symbology description, and the graphic format.

«serv icel nterface» Thematic Map Service					
 get Capabilities(Get CapabilitiesRequest) :GetCapabilitiesResponse get FeatureInf o(GetFeatureInfoRequest) :GetFeatureInfoResponse getMap(GetMapRequest) :GetMapResponse getMapDescription(MapDescriptionRequest) :MapDescriptionResponse get SymbolGraphic(SymbolGraphicRequest) :SymbolGraphicResponse get LegendGraphic(LegendGraphicRequest) :LegendGraphicResponse 					
«type» MapDescriptionRequest	«type» Map Description Response				
 + Lay er :string [0*] + Lay erData :bool [01] + Ref reshLay erData :bool [01] + ServiceChain :bool [01] + Symbology :bool [01] + Legend :bool [01] 	+ Return :string				
«type» SymbolGraphicRequest	≪type» SymbolGraphicResponse				
 + Lay er : string + FeatureId : string + MapDescription : string [01] + Symbology : string [01] + Output : GraphicOutput Type [01] 	+ Return :GraphicOutputType				
«type» LegendGraphicRequest	«ty pe» LegendGraphicResponse				
 + Lay er :string [0*] + MapDescription :string [01] + Symbology :string [01] + Output :GraphicOutputType [01] 	+ Return :GraphicOutputType				

Figure 4.5.: Thematic Map Service Interface

Symbology Description and Repository

Beside the derivation of the data basis for the map rendering, which can be performed with the processing services, the symbolization is an important element of the map creation process. In the generic workflow we mentioned that knowledge of visual variables are needed for the derivation of the symbology parameters, for the symbolization, and for the creation of the legend. Furthermore, the symbology definition is not restricted to one layer. Instead, the symbols of different layers have to be coordinated to provide a coherent map representation.

The symbology description therefore contains symbolizers for point, line, and area features as well as labels. Furthermore, the description includes metadata for variable symbolizer properties to derive the symbology parameters. In addition, matching rules have to be provided to apply the parameters in the rendering process and enable the separation of data and symbology. The used symbology also affects the map legend. Therefore, the description has to include a data independent legend description to provide an automated legend creation. It has to be noted that the specification of the symbology descriptions accounts for the basic functionalities needed for the map production in the SOA. A comprehensive discussion of the symbology description would be beyond the scope of this thesis. Therefore, the symbology description for the further discussion is given by four symbolizer types (point, line, area, and labels), metadata and matching rules for the symbolizer, and one or multiple legend descriptions.

To enable the integration of symbology descriptions in the service chain of map production, symbologies are managed with the Symbology Repository (SR). The interface of the SR is presented in Figure 4.6.

The getCapabilities and getSymbologyDescription requests provide service metadata and information for specified symbologies. These operations are implemented analog to the specification of the processing services. With the getSymbology operation the description of a specified symbology can be requested from the repository. In addition to the symbology name, the request message allows to specify other symbologies (*ParentSymbology*) for inheritance of style properties, to specify the included symbolizer types (*SymbolizerType*), and to select an available legend description (*LegendName*).



Figure 4.6.: Symbology Repository Interface

The components of the SOA can be implemented in numerous ways depending on the needed functionalities and infrastructure. In the following, we will show the realization of the generic workflow with the presented components to demonstrate the operation of the SOA. A further discussion of different implementations is given in section 4.3.3.

The schema in Figure 4.7 presents the components and interactions of the SOA implemented similar to the generic workflow. For the understanding of the further discussion, it is important to known that in addition to the operations of the presented components, the system of the map creator provides basic functionalities for data management and visualization.



Figure 4.7.: Service-oriented Architecture for Thematic Map Production on the Web

In the first step, the data preparation, the map creator selects and imports data from different sources with the DCS (1). The service thereby establishes a data connection to online databases and storages and converts the data to a format suitable for the mapping process. In addition to the usage of online data, a DCS request enables the conversion of local data. The analysis and manipulation of the attributive and spatial data is performed with the methods of the AMS (2). The last step of the data preparation stage is data integration, which is implemented in the production system of the map creator. After combining attributive and spatial data, the data generation step is used to derive specialized map data. The functionalities and procedures are thereby implemented in the CPS (3a), which can utilize the AMS to perform basic processing tasks (3b). In the mapping stage the data is combined with the symbology to render the map representation. The symbology description is thereby requested from the SR (4) and includes point, line, area, and label symbolizers as well as metadata and matching rules for variable symbolizer properties. The methods of the AMS are used to adjust the vector data for the symbolization and derive the symbolizers properties from the vector data (2). Subsequently, the rendering of the map layers is performed with the TMS (5) or directly in the system for map creation. For the last stage of the generic workflow, which includes map composition and map publishing, the map creator specifies the layers and elements of the map in the map description. In addition, the map is published by uploading the map to the TMS (5), which provides the request interface for the map user (6).

The presented implementation for the generic workflow covers the main components of the SOA. It should be noted that a variety of additional Web services can be utilized for the map creation process. Examples are georeference services, which can support the integration of attributive and spatial data, or ontology services, which can provide additional metadata.

4.3.2. The Map Description

With the SOA it is possible to describe the layer creation process from data import to symbolization by specifying the order of performed service requests and the used symbology. Furthermore, a map description can be defined by specifying the service chain for layer creation and the composition of layers and additional map elements. The specification of all elements of the map description would be beyond the scope of this thesis. Therefore, the following discussion is limited to identifying the main elements, which have to be considered for the specification of the map description.

The identified elements of the map description, which are presented in Figure 4.8, are modeled in a hierarchical structure. On the highest level, five instances are specified. Each map description includes one *Map Information* and one *Map Composition* element. In addition, the map description includes one or multiple *Map View* elements and the optional elements *Legend Composition* and *Map Interaction*.

The first two elements, *Map Information* and *Map Composition*, are used to specify overall map information and the setup of the map elements. For the map information examples are the map title, the map projection, and map metadata. The *Map Composition* element defines the arrangement of the map views and additional map elements, such as the legend and imprint.

The *Map View* element defines the extent, the layers, and additional information for a mapped area. The layer definition thereby includes five sub-elements for general layer information, layer properties, layer data, the layer service chain, and the layer symbology and legend description. The map data for the map view is either included by providing the processed layer data or by specifying the service chain for the derivation of the layer data. In the service chain specification, the instances *Data Conversion Service* and *Data Processing Service* define the request to the corresponding services. The import of preprocessed data can be established with the element *Source Data*. In order to perform data integration and simple data manip-

▼ Map Description

- ▶ Map Information (title, description, projection, metadata, ...)
- ▶ Map Composition (map views, legend, impress, ...)
- ▶ Map View^{*}
 - \blacktriangleright Map View Information (title, description, projection, metadata, ...)
 - \blacktriangleright Map View Extent
 - ▶ Map View Layer
 - ► Layer Information (title, description, projection, metadata, ...)
 - ► Layer Properties (layer type, visibility, transparency, ...)
 - \blacktriangleright Layer Data **
 - ▶ Layer Data Processing Chain^{**}
 - ▶ Source Data^{*/**}
 - ► Data Conversion Service^{*/**}
 - ▶ Data Integration^{*/**}
 - ► Data Processing Service^{*/**}
 - ▶ Data Manipulation^{*/**}
 - ► Additional Web Services^{*/**}
 - ► Layer Symbology and Legend Description
- ► Legend Composition^{*/**}
- ▶ Map Interaction^{**}
 - \blacktriangleright Interactive Task^{*}

Figure 4.8.: Elements of Map Description (* multiple instances possible, ** optional)

ulation tasks in the service chain the instances *Data Integration* and *Data Manipulation* are defined. The last element consider the integration of additional services from the Web infrastructure, such as georeference and ontology services. For the definition of the symbology and legend description either the SR is specified or the symbology description is directly included in the map description.

The map description can include multiple *Legend Composition* elements, which define a legend by specifying the composition of individual layer legends. The last element, *Map Interaction*, includes different interactive tasks to account for map interaction in the map description.

This section gave an overview of the elements which should by considered in the specification of a map description. The specified elements can be seen as a basis for the further discussion and for further research.

4.3.3. Different Implementations of the Conceptual Model

The components of the SOA can be implemented in different ways. The implementation for a specific use case thereby depends on the system of production, the system of the map users, and the available infrastructure. In the following, we will discuss two configurations to demonstrate the capabilities of the SOA for different applications. In the first implementation, named as *database-centered* approach, we will discuss the characteristics of a highly centralized system. The second implementation, named as *map-description-centered* approach, will discuss a decentralized implementation of the SOA components.



Figure 4.9.: Database-Centered Implementation of the SOA

The database-centered approach, presented in Figure 4.9, demonstrates a centralized implementation of the functionalities of the SOA. The main component is the map production system which can for example be realized as a Web server software similar to the application GeoServer³. The production system coordinates the components of the SOA and exposes the functionalities for map production to the map creator. The main component of the production system is the database. The DCS and the processing services are thereby directly implemented at the database level. In addition to the database layer, the production system includes the SR and the TMS to render the map representation from the data in the database. The connection to data sources and services on the Web is directly performed by the map production system. To describe the map creation process the production system utilizes the

³http://geoserver.org

map description specification. The distribution of maps to the map users is enabled by the TMS interface.

The database-centered approach is designed to implement all functionalities in one environment. This design provides a broad control over the components and high computational performance. The control over all components allow to minimize security risks by strictly controlling the interfaces to the map creator and the map user. The implementation of the processing functionalities is directly connected to the database. This enables a fast access to the data and allows to optimize the processing functionalities to the actual data representation. Another benefit of the centralized production system is that the system of the map creator can be reduced to functionalities for data visualization if the functionalities for data integration are implemented in the production system.

The disadvantage of a centralized production system is the limited flexibility. The implementation of the DCS and the processing services directly at the database level make it difficult to access the functionalities outside the map production system and therefore difficult to reuse the functionalities in other applications.



Figure 4.10.: Map-Description-Centered Implementation of the SOA

In the map-description-centered approach, presented in Figure 4.10, the components of the SOA are implemented as decentralized Web services. Thereby, the map creator directly interacts with the service interfaces. To describe the map creation process the map creator locally generate the map description, which can be uploaded to a TMS or can be directly sent to the map user.

In the map-description-centered approach the map creator is not dependent from a production system. Instead, the map creator can flexibly combine the components of the SOA, which are implemented at different resources on the Web.

The disadvantages of the decentralized approach are with respect to computational performance and service availability. In the map-description-centered approach the mapping data has to be transferred to the individual service for each processing step. This increases the data transfer between map creator and services and reduces computational performance as a consequence of separating the data representation from the implementation of the processing functionalities. In addition, the map creation process is reliant on the availability of services, which are not managed by the map creator.

It has to be noted, that the presented approaches, which demonstrate idealized implementations, does not exclude each other. The functionalities of a database-centered implementation can be used in a map-description-centered approach by providing additional interfaces to the map creator for all implemented services. In addition, the loosely implemented services of the map-description-centered approach can be accessed by the map production system to integrate new functionalities.

4.3.4. OGC Web Standards and Specifications for the Conceptual Model

In addition to the modeling presented in the previous sections, this section puts the conceptual model in the context of the current Web infrastructure. Therefore, we will discuss the model with respect to OGC Web standards and additional specifications. The discussion intends to demonstrate possibilities for the implementation of the components and to reveal shortcomings of current standards.

The first aspect which has to be considered is the data format to model vector features for the cartographic purpose. Thereby, the data format has to provide interoperability between the components of the SOA and with a variety of services on the Web and local applications. For data representation the main OGC standard is the Geography Markup Language (GML), which is an extensive XML grammar to model geographic objects. In addition to the OGC specification, GML is specified by the International Organization for Standardization (ISO) since the release of version 3.1.1. Furthermore, GML is supported by the majority of geospatial tools and applications as well as the implementations of OGC Web services. For the standard vector data format in the conceptual model, we therefore propose GML, with the restriction to the subset of the Simple Feature Profile. The benefits of this profile are the restriction of geographical objects to simple vector data representations and the prevention of feature nesting, which restricts the data representation to two-dimensional data sets suitable for the layer-based approach in the SOA.

The DCS can be compared to the WFS implementation for vector data and the WMS implementation for raster data. The main difference is thereby that the requested data is not stored on the Web server. Instead, the DCS requests data from different resources on the Web and converts the data according to the conversion parameters. The DCS can therefore enable the import of a variety of different resources and data formats. The main benefit of the DCS is the uniform interface for data import which is required to define the data import in the map description file. The implementation of the AMS and the CPS can be realized with a WPS, which provides a generic specification for the implementation of processing functionalities as a service. The difference of the processing services in the SOA and the WPS specification, is the restriction of request and response messages to a uniform definition for all services, which was included to enhance interoperability.

The design of the TMS represents an extended WMS, similar to the Cartographic Web Service (CartoWS) developed in Iosifescu-Enescu (2011). The difference to the CartoWS is thereby the integration of the layer information and symbology description into the provision of the map description.

To symbolize geographic features OGC specified the XML-based description language Symbology Encoding (SE). The standard specification of SE is limited to the description of cartographic representations. Therefore, the work of Iosifescu-Enescu (2011) and the study of Dietze and Zipf (2007) proposes an enhancement of SE to enable symbolization for topographic and thematic maps. Another approach, which can be used to extend SE to support complex thematic point symbols, is developed in Schnabel (2007) with the description language DiaML. Thereby the definition of symbolizers is performed with a XML-based syntax, which is based on a construction model for symbol generation.

For the implementation of the symbology description, symbolizers for point, line, and area features as well as labels are required. An extended SE specification which allows to specify matching rules to combine data attributes with SE Rules could provide the realization of data independent symbolizers. In addition to the symbolizers, the symbology description includes a specification of map legends. The legend description has thereby be correspondent to the symbolizers of the symbology description. At the moment there is no specification available for the description of legend representations.

The last element of the conceptual model is the map description, which should be implemented with an XML-based syntax to simplify the integration of vector data in the standard format GML and the symbology description in the SE specification. With respect to the definition of the service chain the established standards BPEL and BPML are important. Both specifications define a XML-based syntax for orchestration of Web services described with WSDL and accessed with SOAP messages. The benefits of using BPEL or BPML for the service chain description are the availability of orchestration engines to execute the service chain and the directly support of additional services described with WSDL. The disadvantages are the complexity of both specifications and the requirement of SOAP for the service requests.

A detailed discussion on service chain description and the service interface implementation can not be done at this point. The selection of the technology depends on the requirements to the specific implementation. For complex systems, with requirements for service contracts and security, SOAP and established orchestration languages provide an extensive toolset. In contrast, REST provides a simple and lightweight approach to implement service interfaces. With respect to SOAP and REST it should also be noted that RESTful services can be provided with a interface described in WSDL, as the studies of Sancho-Jiménez et al. (2008) and Fleuren and Müller (2008) showed for OGC Web services such as WCS, WMS and WFS.

5. Implementation

In addition to the modeling, this chapter demonstrates a proof-of-concept by implementing an application which is based on the architecture of the conceptual model. The first section in this chapter provides an overview of the implemented application. In the second section, the capabilities of the system are demonstrated by explaining the creation process of three different map representations.

5.1. Application Design

The main purpose of the client-side application is to demonstrate the operation of the conceptual model. Consequently it has to be possible to import, visualize, and process attributive and spatial data. In addition, the application should provide capabilities for service chaining and map rendering. In the following, we will discuss the implementation of the functionalities of the conceptual model on server- and client-side. The software packages and third-party libraries used to implement the application are listed in section A.1.

5.1.1. Client-side Implementation

The client-side application is realized as a browser-based application which uses server-side services for data processing. The benefits of using browser-based technologies for the implementation of the map production system are the direct integration into the Web infrastructure, which simplifies the implementation of Web service requests, and the platform independent development, which allow to run the application on a variety of devices.

With respect to the development of rich Internet applications (RIA) we can distinguish between plugin-based technologies and applications based on HTML and JavaScript. The most popular plugin-based technologies are thereby Adobe Flex and Microsoft Silverlight, which provide a controlled development environment for the creation of dynamic applications. The disadvantage is the need for additional software on the client side to run the application. In contrast to plugin-based technologies, HTML and JavaScript are natively supported by the majority of browsers. With the implementation of HTML5 functionalities and the efficient execution of JavaScript code in modern Web browsers it is possible to directly provide RIA, without the need of a plugin. The application in this thesis is developed with HTML and JavaScript to provide access for a variety of users. For the implementation the JavaScript framework AngularJS¹ is used, which provides a structure and functionalities for RIA. The benefits are thereby a modular implementation and the possibilities for a test-driven development. In the following we will discuss the developed application by means of the implemented AngularJS modules, which are presented in Figure 5.1.



Figure 5.1.: AngularJS Modules for the Client-side Application

The main modules of the application are the *Editor* and the *Viewer* module, which contain the business logic for the editor and the viewer. The editor is thereby used to perform the

¹http://angularjs.org/

map creation process and to save the map description. The viewer is designed to display a map representation which is defined by a map description.

To perform the tasks for map creation, the Editor module includes one module for data management (DataStore), three modules for the user-interface, and five modules for data import and processing.

The user-interface for the editor is divided into three views (Figure 5.2), which are implemented in the modules Map, Grid, and Tools. The map view is used to visualize the map representation. Therefore the Map module implements the mapping library OpenLayers 3^2 , which provides an extensive toolset to process and render spatial data in the browser. For displaying attributive data, the Grid module implements a data table with the AngularJS extension ng-grid³. To perform the tasks of the map creation process, the Tools module provides four subviews, which are implemented in the respective submodules. The first subview gives an overview of the data layers and provides functionalities for map publishing. In the input view, different data sources can be selected and imported into the application. To process the data, the processing view provides an user-interface to send requests to the processing services. The symbology view provides functionalities to select a map symbology for a spatial data layer.



Figure 5.2.: Editor in the Client-side Application

²http://ol3js.org/

³http://angular-ui.github.io/ng-grid/

In addition to the modules for the user-interface, five modules are implemented to get service descriptions, to import data, to perform service request, to manage and create map symbology, and to process the service chain.

The *DescriptionService* module is used to provide service information. The descriptions are thereby directly provided by the module to simplify the implementation. In a further version of the application, this module can be replaced by directly requesting the service descriptions from the server with the operations getCapabilites, getConversionDescription, getMethodDescription, and getSymbologyDescription as specified in section 4.3.1.

The module *InputHandler* implements functionalities for data import directly in the application and can be compared to the Data Conversion Service. In the current version of the application, it is possible to import data sets that are stored on the server. With respect to the data format the implementation enables the import of spatial data in GML and attributive data in JSON-stat or CSV. The different data sets are thereby defined in the *DescriptionService* module and are imported over the user-interface of the *Input* module.

To process data on the server, the *WebService* module is used. A service request to an Analysis and Manipulation Service or the Cartographic Procedure Service can thereby be performed with the user-interface of the *Processing* module. The definition of available processing services is done in the *DescriptionService* module.

The *Symbolizer* module provides functionalities to manage and create the map symbology. Therefore the module imports a symbology description specified in the *DescriptionService* module and provides the description of the point, line, area, and label symbolizers. In addition, the *Symbolizer* module implements functionalities to match attributive values with the symbolizer description and render the graphical representation.

To demonstrate the capabilities for automated mapping, functionalities to define and execute a service chain were implemented. The individual elements of the service chain are specified with five JavaScript objects, presented in section A.3, which enable to define data import, data integration, basic data manipulation, processing service requests, and data symbolization.

In the current version of the application the service chain definition is used to save and load a mapping project and to specify the map creation process in the map description. To save and open a workflow, functionalities are provided in the overview user-interface. The map description includes the service chain elements and basic map informations and is also specified with a JavaScript object. A definition of the object is given in section A.3.

For the map description the service chain elements are directly transformed into XML. The functionalities to derive a service chain for a created map representation as well as the conversion between JavaScript objects and XML are implemented in the *ServiceChain* module.

To display the map representation defined with a map description, the application includes the viewer. The *Viewer* module thereby uses the functionalities of the *ServiceChain* module to convert the service chain elements from XML to JavaScript objects. The individual tasks of the service chain are executed with the functionalities of the *Editor* module. To display the map representation the *Map* module is included.

5.1.2. Server-side Implementation

The server-side implementation includes server software to provide the application data and to manage service requests. In addition, the server-side system has to provide the processing functionalities for the map creation process.

For the sever software we use the platform Node.js⁴, which utilizes JavaScript as scripting language and therefore enables a fast development in combination with browser applications based on HTML and JavaScript. Another benefit of Node.js is the non blocking input/output model, which enables a parallel processing of requests.

To provide the functionalities of the processing services a REST interface was implemented on the server for the operations required to create the example maps (section A.2). The layer data of the request is thereby processed in a PostgreSQL⁵ database with PostGIS⁶ extension according to the request parameters. For the import of GML data into the database as well as the export the functionalities of the Geospatial Data Abstraction Library (GDAL)⁷ are used. The benefit of the processing with PostgreSQL is the availability of basic functionalities for spatial data processing which can be utilized to implement mapping procedures.

5.2. Map Creation

In this section, we will demonstrate the map creation process of three map representations and discuss the implemented processing functionalities as well as the symbology description. To visualize the map creation process the sequence diagrams according to the operations in the conceptual model are presented. It must thereby be noted that the Data Conversion Service and the Symbology Repository are directly implemented in the browser-based application. Furthermore, the legends for all example maps were created manually, by the reason of the limited development time.

The source data for all example maps come from the Swiss Federal Statistical Office. Attributive data was thereby taken from the STAT-TAB⁸ database. Spatial data for the Swiss cantons and districts come from the GEOSTAT⁹ service.

5.2.1. Choropleth Maps with Standardized Data

The first map representation is the choropleth map with standardized data, which can be created with the sequence shown in Figure 5.3. For the example map, population data and spatial data for the districts of Switzerland are imported and integrated.

⁴http://nodejs.org/

⁵http://www.postgresql.org/

⁶http://postgis.net/

⁷http://www.gdal.org/

⁸http://www.pxweb.bfs.admin.ch/dialog/statfile.asp?lang=1

⁹http://www.bfs.admin.ch/bfs/portal/de/index/dienstleistungen/geostat.html



Figure 5.3.: Sequence Diagram Choropleth Map



Figure 5.4.: Example Choropleth Map

To derive standardized values for the population attribute, three processing steps have to be performed. In the first, the area of each district is calculated with the *measure* operation of the Spatial Analysis Service, which is based on default PostGIS functions. The next step is performed directly in the browser-based application by dividing the population values by the size of each district. In the last step the data set is classified with the *classifyData* operation of the Classification Service. The implementation of the classification method is thereby done with a SQL function.

The symbology description for the choropleth map is specified as a JSON object to enable a direct integration in the application (see section A.4). The object defines overall symbolizer properties, such as the line color for the areas, and variable symbolizer properties which are matched to the areas according to the derived classes. After assigning the symbolizers the map representation can be rendered as shown in Figure 5.4.

5.2.2. Proportional Symbol Maps and Diagrams

For the creation of a proportional symbol map, attributive data, containing figures for established enterprises, and spatial data for Swiss cantons are imported and integrated. In the following processing step, the centroids of the areas in the spatial data set are calculated to derive point representations for the proportional symbol map. The processing is thereby performed with the *analyzeGeometry* operation of the Spatial Analysis Service. In the last step the symbology description is imported and matched with the attributive values. The sequence diagram for map creation is shown in Figure 5.5.

The symbology description for the diagrams of the map representation is defined in DiaML, which was developed in Schnabel (2007) and provides a XML-based syntax to define diagram representations independent from the data (see section A.4). For the rendering of the map, diagrams are created in the *Symbolizer* module by matching the attributive data values with the diagram description. In addition to the diagram symbols derived in the map creation process, the example map, shown in Figure 5.6, includes a base layer from Mapbox ¹⁰.



Figure 5.5.: Sequence Diagram Proportional Symbol Map

¹⁰https://www.mapbox.com/



Figure 5.6.: Example Proportional Symbol Map

5.2.3. The Cartographic Procedure Service and Dot Maps

The last mapping process illustrates the creation of dot maps and includes the Cartographic Procedure Service. The sequence diagram is therefore shown in Figure 5.7. To get the data basis for the calculation of the dot placement, population and area data for the districts of Switzerland are imported and integrated.

The calculation of the dot placement, implemented with the Cartographic Procedure Service, is executed with the *executeMethod* operation. The request message therefore includes the area data with the attribute values, the name of the attribute which defines the dot distribution, and two parameters to define the unit value of one dot and the minimum distance between dots. The response of the procedure is a data set with point features defining the dot positions. The implementation of the processing in PostgreSQL is done with a SQL function, which uses a random function to generate dots and a measuring function to guarantee a minimum dot distance.

After the calculation of the dot distribution, the map representation is rendered by using a simple point symbolizer for the point features. In addition to the dot representation, a spatial data set containing the borders of the Swiss cantons is imported and symbolized with a polygon symbolizer. The symbology description, including both symbolizers, is defined as JSON object analog to the definition for the choropleth map in the first example. The example map containing both layers is shown in Figure 5.8.



Figure 5.7.: Sequence Diagram Dot Map



Figure 5.8.: Example Dot Map

6. Evaluation and Discussion

The conceptual model presented in chapter 4 provides a decentralized approach for thematic map production on the Web. In this chapter we will examine the capabilities of the derived model with respect to the creation of standard thematic map representations and to the application of important concepts for thematic Web cartography.

To discuss the capabilities of the conceptual model for thematic map production, in the first section we will review the suitability of the generic workflow for the production process of the standard thematic map representations. In the second section, we will evaluate the conceptual model with respect to the implementation of the concepts specified in the requirements. In the last section of this chapter, we will review the research questions and objectives to deduce the findings of this work.

6.1. Standard Thematic Map Representations and the Generic Workflow

The generic workflow, presented in section 4.2.2, is the basis for the conceptual model and is developed with respect to the production of standard thematic map representations. For the modeling, the scope of thematic cartography was restricted to two-dimensional map representations based on vector data. In addition to the restrictions, it should be noted that the workflow is designed for automated map production to support an efficient implementation of service chaining in the conceptual model.

Below we will evaluate the suitability of the standard thematic map representations with respect to the generic workflow. A summary of the evaluation is presented in Table 6.1.

Mapping Technique	Eval.
Choropleth Map	++
Dasymetric Map	+
Isarithmic Map	+/-
Cartogramm	+/-
Proportional Symbol Map	+
Dot Map	+
Flow Map	-

 Table 6.1.: Evaluation of Standard Mapping Techniques

The choropleth mapping technique does not include a data generation step in the generic workflow. The uniform data set can directly be mapped by classifying the data and selecting a style (color scheme) for the areas. The steps of the map production process can thereby be implemented with basic processing functionalities and easily be automated. Therefore, the generic workflow is highly applicable to create choropleth maps.

For the dasymetric map, source areas (areas of enumeration unit) and ancillary areas need to be combined in the data generation step. The remaining mapping process is similar to the choropleth mapping technique. The implementation of procedures for data generation requires additional effort, for which reason the evaluation is lower compared to choropleth mapping.

For the isarithmic mapping technique an interpolation has to be performed. The results of this operation can be areas, isolines, raster images or DEMs, which have to be symbolized in the mapping stage. In the generic workflow the interpolation requires specialized implementations for data generation. Furthermore, automation is more difficult if the number and complexity of parameters increases. Therefore, the realization of isarithmic maps with the generic workflow depends on the complexity of the map representation. Isarithmic maps with a simple interpolation step and a single feature type for symbolization get a positive evaluation, whereby maps with multiple processing steps and different features to symbolize get a negative evaluation.

Another similar map representation to choropleth maps is the cartogram. The difference thereby is the transformation of the areas before the mapping, which is based on attributive values. For cartogram mapping, transformation algorithms has to be implemented in the data generation step. The implementation of an automatized workflow is thereby dependent on the complexity of the algorithms and the need for manual editing. Therefore, the evaluation for the creation of cartograms is, analog to isarithmic maps, dependent on the map representation.

The parameters to create the symbols for proportional symbol maps can be derived directly from the data set by classifying, clustering and scaling the data. For the symbolization of the point features the symbology description has to provide complex point symbolizers. In addition, complex maps, with a high number of map features, require a manual arrangement of symbols to solve placement conflicts. The proportional symbol mapping technique gets a positive evaluation similar to the creation of dasymetric maps. The highest evaluation is prevented by the demands for complex point symbolizers and symbol placement in complex maps.

For a dot map the distribution of dots is computed by means of source areas, ancillary areas or a DEM. To determine the position of the dots, point features are derived in the data generation step. In the mapping stage simple point symbolizers are used for symbolization. With the implementation of algorithms to derive point features the map creation process for dot maps is applicable for automation. Therefore, the dot map technique gets a positive evaluation.

The processing of data in the generic workflow is restricted to vector data. The creation of distributive flow maps and network flow maps can thereby be based on line features combined with line symbolizers. In addition to the definition of complex line symbolizers, elaborate map

representations require manual editing of line features. Radial flow maps are mapped with complex point symbolizers including direction and magnitude of flows. Continuous flow maps include the mapping of a vector field and often require data interpolation functionalities. The creation of the different types of flow maps show a great variation of functionalities and the need for manual editing. Therefore, the flow mapping technique is difficult to implement with the generic workflow, which leads to a negative evaluation.

6.2. Implementation of the Concepts for Thematic Web Cartography

The evaluation of the conceptual model in this section will discuss the implementation of the concepts for thematic Web cartography. The evaluation is based on the requirements which were the basis of the modeling process. A summary of the evaluated requirements is presented in Table 6.2, whereby the required functionalities are assessed with respect to the support by the conceptual model. *Directly supported* functionalities are thereby implemented in the architecture of the conceptual model. Functionalities marked as *indirectly supported* are considered in the service-oriented architecture (SOA) but require the implementation of additional concepts and components. Functionalities which are not considered by the conceptual model or can not be implemented are marked as *not supported*.

Functionality	directly	indirectly	not
Functionality	supported	supported	supported
GIS functionalities	•	•	
DTM functionalities	•	•	•
Standard thematic map representations	•		•
Interaction with the data	•		
Interaction with the data representation	•		
Interaction with the temporal dimension		•	
Contextualizing interaction		•	
Collaborative map production		•	
Sharing of intermediate and end products	•		
Sharing of functionalities	•		
Linking of documents and maps		•	
Linking of data	•		
Linking of services	•		
Different publishing concepts		•	
Generation of metadata		٠	

 Table 6.2.: Evaluation of Requirements for Thematic Web Cartography

To perform a comprehensive evaluation of functionalities that are used in GIS and DTM systems would be beyond the scope of this thesis. Therefore, we will limit the discussion to processing functionalities, user-interface functionalities, and functionalities for data visualization.

The support of GIS functionalities with respect to processing is directly implemented in the SOA. The AMS allow to perform basic processing and can be extended to provide the majority of GIS functionalities required for thematic mapping. With respect to user input, simple processing tasks can be specified with a plain service request. For complex processing the user-interface has to provide additional functionalities for tasks such as feature selection and parameter definition. In addition, the map production system has to implement a module for data visualization to enable efficient data analysis and manipulation.

DTM systems primarily use vector graphic formats to create map symbologies. The approach in this thesis creates map representations by matching predefined symbolizers with vector features. In addition, the model strictly separates the vector data from the symbology, which is contrary to cartographic data models used in DTM systems. The support of DTM functionalities is therefore dependent on the complexity of the map symbology. Highly complex and individual map symbologies are not supported by the developed model. Nevertheless, the majority of symbolizers for thematic maps can be defined in the symbology description and are therefore directly supported. In addition components for visual symbology description are not directly implemented in the SOA and have to be provided in a additional user-interface.

The map creation process for the standard thematic map representations was already discussed in the previous section. The discussion showed that the main criteria for the map creation process, to be suitable for the developed system, are the restriction to vector data and the suitability for automation.

The functionalities for providing interactive tasks were introduced in the generic workflow and can be directly applied to the conceptual model. Tasks for interaction with the data can be directly implemented by accessing the map description and changing import and processing parameters in the service chain. Furthermore, the separation of vector data and symbology description reduce the effort to exchange the map symbology and therefore to implement interactive tasks with the data representation. Contextualizing interactions and interaction with the temporal dimension are marked as indirectly supported, by reason of the requirements for a specialized user-interface, which provide functionalities such as arrange multiple views or navigation.

With respect to collaboration the sharing of intermediate products, end products and functionalities is directly supported by the conceptual model. The provision of end products can thereby be performed with the TMS or directly with the map description file. The map description can also be used to share single layers as well as the workflow for map creation. An additional intermediate product of the production process is the map symbology, whereby an exchange can be achieved over a SR or a symbology description file. The sharing of functionalities is directly implemented in the model with the CPS, which enables the provision of specialized processing techniques with a predefined Web service interface. Collaborative map creation is enabled by the conceptual model through the layer-based approach and the separation of feature data and symbology, which allow to separate the tasks for map creation. In addition to the functionalities of the model, collaborative mapping requires a platform which makes it possible to communicate and manage the mapping process. Therefore, the functionalities for collaborative mapping are marked as indirectly supported.

The linking of documents and data is inherent to the Web infrastructure. With the conceptual model linking can be implemented by directly connecting the source data with the mapped data. With respect to the semantic Web, RDF triples can be generated automatically from a vector data set with a DCS or a similar service. It has to be noted that an extensive review of concepts for linking with respect to thematic cartography could not be done in this thesis. Therefore, the linking of documents and maps is marked as indirectly supported for the reason that concepts for the implementation of map and document links are not described in the conceptual model. The linking of services is the basis of the SOA and is defined in the service chain description. The utilization of established standards for service orchestration can thereby enable the integration of a variety of Web services in the map creation process.

The last two requirements involve functionalities to enhance searchability and findability. The publishing of additional formats to optimize the map for search engines is not directly implemented in the model. Therefore, additional conversion functionalities have to be implemented similar to the generation of RDF triples mentioned before. Furthermore, the creation of metadata is not directly considered by the conceptual model, but is enhanced by the possibilities for using additional services in the creation process.

6.3. Discussion of Research Findings

This section examines the findings of this thesis. Therefore, we will review the research questions and objectives formulated in the introduction with respect to the derived workflow and conceptual model as well as the implemented application.

The first research question focuses on the definition of a generic workflow for thematic map production for the Web:

Q1: Can the elements and the sequences of thematic map production for the Web be modeled in one workflow?

The modeling of the generic workflow (Figure 4.1) was based on the creation process of thematic map representations and restricted to two-dimensional map representations. In addition, the data processing was limited to vector data and strictly separated from the map symbology to enable an automated map creation for the Web.

The discussion of thematic map representations shows that the suitability of the generic workflow is dependent on the complexity of the representation, whereby mapping techniques which rely on extensive manual editing are not applicable. Furthermore, suitable mapping techniques have to be based on vector features for the creation of the map representation.

Therefore, it can be concluded that it was not possible to derive a workflow which is applicable to the variety of mapping techniques for thematic cartography.

Nevertheless, the evaluation also showed that the majority of thematic mapping techniques based on vector data can be modeled with the presented workflow. Therefore, the derived workflow can be utilized as a basis for the implementation of map production functionalities in the Web infrastructure.

The second research question is directed to the modeling of the generic workflow with a service-oriented architecture:

Q2: Can the thematic map production workflow be modeled with a service-oriented architecture?

For the development of a service-oriented architecture (SOA) for thematic map production on the Web, the functionalities of the generic workflow were modeled with three different types of Web services (Figure 4.7): The Data Conversion Service implements functionalities for the import of data from different sources, the Analysis and Manipulation Services and the Cartographic Procedure Service implement processing functionalities, and the Thematic Map Service provides publishing functionalities. In addition to the services, a Symbology Repository was introduced to include the map symbology directly in the SOA.

The conceptual model presented in section 4.3 shows that the required functionalities for map production in the generic workflow can be modeled with a SOA. In addition, the examination of different implementations of the conceptual model in section 4.3.3 shows that the map creation process can be realized with decentralized services. The operability of the SOA for map production is demonstrated in the proof-of-concept. Furthermore, the proof-of-concept demonstrates that the concept of service chaining is applicable to the conceptual model and can provide an automated map production process. Therefore, we reason that the generic workflow for thematic mapping is suitable for the implementation with a SOA.

With the third research question we discuss the realization of the conceptual model:

Q3: Are the existing standards and technologies sufficient for the implementation of the service-oriented architecture?

Existing standards for the implementation of the conceptual model were discussed in section 4.3.4. Thereby, it was shown that a data independent specification for the symbology description as well as for the legend description is needed to fully implement the SOA. Furthermore, a specification of the service chain and map description is required to exchange the map creation process.

With respect to technologies for the implementation of the map production system and the components of the SOA, the proof-of-concept demonstrated the map production for three different map representations. The application is thereby limited to selected functionalities and can not provide a comprehensive demonstration of the possibilities of existing Web technologies.

Therefore, the technology question remains open at this point. In addition, further standards and specifications are required to describe the map representation and the map symbology in particular.

The objectives of this thesis were to develop a SOA for map creation and to deliver a proof-of-concept by implementing selected functionalities.

- **O1:** To develop a conceptual service-oriented architecture for thematic map production on the Web which considers important concepts for Web cartography.
- **O2:** To deliver a proof-of-concept by implementing selected functionalities of the service-oriented architecture.

The discussion of the research questions shows that it was possible to develop a conceptual model which is based on a service-oriented architecture. The model thereby includes important concepts for thematic Web cartography as it was examined in section 6.2. In addition to the modeling, the proof-of-concept presented in chapter 5 could demonstrate the capabilities of the SOA and shows the possibilities for automated map production.

The derivation of the conceptual model focuses on the functionalities for the production of two-dimensional map representations based on vector data and provides a elementary concept for thematic map production on the Web. Extensions of the model should therefore include functionalities for the processing of raster data and the support of three-dimensional map representations. With respect to the implementation, an extension of the functionalities of the proof-of-concept could demonstrate the capabilities of map creation on the Web on a larger scale.

7. Conclusions and Outlook

The last chapter of this thesis concludes the main findings of the work and draws out their implications. In addition, we will give an outlook to further developments and possibilities for research.

7.1. Conclusions

The thesis is focused on the development and examination of a concept for thematic map production on the Web. The aim was thereby to develop a conceptual model, which involves the main components for thematic map creation and is based on a service-oriented architecture suitable for the Web infrastructure. For the derivation of the model, the first step involved a review of important concepts for thematic Web cartography and the production of standard thematic map representations. Based on these findings a map creation workflow was derived, which provided the basis for the conception of the service-oriented architecture. In the conceptual model, the required functionalities for map creation were identified with specific Web services and extended with additional components of the map creation process.

The results of the evaluation show that the conceptual model is suitable for the creation of the majority of thematic map representations and is capable to support important concepts for Web cartography. The main limitations of map creation are the restriction to two-dimensional map representations and to a creation process based on vector features. With respect to the operability of the conceptual model, the implementation of the creation process for three example maps tests the capabilities of the model. Although the implementation is limited to selected functionalities, the application shows that the map creation process can be performed with decentralized services. Furthermore, it could be demonstrated that automated map production is possible. The main obstacle for the implementation is the lack of specifications. Therefore, a data independent definition of the map representation, including the definition of the steps for map creation, is needed to implement automated mapping and to exchange production workflows.

For the field of Web cartography this thesis provides a concept for the production of thematic maps on the Web. The work, which includes the map creation process from data import to map publishing, can be seen as a continuation of service-oriented approaches, and thereby extends current map provision approaches by an extensive set of processing functionalities for map creation. Furthermore, this thesis provides a basis for further research on decentralized and automated map production on the Web by outlining the functionalities and components required for thematic Web cartography.

The findings demonstrate the capabilities of a server-oriented architecture when it comes to implementing Web based map production systems. The main benefits are the direct integration in the Web infrastructure and the capabilities with respect to the description of the map creation process. Thereby allowing a decentralized implementation of the components and enabling efficient integration of various Web based sources and services. Furthermore, a description of the creation process enables the exchange of production workflows and offers possibilities for automated mapping.

7.2. Outlook

With further developments in the Web infrastructure more and more services will become available on the Web supported by enhanced technologies and increased coverage and bandwidth. For the field of thematic Web cartography these developments bring new opportunities for research and development of cartographic mapping systems.

Further research with respect to this work has to address a data independent symbology and legend description. In addition, the derivation of an extensive map description including concepts for service chaining is necessary to enable automated map creation. With respect to the extension of the approach, processing functionalities for raster data and for the creation of three-dimensional map representations has to be evaluated.

For the implementation of map production systems on the Web this study provides an elementary concept. A continuation of this approach can enable a comprehensive implementation of map production with decentralized services. In addition, the adoption of the concept on a larger scale enhances the interoperability between different implementations and can provide new possibilities for automated map creation. Furthermore, enhanced interoperability can increase networking between cartographers for the creation of maps and the exchange of functionalities.

Bibliography

- Ankolekar, A., M. Krötzsch, T. Tran, and D. Vrandečić. (2008): The two cultures: Mashing up Web 2.0 and the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web, 6 (1), 70–75.
- Asche, H. and R. Engemaier. (2012): From Concept to Implementation: Web-Based Cartographic Visualisation with CartoService. In: Murgante, B., O. Gervasi, S. Misra, N. Nedjah, A. M. A. C. Rocha, D. Taniar, and B. O. Apduhan (Ed.) Computational Science and Its Applications – ICCSA 2012. Berlin Heidelberg. Springer. 414–424.
- Asche, H., S. Stankute, M. Mueller, and F. Pietruska. (2013): Towards Developing an Integrated Quality Map Production Environment in Commercial Cartography. In: Murgante, B., S. Misra, M. Carlini, C. M. Torre, H.-Q. Nguyen, D. Taniar, B. O. Apduhan, and O. Gervasi (Ed.) Computational Science and Its Applications – ICCSA 2013. Berlin Heidelberg. Springer. 221–237.
- Berners-Lee, T., J. Hendler, and O. Lassila. (2001): The Semantic Web. Scientific American, 284 (5), 34–43.
- Bizer, C., T. Heath, and T. Berners-Lee. (2009): Linked Data The Story So Far. International Journal on Semantic Web and Information Systems, 5 (3), 1–22.
- Broder, A. (2002): A taxonomy of web search. ACM SIGIR Forum, 36 (2), 3–10.
- Bär, H. R. and R. Sieber. (1999): Towards High Standard Interactive Atlases The "GIS and Multimedia Cartography" Approach. Proceedings of the 19th International Cartographic Conference ICC 1999. Ottawa.
- Ceri, S., A. Bozzon, M. Brambilla, E. D. Valle, P. Fraternali, and S. Quarteroni. (2013): Publishing Data on the Web. In: Carey, M. and S. Ceri (Ed.) Web Information Retrieval. Berlin Heidelberg. Springer. 137–159.
- Crampton, J. W. (2009): Cartography: maps 2.0. Progress in Human Geography, 33 (1), 91–100.
- Crampton, J. W. (2002): Interactivity Types in Geographic Visualization. Cartography and Geographic Information Science, 29 (2), 85–98.
- Csaba, V. (2013): Crowdsourced Semantics with Semantic Tagging. Proceedings of the 1st International Workshop on Crowdsourcing the Semantic Web. Sydney.

- Dent, B. D., J. Torguson, and T. W Hodler. (2008): Cartography Thematic Map Design. New York: McGraw-Hill Higher Education, P. 368.
- Dietze, L and A Zipf. (2007): Extending OGC Styled Layer Descriptor (SLD) for Thematic Cartography. Proceedings of the 4th International Symposium on LBS & TeleCartography. Hong Kong.
- Engemaier, R. and H. Asche. (2011): CartoService: A Web Service Framework for Quality On-Demand Geovisualisation. In: Murgante, B., O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan (Ed.) Computational Science and Its Applications - ICCSA 2011. Berlin Heidelberg. Springer. 329–341.
- Farazi, F., V. Maltese, B. Dutta, A. Ivanyukovich, and V. Rizzi. (2013): A semantic geocatalogue for a local administration. Artificial Intelligence Review, 40 (2), 193–212.
- Fleuren, T. and P. Müller. (2008): BPEL Workflows Combining Standard OGC Web Services and Grid-enabled OGC Web Services. Proceedings of the 34th Euromicro Conference Software Engineering and Advanced Applications. Parma.
- Hake, G., D. Grünreich, and L. Meng. (2002): Kartographie: Visualisierung raum-zeitlicher Informationen. Berlin: Walter de Gruyter & Co, P. 624.
- Haklay, M., A. Singleton, and C. Parker. (2008): Web Mapping 2.0: The Neogeography of the GeoWeb. Geography Compass, 2 (6), 2011–2039.
- Hardy, P. and T. Kressmann. (2005): Cartography, Database and GIS: Not Enemies, but Allies! A Coruña.
- Hudson-Smith, A., A. Crooks, M. Gibin, R. Milton, and M. Batty. (2009): NeoGeography and Web 2.0: concepts, tools and applications. Journal of Location Based Services, 3 (2), 118–145.
- Hurni, L. and E. Hutzler. (2008): Implementation kartographischer Funktionen als Adobe Illustrator[™]-Plugins. KN Kartographische Nachrichten, 58 (5), 248–255.
- ITU. (2013): ITU Internet Statistics. http://www.itu.int/en/ITU-D/Statistics/Pages/ stat/default.aspx (visited on 12/12/2013).
- Iosifescu-Enescu, I. (2011): Cartographic Web Services. Dissertation, ETH Zürich, Zürich, Switzerland.
- Kiehle, C., K. Greve, and C. Heier. (2006): Standardized Geoprocessing- Taking Spatial Data Infrastructures one Step Further. Proceedings of the 9th AGILE Conference on Geographic Information Science. Visegrád.
- Kohlborn, T. and M. L. Rosa. (2012): SOA Approaches. In: Barros, A. and D. Oberle (Ed.) Handbook of Service Description. New York. Springer. 111–133.

- Kramers, E. R. (2008): Interaction with Maps on the Internet A User Centred Design Approach for The Atlas of Canada. The Cartographic Journal, 45 (2), 98–107.
- Lienert, C., B. Jenny, O. Schnabel, and L. Hurni. (2012): Current Trends in Vector-Based Internet Mapping: A Technical Review. In: Peterson, M. P. (Ed.) Online Maps with APIs and WebServices. Berlin Heidelberg. Springer. 23–36.
- MacEachren, A. M. and D. R. F Taylor. (1994): Visualization in Modern Cartography. Oxford, UK: Pergamon, P. 368.
- Norman, D. A. (2002): The Design of Everyday Things. New York: Basic Books, P. 192.
- Perkins, B. (2010): Have you mapped your data today? http://www.computerworld.com/ s/article/350588/Have_You_Mapped_Your_Data_Today_ (visited on 12/09/2013).
- Persson, D., D. G. Gartner, and D. M. Buchroithner. (2006): Towards a Typology of Interactivity Functions for Visual Map Exploration. In: Stefanakis, D. E., D. M. P. Peterson, D. C. Armenakis, and D. V. Delis (Ed.) Geographic Hypermedia. Berlin Heidelberg. Springer. 275–292.
- Peterson, M. P. (2008): Trends in Internet and Ubiquitous Cartography. Cartographic Perspectives, 61, 36–49.
- Peterson, M. (2003): Maps and the Internet: An Introduction. In: Peterson, M. (Ed.) Maps and the Internet. Oxford. Elsevier Science. 1–16.
- Rautenbach, V., S. Coetzee, and A. Iwaniak. (2013): Orchestrating OGC web services to produce thematic maps in a spatial information infrastructure. Computers, Environment and Urban Systems, 37, 107–120.
- Roth, R. E. (2012): Cartographic Interaction Primitives: Framework and Synthesis. The Cartographic Journal, 49 (4), 376–395.
- Sancho-Jiménez, G., R. Béjar, M. A. Latre, and P. R. Muro-Medrano. (2008): A Method to Derivate SOAP Interfaces and WSDL Metadata from the OGC Web Processing Service Mandatory Interfaces. In: Song, I.-Y., M. Piattini, Y.-P. P. Chen, S. Hartmann, F. Grandi, J. Trujillo, A. L. Opdahl, F. Ferri, P. Grifoni, M. C. Caschera, C. Rolland, C. Woo, C. Salinesi, E. Zimányi, C. Claramunt, F. Frasincar, G.-J. Houben, and P. Thiran (Ed.) Advances in Conceptual Modeling – Challenges and Opportunities. Berlin Heidelberg. Springer. 375–384.
- Schnabel, O. (2007): Benutzerdefinierte Diagrammsignaturen in Karten Konzepte, Formalisierung und Implementationen. Dissertation, ETH Zürich, Zürich, Switzerland.
- Schnabel, O. and L. Hurni. (2009): Cartographic Web Applications Developments and Trends. Proceedings of the 24th International Cartographic Conference. Santiago de Chile.

- Sieber, R., R. Geisthövel, and H. Lorenz. (2009): Atlas of Switzerland 3 A Decade of Exploring Interactive Atlas Cartography. Proceedings of the 24th International Cartographic Conference. Santiago de Chile.
- Sieber, R., L. Hollenstein, B. Odden, and H. Lorenz. (2011): From Classic Atlas Design to Collaborative Platforms – The SwissAtlasPlatform Project. Proceedings of the 25th International Cartographic Conference. Paris.
- Slocum, T. A., R. B. McMaster, F. C. Kessler, and H. H. Howard. (2008): Thematic Cartography and Geovisualization. Upper Saddle River, NJ: Pearson Prentice Hall, P. 576.
- Spence, C., J. Devoys, and S. Chahal. (2009): Architecting Software as a Service for the Enterprise. IT@Intel White Paper. http://www.intelcore.co.hu/content/dam/ doc/white-paper/cloud-computing-intel-it-architecting-software-as-aservice-paper.pdf (visited on 12/12/2013).
- Supavetch, S. and S. Chunithipaisan. (2011): Interface Independent Geospatial Services Orchestration. Information Technology Journal, 10 (6), 1126–1137.
- Turner, A. (2006): Introduction to Neogeography. Sebastopol, CA: O'Reilly, P. 54.
A. Appendices

A.1. Third-party Libraries and Software

	Table A.1.: Software		
Software	Version	License	Description
Node.js	0.11.0	MIT	Server software
http://nodejs.org PostgreSQL http://www.postgresql.org	9.1.9	PostgreSQL (MIT)	Database software
PostGIS http://postgis.net/	2.0.3	GPL Version 2	Geographic functions and objects for PostgreSQL
GDAL http://www.gdal.org	1.9.0	MIT	Geospatial Data Abstraction Library

Table A.2.: Server-side Libraries

Bibliothek	Version	Lizenz	Beschreibung
Grunt http://gruntjs.com	0.4.1	MIT	Taskrunner, provide production environment
Grunt-* https://github.com/gruntjs	*	MIT	Various Grunt Tools, see Web page
Express http://expressjs.com	3.4.4	MIT	Framework for Node.js
node-postgres https://github.com/brianc/ node-postgres	2.8.2	MIT	Library to connect with PostgreSQL

14	bic 11.0 OI	ient side Libia	1105
Bibliothek	Version	Lizenz	Beschreibung
AngularJS http://angularjs.org	1.2.0	MIT	Application Framework (MVVM)
Angular UI ng-grid http://angular-ui. github.io/ng-grid/	0.0.2	MIT	Data grid component for AngularJS
Angular UI Sortable https://github.com/ angular-ui/ui-sortable	0.0.1	MIT	User-interface component for AngularJS
Angular UI Select2 https://github.com/ angular-ui/ui-select2	0.0.2	MIT	User-interface component for AngularJS
Select2 http://ivaynberg.github. io/select2	3.4.4	GPL Version 2	Library for select inputs
Bootstrap http://getbootstrap.com/	3.0.1	MIT Version 2	User-interface framework
OpenLayers http://ol3js.org/	3.0.0- beta	BSD 2-clause	Library for map display
jQuery http://jquery.com	2.0.3	MIT	Library for DOM manipulation
jQuery UI http://jqueryui.com	1.10.3	MIT	User-interface components for jQuery
jQuery CSV https://github.com/ mirlord/jquery-csv	0.7.0	MIT	Library for CSV conversion
JSON-stat https://github.com/ badosa/JSON-stat	0.6.1	Apache Version 2	Library for JSON-stat conversion
X2JS http://code.google.com/ p/x2js/	1.1.3	Apache Version 2	Library for XML conversion
es5-shim https://github.com/ kriskowal/es5-shim	2.1.0	MIT	Cross-browser EcmaScript 5 support
json3 http://bestiejs.github. io/json3	3.2.5	MIT	Cross-browser JSON implementation

Table A.3.: Client-side Libraries

A.2. Server-side API

Host Url	Request method	Parameter
/SAS/analyzeGeometry	POST	method=centroid
/SAS/measure	POST	method=area targetColumn=(Column Name)
/CS/classifyData	POST	method=quantile attributeColumn=(Column Name) classCount=(Integer number)
/CPS/executeMethod	POST	method=dotFromArea attributeColumn=(Column Name) dotValue=(Float number) minDotDistance=(Float number)
/TMS/saveMapDescription	POST	mapId=(Map Id)
$/\mathrm{TMS/getMapDescription}$	GET	mapId=(Map Id)
SAS Spatial Analysis Servi CPS Cartographic Procedu	ce re Service	CS Classification Service TMS Thematic Map Service

A.3. Service Chain Objects and Map Description

```
Listing A.1: Service Chain Object - Data Import
```

```
1
        'inputService': {
\mathbf{2}
3
            'sourceId': 'Source Id',
            'name': 'Source Name',
4
            'desc': 'Source Description',
5
            'type' :'local'
\mathbf{6}
                              ,
            'param': {
7
                'files': {
8
                     '1':{
9
10
                         'layerId': 'Layer Id',
                         'type': 'Layer Type',
11
                         'name': 'Layer Name',
12
13
                         'path': 'File Path',
                         'fileType': 'File Type'
14
15
                    }
16
17
18
        },
        'config':
19
20
21
            'layer':'Layer Id'
22
23
```

The service chain object for data import includes an input service and a configuration to select a specific data set. In the definition in Listing A.1 a local input service is specified, whereby the data sets are defined in a file object which includes the file path.

```
Listing A.2: Service Chain Object - Data Integration
```

```
{
1
       'mappingTable' :{
\mathbf{2}
           'layers' :{
3
               '(Layer Id)' :{
4
                   'layerId' :'Layer Id',
5
                   'column' :'Column Id',
6
                   'columnNewTable' :['Column Id','Column Id','...']
7
               },
8
               '(Layer Id)' :{
9
                   'layerId' :'Layer Id',
10
                   'column' :'Layer Id',
11
                   'columnNewTable' :['Column Id','Column Id','...']
12
13
               }
14
15
       },
16
       'layerName' :'New Layer Name',
       'layerId' :'New Layer Id'
17
18
   ł
```

For data integration the service chain object (Listing A.2) has to specify the new layer name and id. In addition, the *mappingTable* object declares the data layers for the integration.

Listing A.3: Service Chain Object - Data Manipulation

```
{
1
       'layerId' :'Layer Id',
\mathbf{2}
       'action' :'/,*,+, or -',
3
       'config' :{
4
           'column1' :'Column Id',
5
           'column2' :'Column Id',
6
           'targetColumn' :'New Column Id',
7
           'targetColumnName' :'New Column Name'
8
9
10
   }
```

With the service chain object for basic data manipulation (Listing A.3) arithmetic operations can be applied to the columns of a data layer. Therefore, the object has to declare the layer id, the arithmetic operation, and the involved columns.

```
1
   {
        'processingService' :
\mathbf{2}
3
            'type' :'Service Type',
4
            'serviceId' :'Service Id',
5
            'name': 'Service Name',
6
            'url' :'Service URL',
7
            'methods' :
8
9
10
                '(Method Id)' :{
11
                    'methodId' :'Method Id',
                    'operation' :'Operation Name',
12
                    'method' :'Method Name',
13
                    'methodDesc' :'Method Description'
14
15
16
17
        },
        'config': {
18
            'methodId' :'Methode Id',
19
            'requestData' :{
20
21
                'layersId' :['Layer Id']
            },
22
            'requestParam' :{
23
                'param1' :'Parameter 1',
24
                '...' <mark>:</mark>'...'
25
26
            }
27
28
```

Listing A.4: Service Chain Object - Processing Service

The service chain object to perform service requests is given in Listing A.4. Thereby, the processing service and the available methods are specified in the *processingService* object. The configuration object includes the method id, the id of the layer for processing, and the parameters.

```
1
   {
        'symbologyRepository' :{
\mathbf{2}
            'sourceId': 'Repository Id',
3
            'name': 'Repository Name',
4
            'desc': 'Description',
5
            'type' :'local' ,
\mathbf{6}
            'param': {
7
                'symbologyDescriptions': {
8
                    '1':{
9
10
                        'symbologyId': 'Symbology Description Id',
                        'path': 'File Path'
11
12
13
14
15
       },
16
        'config' :{
            'layerId' :'Layer Id',
17
            'symbologyId': 'Symbology Description Id',
18
            'type' :'Symbolizer Type',
19
            'columns' :['Column Id','Column Id','...']
20
21
       }
22
```

Listing A.5: Service Chain Object - Symbology Description

The symbology for a data layer is specified with the service chain object presented in Listing A.5. The object declares a local symbology repository with a symbology description referenced by a file path. The configuration object specifies the layer for symbolization, the id of the symbology description, the symbolizer type, and the columns for a variable symbolizer.

```
{
1
        'mapInfo' :{
\mathbf{2}
            'title' :'Map Title',
3
            'imprint' :'Imprint Text',
4
            'id' :'Map Id',
5
            'map' :{
\mathbf{6}
                 'zoom' : 'Map Zoom',
7
                 'center' :{
8
                     'lat' :'Latitude',
9
10
                     'lng' :'Longitude'
11
12
            },
             'layer' :[
13
14
                     'id' :'Layer Id',
15
16
                     'type': 'Layer Type',
                     'visibility': 'Boolean'
17
18
                 },
19
                     '...' :'...'
20
21
            ]
22
23
        },
        'serviceChain' :[
24
25
            {
                 'id' :'Service Chain Object Id',
26
                 'type' :'Service Chain Object Type',
27
                 'config' :'Service Chain Object'
28
29
30
            ł
                 '...' <mark>:</mark>'...'
31
32
        ]
33
34
```

```
Listing A.6: Map Description Object
```

The map description object (Listing A.6) includes the map information and the service chain. The *mapInfo* object thereby defines meta informations, the map extent, and the map layers. To derive the layer data, the *serviceChain* object defines the map creation process.

A.4. Symbology Descriptions for Example Maps

```
Listing A.7: Symbology Description Choropleth Map
```

```
1
     'pointSymbolizer' :{},
\mathbf{2}
     'lineSymbolizer' :{},
3
     'polygonSymbolizer' :{
4
       '1':{
                                               // Polygon Symbolizer
5
         'symbologyId' :'1',
                                               // with the name 'Sequential'
\mathbf{6}
         'name' :'Sequential',
7
         'styleType' :'json',
8
         'style' :{
                                               // Global Symbolizer style
9
           'polygon-fill' :'#ffffff',
10
           'polygon-opacity' :0,
11
           'line-color' :'#5cb85c',
12
13
           'line-width' :1
14
         },
         'variableSymbology' :[
                                                // Variable Symbolizer style
15
                                                // and metadata
16
           ł
             'columnType' :'nominal',
17
             'maxValues' :5,
18
             'minValues' :2,
19
20
             'styles' :{
21
               '1' :{'polygon-fill' :'#FFFFD4'},
22
               '2' :{'polygon-fill' :'#FED98E'},
               '3' :{ 'polygon-fill' :'#FE9929' },
23
               '4' :{'polygon-fill' :'#D95F0E'},
24
               '5' :{'polygon-fill' :'#993404'}
25
26
27
         ]
28
29
30
     },
     'labelSymbolizer' :{}
31
32
   ł
```

```
Listing A.8: Symbology Description Dot Map
```

```
1
   ł
     'pointSymbolizer' :{
\mathbf{2}
       '1':{
                                                // Point Symbolizer
3
         'symbologyId' :1,
4
         'name' :'Dot Map Small',
5
         'styleType' :'json',
\mathbf{6}
         'style' :{
7
           'marker-type' :'ellipse',
8
           'marker-fill' :'#570387',
9
10
           'marker-fill-opacity' :'0',
           'marker-width' :'4',
11
           'marker-height' :'4'
12
13
14
15
     },
16
     'lineSymbolizer' :{},
     'polygonSymbolizer' :{
17
       '1':{
                                                // Polygon Symbolizer
18
         'symbologyId' :1,
19
         'name' :'Light Boarder',
20
21
         'styleType' :'json',
22
         'style' :{
           'line-color' :'#000000',
23
           'line-width' :2,
24
           'line-opacity' :0.8
25
26
         ł
27
       }
28
     },
29
     'labelSymbolizer' :{}
30
```

```
1
     'pointSymbolizer' :{
2
3
       '1':{
                                                // Point Symbolizer
          'symbologyId' :1,
                                                // DiaML
4
          'name' :'Two Values Grouped',
5
          'styleType' :'diaML',
\mathbf{6}
          'diaML' :{
7
            'path' :'Symbology/DiaML/pie.xml',
8
           'type' :'diagram'
9
10
         },
          'variableSymbology' :[
11
                                                 // Variable Symbolizer metadata
12
              'columnType' :'metric',
13
             'diaMLRef' :'column1'
14
15
16
             'columnType' :'metric',
17
              'diaMLRef' :'column2'
18
19
20
              'columnType' :'metric',
21
             'diaMLRef' :'column3'
22
23
24
              'columnType' :'metric',
25
             'diaMLRef' :'column4'
26
27
28
         ٦
29
30
     'lineSymbolizer' :{},
31
     'polygonSymbolizer' :{},
32
     'labelSymbolizer' :{}
33
34
   ł
```

```
Listing A.9: Symbology Description Proportional Symbol Map
```

Listing A.10: DiaML Pie Chart Symbolizer

```
<symbol xmlns='http://www.carto.net/schnabel/mapsymbolbrewer'</pre>
1
          xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
\mathbf{2}
          xsi:schemaLocation='http://www.carto.net/schnabel/mapsymbolbrewer
3
          http://www.carto.net/schnabel/mapsymbolbrewer/schemas/diaml.xsd'>
4
      <style id='s0'>
\mathbf{5}
          <fill>#CC7D7D</fill>
6
7
          <fill-opacity unit='percent'>80</fill-opacity>
          <stroke>#333333</stroke>
8
          <stroke-width unit='pixel'>2</stroke-width>
9
```

10	
11	<style id="s1"></td></tr><tr><td>12</td><td><fill>#7DCC7D</fill></td></tr><tr><td>13</td><td><fill-opacity unit='percent'>80</fill-opacity></td></tr><tr><td>14</td><td><stroke>#333333</stroke></td></tr><tr><td>15</td><td><pre><stroke-width unit='pixel'>2</stroke-width></pre></td></tr><tr><td>16</td><td></style>
17	<primitive id="p0"></primitive>
18	<pre><sector proportional="area"></sector></pre>
19	<r scale="partSum">20,25</r>
20	<pre><angle scale="dataValue">40,140,70,110</angle></pre>
21	
22	
23	<pre><diagram <="" pre="" rotation="0" translationx="0" translationy="0"></diagram></pre>
24	<pre>minSize='10' maxSize='70' areaRatio='10'></pre>
25	<pre><diagramarrangement></diagramarrangement></pre>
26	<polar></polar>
27	<pre><groups>2</groups></pre>
28	<centerdistance>3</centerdistance>
29	<totalangle unit="degree">360</totalangle>
30	
31	
32	<pre><diagramrelation></diagramrelation></pre>
33	<pre><dataref>column1</dataref></pre>
34	<primitiveref>p0</primitiveref>
35	<styleref>s0</styleref>
36	
37	<pre><diagramrelation></diagramrelation></pre>
38	<pre><dataref>column2</dataref></pre>
39	<primitiveref>p0</primitiveref>
40	<styleref>s1</styleref>
41	
42	<pre><diagramrelation></diagramrelation></pre>
43	<pre><dataref>column3</dataref></pre>
44	<primitiveref>p0</primitiveref>
45	<styleref>s1</styleref>
46	
47	<pre><diagramrelation></diagramrelation></pre>
48	<pre><dataref>column4</dataref></pre>
49	<primitiveref>p0</primitiveref>
50	<styleref>s0</styleref>
51	
52	<labeldata>no</labeldata>
53	
54	
54	

B. Declaration of Originality

I hereby declare that the written work I have submitted entitled

Service-oriented Architecture for Thematic Cartography on the Web

is original work which I alone have authored and which is written in my own words.

Author	
Last name	First name
Krimbacher	Andreas
Supervisor	
Last name	First name
Hurni	Lorenz
Supervisor	
Last name	First name
Bär	Hans-Rudolf
Iosifescu Enescu	Ionut
Straub	Florian

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette'¹. The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zürich, January 27, 2014 Place and Date

Signature

¹http://www.ethz.ch/students/exams/plagiarism_s_en.pdf