Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Pyry Kröger

Visualizing Geographical Data on the Web

Reducing the work needed by eliminating boilerplate

Master's Thesis
Espoo, May 28, 2014

DRAFT! — August 9, 2014 — DRAFT!

Supervisor: Professor Petri Vuorimaa
Instructor: N.N. M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Pyry Kröger		
Title:			
Visualizing Geographical Data on the Web - Reducing the work needed by eliminating boilerplate			
Date:	May 28, 2014	**Pages:**	vii + 27
Professorship:	Media	**Code:**	T-111
Supervisor:	Professor Petri Vuorimaa		
Instructor:	N.N. M.Sc. (Tech.)		

A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.

!FIXME **Abstract text goes here (and this is an example how to use fixme).** FIXME! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom `mydraft` mode, text parts tagged with fixmes are shown in bold and with fixme tags around them. When compiled in normal mode, the fixme-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom `mydraft` mode is selected by the `mydraft` option given for the package `aalto-thesis`, near the top of the `thesis-example.tex` file.

The thesis example file (`thesis-example.tex`), all the chapter content files (`1introduction.tex` and so on), and the Aalto style file (`aalto-thesis.sty`) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!

Keywords:	work, in, progress
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan tutkinto-ohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Pyry Kröger		
Työn nimi: Geografisen datan visualisointi webissä - Tarvittavan boilerplate-koodin vähentäminen			
Päiväys:	28. toukokuuta 2014	**Sivumäärä:**	vii + 27
Professuuri:	Media	**Koodi:**	T-111
Valvoja:	Professori Petri Vuorimaa		
Ohjaaja:	Diplomi-insinööri N.N.		

Kivi on materiaali, joka muodostuu mineraaleista ja luokitellaan mineraalisisältönsä mukaan. Kivet luokitellaan yleensä ne muodostaneiden prosessien mukaan magmakiviin, sedimenttikiviin ja metamorfisiin kiviin. Magmakivet ovat muodostuneet kiteytyneestä magmasta, sedimenttikivet vanhempien kivilajien rapautuessa ja muodostaessa iskostuneita yhdisteitä, metamorfiset kivet taas kun magma- ja sedimenttikivet joutuvat syvällä maan kuoressa lämpötilan ja kovan paineen alaiseksi.

Kivi on epäorgaaninen eli elottoman luonnon aine, mikä tarkoittaa ettei se sisällä hiiltä tai muita elollisen orgaanisen luonnon aineita. Niinpä kivestä tehdyt esineet säilyvät maaperässä tuhansien vuosien ajan mätänemättä. Kun orgaaninen materiaali jättää jälkensä kiveen, tulos tunnetaan nimellä fossiili.

Suomen peruskallio on suurimmaksi osaksi graniittia, gneissiä ja Kaakkois-Suomessa rapakiveä.

Kiveä käytetään teollisuudessa moniin eri tarkoituksiin, kuten keittiötasoihin. Kivi on materiaalina kalliimpaa mutta kestävämpää kuin esimerkiksi puu.

Asiasanat:	vähän, vielä, kesken
Kieli:	Englanti

Acknowledgements

So long, and thanks for all the fish.

Espoo, May 28, 2014

Pyry Kröger

Abbreviations and Acronyms

API	Application Programming Interface
POI	Point of Interest; a piece of data with geospatial dimension
GIS	Geographic Information System

Contents

Chapter 1

Introduction

The introduction in itself is rarely very long; two to five pages often suffice.

!Fixme **Here maybe something related to why I chose this topic, i.e. why it is beneficial for me/others to have a framework that eases the map visualization process.** Fixme!

Geographical data is data with geospatial dimension, such as POI with location data as coordinates. The most natural method for visualizing geographical data is usually with various maps. In the past, geographical data was predominantly visualized by cartographers, but it has been recognized (Kraak and MacEachren, 1999) that the situation has changed, with people from increasing number of fields having a need for visualizing geographical data. Moreover, the popularity of Google Maps (Google, 2005a) along with its API (Google, 2005b) has proved that in addition to experts of other academic fields, there is a definite demand for web map visualizations within consumers as well.

The web makes publishing and bundling map visualizations extraordinarily straightforward when compared to traditional desktop-based GIS applications, which is especially important when the visualizations are made by non-cartographers !Fixme **find a reference for this maybe** Fixme!. However, as the web platform is primarily designed for static documents instead of dynamic applications (Berners-Lee, 1989; Berners-Lee et al., 1992), there are some additional concerns to address when making a complex data visualization on the web.

!Fixme **Why geodata should be visualized? Kraak and Ormeling (2011, chap. 1.1) or Bartz Petchenik (1979) should help.** Fixme!

!Fixme **Needs more flesh** Fixme!

1.1 Problem Statement

Currently, there are several libraries available for displaying maps and simple visualizations !Fixme **add references to the libraries. GMaps, Leaflet, OpenLayers etc.** Fixme!. However, none of the mainstream libraries is of sufficiently high abstraction level for building map visualizations effectively, resulting in the need for writing *boilerplate* code that does not directly contribute to the visualization. Moreover, the libraries are not designed primarily for visualizations and therefore do not encourage or push the visualizer to create visually and cognitively effective visualizations.

We plan to evaluate means to make creating map visualizations for the web more efficient by building a higher abstraction level software framework for map visualizations. This framework should provide the structure for creating the visualization as well as common web application features needed in modern web applications.

In order to find the solution for the problem, it is necessary to study geographical visualizations and software frameworks. The process of making geographical visualizations should be studied to ensure that the framework enables creating *effective* visualizations. In addition, software frameworks and software reuse should be studied in order to be able to create visualizations *efficiently*. Therefore, we select the following research questions for this thesis:

RQ1 How to build effective geographical visualizations in the web?

RQ2 How to build a web software framework which enables creating effective geographical visualizations efficiently?

!Fixme **Would it make sense to drop the RQ1 since RQ2 pretty much covers it? Or rephrase the questions to overlap less?** Fixme!

1.2 Structure of the Thesis

Chapter 1 (this introduction) presents the motivation for this thesis as well as the problem statement. Chapter 2 describes the background of the work. In particular, the chapter describes how to visualize geographical data and the essence of web software frameworks. Chapter 3 presents the web technology, standards and other needed material for building a web visualization as well as describing some existing map visualizations !Fixme **Maybe rephrase** Fixme!.

In chapter 4, we discuss the methods used to examine the problem and evaluate solutions we will propose. In chapter 5, we describe the methods used to solve the problem. In chapter 6, we evaluate the implementation and its results. !Fixme **Chapters 7 and 8 missing. Also, maybe elaborate description about chapters 4-6 a bit** Fixme!

!Fixme **After writing each chapter, check the description in this section** Fixme!

Chapter 2

Background

Also known as “literature review”/“Kirjallisuuskatsaus”. About 20 pages long.

In order to build an efficient framework for visualizing geographical data, it is needed to study (a) how to visualize geodata and (b) how to build frameworks. We are going to tackle this problem by first studying the basics of data visualization with an emphasis on geographical data, maps and the visualization process. After visualization, we are going to study the essence of software reuse, focusing on building and evaluating reusable software, also known as software frameworks.

2.1 Data Visualization

2.1.1 Definition

According to Kosara (2007, chap. 3), there is no universally accepted definition of visualization. He proposes the following for a “minimal set of requirements for any visualization”:

- It is based on (non-visual) data
- It produces an image
- The results are readable and recognizable

According to him, while visualizations can also have other properties or qualities, such as interaction or visual efficiency, the requirements above are the ones needed for technical definition of the term. Moreover, it should be emphasized that according to this definition, visualization is the *process* itself, not the result of it.

Kosara (2007, chap. 4) argues that visualization is separated into two types, *pragmatic* and *artistic* visualization. Pragmatic visualization focuses on the analysis of the data in order to show its relevant characteristics as efficiently as possible. Artistic visualization on the other hand concentrates on the communication of the concern behind the data, not the display of the actual data. Kosara states that while these types focus on the opposite sides of the visualization spectrum, it may be possible to close the gap using e.g. interaction.

The first requirement for visualizations by Kosara (2007) dictates that the visualization is based on data. This is an essential characteristic of *data* visualizations: the visualization is a function which takes data as an input and produces a visual object as an output. In less technical terms, this means that the visualization turns data into visual, effortlessly and efficiently digestible format.

We argue that this leads to the fact that the data and visualization are not inherently tied to each other; the visualization "function" can be independent of the data and thus it may be possible to create a visualization framework or platform which is able to function on a potentially wide range of data.

!FIXME **The goal of visualization is (usually) better understanding of the data.** FIXME!

2.1.2 Principles for Successful Data Visualization

The requirements presented in the previous section are sufficient for the definition of data visualization. However, they do not convey any information about visualization quality. In order to discover the characteristics for successful data visualization, additional principles are needed. Tufte (1986, p. 13) states that excellent graphics (i.e. results of visualizations) consist of "complex ideas communicated with clarity, precision and efficiency". In practice, this means that the graphics should emphasize the actual data and its nuances above everything else, while serving a clear purpose.

In addition to graphics principles presented in the previous paragraph, Tufte (1986, p. 93) presents the concept of *data-ink*. Data-ink represents the ink used for displaying the data in a visualization. He argues that in an excellent visualization, most, if not all, ink used should contribute to display of the data. However, research by Inbar et al. (2007) suggests that maximizing the share of data-ink may not be beneficial to the user experience of the visualization.

The principles presented above are essential, but too abstract in order to be used as a sole basis for defining a good visualization. However, when combined with the data visualization definition stated above, the principles become considerably more useful and concrete. Azzam and Evergreen (2013) propose an adapted version of the definition by Kosara (2007), complementing the second requirement by requiring the produced image to represent the data truthfully. We argue that this definition in effect combines the definition by Kosara (2007) with the principle of showing data introduced by Tufte (1986). The adapted definition facilitates the process of creating a successful data visualizations by offering a more concrete version of Tufte's principles. It gives the developer of the visualization a concrete checklist for representing the data: make sure it does not (a) omit or (b) overrepresent any information (Azzam and Evergreen, 2013).

!FIXME **If desperately in need of more background, add human perception in relation to information visualization (from Ware).** FIXME!

2.1.3 Visualizing Geographical Data

!FIXME **Visualization - Scientific Visualization - Map Visualization - Kraak (1998) has a good overview.** FIXME!

The most natural way of visualizing geographical data is by using a map(Kraak, 1998; Kraak and Ormeling, 2011, chap. 1). This technique is called *thematic mapping* (Slocum and McMaster, 2014, chap. 1). Thematic mapping does not require any specific format of data, except for the geographical dimension (Kraak and Ormeling, 2011, chap. 1). However, the nature of the data has a great effect on the method, or type, of thematic mapping.

2.1.3.1 Methods for Thematic Mapping

As stated above, there are several types of geographical data, many of which are fundamentally different requiring different visualization methods. Therefore, several different thematic mapping methods have been developed. Slocum and McMaster (2014, chap. 14-18) list some of the most typical ones: !Fixme **rephrase the descriptions below** Fixme!

- Choropleth map - Shows data aggregated for a set of predefined areas (countries, regions etc)
- Isarithmic map - Maps with areas separated by contour lines.
- Dasymetric map - !Fixme **what's this?** Fixme!
- Proportional symbol map - like dot map, but replaces dots with relevant symbols of sizes proportional to the data
- Dot map - simple maps with dots on relevant locations
- Multivariate mapping - A map that shows data of several dimensions (in addition to location).
- Flow map - a map that shows "flows" from one area to another. Napoleon Russian campaign map.

When designing a software framework for geovisualization, it is not necessary to support every method above. However, as those are some of the most used ones, omitting any must be a conscious decision.

!Fixme **describe the use cases for each type** Fixme!

Even a single thematic map is often used for multiple different purposes (Schlichtmann, 2002, chap. 2). For instance, a single map can be read on the *overall level* ("where are the primary schools located in Helsinki metropolitan area?") and *elementary level* ("is there a primary school in Punavuori?").

Furthermore, some possible uses for a thematic map are "what is the ratio and distribution of Finnish schools compared to Swedish schools in Helsinki" or "what is the spatial distribution of sizes of schools in Helsinki". Therefore, an efficient map visualization should not lock the user to any single perspective. !FIXME **maybe move somewhere? Create a separate (sub)section for interactivity?** FIXME! !FIXME **Interactivity could help here? See (Andrienko and Andrienko, 1999)** FIXME!

!FIXME **Analysis part of geovisualization - "...is considered out of the scope of this work."** FIXME!

!FIXME **To Do** FIXME!

- Geographic visualization (e.g. in relation to scientific visualization)
- Map visualization vs. map (thematic map vs general-reference map) (Bartz Petchenik, 1979)
- How do the principles introduced in the previous subsection apply to geographical data? Is there anything else to consider?
- Thematic map interactivity (Andrienko and Andrienko, 1999) (Where should this go? Is this an env related thing or here somewhere)

Tufte (1986, p. 16) May help here as well.

2.2 How Thematic Maps Are Made

Schlichtmann (2002) describes making thematic maps as a six-step process. The first four steps involve deciding on and obtaining the data which are not relevant when building a software framework for visualization. Therefore, we ignore those steps. The step five consists of selecting the visualization method and using it to produce a meaningful visualization from the data. The step six involves explaining the visualization in legend. !FIXME **Schlichtmann not concerned with the step 6, find some other source of displaying the legend.** FIXME!

In the map visualization process, several identified objectives for the resulting graphic exist (Schlichtmann, 2002). The objectives are presented in the table below.

Name	**Description**
Clarification	Making the map clear and readable. In practice, this means that the topemes (symbols) in a map should be easily detectable and distinguishable from each other
Emphasis	Making topemes and other important characteristics of the visualization to stand out visually
Types of Entries	Having a clearly distinguishable type for each topeme.
Sets of Types	Grouping data points and symbols with similar traits in order to make them belong together visually. Ideally, the visual similarity should be related to the conceptual similarity.
Cross-Relations	Visually indicating the potential relations and similarities between different types or between entries of different types.
Local Syntax	Aligning visual properties of the topemes to prevent unintentional emphasis of single topemes.
Local Ensembles	Supporting topemes with multiple properties (such as the numbers of children and adults in an area) so that the topeme visually reflect both the individual properties and the combination of all properties.
Multilocal Ensembles	Supporting topemes with multiple geographical properties (such as spatial distribution of people)
Addable and Non-Addable Quantities	Differentiating addable and non-addable properties. Typically absolute quantitative properties are addable while relative and qualitative properties are non-addable. Addable properties should be visualized in a way that cognitively supports addition (e.g. with sizes of elements) while non-addable quantities should be visualized without said feature (e.g. with colors.)

The Surface Illusion	Creating an illusion of surface on the map. This can be achieved for example by using illumination and shadowing. These visual traits can convey a meaning themselves and often naturally do so.

Table 2.1: Map visualization objectives as per Schlichtmann (2002)

The objectives above are important when visualizing geographical data on a map. Therefore, it is needed to take those into account when creating a visualization tool or framework in order to enable or even encourage the visualizers to reach as many of the objectives as possible.

!Fixme **This could use an opinion from some other source as well (Maybe (Slocum and McMaster, 2014, p. 5))** Fixme!

2.3 Software Reuse

In order to create a reusable software framework for visualization, it is necessary to study software reuse along with different reuse techniques and their characteristics, advantages and disadvantages.

Krueger (1992) presents software reuse as a process of reusing existing software code (applications, libraries, functions or single lines) when building new software. Therefore, software reuse combines often several different existing pieces of code along with new code which is specific for this application. According to Mcilroy (1969) and Boehm (1999), it is one of the most effective techniques of reducing the development time and cost of complex software products.

2.3.1 Software Reuse Advantages & Disadvantages

Advantages and disadvantages in general about reuse, not about different methods as those are described in the reuse methods chapter. Software reuse metrics research (Mohagheghi and Conradi, 2008; Frakes and Terry, 1996; Selby, 2005) probably has some good points. Also, Johnson (1997) may have something.

2.3.2 Dimensions for Software Reuse

Different reuse methods excel at different areas. (Krueger, 1992) separates making reusable software into four areas, or dimensions: abstraction, selection, specialization and integration. Different reuse methods use these dimensions to a varying degree. **Abstraction** is the process of making a piece of software more generic, thus making it applicable to a wider range of software projects. Software reuse is almost always based on abstraction, but according to (Krueger, 1992), raising the abstraction level has proven to be difficult, thus making building reusable software hardly a trivial process. **Selection** facilitates finding, comparing and choosing suitable pieces of software. **Specialization** is the process of making the abstracted component more specific, usually by parameterizing the software or making it transformable. **Integration** means providing the software with components to be reused, such as a mechanism to import relevant functions to the software.

2.3.3 Software Reuse Methods

Libraries, components, frameworks... Use Krueger (1992). Also why choose framework? Johnson (1997) has some reasons.

Krueger (1992) has a list of different methods along with their advantages and disadvantages. Includes frameworks in chapter 10. Johnson (1997) extends the description. Using the characteristics along with praise of Johnson (1997), it should be possible to reason about going with frameworks.

2.3.4 Software Frameworks

Frameworks more in depth. Maybe how to build them.

2.4 Evaluating Software Framework Effectiveness

Here should be the whole research on how to measure if it is beneficial to reuse software instead of writing from scratch. At least Mohagheghi and

Conradi (2008); Frakes and Terry (1996); Selby (2005) are useful here.

Chapter 3

Environment

This could be entirely about web tech for maps? About 5 pages long.

3.1 Related Web Technology

Describe HTML5, JS and other related technology, because these are really the things that restrict the implementation.

Visualizations need to be implemented using web technology, because it is crucial for distributability and discoverability that the system works in web browsers... !FIXME **continue...** FIXME!

3.2 Using Web Technology for Geographic visualization

Probably should describe the relevant web technology, some needed HTML5/JS features etc. !FIXME **to subsection of related webtech?** FIXME!

3.3 Current Map Visualizations

Describe the map visualizations that sparked the interest for this kind of research. FindBooze, Peruskartta and Ottoapp. Also, reason why it is unnecessarily laborious to write map visualizations every time from scratch.

!Fixme **May be moved to implementation** Fixme!

3.4 LaTeX Graphics

When you use `pdflatex` to render your thesis, you can include PDF images directly, as shown by Figure 3.1 below.

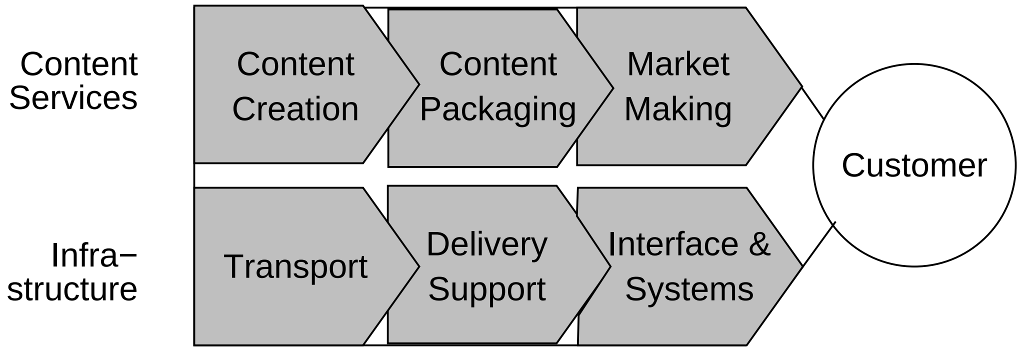
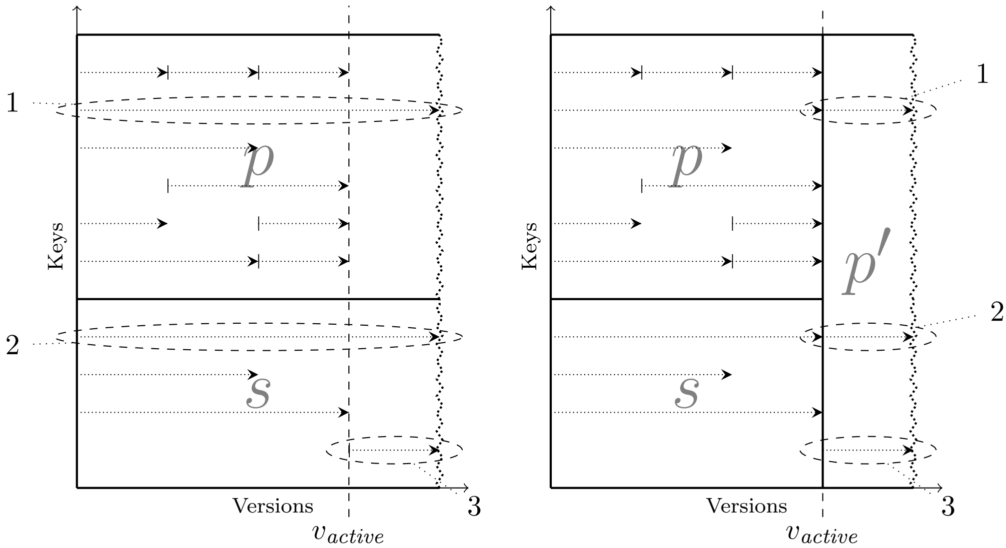


Figure 3.1: The INDICA two-layered value chain model.

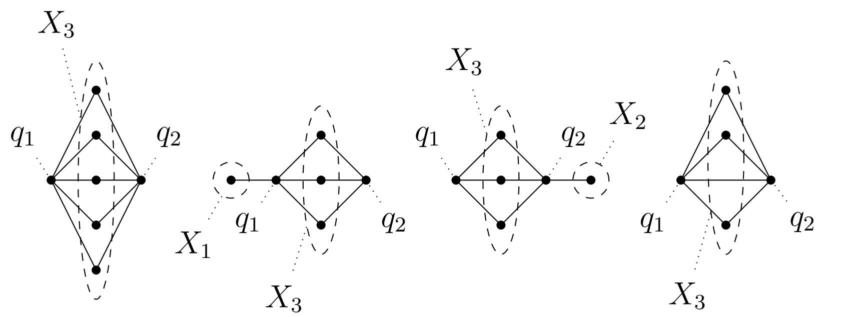
You can also include JPEG or PNG files, as shown by Figure 3.2.

Figure 3.2: Eeyore, or Ihaa, a very sad donkey.

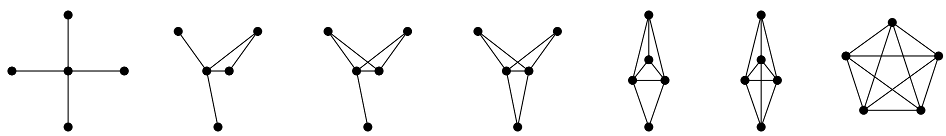


(a) Before (b) After

Figure 3.3: Example of a multiversion database page merge. This figure has been taken from the PhD thesis of Haapasalo (Haapasalo, 2010).



(a) Examples of obstruction graphs for the Ferry Problem



(b) Examples of star graphs

Figure 3.4: Examples of graphs draw with TikZ. These figures have been taken from a course report for the graph theory course (Göös et al., 2010).

Chapter 4

Methodology

About 5-10 pages long?

In this thesis, we are going to use method X (cost-benefit analysis with Y for calculating costs and benefits?) for evaluating the effect the framework has on the visualizing, because according to Doe & Smith (2010), it is applicable to this kind of problem. Also something why it is the best format for this, and while it is not perfect, it is not reasonable or feasible to accomplish more precise measurements.

For gathering data, we implemented three(?) map visualizations with and without the framework. The first visualization is a simple use case which concentrates on the display of a custom map. The second visualization highlights POI data on a map, and the third one is the most complex, consisting of POI data, relations between POIs and a custom backend serving the data with real-time updates from a mobile client. The types of visualizations were selected to obtain data about a wide variety of different map visualizations.

If you have not yet done any (real) metholodogical courses (but chosen introduction courses of different areas that are listed in the methodological courses list), now is the time to do so or at least check through material of suitable methodological courses. Good methodologial courses that consentrates especially to methods are presented in Table 4.1. Remember to explain the content of the tables (as with figures). In the table, the last column gives the research area where the methods are often used. Here we used table to give an example of tables. Abbreviations and Acronyms is also a long table.

Code	**Name**	**Methods**	**Area**
T-110.6130	Systems Engineering for Data Communications Software	Computer simulations, mathematical modeling, experimental research, data analysis, and network service business research methods, (agile method)	T-110
Mat-2.3170 Simulation (here is an example of multicolumn for tables)		Details of how to build simulations	T-110
S-38.3184	Network Traffic Measurements and Analysis	How to measure and analyse network traffic	T-110

Table 4.1: Research methodology courses

The difference is that longtables can continue to next page.

Chapter 5

Implementation

10-15 pages?

You have now explained how you are going to tackle your problem. Go do that now! Come back when the problem is solved!

Now, how did you solve the problem? Explain how you implemented your solution, be it a software component, a custom-made FPGA, a fried jelly bean, or whatever. Describe the problems you encountered with your implementation work.

Chapter 6

Evaluation

5 pages?

You have done your work, but that's[1] not enough.

You also need to evaluate how well your implementation works. The nature of the evaluation depends on your problem, your method, and your implementation that are all described in the thesis before this chapter. If you have created a program for exact-text matching, then you measure how long it takes for your implementation to search for different patterns, and compare it against the implementation that was used before. If you have designed a process for managing software projects, you perhaps interview people working with a waterfall-style management process, have them adapt your management process, and interview them again after they have worked with your process for some time. See what's changed.

The important thing is that you can evaluate your success somehow. Remember that you do not have to succeed in making something spectacular; a total implementation failure may still give grounds for a very good master's thesis—if you can analyze what went wrong and what should have been done.

[1] By the way, do *not* use shorthands like this in your text! It is not professional! Always write out all the words: "that is".

Chapter 7

Discussion

At this point, you will have some insightful thoughts on your implementation and you may have ideas on what could be done in the future. This chapter is a good place to discuss your thesis as a whole and to show your professor that you have really understood some non-trivial aspects of the methods you used…

Chapter 8

Conclusions

Time to wrap it up! Write down the most important findings from your work. Like the introduction, this chapter is not very long. Two to four pages might be a good limit.

Bibliography

Gennady L. Andrienko and Natalia V. Andrienko. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4):355–374, 1999. ISSN 1365-8816. doi: 10.1080/136588199241247. URL `http://dx.doi.org/10.1080/136588199241247`.

Tarek Azzam and Stephanie Evergreen. *J-B PE Single Issue (Program) Evaluation, Volume 139 : Data Visualization, Part 1 : New Directions for Evaluation*. John Wiley & Sons, Somerset, NJ, USA, 2013. ISBN 9781118793374. URL `http://site.ebrary.com.libproxy.aalto.fi/lib/aalto/docDetail.action?docID=10768989`.

Barbara Bartz Petchenik. From place to space: The psychological achievement of thematic mapping. *The American Cartographer*, 6(1):5–12, 1979. ISSN 0094-1689. doi: 10.1559/152304079784022763. URL `http://www.tandfonline.com/doi/abs/10.1559/152304079784022763`.

Tim Berners-Lee. Information management: A proposal, March 1989. URL `http://www.w3.org/History/1989/proposal.html`.

Tim Berners-Lee, Robert Cailliau, Jean-François Groff, and Bernd Pollermann. World-wide web: The information universe. *Internet Research*, 2(1):52–58, December 1992. ISSN 1066-2243. doi: 10.1108/eb047254. URL `http://www.emeraldinsight.com.libproxy.aalto.fi/journals.htm?articleid=1670698&show=abstract`.

B. Boehm. Managing software productivity and reuse. *Computer*, 32(9): 111–113, September 1999. ISSN 0018-9162. doi: 10.1109/2.789755.

William Frakes and Carol Terry. Software reuse: Metrics and models. *ACM Comput. Surv.*, 28(2):415–435, June 1996. ISSN 0360-0300. doi: 10.1145/234528.234531. URL `http://doi.acm.org/10.1145/234528.234531`.

Google. Maps, February 2005a. URL `http://maps.google.com`. Accessed 25.7.2014.

Google. Maps API, June 2005b. URL `http://maps.google.com`. Accessed 23.7.2014.

Mika Göös, Tuukka Haapasalo, and Leo Moisio. Finding hard instances of the ferry problem. Course report for the course T-79.5203/S-72.2420 Graph theory, Aalto SCI, 2010.

Tuukka Haapasalo. *Accessing Multiversion Data in Database Transactions*. PhD thesis, Department of Computer Science and Engineering, Aalto University School of Science and Technology, Espoo, Finland, 2010. `http://lib.tkk.fi/Diss/2010/isbn9789526033600/`.

Ohad Inbar, Noam Tractinsky, and Joachim Meyer. Minimalism in information visualization: Attitudes towards maximizing the data-ink ratio. In *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore!*, ECCE '07, page 185–188, New York, NY, USA, 2007. ACM. ISBN 978-1-84799-849-1. doi: 10.1145/1362550.1362587. URL `http://doi.acm.org/10.1145/1362550.1362587`.

Ralph E. Johnson. Frameworks=(components+ patterns). *Communications of the ACM*, 40(10):39–42, 1997. URL `http://dl.acm.org/citation.cfm?id=262799`.

R. Kosara. Visualization criticism - the missing link between information visualization and art. In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 631–636, July 2007. doi: 10.1109/IV.2007.130.

Menno-Jan Kraak. The cartographic visualization process: From presentation to exploration. *The Cartographic Journal*, 35(1):11–15, June 1998. ISSN 0008-7041. doi: 10.1179/caj.1998.35.1.11.

URL http://www.maneyonline.com.libproxy.aalto.fi/doi/abs/10.1179/caj.1998.35.1.11.

Menno-Jan Kraak and Alan MacEachren. Visualization for exploration of spatial data. *International Journal of Geographical Information Science*, 13(4):285–287, 1999. ISSN 1365-8816. doi: 10.1080/136588199241201. URL http://dx.doi.org/10.1080/136588199241201.

Menno-Jan Kraak and Ferjan Ormeling. *Cartography, Third Edition: Visualization of Spatial Data*. Guilford Press, June 2011. ISBN 9781609181949.

Charles W. Krueger. Software reuse. *ACM Comput. Surv.*, 24(2):131–183, June 1992. ISSN 0360-0300. doi: 10.1145/130844.130856. URL http://doi.acm.org/10.1145/130844.130856.

Doug Mcilroy. Mass-produced software components. pages 138–155, January 1969. URL http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF.

Parastoo Mohagheghi and Reidar Conradi. An empirical investigation of software reuse benefits in a large telecom product. *ACM Trans. Softw. Eng. Methodol.*, 17(3):13:1–13:31, June 2008. ISSN 1049-331X. doi: 10.1145/1363102.1363104. URL http://doi.acm.org/10.1145/1363102.1363104.

Hansgeorg Schlichtmann. Visualization in thematic cartography: towards a framework. *The Selected Problems of Theoretical Cartography*, page 49–61, 2002. URL http://rcswww.urz.tu-dresden.de/~wolodt/tc-com/pdf/sch_2000.pdf. Accessed 2014-07-21.

R.W. Selby. Enabling reuse-based software development of large-scale systems. *IEEE Transactions on Software Engineering*, 31(6):495–510, June 2005. ISSN 0098-5589. doi: 10.1109/TSE.2005.69.

Terry A. Slocum and Robert B. McMaster. *Thematic Cartography and Geovisualization: Pearson New International Edition*. Pearson Education, 3rd edition, 2014. ISBN 9781292055442. URL https://www.dawsonera.com/abstract/9781292055442.

Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. ISBN 0-9613921-0-X.

Appendix A

First appendix

This is the first appendix. You could put some test images or verbose data in an appendix, if there is too much data to fit in the actual text nicely.

For now, the Aalto logo variants are shown in Figure A.1.

Aalto University
School of Science

(a) In English

Aalto-yliopisto
Perustieteiden
korkeakoulu

(b) Suomeksi

Aalto-universitetet
Högskolan för
teknikvetenskaper

(c) På svenska

Figure A.1: Aalto logo variants