

Effective use of open source GIS in rural planning in South Africa

Patrik Andersson
Anders Jürisoo

Copyright © Patrik Andersson & Anders Jürisoo

Title: Effective use of open source GIS in rural planning in South Africa

Author: Patrik Andersson and Anders Jürisoo

Year of publication: 2011

Master thesis in geographic information No: 04

Department of Earth and Ecosystem Sciences

Lund University

Sölvegatan 12

223 62 Lund

Telephone: +46 (0) 46-222 30 30

Fax: +46 (0) 46- 222 03 21

Homepage: <http://www.geko.lu.se>

Printed by *Media Tryck*, 2011.

Effective use of open source GIS in rural planning in South Africa

Master of Science Thesis by:

Patrik Andersson, Master of Science program Surveying and land management, LTH, Sweden

Anders Jürisoo, Master of Science program Surveying and land management, LTH, Sweden

Supervisor:

Lars Harrie, Senior Lecturer, Earth & Ecosystem Sciences, Lund University

Dean Leigh Barnes, Enhance Strategies, South Africa

Examiner:

Ali Mansourian, Senior Lecturer, Earth & Ecosystem Sciences, Lund University

Keywords:

Web GIS, informal settlements, maps, open source. rural planning.

Nyckelord:

Web GIS, informella bosättningar, kartor, öppen källkod, landsbygdsplanering.

Abstract

The purpose of this study is to develop a method on how to collect, store and visualize geographic data in an easy and inexpensive way, and to enable a wide audience to reach the information. The focus of this study is mainly on projects that are taking place in developing countries and in environments where one usually works with limited resources. Many of the organizations that are working in these environments cannot afford commonly used GIS programs like ArcGIS to visualize their geographic data.

Organizations that are active in developing countries and are executing projects there have access to geographic data, but many of them have no knowledge on how to visualize these in a good way. For a nongovernmental organization (NGO) it could prove useful to visualize these geographic data for many purposes. One of the reasons for this is that the organizations are working with limited resources and they must be able to show results to their donors. Being able to visualize geographic data from various projects around the world could be one way to inform donors about the work progress.

This study took place in Dukuduku forest located in KwaZulu-Natal in South Africa. It has been funded by a Minor Field Study, which is a scholarship given by the Swedish International Development cooperation Agency (SIDA). We also collaborated with a local consulting company named *Enhanced Strategies*. They are project managers within the Dukuduku forest where around 50 projects, funded by the government and NGOs, are taking place. The aim has been to create a workflow that can help to visualize these projects and the progress in the area to all the concerned parts.

A workflow has been developed that describe in six steps how to gather, store and visualize geographic data. The intention with the workflow has been to enable development of cheap products that can reach a group of people as large as possible. Open source programs have been used to reduce the cost and Web GIS has been used to reach a wider audience.

The results of this study are, in addition to the workflow, three examples that are based on the workflow. The first intends to be a basic map viewer where the user can decide which layers that should be visualized. The second example describes how geographic data can be edited through a web browser and consequently updating a database remotely. The last example is intended to describe how the workflow can be used for visualize the development in one specific project.

Sammanfattning

Syftet med denna studie är att ta fram en arbetsmetod för att samla in, lagra och visualisera geografisk data på ett enkelt och billigt sätt, samt att man når en stor målgrupp med denna geografiska data. Studien riktar sig främst till arbete som pågår i utvecklingsländer och i miljöer där man jobbar med begränsade resurser. Många organisationer som jobbar under dessa förutsättningar har inte råd att använda vanliga GIS program som exempelvis ArcGIS för att visualisera den geografiska data.

Många organisationer runt om i världen har tillgång till geografisk data, men de flesta av dessa har inte kunskap om hur denna kan visualiseras på ett smidigt sätt. Det kan dock vara viktigt att visualisera denna geografiska data för exempelvis en hjälporganisation. Anledningen till detta är att många av dessa organisationer jobbar med begränsade resurser och att de har någon form av prestationskrav gentemot sina finansörer. Att kunna visualisera geografiska data från olika projekt runt om i världen skulle kunna vara ett sätt att informera berörda parter om hur arbetet går.

Denna studie har ägt rum i Dukuduku skogen som ligger i KwaZulu-Natal provinsen i Sydafrika. Den har varit finansierad av ett Minor Field Study stipendium som ges ut av SIDA. Vi har samarbetat med ett lokalt konsultbolag vid namnet *Enhanced Strategies*. De är projektledare i Dukuduku skogen där ett 50-tal projekt, finansierat av regering och hjälporganisationer, äger rum. Syftet har varit att skapa en arbetsmetod som kan hjälpa till att visualisera dessa projekt och framstegen i området till berörda parter.

En arbetsmetod har tagits fram med avsikt att vara så billig och att nå en så stor grupp människor som möjlig. För att minska kostnaderna i denna arbetsmetod har program baserat på öppenkällkod använts och för att nå en stor målgrupp har program inom webb baserat GIS använts i så stor utsträckning som möjligt. Den framtagna arbetsmetoden beskriver i sex olika steg hur man kan hämta, lagra och visualisera geografisk data på en hemsida.

Resultatet av denna studie är, förutom den framtagna arbetsmetoden, tre stycken exempel där den framtagna arbetsmetoden används. Den första avser att vara en allmän karta över området där användaren själv väljer vad som ska visualiseras. Det andra exemplet beskriver hur geografisk data kan editeras över webben och då följdaktningen uppdatera en databas på distans. Det sista exemplet avser att beskriva hur arbetsmetoden kan användas för att visa framstegen i ett specifikt projekt.

Foreword

This master thesis is produced as a part of the program *Surveying and land management* at Lund university's faculty of engineering, LTH. The work has been done in cooperation with Enhance Strategies, Durban, South Africa, and the department of Earth and Ecosystem Sciences at Lund University. The financing for the project was provided by SIDA, the Swedish international development cooperation agency.

In the process of the study we have received invaluable help from several people and organizations. First of all we would like to thank our supervisors Lars Harrie representing Lund University and Dean Barnes representing Enhance Strategies. Expressions of gratitude are also sent to MHP Geospace, African Conservation Trust and the people of Khula Village.

Patrik Andersson and Anders Jürisoo
Lund, Sweden
September 2011

Table of Contents

Effective use of open source GIS in rural planning in South Africa.....	iii
Foreword.....	xi
Table of Contents.....	xiii
1 Introduction.....	1
1.1 Background.....	1
1.2 Purpose.....	2
1.3 Research problem.....	2
1.4 Method.....	2
1.5 Selection and limitations.....	3
1.6 Disposition.....	3
2 Development projects in the Dukuduku forest, South Africa.....	4
2.1 Study area.....	4
2.2 Projects in Dukuduku forest.....	5
3 Geographic information.....	8
3.1 A model of reality.....	8
3.2 Raster and vector format.....	8
3.3 Spatial reference and map projections.....	10
3.4 Geographic datasets.....	13
3.4.1 Google Maps.....	13
3.4.2 OpenStreetMap.....	13
4 Methods for handling geographic information.....	14
4.1 Collecting the data.....	14
4.1.1 Remote sensing.....	14
4.1.2 Geodetic surveying and GPS.....	15
4.2 Storing the data.....	18
4.3 Visualizing the data.....	18
4.3.1 Platforms.....	18
4.3.2 Visual hierarchies.....	19
4.4 Geographic information system – GIS.....	20
4.5 Open standards for web GIS.....	21
4.5.1 Web Map Services – WMS.....	21
4.5.2 Web Feature Service – WFS.....	24
4.5.3 Styled Layer Descriptor - SLD.....	27
5 Open Source.....	28
5.1 <i>Open source licenses</i>	28
5.1.1 Apache License.....	28
5.1.2 MIT license.....	28
5.1.3 BSD licenses.....	28
5.1.4 The GNU General Public License – GPL.....	29
5.1.5 The GNU Lesser General Public License (LGPL).....	29
5.1.6 Creative Commons Attribution-ShareAlike 2.0 license (CC-BY-SA).....	29
5.1.7 Google Maps API license.....	29

5.2 Open source software for geographic information.....	30
5.2.1 GeoTools.....	30
5.2.2 Geoserver.....	30
5.2.3 Openlayers.....	30
5.2.4 PostGIS.....	31
5.2.5 Atlas Styler.....	31
6 Workflow for web based open source GIS.....	32
6.1 Incoming data.....	35
6.2 Outgoing data.....	36
6.3 Distributing the data to a network.....	37
6.4 Styling the distributed data.....	39
6.5 Publishing of the web application.....	40
6.6 Core functionality and editing of geographic data.....	41
7 Evaluations of the model in Kwazulu Natal.....	42
7.1 Application 1: Custom map viewer.....	42
7.1.1 Background.....	42
7.1.2 Requirement specification.....	42
7.1.3 Input data.....	43
7.1.4 Functionality.....	43
7.1.5 Outcome and application scope.....	45
7.2 Application 2: Mapping informal settlements.....	45
7.2.1 Background.....	45
7.2.2 Requirement specification.....	46
7.2.3 Input data.....	46
7.2.4 Functionality.....	46
7.2.5 Outcome and application scope.....	48
7.3 Application 3: Garden project data viewer.....	48
7.3.1 Background.....	48
7.3.2 Requirement specification.....	51
7.3.3 Input data.....	52
7.3.4 Functionality.....	52
7.3.5 Outcome and application scope.....	55
8 Discussion.....	56
8.1 Using open source.....	56
8.2 Implementation model.....	56
8.3 Application 1 – Custom map viewer.....	56
8.4 Application 2 – Mapping informal settlements.....	57
8.5 Application 3 – Garden project data viewer.....	58
8.6 Future work.....	59
9 Conclusions.....	61
References/Bibliography.....	63
Appendix 1: EPSG SRID Example.....	68

1 Introduction

1.1 Background

Rural planning in South Africa is characterized by large economic, social and environmental interest. The modern history of the country has led to complex situations regarding legal security and housing for the common citizen. The housing situation and especially informal settlements is one of the problems with rural planning in today's South Africa.

An informal settlement refers to land that is occupied by someone who does not have the legal right to live there. There are many different definitions of informal settlements, but the most common one, which also have been used for this study, is the definition from the UN habitat program. This says that an informal settlement is an area that is inhabited by someone who doesn't have the legal right to live there or that a property that is not intended for housing is being used for this purpose (World Health Organization 2010).

Worldwide there are around 1 billion people living in informal settlements, which is almost 17% of the world's population (Shah 2011). In South Africa 25-30 % of the population lives in informal settlements outside the existing cadastral system (UN habitat 2003). There are several beneficial reasons for both the residents and the government to legalize these settlements. The residents will benefit from a transition from an informal to a formal settlement by receiving new services and government support that will result in improved socio-economic, environmental and health related benefits (Adams, Sibanda & Turner 13).

The UN Habitat recognizes from a conference in Nairobi, Kenya, 2003 that a country needs non-cadastral geographic data in order to tackle the problems with informal settlements (UN habitat 2003). This kind of geographic data could be obtained in many different ways. One easy and accessible way of doing this is through an ordinary handheld GPS device, which can then be put into a geographical information system.

A geographic information system (GIS) provides a good solution for visualizing the collected geographic data. In addition to this it can also provide many different kinds of spatial and geometric operations on the collected data. A GIS supports many different kinds of input file format and it can visualize the geographic data in both raster and vector format.

A Web based GIS is a good solution when it comes to publishing geographic data on the web. There are many alternatives, both commercial and open source, that allow this publication of geographic information on an ordinary webpage. Open source programs are usually free of charge, but they might have some restriction on how you may use them. The commercial alternatives cost money, but a web based GIS usually have cheaper licenses than a desktop GIS. The reason for this is that maps published

on the web are mostly applied for display purposes and the users are expected to use less functionality (Jiang 2003).

In KwaZulu Natal a Web based GIS could be used when working with geographic data that concerns rural planning and informal settlements. It could for instance be used as recourse to facilitate inventory and mapping of rural areas. In addition to this application a web based GIS could be used within the local development projects with a need of geographic information.

1.2 Purpose

The purpose of this study is to develop methods on how to collect, store and visualize geographic data in applications of rural planning and informal settlements. The methods should be simple and cost efficient.

1.3 Research problem

Is it possible to develop a workflow for surveying and visualization of geographic data, in areas with informal settlements? The result from using this workflow should be a cost effective product that is efficient and simple to use.

1.4 Method

The methods used are illustrated in figure 1.1 and explained in text below.

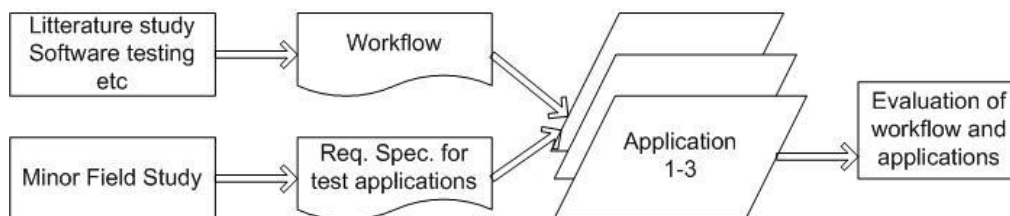


Figure 1.1. Method.

A literature study was performed to acquire the base knowledge used for the theory background. GIS books and some articles were used. The techniques used while developing the workflow was found and learned in the Open Source Community on the web. Forums, guides and tutorials were undertaken at websites like OSgeo.org, GeoServer.org and OpenLayers.org. A field study was also performed parallel where a great deal of information received by personal communication with people working in the area. This communication was important when deciding the direction of the example applications created. Further on number of reports and project descriptions regarding the Dukuduku area was used. After the elaboration of the workflow and the requirement specifications the applications where developed, tested and evaluated.

1.5 Selection and limitations

The study was performed and restricted to an area called the Dukuduku forest in KwaZulu-Natal, South Africa.

Open source programs have been used for this study in order to reduce the cost for licenses. The reason for this is that this study have been carried out in a low income area. The program Geoserver have been used, because it supports the *transactional WFS*. Atlas Styler have also been used for styling the data from Geoserver. PostGIS have been used due to its compability with Geoserver and finally have the JavaScript library Openlayers been used since its helpful when building web applications.

The visualization in this study is strictly performed on the web. Web GIS programs were selected for this purpose. These programs are mentioned in previous paragraph.

All geographic data used in the test applications were supplied by a local surveying company. Survey within the study area was not reccomended due to the political situation.

One limiting factor has been time. Because of the broad topic of the study, focus has been directed more to the gather, store and visualization aspects, rather than implementing for example PostGIS functionality in the web application.

1.6 Disposition

Chapter 2-5 presents the background used in the paper. At first, the study area and ongoing local projects are described in chapter 2. Chapter 3 explains the concepts of geographic information. Chapter 4 discusses general methods for collecting, storing and visualizing geographic information. Finally open source is reviewed in chapter 5.

Chapter 6-7 contains the result of the study. In chapter 6 a model is presented with the purpose to support developing projects with open source web GIS resources. The model is then applied in chapter 7 where three example web applications are presented.

Chapter 8-9 contains the discussion, conclusions and references.

The appendix holds application source code, illustrated examples and specifications used in the paper.

2 Development projects in the Dukuduku forest, South Africa

The following section provides a detailed background on the area of study and more specifically the projects where open source web GIS software can be applied. Implementations of map applications supporting the projects can be found in chapter 7.

2.1 Study area

The study area of this thesis is the Dukuduku forest, within the Isimangaliso Wetland Park, which is located approximately 300 km north of Durban, South Africa. The wetland park, which hosts several types of ecosystems and a variety of species, was inscribed as a World Heritage site in 1999 (UNESCO 2011).



Figure 2.1 (left) and figure 2.2 (right) location of the Isimangaliso wetland park and the study area.

The Dukuduku forest is not only a part of a World Heritage site it is also a state owned forest, which means that it falls under the rules and restrictions of the National Forests Act no. 84 from 1998 (ACER, 2006). It is because of these that the government has introduced numerous projects to address the issues of informal settlements. Paragraph 7 in *Chapter 3 special measures to protect forests and trees part 1* from the National Forests Act no. 84 declares that:

7. (1) No person may cut, disturb, damage or destroy any indigenous, living tree in, or remove or receive any such tree from, a natural forest except in terms of—
(a) a license (*ibid*) issued under subsection (4) or section 23; or
(b) an exemption from the provisions of this subsection published by the Minister in the Gazette on the advice of the Council (National forests act 1998)

No special permission to cut down the indigenous forest, within the Dukuduku forest, has been granted to the inhabitants by the government. This results in that all forestry in the area is illegal.

The forest itself is home to over 60,000 people, a majority of whom are from the Zulu tribe who represent one of the largest ethnic groups in the province KwaZulu-Natal, South Africa (Lewis 2009). A majority of the inhabitants within the Dukuduku forest reside in informal settlements and are constantly moving around, cutting down indigenous forest and building new structures to meet their needs (ACER, 2006). This is, as mentioned, in direct conflict with the rules and regulations set out in the National Forest Act no.84.

The problematic living conditions have resulted in several attempts by the government to move the inhabitants out of the forest and into established communities. For instance, in 1994 the government was successful in relocating numerous inhabitants into the Khula village where the population now is approximately 15,000 people (ACER, 2002).

With such a strong focus on this area the government and various nongovernmental organizations, NGO's, have introduced over 50 projects aimed at improving the socio-economic status of the individuals while protecting and improving the surrounding environment.

2.2 Projects in Dukuduku forest

The following section will introduce some of the projects in the Dukuduku forest. The usefulness of an open source web GIS in one of these projects will be described further in chapter 7.

As mentioned, there are over 50 development projects in the Dukuduku forest, all of which are managed by a consulting company called *Enhanced Strategies* who work in collaboration with other implementing organizations. All these projects take place within an established project area, which is illustrated in figure 2.3.



Figure 2.3. Project boundaries in the Dukuduku forest.

The study area was chosen because of the potential benefits a web based GIS system could provide for projects that focus on social, economic and environmental aspects. There are many different actors involved in the projects that need up to date information about the constant changing environment. An example of this could be everything from the people in the village that are working with the actual projects to the government or different NGOs that are providing the funding. A web based GIS enables easy distribution of data and it can easily be updated with new information.

Some of the main projects in the Dukuduku forest are (Barnes, 2011):

- *Environmental conversation.* The aim of this project is to determine the boundary between the Isimangaliso wetland park and the development that are taking place in the area. Protection of indigenous forest is also included in this project.
- *Water supply.* The aim of this project is to provide 5000 households with water.
- *Electricity.* The aim of this project is to provide 5000 households with electricity.
- *New housing.* The aim of this project is to upgrade 5000 existing homesteads to a more permanent structure.
- *Tourism.* The aim of this project is to boost the local economy by tourism. This project got potential because of the areas closeness to St Lucia and that all visitors there need to pass through the project area.
- *New community facilities.* The aim of this project is to provide the project area with new schools.

There are also other minor projects going on in the area. Example of this is:

- *180 food gardens.* The aim of this project is to create 180 food gardens within the project area. These gardens will provide crop with good nutritional value and they should also be constructed in way that requires none or little future maintenance. A more detailed explanation about this project and how the data could be visualized in a web GIS is given in chapter 7.

3 Geographic information

To work with geographic information one need suitable data structures, reference systems and methods for visualization. Among these needs is the mathematical theory of how to represent a spatial object and further on how to store the object digitally. Examples of application handling geographic information could be to measure, save and then display features from the study area as described in chapter 2. These features could for instance describe the extent of the Dukuduku forest or the positions of the dwellings.

In the first sections of the chapter a general introduction describes what kind of geographic information we can save and what data structures are available. Thereafter coordinate systems are explained along with reference systems and map projections. The last sections introduces reference systems commonly used for web GIS and some examples of web map service providers.

3.1 A model of reality

To collect and subsequently store an *exact* representation of an area or feature is practically not possible. Every method used to collect geographic data has its limit in precision (as explained in chapter 4). In addition to the challenge of collecting high precision data, one must also consider the storage space demanded by such information, and the final accuracy limits set by computer arithmetic. Given these limitations the conclusion is one can only store a somewhat partial and simplified *model* of reality (Eklundh, 2006).

3.2 Raster and vector format

When storing geographic information digitally, two general data types can be identified, *raster* and *vector*. The principles of the two are illustrated and compared in figure 3.1. Advantages and disadvantages of the storage methods are presented in table 3.1.

Raster data divides the mapped space into smaller cells, each with the same shape. Usually these cells are quadratic, which enables for a row-column index for each cell. This index in combination with the grid properties determines the position of the cell content, which is considered to be homogenous.

Vector data are defined by standalone points and/or points connected to each other. By this means it is possible to store points, lines and polygons as unique objects. Compare to the raster model where an object can only be defined more loosely as a set of connected pixels. An objects position can be defined by the points it contains as each of those hold coordinates in the given reference system.

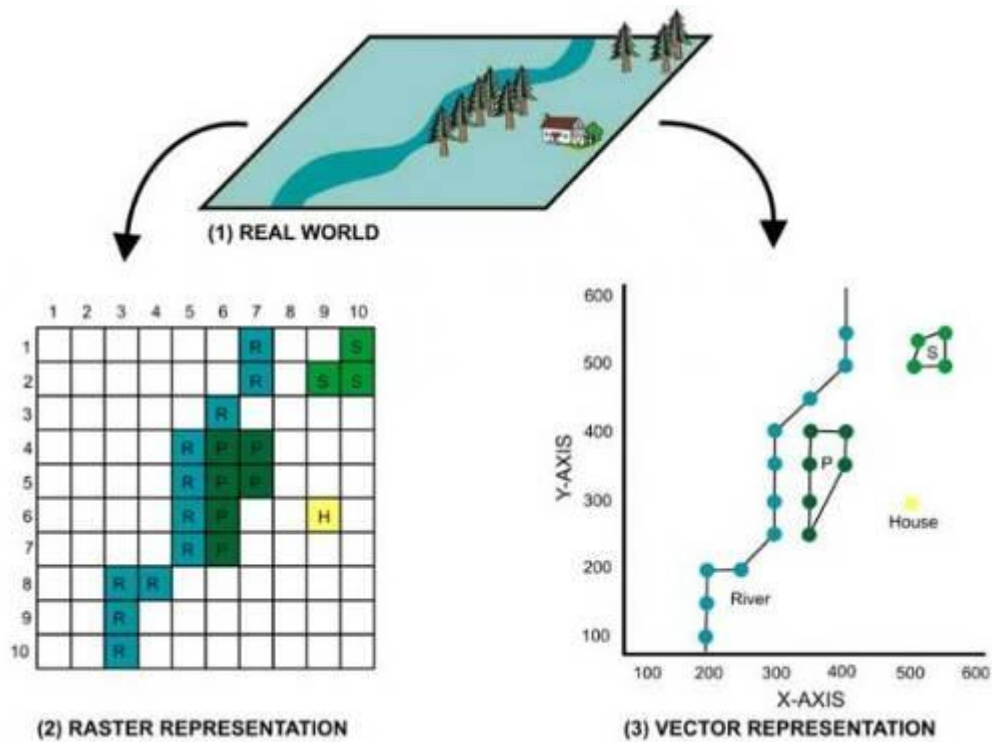


Figure 3.1. Comparison of the raster and vector data types. The information collected from the real world (1) can be saved in raster format (2), or in vector format (3). (Housing and Land Use Regulatory board 2007).

Table 3.1. Raster and vector data structures advantages and disadvantages (Housing and Land Use Regulatory board 2007).

	Raster	Vector
Advantages	<ul style="list-style-type: none"> Good for complex analysis Efficient for overlays Data structure common for imagery Good for continuous surfaces 	<ul style="list-style-type: none"> Compact data structure Efficient for encoding topology Good for objects and fixed shapes
Disadvantages	<ul style="list-style-type: none"> Large datasets Topology hard to represent Maps less "realistic" 	<ul style="list-style-type: none"> Complex structure Overlay operations difficult Might imply false sense of accuracy

3.3 Spatial reference and map projections

To represent a position, a common method is to use numerical values within the context of a *reference system*. In order to fully explain the concepts of such a system, we first need to review some properties of a *coordinate system*.

3.3.1 Coordinate systems

A coordinate system uses one or more numbers to describe a point's position in relation to the *origin*. The numbers are separated by their internal order or by a letter reference, as in 'the x-coordinate'. In order to represent a point in three dimensional space one can use a system with orthogonal X, Y and Z axes. If we also assign the same unit of length to these axes, one says the coordinate system uses *Cartesian coordinates*. See figure 3.2 for an illustration of such a system. While this approach is easy to understand and communicate, it is not very intuitive for the application of describing locations on the earth's surface (Harrie, 2008).

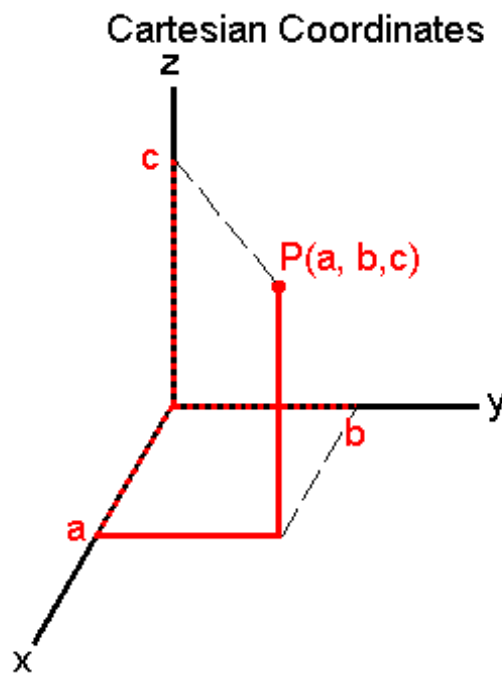


Figure 3.2 Cartesian coordinates: Orthogonal axis with uniform scale (Marianne Breinig 2011).

An alternative to the Cartesian coordinate system is to make use of an ellipsoid earth model. The coordinates of such a model is referred to as *geodetic coordinates*. To obtain the ellipsoid a suitable ellipse is rotated around its minor axis and to specify a position on this ellipsoid one uses a latitude and longitude grid, as illustrated in figure

3.3. The ellipsoid is designed to conform to the mean ocean level or more precisely to the *geoid*. A third coordinate h is also added which describes the perpendicular distance, or the height, from the spheroid surface (Harrie, 2008).

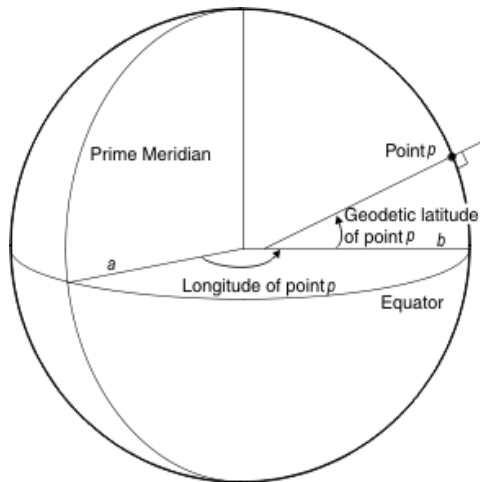


Figure 3.3 Geodetic coordinates (IBM, 2011).

3.3.2 Reference systems

To make use of the coordinate systems described in the previous section, one also needs to relate it to the Earth. This is easily understood by posing the questions “*where is the origin?*” or “*which orientation and scale does the axis have?*”. Solving these matters for a coordinate system results in a *geodetic reference system*. Or in other words, a geodetic reference system is a coordinate system related to the Earth. To link the coordinate system to the earth and thus create this relation, a net of reference points are assigned coordinates in the actual system. This set of reference points is called the *control network* (Harrie, 2008).

3.3.3 Map projections

To visualize geographic information on a paper map or on a computer screen, the three dimensional data needs to be projected to a two dimensional space. Performing a projection like this inevitably comes with a certain degree of deformation and depending on what type of projection used this will affect different properties of the map. Therefore it is always an important decision what kind of projection to choose (Harrie, 2008). Some of the available projection methods are illustrated in figure 3.4.

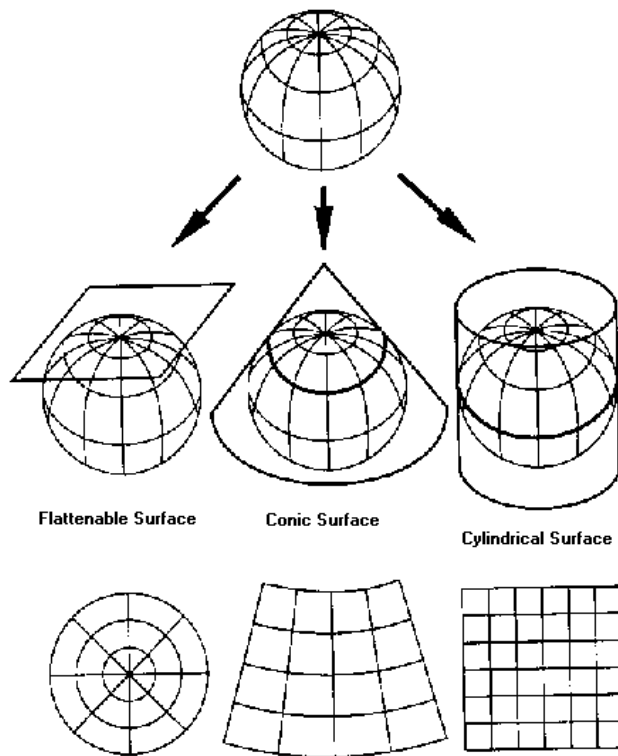


Figure 3.4 Basic geometric forms in map projections. Note the differences in deformation depending on projection type. (Eliris, 2011)

3.3.4 Spatial Reference and map projections in web GIS

Virtually all GIS applications use a *Spatial Reference System Identifier*, SRID, to identify reference system and map projections. The software either has its own SRID implementation or it refers to one issued by an authority. One of the main distributors of SRID identifiers was the European Petrol Survey Group¹, whose SRID values are commonly written in the form “EPSG:12345”.

One SRID used for many web application is EPSG:3785, or more known as EPSG:900913. This SRID uses the reference system WGS84 and a Mercator projection. It is used by Google Maps, OpenStreetMap and many other web map service providers. The full definition behind this SRID is supplied in *Well Known Text* format in Appendix 1 (Openlayers, 2011 a). For a more detailed description on the web map services mentioned, see next section.

¹ Since 2005 the EPSG SRID values are distributed by the International Association of Oil & Gas Producers (OGP) Surveying & Positioning Committee. (OGP Surveying & Positioning Committee, 2011)

3.4 Geographic datasets

The following section introduces a couple of web services that supply geographic data and tools for creating online maps.

3.4.1 Google Maps

Google Maps is web mapping service free of charge for non-commercial users. The data consist of satellite imagery, street and overview maps accessible from the Google maps website or through external sites. There is an API available for incorporation the Google Maps interface into other web sites, and it is also possible to use open source tools such as OpenLayers to retrieve Google maps data for use in a customized application (Google maps 2011).

3.4.2 OpenStreetMap

OpenStreetMap is a map service that is created by the users. The idea behind it is a worldwide map free of charge, where the data comes from the users. Everyone with geographic data may contribute to this map service. This data are stored in a database and all the users have the right to access and modify it. This makes the structure of OpenStreetMap very similar to the concept of *Wikipedia*, which is a free online encyclopedia (*OpenStreetMap, 2011*).

4 Methods for handling geographic information

Suppose we want to visualize a rural area as a digital map. The purpose of the map could for example be to clearly display villages and informal settlements, the surrounding vegetation, roads and other infrastructure. To achieve this, three general steps can be identified as illustrated in figure 4.1. Initially, information on the settlements extent and location needs to be *collected*. Secondly, this information needs to be *stored* for future access, in e.g. a geographical database. The last step is to *visualize* the geographic data to create a comprehensive map for use in a computer or mobile device. The following sections explain and compare some of the available techniques used to achieve the three steps.

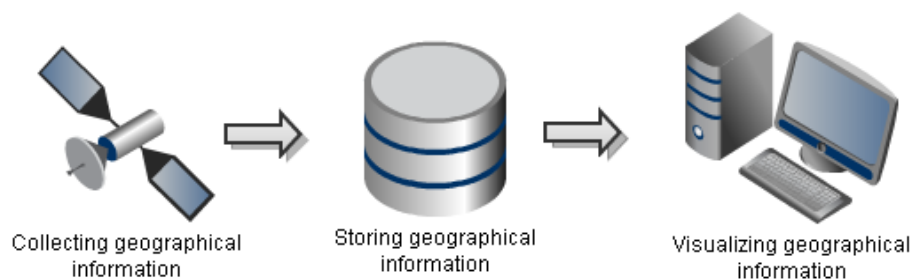


Figure 4.1. The process of visualizing informal settlements.

4.1 Collecting the data

When collecting geographic information several techniques can be utilized. One possibility is to extract the data from already existing sources, e.g. paper maps or statistical tables. This is not always an option, leaving the need for first hand data collection (Eklundh, 2006). A features position on the earth's surface can be measured either on site with geodetic surveying instruments or by the use of remote sensors assisted by some computer software.

4.1.1 Remote sensing

Remote sensing is an umbrella term for data collection made on a distance, for example by sensors mounted on an aircraft or satellite. These methods are often preferred when collecting data over a large region or a continuous surface. One common application is map making.

Orthophotos are often used in maps as background or base layer to give the user an overview of the terrain type. By mounting cameras on an aircraft and let it fly in overlapping stripes, images with a bird's eye view are acquired. One of the reasons of the overlap is to enable creation of corrected images, where errors depending on elevation differences and the Earth's bulge are removed. Such an image, a geometrically corrected aerial photography, is called an *orthophoto* (Eklundh, 2006).

4.1.2 Geodetic surveying and GPS

Geodetic surveying is to accurately position specific objects or points on the earth's surface in a given coordinate system. It is used frequently to carefully determine the position of objects in the built environment, such as roads and buildings.

Geodetic surveying is often performed with a total station, which measures lengths and angles. Together with some previously known point benchmark, other positions can be determined with high accuracy. Another common method is to use a satellite positioning system such as the Global Positioning System.

The Global Positioning System (GPS) is a global navigation satellite system developed by the U.S. Department of Defense in the seventies. It went fully operational in 1994 and since May 2000 it allows for unrestricted public access. The system consists of 24 satellites orbiting the earth in a very precise manner. With a free line of sight to at least four of these satellites, a user can acquire high precision positioning with meter accuracy. Figure 4.2 shows a user of the system with a hand held receiver (Horemuz, 2010).

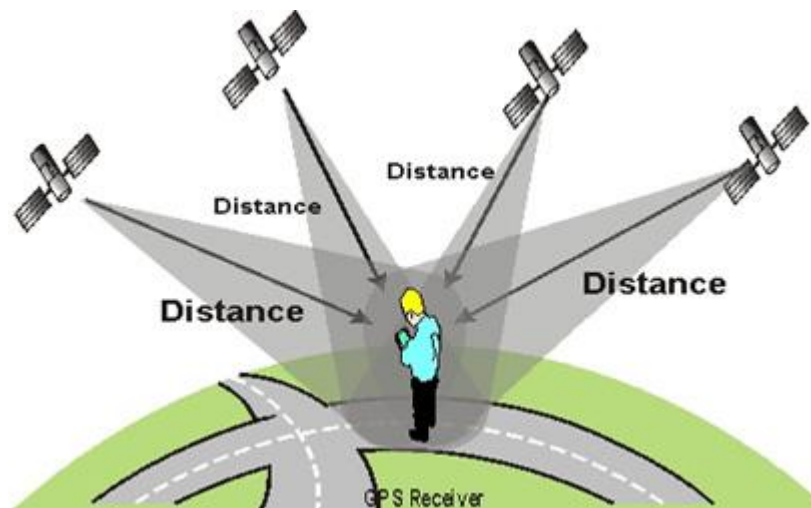


Figure 4.2. GPS satellites and user with hand held receiver (Althos, 2011).

Basic concept

The basic principles of the GPS functionality can be explained by a relationship of time, distance and velocity. Assume at least four satellites are within range of a given receiver. Each one of these satellites transmit a signal which is observed by the receiver. By measuring the travel time of these signals one can calculate the distance from the receiver to the individual satellites as following²

$$\text{Distance(receiver-satellite) = Speed of light} \cdot \text{Signal travel time}$$

Knowing the distance to three of the satellites allows the receiver to trilaterate its position to two possible cases. In fact, one of these candidates can directly be written off as an impossible position, leaving only one feasible point. The purpose of the fourth satellite distance is to allow synchronization of clocks on the satellites and in the receiver, which lead to more accurate distances and thus a better solution (*Horemuz, 2010*).

Phase and code measurements

The code transmitted by the satellites is called a *pseudo random code*. At the same time this code is also generated in the receiver. When the satellites signal reaches the receiver the two signals are out of phase due to the travel time. However, by shifting the receiver's version of the signal until the two match, the delay - equal to the travel time - is found. This approach is called code measurement (*Horemuz, 2010*).

Another approach, known as carrier-phase measurement, relies on the carrier wave of the signal. As the frequency of the carrier signal is much higher than the that of the pseudo random code, it gives a much higher accuracy (*Horemuz, 2010*).

DGPS and RTK

Differential GPS or DGPS is a system used to improve the accuracy of GPS code measurements. The system relies on at least two receivers, one at a base station and one mobile as illustrated in figure 4.3. The position of the base station receiver is known since before and the station is connected via a data link to the mobile receiver. Using this technique the uncertainty can be reduced to around one meter. To improve accuracy even further one can use receivers measuring both on code and carrier wave. This is called RTK or real time kinematics (*Horemuz, 2010*).

² In this simplified example we disregard minor error sources such as the fact that the velocity of radio signals are only constant in vacuum.

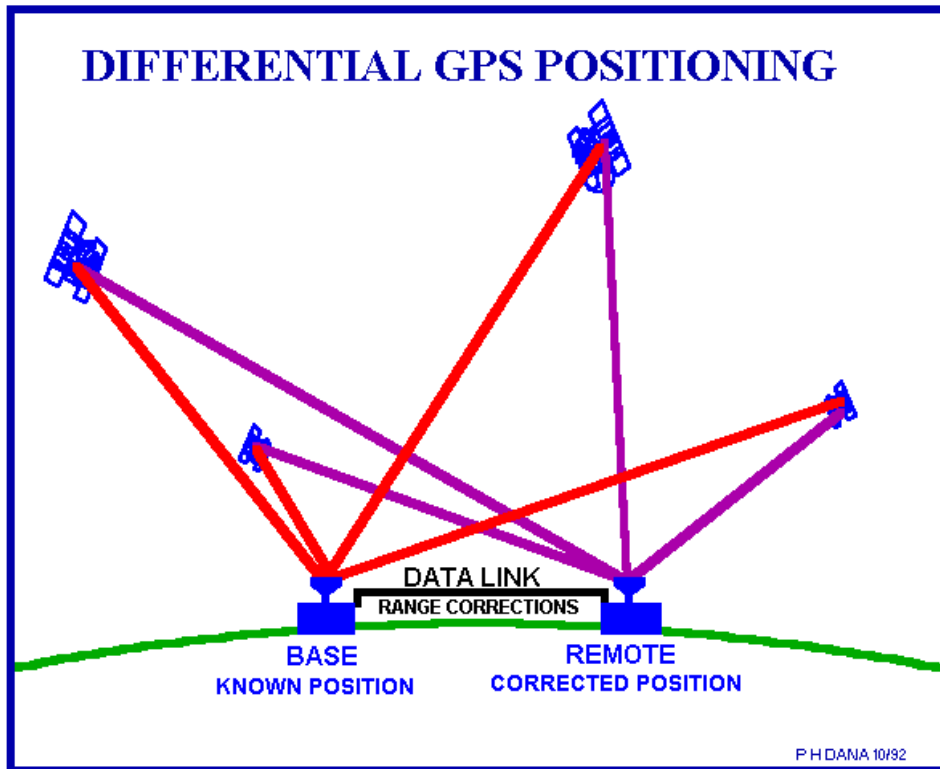


Figure 4.3. The principle behind DGPS. (Samhassan, 2011).

Comparison and accuracies

The GPS methods discussed above are compared with regards to accuracy in table 4.1. The reader should be aware there are also other methods that will not be treated here, such as static measurements where a receiver will stay at a location, sometimes for one hour or more (Horemuz, 2010).

Table 4.1 Comparison between methods (Horemuz, 2010).

Method	Horizontal accuracy (95%) (m)	Vertical accuracy (95%) (m)
Absolut	3	6
DGPS	1	2
RTK	0.03	0.05

4.2 Storing the data

After acquiring the geographic data in suitable format the next question would be how to store the data in an effective way. One way to manage the information is to use a *file based data handling system*. This means a scheme is created to be used by programs directly accessing and processing the storage files. Usually, a drawback of such a system is that it can be very person dependent and challenging when update of the functionality is needed (Eklundh, 2006).

A better approach is to import the files to a database with a DBMS, *Database management system*. By a set of routines the DBMS control the maintenance and use of the database, making the technical details more transparent through a user interface. The actual type of database most suitable may vary depending on application (Eklundh 2006).

A common type of system structures is the *relational database*. In the relational database data are stored in tables, also called relations. Every relation has a number of attributes which corresponds to the columns in that table. By linking the relations together an effective system can be created allowing for fast processing.

In the application of storing spatial data, a *spatial database* is preferable. In addition to the common data types of numbers, characters and text strings a spatial database also supports geometric data types, such as points, lines and polygons. For a description of the specific spatial database software's used in this study, see chapter 5 and 6. These sections also include some information on the formats relevant for a spatial database.

4.3 Visualizing the data

This section describes how to visualize the data to an end user. It also describes important techniques on how a map should be constructed in order to make it easy to interpret the information.

4.3.1 Platforms

In this study the focus of map visualization is on computer platforms rather than other media. Among other benefits these platforms give the option to include interactivity in the map interface. Maps could be created for display in advanced GIS programs or in a simple image software. Another possibility is to make use of libraries for viewing maps in web browsers. For every one of these methods it is important how the information in the map is presented. In the next section some rules are presented that will help a user interpret the map. Section 4.4 and 4.5 will describe GIS and web GIS in more detail.

4.3.2 Visual hierarchies

Visual hierarchies describe how a user interpret and extract information from a map. It also explains how a map should be constructed in order to emphasize important data. There are seven categories that are related to visual hierarchies that are important to keep in mind when creating a map. These are:

- Size
- Color
- Form
- Density
- Structure
- Direction
- Position

These seven categories, also called graphic variables, determine how a user interprets the map. None of these graphic variables are more important than another when making a map. It is however very important that all these variables have been considered before the map is published in order to make the context easy to grasp for the user. (Keates, 1996).

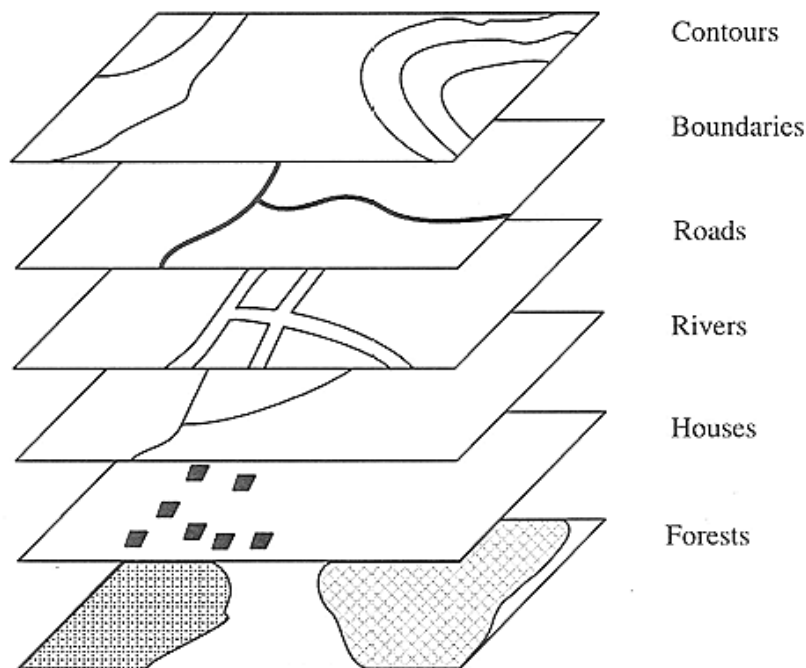


Figure 210 Map Layers

Figure 4.4. An example on how to create a map by using visual hierarchies (FOA, 2011).

4.4 Geographic information system – GIS

Geographic information system (GIS) is a tool for analyzing and working with geographic data. Queries and calculations can be used to solve a problem and get a solution. A map can be made from the results. This map can be used to explain or clarify the discoveries to a recipient (GIS, 2011). There are several GIS programs available for the user that supports these operations and all these programs can be described as four individual types.

There exists, as mentioned, four individual types that a GIS program can be described as. These programs can be published either as an open source or proprietary program and the aim of the program could either be a web based GIS or a desktop GIS. See *figure 4.5*

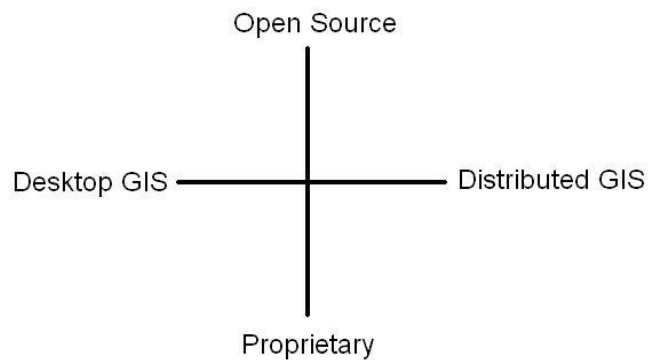


Figure 4.5. Four individual types of GIS program classifications. The horizontal axis describe the architecture and the vertical describe the license type.

A proprietary GIS refers to a GIS system where the source code doesn't come with the program. Licenses for these programs usually cost a lot of money. An example of this would be the license for ArcGIS, which is one of the most used GIS programs. The license for this program costs a couple of thousand dollars (CMC, 2011). The users have no possibility to develop a proprietary GIS program further, since they have no access to the source code. That is the reason why several open source GIS programs have arrived on the market. More about open source programs can be found in chapter 5.

A desktop GIS refers to a GIS program where the program runs locally on the computer. This usually gives the user a powerful program, but the data and results from analysis are very limited to the local computer where the program is running. The results from a desktop GIS could be distributed to a receiver as a map in for example JPEG or PDF format. Publication of these data on the web could be preferred if the creator of a map wish to distribute the data to a larger audience. This could be achieved by a web based GIS.

A web based GIS refers to a program where the data is distributed and handled on the web instead of on the local computer. The data are broadcasted from a server to the web and this allows the users to perform several GIS commands on the web instead of on the local computer. There are several schemas that describe how request to the web GIS server should be constructed in order to perform an action. Examples of the most common schemas are explained in the following section.

4.5 Open standards for web GIS

This section describes some common web GIS specifications and explains what purpose they are used for. Web map service, web feature service and styled layer description will be explained in the sections below.

4.5.1 Web Map Services – WMS

A request sent over the web map service (WMS) returns an image over the requested area. This image shows geographical data over the area. This image layer, and potentially more layers, can be used in order to create a map. The WMS interface is a standard defined by the Open Geospatial Consortium (OGC). This interface supports three requests that are described in the text below.

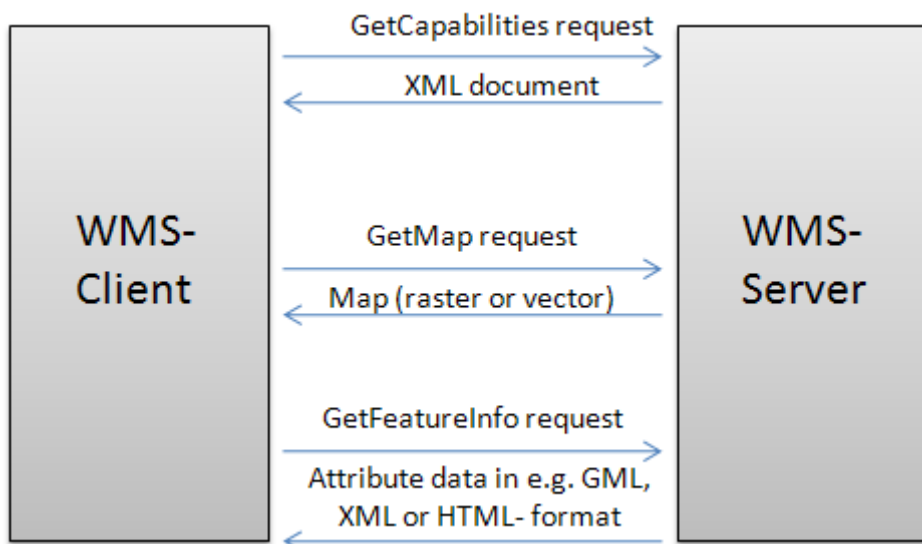


Figure 4.6 Requests protocols for a WMS.

The WMS standard supports the *getCapabilities*, *getMap* and *getFeatureInfo* requests. All these requests are sent through the Hypertext Transfer Protocol (HTTP) and the result could be used to create an online map. Figure 4.5 illustrates how these requests works.

The *getCapabilities* is a mandatory WMS request that returns the metadata in XML format. This metadata is both machine readable and human readable and it provides information about all operations that are supported for the WMS requests. See table 4.2 on how the *getCapabilities* requests are formed.

Table 4.2. The parameters of a *getCapabilities* request URL (Open Geospatial Consortium, 2006 page 21).

Request/parameter	Mandatory/optional	Description
VERSION=version	O	Request version
SERVICE=WMS	M	Service type
REQUEST=GetCapabilities	M	Request name
FORMAT=MIME_type	O	Output format of service metadata
UPDATESEQUENCE=string	O	Sequence number or string for cache control

The *getMap* request, which is also a mandatory WMS request, uses the parameters from the *getCapabilities* request. This is also sent as a HTTP request and the output is georeferenced image over the specified area. See table 4.3 on how the *getMap* request are formed.

Table 4.3. The parameters of a GetMap request URL (Open Geospatial Consortium, 2006 page 33).

Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetMap	M	Request name.
LAYERS=layer_list	M	Comma-separated list of one or more map layers.
STYLES=style_list	M	Comma-separated list of one rendering style per requested layer.
CRS=namespace:identifier	M	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	M	Bounding box corners (lower left, upper right) in CRS units.
WIDTH=output_width	M	Width in pixels of map pictures.
HEIGHT=output_height	M	Height in pixels of map pictures.
FORMAT=output_format	M	Output format of map.
TRANSPARENT=TRUE FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red-green-blue color value for the background color (default=0xFFFFFF).
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=XML).
TIME=time	O	Time value of layer desired.
ELEVATION=elevation	O	Elevation of layer desired.
Other sample dimension(s)	O	Value of other dimensions as appropriate.

The getFeatureInfo is an optional request that will allow the user to retrieve information about an object. This request returns the attribute values from a certain location in the image retrieved from the getMap request. See table 4.4 on how the getFeatureInfo request are formed.

Table 4.4. The parameters of a *GetFeatureInfo* request URL (*Open Geospatial Consortium 2006 page 33*).

Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetFeatureInfo	M	Request name.
map request part	M	Partial copy of the Map request parameters that generated the map for which information is desired.
QUERY_LAYERS=layer_list	M	Comma-separated list of one or more layers to be queried.
INFO_FORMAT=output_format	M	Return format of feature information (MIME type).
FEATURE_COUNT=number	O	Number of features about which to return information (default=1).
I=pixel_column	M	i coordinate in pixels of feature in Map CS.
J=pixel_row	M	j coordinates in pixels of feature in Map CS.
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default= XML).

All these requests could be used in order to create a basic map where the user can identify a feature, get some basic information from it and also perform some basic queries. There might however be a scenario where the user wants to be able to modify, create and delete features. Then the user needs access to the actual data instead of just an image of it. This is achieved by the web feature service (*Open Geospatial Consortium, 2006*).

4.5.2 Web Feature Service – WFS

The Web Feature Service a standardized recipe also called an interface, on how to distribute geographic information over a network. This standard is, as well as the WMS standard, developed by the *Open Geospatial Consortium* and both these are sent as a HTTP request. These two components make the foundation on how to create maps and distribute geographic data on the internet, see *figure 4.6* on how this request is made. The WMS and the WFS interface are similar in that sense that they both can be requested over a network, but the result of these request are quite different.

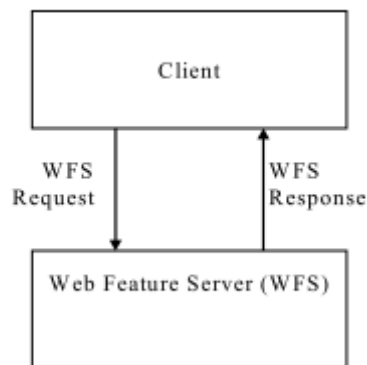


Figure 4.7. Basics understanding of the WFS protocol

A request sent as a web feature service (WFS) returns the data over a specified geographic area. This differ from the WMS request, *See chapter 4.5.1*, that only returns an image over a geographic area. The retrieved data from the WFS request is a file in the eXtensible Markup Language (XML) format. This XML file contains all the information over a geographic area. Additionally are geographic features within this XML document expressed in the Geographic Markup Language (GML) format. All geometries are assumed to be in 0, 1 or 2 dimensions, also called simple geometries. This refers to points, linestrings and polygons. Additionally to this is a response sent back to the user once the request is processed. This response will describe if the request was processed successful or if there were any problems encountered. To reduce the amount of information within the retrieved files a filter can be used.

A filter can be applied to the WFS request to reduce the amount of data within the requested area. This filter is transmitted in XML format and it supports the CQL standard, which also is a standard from Open geospatial Consortium. The CQL standard supports logical operators and can in that way limit the amount of information. An example of when this could be useful would be selecting only the highways from a road network layer. Five operations are supported to retrieve and handle the data over a network.

The operations *getCapabilities*, *DescribeFeatureType*, *GetFeature*, *GetGmlObject*, *Transaction* and *LockFeature* are used for retrieve handle and get information about the original data set. The *getCapabilities* operation returns an XML file that describes the data set. It provides all the information about e.g. the feature types, coordinate systems and name of the layers that can be accessed by a WFS request. It also provides all the operations that are supported by a WFS request. The *DescribeFeatureType* gives a description about one or more feature types. That differs from the *getCapabilities* operation that contains much more information etc. supported operations on the data set. The *DescribeFeatureType* returns an XML document with the information to the user. The *GetFeature* operation will send a query to the original data set and then return all data that fulfill these requirements set by the query. The returned data will contain features and it will be distributed in GML format. The *GetGmlObject* operation allows the user to retrieve element instances depending on their XML ID.

The *Transaction* operation supports the *create*, *delete* and *update* operations on geographic data. These operations allow the user to modify the original data set and these are described further in next paragraph. The *LockFeature* operation can lock one or more features. These features cannot be modified from a *Transaction* operation. The *LockFeature* operation is described in the next paragraph. There are five additional operations that are available for the WFS request once the data have been retrieved.

The web feature service supports the operations *create*, *delete*, *update*, *lock* and *query* on the retrieved data. The *create* operation allows the user to add information to the retrieved data. The *delete* operation allows the user to remove information from the retrieved data. The *update* operation transmits the modifications done from the *create* and *delete* operation to the source and saves the changes to the original data set. The *lock* operation assures that features in the original data set cannot be modified as long as the lock is active. The reason for this operation is to assure the quality of the data and make sure that none updates the data at the same time as another user is accessing it. The *query* operation can answer logical questions about the geographic data that the user might have. The *create*, *delete* and *update* can all modify the data, while the *lock* and *query* cannot. This divides the WFS into three categories.

The web feature service can be divided into *basic WFS*, *XLink WFS* and *Transactional WFS*. The *basic WFS* can be represented by the *getCapabilities*, *DescribeFeatureType* and *GetFeature* operations when retrieving the data. Additionally to this it also supports the *lock* and *query* operations on the retrieved data. The *XLink WFS* supports all the operations from *basic WFS* and addition to this it also supports the *GetGmlObject* operation. The *basic WFS* and *XLink WFS* does not allow modification of the original data set, while the *transaction WFS* does, see figure 4.7.

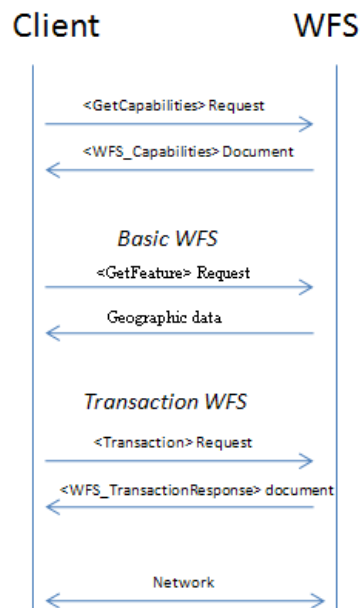


Figure 4.8 Difference between *basic* and *transaction* WFS.

The *Transaction WFS* allow a user to remotely modify a data set. This can be an improvement to a map viewer, since certain user might be able to add or remove geographic data. A result of this could be a map where the data is more accurate and up to date. The *Transaction WFS* can be described by two operations when handling the data.

The *Transaction WFS* can be represented by the *Transaction* and *LockFeature* operation. The *Transaction* operation allows the user to modify the original data set. That is done through the *create*, *delete* and *update* operation on the retrieved data. The *LockFeature* operation can lock features in the data set so it cannot be modified from a *Transaction* operation. Both the *Transaction* and *LockFeature* operations are useful to add additional GIS functionality, like add and remove point, to a web map application. (Open Geospatial Consortium, 2005).

4.5.3 Styled Layer Descriptor - SLD

Styled layer descriptor is an XML schema that determines how the output data from a WMS request should be presented. This XML schema controls how the geographic data are visualized and labeled (Open Geospatial Consortium, 2007). An example of this would be that a road could be labeled and displayed in different colors and thickness depending on what kind of road it is. Another cartographic operation that is supported with the SLD is the operation to determine on which zoom levels spatial data should be visible at. All these operations are supported for points, lines and polygons. The SLD standard is implemented in e.g. the web GIS program Geoserver (Geoserver, 2011c).

5 Open Source

This chapter describes the open source licenses and programs that are used in this study. Open source refers to programming code that is freely available for the public to study and improve. There are several open source licenses and standards that define how you are allowed to distribute and handle the code. A program based on any of these licenses is called an open source program.

5.1 *Open source licenses*

The most relevant open source licenses to this study are briefly explained in the section below. For each license type an example of a tool that uses this license is given. The tools are then discussed in section 5.2.

5.1.1 Apache License

A program under the Apache license may be used and distributed, free of charge or for a fee, as long as the Apache license comes with it. The user is also allowed to modify the program as long as it is clear which parts that have been modified. No warranty comes automatically with products under Apache, but the creator of the product can take out a fee in order to offer a warranty (Apache, 2004).

Rico, which is used in the Openlayers library, are using this license type.

5.1.2 MIT license

The user of a product under the MIT license is allowed to copy and modify the program free of charge. It also allows the user to sell this product to other users. Products under this license do not have any warranty at all (MIT, 2011).

The prototype.js file in OpenLayers is an example of a product using the MIT license.

5.1.3 BSD licenses

The BSD license allows the user to modify, change and distribute a product as long as it is clear who is the copyright holder and the user can never suggest that copyright holder approve of the users product (BSD, 1998).

Openlayers is an example of a product that is using the BSD license.

5.1.4 The GNU General Public License – GPL

The aim of the GPL license is to guarantee that the user can access to the code. This does not mean that programs based on the GPL license have to be free of charge, but rather that all code for the program should be available. This could be done either by all the source code comes with the program or that you can buy it from the developer. There are no warranties for programs under this license since the license is obtained free of charge. A distributor of software under GPL can charge a fee for warranty if one is responsible for the product. All programs that are released under the GPL license must stay under the GPL license. That means that programs that are using parts of a program under GPL license or that origin from a modified GPL licensed program must stay under the GPL license (GPL, 1991).

PostGIS and Geoserver are examples of products that are using the GPL license.

5.1.5 The GNU Lesser General Public License (LGPL)

The main difference between the LGPL and the GPL license is when you modify a program under LGPL you have the option to release it under another license type. This is also true when a program is built with another program that is released under LGPL (LGPL, 2007).

Atlas styler and Geotools are examples of programs under the GNU Lesser General Public License.

5.1.6 Creative Commons Attribution-ShareAlike 2.0 license (CC-BY-SA)

The CC-BY-SA license grants you the right to copy transmit and distribute your work as long as you contribute the creator of the program. This is done free of charge and the users will have full access to all the source code. It also allows the user to modify and then distribute the work as long as the modified program is under the same, or a similar, license. The user may never suggest that the creator of the program approve of the users program. The CC-BY-SA license could also be applicable on other things than computer program. Example of this would be license for music (CC-BY-SA, 2011).

OpenStreetMap is an example of a product that are using the CC-BY-SA license.

5.1.7 Google Maps API license

The user needs to have a Google account in order to use the Google Maps API and logging in with this account automatically confirm the terms for this license. The user might use material under this license free of charge as long as they not in any way suggest that Google approve the product. The Google logo and the terms and

condition link may never be removed. By publishing a product under the Google Maps API license you grant Google the right to replicate your work and add it as a hit to their search engine (Google, 2011).

Google maps are an example of a product that is using the Google Maps API license.

5.2 Open source software for geographic information

This section describes the different open source programs that are relevant for this study. Chapter 6 explains further how these programs are connected.

5.2.1 GeoTools

GeoTools is a Java based library for handling and distributing geographic data. The main purpose is not to visualize the geographic data to end users, but to mediate the data to other applications. The GeoTools library is built on top of the JTS from Vivid Solutions and it follows the structure of the GeoAPI schema from the Open Geospatial Consortium. The result from this is a product that hides the complexity of the data source, feature models and projections from the end user. The styling of a layer in GeoTools follows the OGC Styled Layer Description, which is described in section 4.5.3. Examples of applications that are using the GeoTools standards would be Geoserver, uDig and GeoVISTA studio. The geotools library is released under the GNU Lesser General Public License (Hall and Leahy 2008).

5.2.2 Geoserver

Geoserver is, as the name indicates, a server client that allows geographic data to be distributed from a server computer out on a network. The main purpose behind this program is to distribute geographic data in a free of charge and efficient way. Its origin is from a non-profit organization in New York City called The Open Planning Project. Their idea was to make urban planning data more open and accessible to the public (Geoserver 2011a). Geoserver is based on the GeoTools library, which supports Web Map Services (WMS), Web Feature Services (WFS) and Web Coverage Service (WCS), which all are interface standards issued by the the Open Geospatial Consortium (OGS) (Geoserver 2011b).

5.2.3 Openlayers

Openlayers is an open source JavaScript text file, or library, that allows publication of spatial data on the web. OpenLayers is free of charge and it support many simple map making tools like panning, zooming, but also some more advanced operations like area calculation. Openlayers supports request done by the Web Map Services (WMS) and Web Feature Services (WFS), which makes this library compatible with Geoserver and data from here could be loaded onto a map. This map can be published on the web since it is retrieved with JavaScript functions, which is supported by web browsers. Openlayers is released under the BSD license, but some part of the code are

based on other licenses as well. The other kind of licenses that are used in the program are Apache and MIT license (Openlayers, 2011).

5.2.4 PostGIS

PostGIS is a spatial SQL database that is an extension to PostgreSQL. The main difference between the two is that PostGIS can handle geometric data types in form of points, lines and polygons. PostGIS is, as mentioned WHERE?, a SQL database that follows the “simple feature SQL”, which is a standard defined by the OGC. The program also supports several GIS operations. Examples of this would be area calculation and buffering. PostGIS is published under the GNU general public license (PostGIS, 2011).

5.2.5 Atlas Styler

Atlas styler is a program that is used for styling and labeling of spatial data. This is done through an XML file that is based on the Styled Layer Description (SLD), which is a standard set by the OGC. This XML file contains all information on how the geographic data should be visualized and it is compatible with both PostGIS and Geoserver. Atlas Styler is based on the GNU Lesser General Public License (LGPL).

6 Workflow for web based open source GIS

This chapter will describe an implementation model, based on open source programs, how to gather, store and visualize geographic data on the web. This workflow is meant to be general and it can be used for many purposes. Chapter 7 will describe three scenarios where this workflow can be applied. The workflow can be divided into six steps. A For a full view of the implementation models details see figure 6.1. The six steps are:

- 1) Incoming data
- 2) Outgoing data
- 3) Distribution
- 4) Styling
- 5) Publishing
- 6) Core functionality

The first step in the implementation model is how to handle and store incoming data. That could be geographic data, which means that it refers to some place on earth, or it could be non geographic data. This data needs to be structured and stored in order to be useful in a digital world. Additional to this geographic data also needs a reference system attached to it in order so a future user can localize where the data have been collected. This implementation model has worked with two open source program for storing and distributing these data. The first one is an open source spatial database called *PostGIS* and the other program is the open source program *GeoServer*. Both these programs can store and handle the data in a satisfactory way. Once this data is stored the need might come for querying and working with the data at this stage and that leads to the second step in the implementation model, which is *how to query and work with incoming data*.

The second step in the implementation model is how to query and work with the incoming data. This step can answer simple questions and it can also allow the users to work with the data at a local computer. A user with the access to the database *PostGIS* can query a data set and in that way get adequate answers without even using the web application, which is described more in step 4. Users with access to the program *Geoserver* are also given the possibility to download all the stored data into several file formats. This allows the user to work further with these data in for example a desktop GIS. If no access are given to these programs, then the user needs to access them through a network and that is described in the third step *how to publish the stored data onto a network*.

The third step in the implementation model is how to distribute the stored data onto a network. A web application is built with the help of an open source JavaScript library called *Openlayers*. This library is compatible with the WMS and WFS, which both are interfaces that are designed for distributing geographic data and images over geographic data. *Geoserver* and *PostGIS* are also compatible with these interfaces and

that results in that data from these programs can be added into the web application. Openlayers also support map operations like zooming and panning and these can be added to the map. One thing however that Openlayers not provide is styling of the data. How this is done will be described in the fourth step *how to style the geographic data*.

The fourth step in the implementation model is how to style the geographic data. Data that are called with the WMS request have a basic design what it will be displayed as. This basic style can be altered with a request sent in XML format. An open source program called *Atlas Styler* allows the user to define a style and then it helps to generate this XML code. Once all the WMS data is styled the map can be ready to use and this leads to the fifth step *how to publish the web application on the web*.

The fifth step in the implementation model is how to publish the web application on the web. This can be done in two ways. Either as the application is being used as a standalone application with just some explanatory text, or it can be integrated with an existing homepage. Once this decision is made the question might rise if the stored data can be modified and updated from the web application? This leads to the sixth and final step *the core functionality of the model*.

The sixth step in the implementation model describes the core functionality of the model. This describes how stored data can be modified from the web application by using the *transaction* WFS request. This step could make the web application more user friendly and information could be spread more rapidly and be up to date.

Step 1-6 will be described further in the section below and figure 6.1 illustrates what the implementation model looks like.

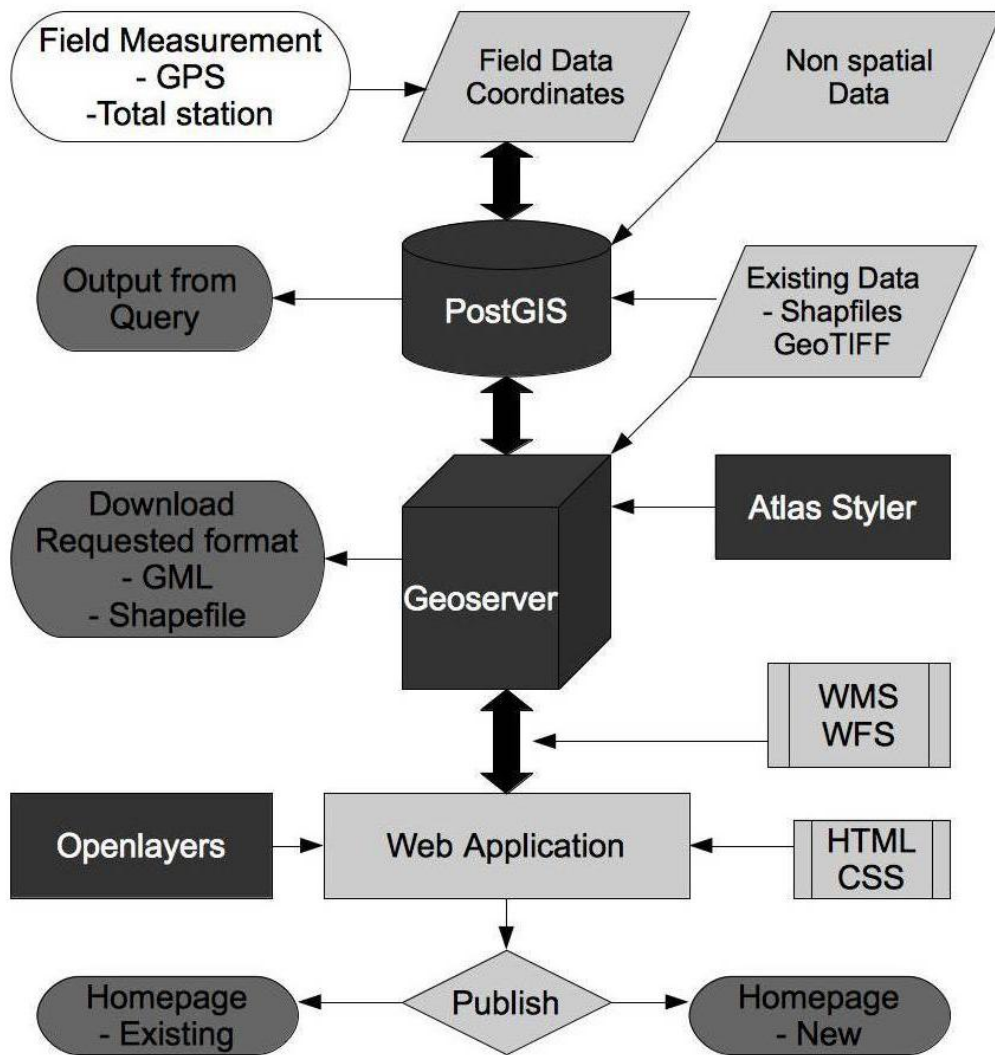


Figure 6.1 Implementation model over how to gather, store and visualize geographic data. All open source programs are displayed in dark gray with white text.

6.1 Incoming data

This section describes the first step in the implementation model, which is handling the incoming data. The parts from the implementation model that are concerned in this section are illustrated in figure 6.2.

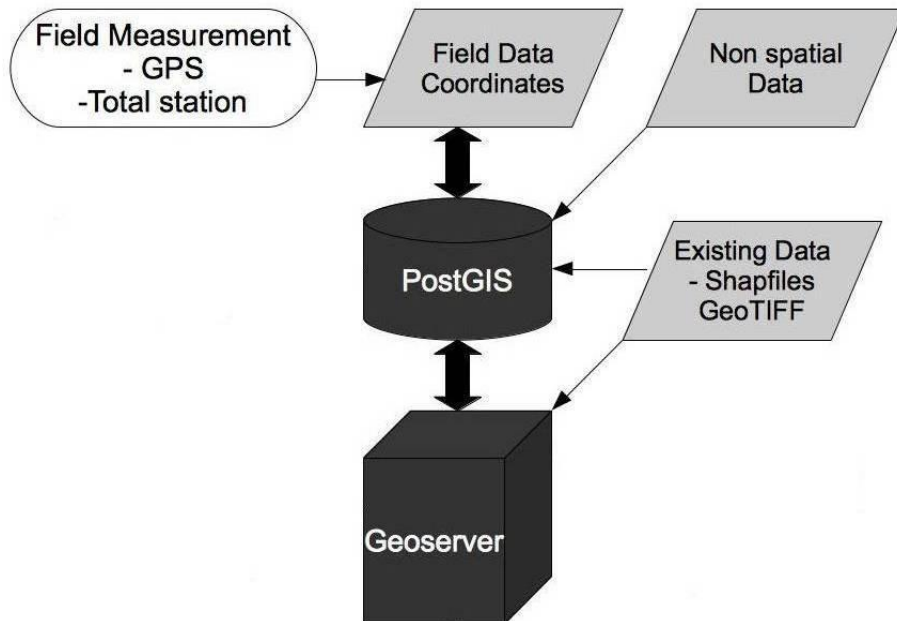


Figure 6.2 Step 1 in the implementation model – incoming data.

The incoming data to the implementation model needs to be stored and recognized by software before it can be used further. Data in this context refers to both geographic data and non geographic data. The programs that are used for storing data in this study are two open source programs called PostGIS and Geoserver. Both these programs are running locally on a server. PostGIS, which is a spatial database program, will allow storage of both geographic and non geographic data, while Geoserver only will create a link between the program and the geographic data.

The files with geographic information that are recognized by Geoserver will be stored as shown in figure 6.3. First a workspace is selected where the file should be stored. After this a store is established to a file or folder that contains geographic information. The file(s) s in that folder can later be recognized as a layer by Geoserver. Projection system and bounding box are defined in this step. Once this is done one can access the data over a network.

Data






-  Workspaces
-  Stores
-  Layers
-  Layer Groups
-  Styles

Figure 6.3 Data store in the Geoserver interface. The data store in GeoServer is arranged in workspaces, stores and layers. Further on layers can be added to group layers and they can be assigned styles.

6.2 Outgoing data

This section reviews the second step in the implementation model, which describes what operation you can perform on the incoming data. The part from the implementation model that are concerned in this section are illustrated in figure 6.4.

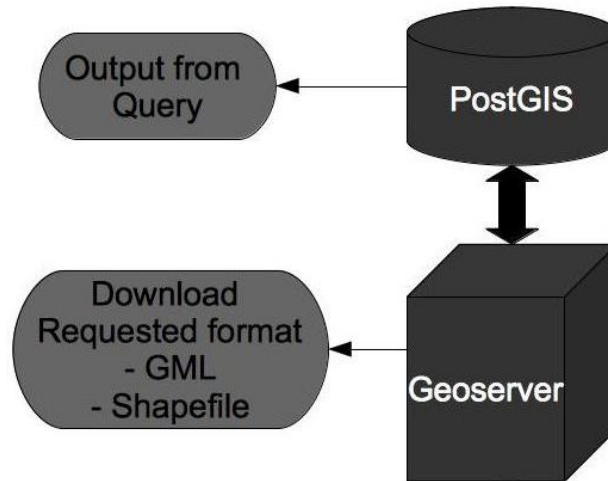
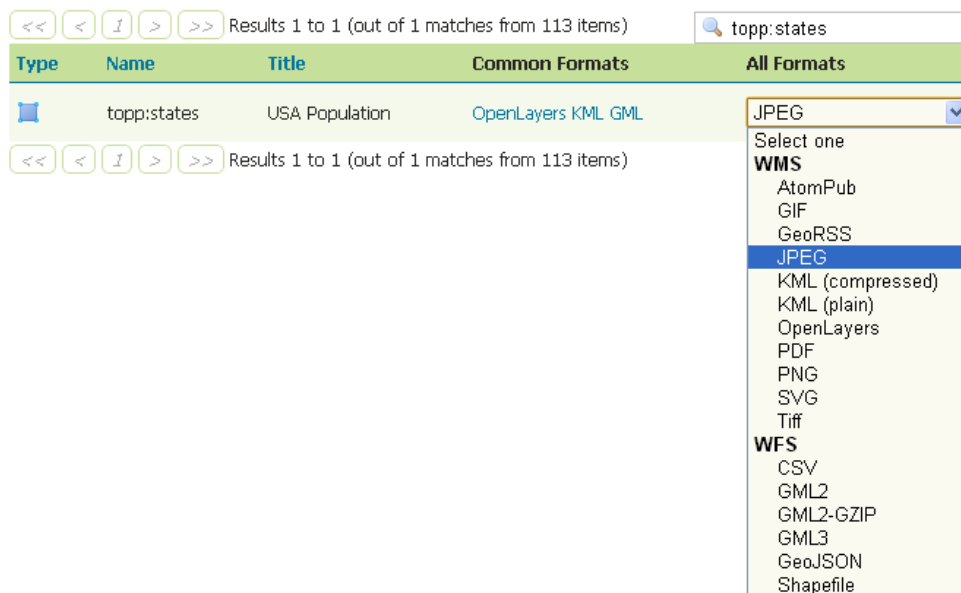


Figure 6.4 Step 2 in the implementation model – operations that can be done on incoming data.


One can perform several actions on the incoming data once a store has been established. The data stored in PostGIS can be queried and output of this could answer simple questions like who have paid for a product and which garden has the biggest area. The data files stored in Geoserver can be downloaded so one can continue to work with the data in for example a desktop GIS. Figure 6.5 illustrates how that data can be previewed and downloaded from Geoserver.

Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.



Results 1 to 1 (out of 1 matches from 113 items) topp:states

Type	Name	Title	Common Formats	All Formats
	topp:states	USA Population	OpenLayers KML GML	JPEG Select one WMS AtomPub GIF GeoRSS JPEG KML (compressed) KML (plain) OpenLayers PDF PNG SVG Tiff WFS CSV GML2 GML2-GZIP GML3 GeoJSON Shapefile

Results 1 to 1 (out of 1 matches from 113 items)

Figure 6.5 Interface for previewing data in Geoserver. Note the list of formats to the right available for download.

PostGIS can be accessed through a network if one has an authorized log in account. Geoserver can also be accessed remotely and restriction can be set to the different users. In order to download a file from Geoserver, as described above, one need to have a log in and the permission to perform this action (Geoserver, 2011 d).

The actions described in this section only target a small group that has access to the programs and the action they can perform results in something that not automatically is freely available to the public. If one like to target a wider crowd the data needs to be transmitted to an application that can be reached from either a network or the web.

6.3 Distributing the data to a network

This section describes the third step in the implementation model, which describes how to distribute the incoming data onto a network. The parts from the implementation model that are concerned in this section are illustrated in figure 6.6.

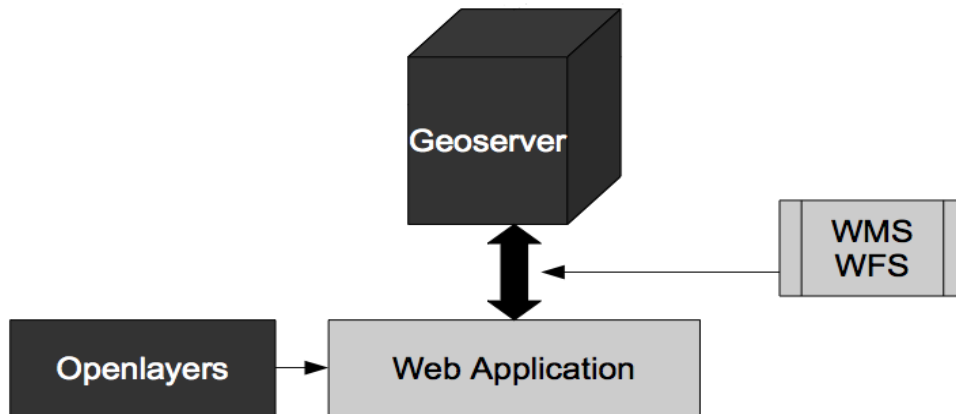


Figure 6.6 Step 3 in the implementation model – How to transfer the incoming data onto a network.

Data files with geographic information can be called from Geoserver to a web application where they can be visualized on a network. This is done through the WMS and WFS protocols that are a schema for how to distribute geographic data. The WMS request will return an image over the geographic area, while the WFS request will return the actual data over a geographic area. Both the open source JavaScript library Openlayers and Geoserver are compatible with these protocols.

Openlayers is an open source JavaScript library that can provide several map making tools to the web application. Several predefined JavaScript functions can be called and loaded onto the web application to create a web map. Example of these functions would be zooming and panning. However, Openlayers provide more functionality than styling to a map so one needs HTML and CSS in order to style the data in a preferable way.

6.4 Styling the distributed data

This section is about the fourth step in the implementation model i.e. *how to style the transferred data*. The part from the implementation model that are concerned in this section are illustrated in figure 6.7.

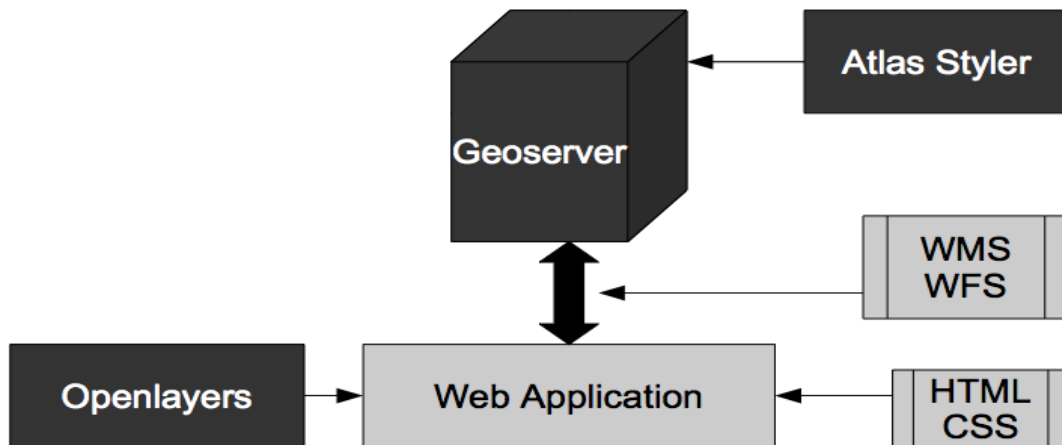


Figure 6.7 Step 4 in the implementation model – How to style the transferred data.

The web application can be styled in order to get more cartographic characteristics. The styling of the web application is done through HTML and CSS code. This code can create buttons, allow the users to add own defined symbols and much more. One thing, however, that HTML and CSS not is providing is styling on the layers from Geoserver. Styles for the files that are loaded from here in WMS format is defined XML code. The open source program Atlas Styler can generates this code for a user. This XML code can define at which zoom levels a layer should be displayed, what texture it is having and much more. Once the data is styled it is ready for publication and that is described in the next section.

6.5 Publishing of the web application

The fifth step in the implementation model is how to publish the web application. The part from the implementation model that are concerned in this section are illustrated in figure 6.8.

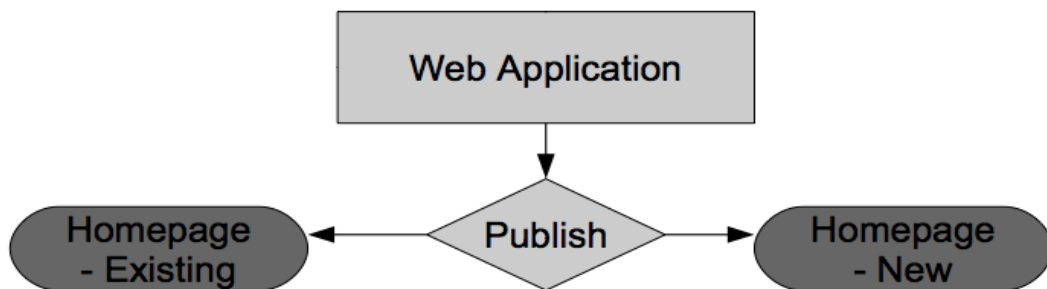


Figure 6.8 Step 5 in the implementation model – How to publish the web application.

Several users can access the web application once it has been published on a network. This can be done in two ways. The first way is that the web application is published as it is and a web based interface is built around that. The second alternative would be to integrate the created map window with an existing webpage.

To reach an even wider audience with the application it can be published on the internet. Additional security needs to be added to the application in this stage. Example of this would be to have some kind of password security in order to access the application. This can be good when working with data that is not freely available for the public.

6.6 Core functionality and editing of geographic data

The last step in the implementation model describes the core functions of the model. The parts from the implementation model that are concerned in this section are illustrated in figure 6.9.

The main workflow refers to how PostGIS, Geoserver and the web application work together. As the arrows show they are fully compatible both ways in this implementation model. That means that data added into PostGIS will automatically be updated in the web application, but it also means the opposite. Data added in the web application can also modify and update the database.

In order to update and modify PostGIS from the web application one needs to use the *Transaction WFS* protocol. The WFS protocol *GetFeature* can retrieve the actual data in GML format from PostGIS. This data can be modified and then sent back to PostGIS with the *Transaction* protocol. The *Transaction* protocol supports the operations *delete*, *create* and *update* on the retrieved data. That a user can contribute to the data set adds another dimension to the workflow, since the application is no longer just read only. One example of this could be that a form could be added to the webpage where the users could add information about objects, for example their property. These data could be sent by *Transaction WFS* to update the *PostGIS* database once the formula is completed. This could lead to a quick, efficient and cheap way to update and handle a database.

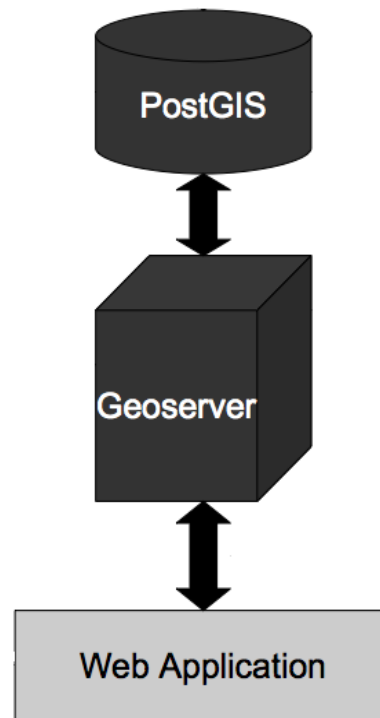


Figure 6.9 Step 6 in the implementation model - Core function of the model.

The *Transaction WFS* protocol allows editing of geographic data over a network and this can improve functionality to a web application and make it more users friendly. The functionality can be increased, as mentioned above, since *Transaction WFS* supports *Delete*, *create* and *update* of the existing data. These commands can also make the web application more users friendly since data can be added into a simple format in the web application and then be converted into the right format so the database can be update in a proper way. It is also worth mentioning that *Transaction WFS* is the only way, from the techniques described in this study, to update *PostGIS* from the web application.

7 Evaluations of the model in KwaZulu-Natal

This chapter aims to show how the workflow in chapter 6 can be used in practice when creating applications handling geographic information. As a part of the study, three test applications have been developed using geographic data from local actors in the Dukuduku area. This chapter presents the background, requirement specifications, incoming data and the functionality for each one of these applications.

7.1 Application 1: Custom map viewer

The purpose of this application is to distribute and visualize geographic information in a simple way. The application is to be broad and usable in many projects.

7.1.1 Background

When asking development project personnel about their GIS needs, one of the concerns from the project managers side was to get an easy way of displaying maps. These persons may have limited knowledge of GIS, but on regular basis need to review shape files and other sources of geographic information. Today, a method used frequently to meet these needs in KwaZulu-Natal is to distribute layered PDF files. Obviously this is not an optimal solution, leaving a motivation for the study of different approaches (Barnes, 2011).

7.1.2 Requirement specification

An application is to be developed for distribution and visualization of maps. By using a standard web browser a user should be able to connect to the application. The user should be allowed to customize a map and specify which of the layers available on the server to view. Below are the requirements in detail.

- The application should at least be compatible with Internet Explorer 8 and Google Chrome 11.
- No add-ons or extra plug-in should be required.
- The application must be executable both locally and via an Internet connection.
- The application must support common formats for geographic information.
- The map window should allow panning, zooming and the option to switch layers on or off in the current view.
- A legend should be provided with the map.
- The user should be able to view single layers or do overlays with several files at once.
- The user should not be required to have great GIS experience.
- No programming should be needed to administrate the system, for example when updating the application with layers or styles.

7.1.3 Input data

The application can display any file that is compatible with Geoserver. This means it is also possible to connect via Geoserver to remote resources such as another server or a PostGIS database. The input data used in the illustrations are mostly Shape files and GeoTIFF. These files come from MHP Geospace, one of the GIS companies working in the area. For the map background Google Maps satellite imagery is used.

7.1.4 Functionality

The functionality is described by a series of screen shots as available in Figure 7.1 - 7.4. When the user starts up the application a list of available layers from the server is presented. If needed the user can select a specific workspace from a drop down list and in that way filter the viewed list of layer. The next step is to check a number of boxes next to the layers and press the submit button. By submitting the form with the selected layers an overlay map is created together with a legend. In the resulting map the user can freely navigate, zoom and turn layers on or off in the view.

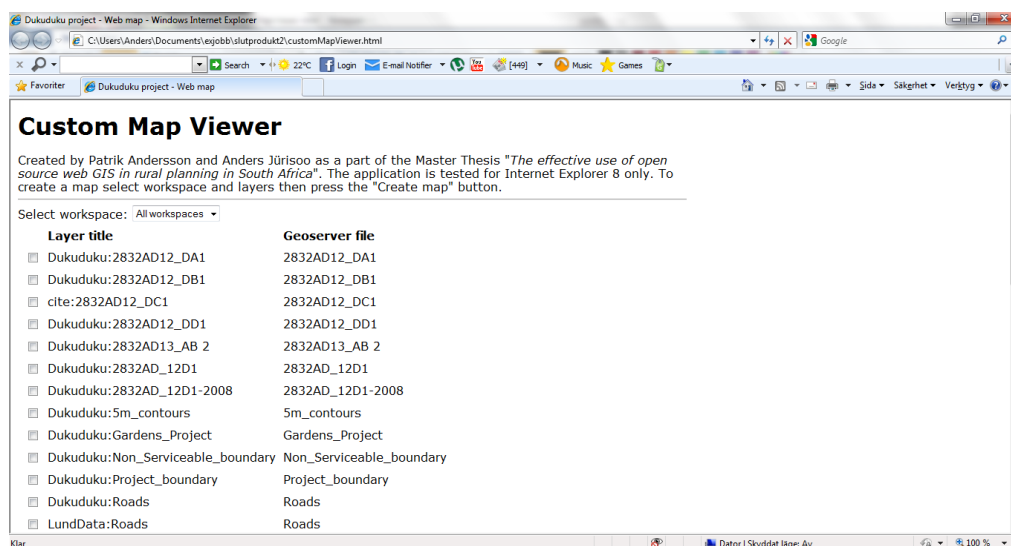


Figure 7.1. The application start up view. Here the user can see all available layers on the server.

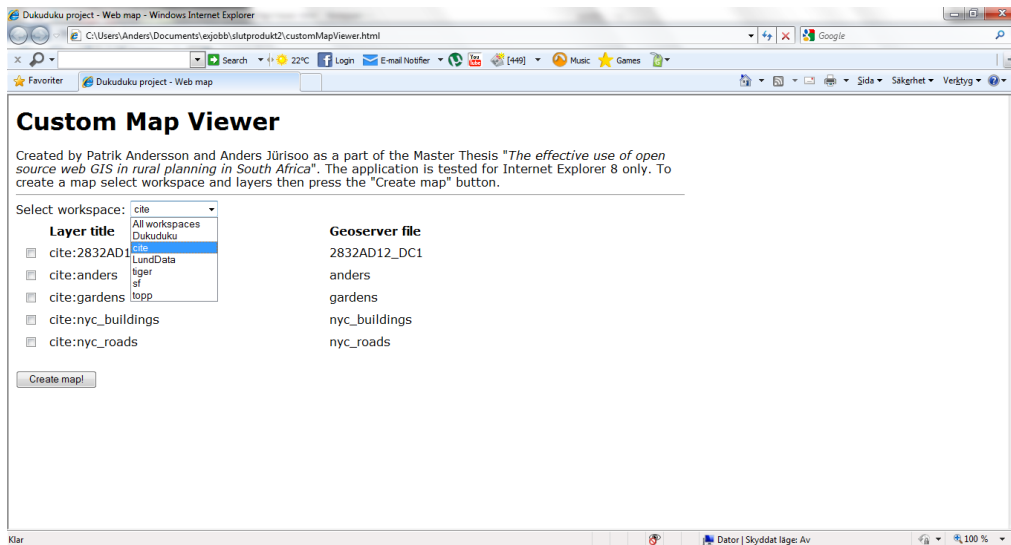


Figure 7.2. Filtering. By selecting a particular workspace from the drop down list only relevant layers are shown.

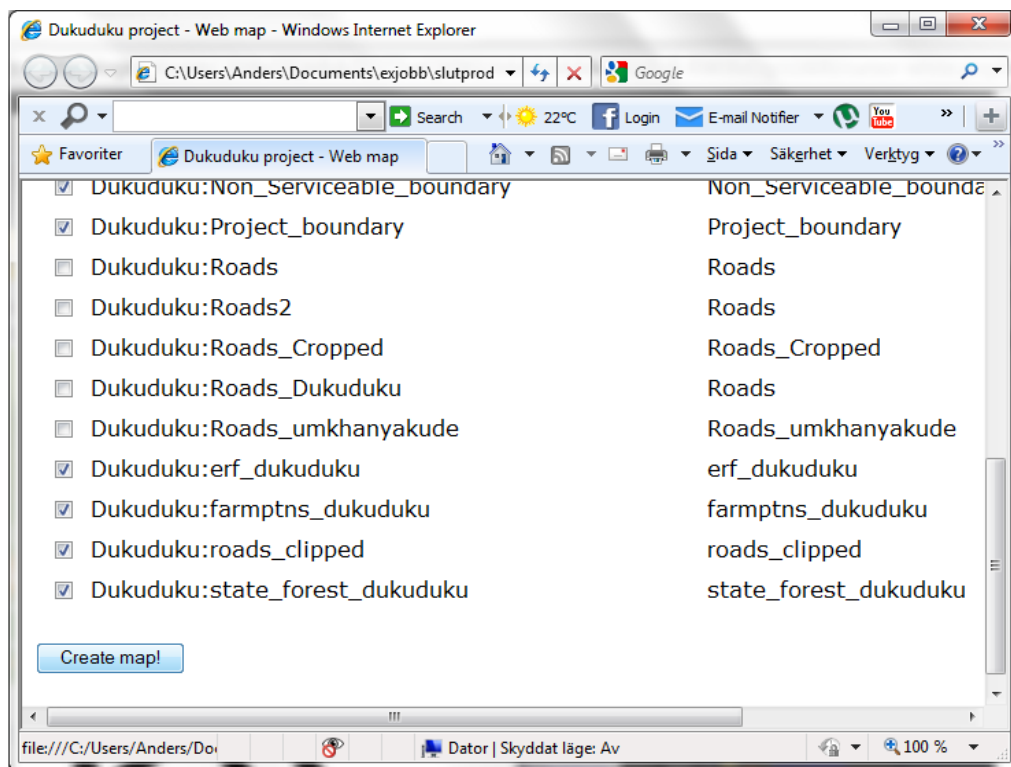


Figure 7.3. Selection. By selecting the wanted layer(s) and submitting the form the map can be created.

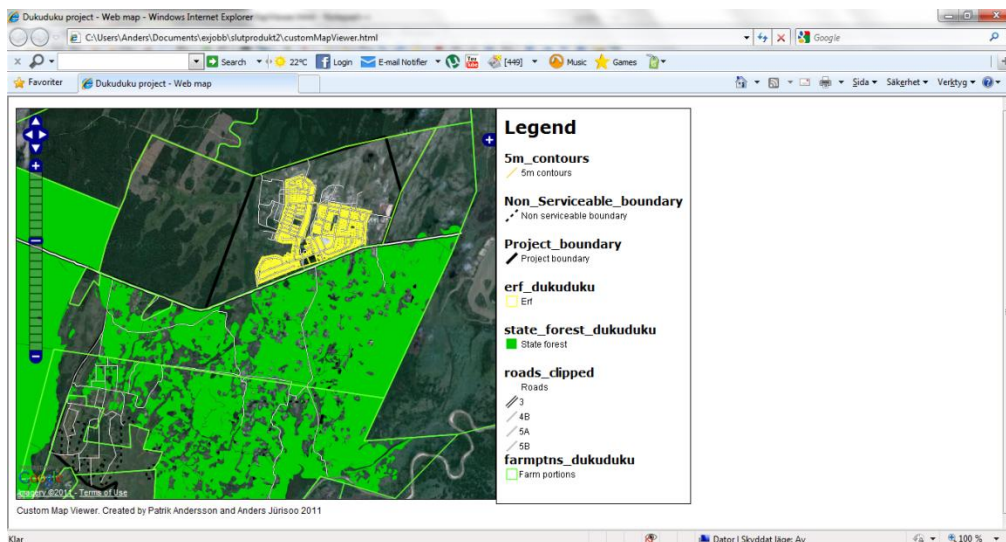


Figure 7.4. The resulting map. The user interface include basic GIS functionality such as zooming, panning and the option to turn layers on or off. On the right is the automatically generated legend.

7.1.5 Outcome and application scope

The resulting application can potentially be used within a variety of local projects. Examples could be the electricity and water projects mentioned in chapter 2, where many layers of geographic information are needed for review in the same map. The nature of the application allows any material on the server to be used for overlays which makes this application very broad.

The resulting application fulfills the most vital parts of the requirement specification, but there is some problem with browser compability. The resulting program can therefore only be used with Internet Explorer. This application is inspired by code examples from the Openlayers community (OPENLAYERS 2011b).

7.2 Application 2: Mapping informal settlements

The aim of this application is to demonstrate how the WFS-T protocol can be used for the client to remotely update the database at the server.

7.2.1 Background

As described in the introduction the study area are facing a lot of problems regarding people settling in areas where housing is not permitted. In order to keep the geographic information updated regular field measurements must be performed. The

application developed could be used as an inventory sketch tool for adding the settlements positions in preparation for future more precise measurements.

7.2.2 Requirement specification

An application is to be developed for displaying, modifying and insertion of settlements location in the Dukuduku area. By using a standard web browser a user should be able to connect to the application. Below are the requirement details.

- The application should at least be compatible with Internet Explorer 8 and Google Chrome 11.
- No add-ons or extra plug-in should be required.
- The application must be executable both locally and via an Internet connection.
- The map should view the settlements measured previous years and a suitable background layer.
- A sketch layer should be available for insertion of new settlements and editing of existing ones.
- The map window should allow panning, zooming and the option to switch layers on or off in the current view.

7.2.3 Input data

The data on the settlements are retrieved from shapefiles provided by MHP Geospace. As background Google Maps is currently used, but it is also possible to change this to OpenStreetMaps or another service provider.

7.2.4 Functionality

The functionality of the application is described by a series of screenshots as available in Figure 7.5 - 7.7. When the user starts up the application a layer with point location of settlements is presented with easily understandable symbology. These are the newest information in the application and this layer is editable. By pressing the “+” sign layers from previous measurements can be viewed together with the most up to date information.

The user can select what tool to use from a panel in the maps upper right corner. New points can be added, existing ones could be deleted or moved and the changes can be saved to persist against the database. These actions are all performed by clicking on the tool symbols and the desired location or object.



Figure 7.5. Map functionality. By choosing tools in the upper right corner the user can add points, edit existing settlements or delete features.



Figure 7.6. The available tools and the associated buttons (compare figure 7.5). From left to right: add a point (i. e. a settlement), edit a point (i. e. move a settlement), delete a point (i.e delete a settlement), save changes and finally: pan the map.

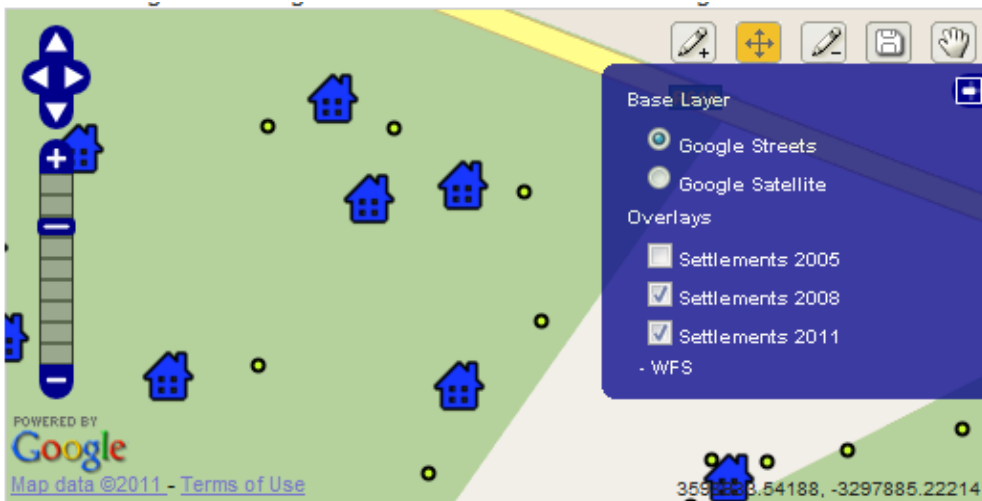


Figure 7.7. Map interactivity. The map window allows the user to zoom, pan and turn layers on/off.

7.2.5 Outcome and application scope

The resulting application can potentially be used to support mapping and inventory of informal settlements. One example of this could be when monitoring the *project serviceable borders*, where it is important to locate new settlements in restricted areas. This application allows quick marking of the rough positions of these new settlements without bringing out a geodetic measurement unit. At a later time the positions can be measured more accurately if needed with the editable layer as a starting point.

The resulting application fulfills the most vital parts of the requirement specification, but there are several improvements that could be done. For instance, create a form where the user can paste coordinates or write them together with other information. This would lead to a more accurate system compared to just clicking them out. If wanted, small configurations will also allow line or polygon data to be inserted and updated.

There is some problem with browser compability. The resulting program can therefore only be used with Internet Explorer. This application is inspired by code examples from the Openlayers community (OPENLAYERS 2011c).

7.3 Application 3: Garden project data viewer

The purpose of this application is to distribute and visualize geographic information regarding the permaculture homesteads project. The application enables high interactivity and access to layer attributes.

7.3.1 Background

A vast area of the Dukuduku forest falls either within the Isimangaliso wetland park or other environmentally sensitive areas. This causes a big problems for the farming and agricultural activity in these area since farming within these zones are illegal.

Permaculture gardens could provide a solution to the farming dilemma described above. The reason for this is that a permaculture garden takes up less space, have no need for pesticides and it provides the household with variety of fruits and vegetables. Permaculture comes from a combination between the words permanent and agriculture. The meaning of permaculture gardens is a garden that can handle itself and provide crops year after year without human impact.

Some of the main difference between a permaculture system and traditional farming is that the permaculture gardens take up less space physical space and they provide a wider variety of crops. The traditional farming is usually a field planted with one kind of crop that could be supplied to a market, while the permaculture in first hand provide the need of a single household. A well maintained permaculture could also give excess crop that could be sold on the local markets.

The idea of the permaculture system is that the ground is prepared in a certain way and the planted with different kind of fruit, vegetables and other crop so that the system is handling itself with minor or none human impact. A permaculture system could more or less be established anywhere in the world, but the construction of it varies depending on the local climate.

The location of the permaculture gardens is located close to the homestead. The reason for this is so that the rain water from the roof can be led to a ditch and then into the actual system.

The permaculture system used for the Khula village and Dukuduku forest could be described in two phases. The first one would be construction of the system and the other one would be planting of the system.

The construction of the permaculture system that is used in Khula village and Dukuduku forest starts on ground level and are then built up from there by different favorable and useful materials. Cardboards are laid on the ground. Grass, sawdust, chicken manure and other beneficial materials are added to the structure. Tree stumps are added the constructions in order to create a walk way. These will eventually molder down and provide the system with nutrient. Plastic bottles, with the bottom removed, are also used in the construction in order to create a water mechanism for manual irrigation. A permaculture system should need no irrigation once there are big trees like the banana and palm trees that are full grown and can provide shade with their leaf and in that way prevent evaporation, but until then extra watering might be needed. See figure 7.8.



Figure 7.8. The construction of a permaculture garden. Note the spiral towards the middle

The construction is built in a spiral with a ditch that loops from where the rainwater falls down from the roof until the middle of the permaculture garden. The ditch is constructed with a certain elevation in order for the water to flow smoothly into the middle. Figure 7.9 displays what the permaculture system looks like when it has been covered by soil. Bigger trees, like banana trees are planted in this stage in order to provide shade for the under growing vegetation. The leaves and fruits that are falling down into the permaculture garden will also provide the system with nutrient.



Figure 7.9. A newly finished permaculture garden. With time the soil will be completely covered with plants and trees.

The permaculture project involves 180 permaculture systems or also called food gardens. All these gardens are measured with a GPS once they are done in order to keep track of them. These GPS points are added into a desktop GIS and information about the garden, owner picture of owner and much more is added into this system.

The permaculture project involves several actors. It is all from the Department of Cooperative Governance and Traditional Affairs that provides the funding for the project to the local beneficiaries that are working with the project.

A web based open source GIS system could easily keep all these people involved in the project updated about the current status and it could provide the same functionality as the desktop GIS, but with a much lower cost. The GIS costs for the project are budgeted for R 12,000, or approximately 1200 Euros, and that cost could almost be eliminated with a web based GIS, with some exception for minor maintenance costs.

7.3.2 Requirement specification

An application is to be developed for handling the geographic information and metadata regarding the permaculture homesteads project. By using a standard web browser a user should be able to connect to the application. The user should be

allowed to search and query the map for specific attributes regarding the gardens. A high level of interactivity is required. Below are the requirement details.

- The application should at least be compatible with Internet Explorer 8 and Google Chrome 11.
- No add-ons or extra plug-in should be required.
- The application must be executable both locally and via an Internet connection.
- The map should view the garden locations and a suitable background layer.
- The map window should allow panning, zooming and the option to switch layers on or off in the current view.
- By clicking the point location of a garden attribute data should be viewed. A photograph should also be viewed.
- The user should be able to query the gardens on their attribute using CQL.

7.3.3 Input data

The data on the gardens used in the application is a shapefile supplied by the local actor MHP Geospace. The shapefile contains several attributes for example the person responsible for the garden, a score or ranking and what plants currently growing in the garden.

There is also a project boundary layer, which describes the project area. As background Google Maps is currently used, but it is also possible to change this to OpenStreetMaps or another service provider.

7.3.4 Functionality

The functionality of the application is described by a series of screenshots as available in Figure 7.10 - 7.13. When the user first starts up the application a layer with the permaculture gardens is presented together with some instructions. From here the user can click on gardens to get more information about them, or query the layer. When a garden is being clicked the associated attributes is shown in a table below the map window. On the side of the map is the image section where photography is also showed when a point is clicked. If the user wishes to query the layer, this is done in the form above the map window. One of the options is to use CQL, common query language which is similar to the SQL syntax. Another option is to query on the geometries ID, or ID from the database table.

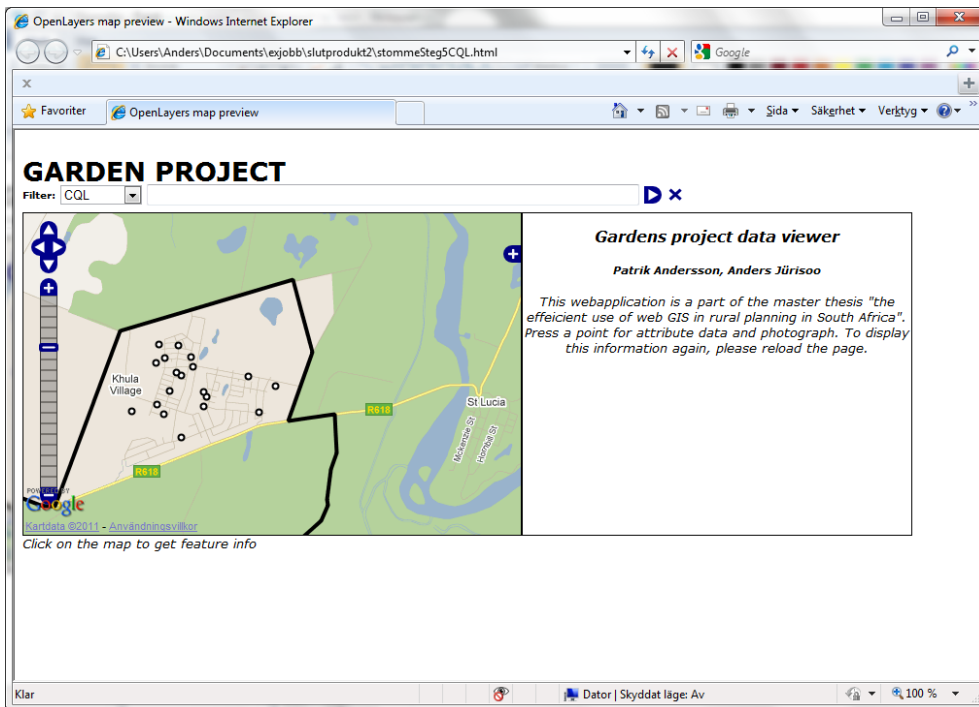


Figure 7.10. Application at start up. The user can move in the map, click points or query the layer using the filter tool.

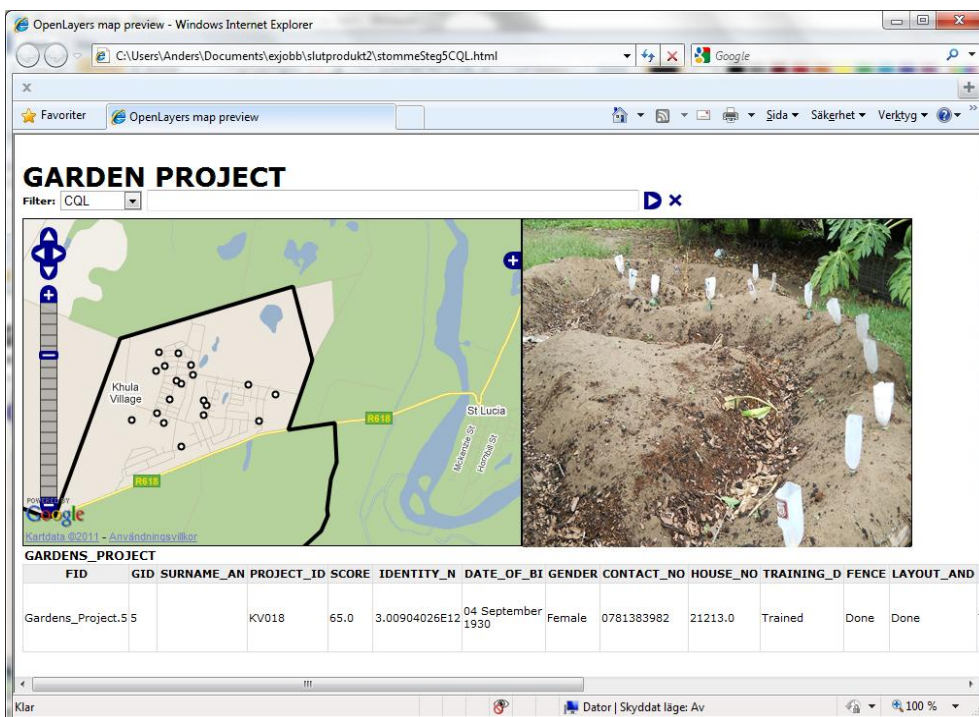


Figure 7.11. Attribute data. When clicking a point a photograph is shown together with all the attributes.

OpenLayers map preview - Windows Internet Explorer

C:\Users\Anders\Documents\exjobb\slutprodukt2\stommeSteg5CQL.html


Google

OpenLayers map preview

GARDEN PROJECT

Filter: CQL score < 100 AND gender = 'Female'

CQL
OGC
FeatureID



Kisumu Village

PROVIDED BY Google

Kardata@2011 - Användningsvillkor

GARDENS_PROJECT											
FID	GID	SURNAME_AN	PROJECT_ID	SCORE	IDENTITY_N	DATE_OF_BI	GENDER	CONTACT_NO	HOUSE_NO	TRAINING_D	FENCE_LAYOUT_AND
Gardens_Project.13	13		KV057	75.0	7.80710063E12	10 July 1978	Female	0795634493	20728.0	Trained	Done Done

Filter: CQL score < 100 AND gender = 'Female'

Figure 7.12. Queries. By using the filter tool the user can create queries or search after a specific id.

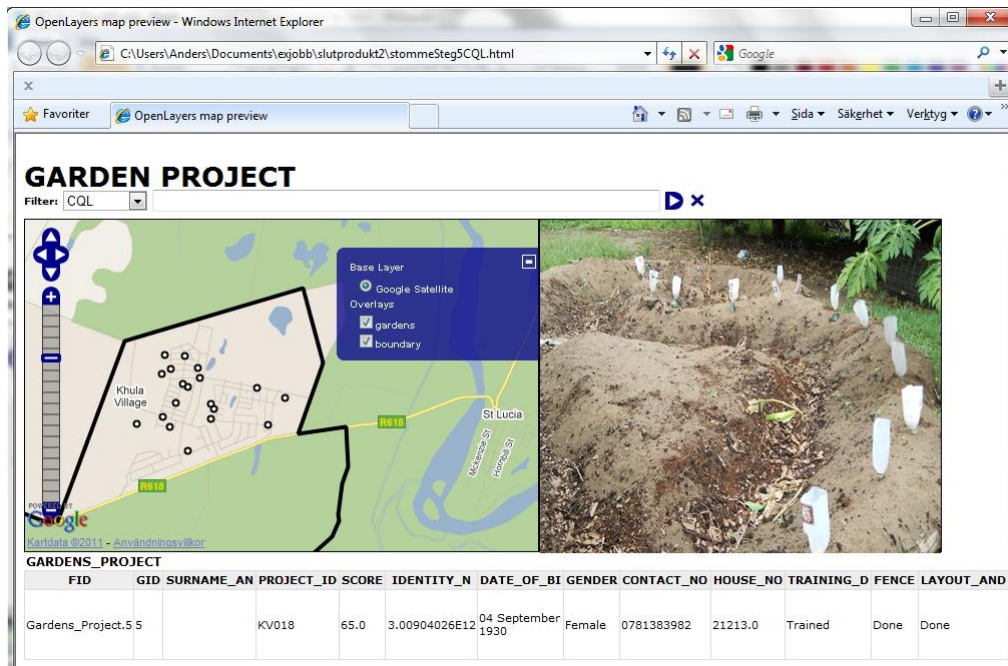


Figure 7.13. Map interactivity. The map window allows the user to zoom, pan and turn layers on/off.

7.3.5 Outcome and application scope

The resulting application has a natural field of use in the 180 food gardens project. By using the web GIS solution of the application it will be accessible by a larger group of potential users and administrators. Previously the system was managed with a Microsoft Excel file, where no map was available. While the map offers greater clearness and overview of the data, a drawback is that new information must be added directly to the database. See application 2 for an example of insertion directly through the web browser.

In addition to the 180 food gardens project, the system could also be applicable in other scenarios. One example of this is the *African Conservation trusts* wish for a tool displaying locations of cave paintings together with photographs.

The resulting application fulfills the most vital parts of the requirement specification, but there is some problem with browser compability. The resulting program can therefore only be used with Internet Explorer. This application is inspired by code examples from the Openlayers community (OpenLayers 2011d).

8 Discussion

8.1 Using open source

Using open source tools are often a cheap alternative to commercial proprietary applications. Other benefits are the insight in how the software is implemented and the fact that open source is supported by a growing community of users and developers. At this stage all the software used is free of charge, even though there is sometimes no guarantee they will stay free in the future.

Problems concerning open source software include the lack of documentation and help sections. The developer sometimes must rely on the knowledge of the open source community which is spread in forums and mailing lists. This in combination with slim interfaces and often unstable versions can lead to hardships when developing applications. On the other hand, there are also many code scraps and examples available on the internet. This was utilized when creating the test applications.

8.2 Implementation model

The implementation model gives a template for how to implement web GIS applications with open source tools. The software and connections described in the workflow (figure 6.2) is already used widely today, but this study has also aimed to present the whole process in a clear and pedagogic way. This means the model is suitable for education purposes or for a developer looking for more insight in the open source web GIS field.

The components in the implementation model were selected to enable the creation of the three test applications. With the functionality stated in the requirement specifications the model needed to work as a framework with wide openings for variations and different implementations. The details and the justification of the technical choices are presented in the following sections.

As the discussion concerning the applications will show the model can be used with great success. The use of a model like this can speed up the development of an application, especially since the documentation of open source software is not always the best.

8.3 Application 1 – Custom map viewer

The purpose of the Custom Map Viewer is to allow easy distribution and visualization of geographic information. By using a standard web browser the user can create a map and choose freely from the available layers on the server. This can be valuable when project maps are frequently used and updated.

The custom map viewer could potentially be used as a tool to help tackle problems regarding rural planning and areas of informal settlements. It can be used to visualize infrastructure, terrain and other features in areas of interest. This application is broad in the sense that it can be used in any context where there is a need for layered map viewing, that means not only in areas of rural planning and informal settlements.

The application was created using the implementation model discussed in chapter 6. Alternatives considered included MapServer and Tiny-OWS but while they meet the needs of the custom map viewer it did not support operations needed by other planned applications. This is also reviewed in the discussion for application 2, mapping informal settlements.

Among the benefits of the developed application is the *accessibility*. As described in the application background many actors need to access the geographic information at different times and places. These people may not have access to standard desktop GIS software why the simple web browser platform is very effective. All that is needed is an internet connection. Furthermore the user can directly accesses all the layers on the server so there is no problem with circulating files or versioning issues.

The drawbacks of the system in its current form are partly connected to the accessibility. The user needs a good internet connection, especially to view raster images such as *orthophotos*. A method to reduce the loading time is to implement *caching* of big files. In this way the tiles creating the map will be prepared, stored and loaded more efficiently.

In addition to the caching mentioned above, the application functionality could be improved in particular regarding information on the layer attributes. The attribute tables could be viewed and the option to query the data could be implemented as in the case of application 3.

8.4 Application 2 – Mapping informal settlements

While application 1 and 3 can be very useful in an area with informal settlements, this application deals directly with the settlements. A user can add new points and directly save them to a database. In its current form this is done by clicking out settlement locations on a map. The accuracy of this method is not very high why the applications main use is that of a rough sketch tool.

The techniques and software used to create the application are meeting the demands of the functionality in the requirement specification. It also matches the criteria of open source which is the theme of the study. Compared with MapServer, which is another market-leading environment for building web map services, it is undisputable better since PostGIS-GeoServer-OpenLayers support the WFS-T schema needed for database insertion. This is not the case for a MapServer implementation. Another examined option is using postgis and Openlayers together with Tiny-OWS. This

alternative was rejected due to a smaller community support and more advanced employment.

Another issue is whether this application can be used to counteract the problems regarding informal settlements. In South Africa informal settlements are a substantial part of the population housing. Not only in the study area but in all mayor rural and urban areas shanty houses and improvised structures can be found. To deal with this housing situation and improve it one needs to consider the economic, cultural, social and technical aspects of a vast and complex problem. When discussing these issues with local development personnel they were especially interested in the opportunities of open source GIS. One of the topics was licenses and the costs of GIS software and if it is possible to see a cheaper, more effective and open GIS system in the future. This application gives a partial answer to the question and serves as an example of what can be achieved at this stage with a moderate work effort. The conclusion must be that it is possible; the future will hold more options regarding open source web GIS applications.

The authors view of the usefulness of this application is that while it shows some interesting concepts, it need to be developed further and made more flexible to be used in reality. However, this application points out interesting questions such as how users can contribute to a map service. Allowing users to directly insert data is certainly a delicate matter. The GIS professionals at local companies in South Africa was interested in this application but raised the question about security and reliability. What is really intriguing is the increase of mobile technology and real time map services available to a large group of users. Even in the informal settlements areas one can find cell phones with built in GPS technology. Maybe this can be utilized in the future for mapping informal settlements.

8.5 Application 3 – Garden project data viewer

The purpose of the Garden project data viewer is to distribute and visualize geographic information regarding the permaculture homesteads project. The application enables high interactivity and access to layer attributes. In this application the focus are on a few layers and their attribute data. The user can interact and click points as well as query the data. Images are shown as part of the attributes for point objects.

The techniques used were chosen since the database needed to be capable of holding spatial objects. It also needed to be open source and compatible with a server and a suitable back end. Taking the other applications into consideration the workflow model in chapter 6 was assessed to be the most effective option.

How can this application be used to assist the development work and rural planning in areas of informal settlements? GIS software utilizes attributes in maps as a powerful way to communicate information. This can in turn help with analysis and decision making for instance in the process of carrying out development projects. In this

application a regular web map is enhanced with attributes and images, allowing the user to search and query the material. Like the other applications this one also uses the web browser platform which is widely spread. However, this application merely demonstrates the attribution of maps and how it can be incorporated into web map services.

The general methodology could be utilized in many ways, for instance in marketing purposes. Another example is the organization *African Conservation Trust* who was looking for a map tool to view cave paintings based on location. This could be achieved using the same system. Numerous other adaptations could be mentioned and the garden project is a good example of how the system could be used to support activities in informal settlements and the local communities.

An interesting end note to this discussion is how the test applications can be seen all together and the possibilities to merge functionality. The first application has shown how a general viewer could work with multiple layers and legend functionality. The second application examined the area of user input of geographic data. The last application described above show how a spatial database can be connected to the system and how attributes can be retrieved and presented. To merge these functionalities could allow for creation of a broad and complete system for use in a variety of tasks.

8.6 Future work

While this study has been very broad and investigating, the application developed also shows that the workflow model can be used to create powerful applications. The test applications all have areas they can be improved at, but the concepts of the model has been shown to work.

Areas where the applications can be developed further are functionality, design and methods for increasing the scale of users and amount of information. At this stage the model has only been tested by a handful of users at the same time and all at a good internet connection. As mentioned in the custom map viewer the ability to use aerial photography or raster images is also limited. This is a big setback and need to be fixed to put a system to use. Solving this issue can improve application 1 further, since the othophotos have better resolution and are more up to date than the Google satellite image.

Working further with the *Transaction WFS* on application 2 would improve the usefulness and make it more user friendly. With basic programming a form can be created for use in the application. Here can data be added about location of coordinate, coordinate system, the person who measured it on what date and so on. This data can be sent through the *Transaction WFS* to PostGIS and it that way allow remote updates to a database.

Another area that has been growing is the use of maps in smart phones. Development of a application that could fit to use on a small screen would be helpful. One of the reasons that this is really interesting is because of the smart phones built in GPS and the portability of the phone itself. This development combined with the *Transaction WFS* can create an excellent tool for doing quick mapping in the field. It will not require any large amount of equipment and all the notes taken can be viewed from several people at once.

9 Conclusions

The purpose of this study was to develop a method how to collect, store and visualize geographic data in applications of rural planning and informal settlements. This purpose was solved by creating a workflow (*figure 6.2*) and three implementations were made from this workflow (*chapter 7*). During this study it has been shown that the developed workflow is a good and efficient method for creating interactive web GIS systems. An indication that confirms this is that the produced applications could meet the goals in the requirement specifications that were inspired from the needs of local actors. Many other fields of application are possible why these examples should only be seen as a demonstration of the workflow capabilities.

The workflow is open source based, which means that it provides a free of charge platform. It is supported by a large and growing open source community. An effect of this is that the functionality in the software used is expanded rapidly by a large group of developers. Another benefit is the insight and adaptability of software as all the code is freely accessible. The development, maintenance and updating of an open source GIS application is of course a cost, but with the license types proposed in the model a finished application can be distributed and widely used without any further cost.

The study area is subject to many social, economic and environmental development projects where geographic data are handled on a daily basis. The rural part of KwaZulu-Natal is a dynamic area with many actors and stakeholders. The GIS solutions that are achievable using the workflow are suitable for this group of users as they utilize a common interface in form of a standard web browser. In that way governments, NGOs, companies and locals from the area can access the applications as long as a computer with internet connection is available. An effect of this is that the workflow allows a large group of people to access the geographic data.

The open source workflow could potentially facilitate operations concerning rural planning. In every stage of rural planning geographic information is a vital tool for visualization and decision making. For example, by using the workflow a workgroup can be given access to up to date geographic information at any place and time. This could provide a great tool for planning meetings and distribution of maps to the public and other involved actors.

Another goal of the study was to allow development of applications that are easy to use. This has been fulfilled as the test applications interfaces are based on basic actions as clicking or scrolling with the mouse. No programming or advanced computer skills are required. The system is also easy to administrate for example when adding new geographic data.

The first implemented example from the workflow shows a basic map viewer over the Dukuduku forest. This example describes the usefulness of an open source web based workflow. This application could for instance be a useful tool in rural planning

meetings in developing countries. The reason for that is that everyone involved in the projects can have access to all geographic data over the project area as long as there is an internet connection provided.

The second example that was implemented from the workflow describes how geographic data can be edited remotely in a web browser. This is done through the *Transaction WFS* protocol and it expands the users' interactivity with the map as a database can be updated remotely. In the example implementation this is achieved by clicking out new points, but a system could also be designed where a user enter numerical values, coordinates and text in forms. This can be proven very useful when dealing with many people and want to have quick results.

The last implemented example shows the usefulness of the workflow in one specific project. The concerned project is a garden project and functionality fit for the purpose has been added to that example. A CQL filter has been added to allow a user to find gardens that fulfill specified requirements. It will also provide information and a photo of the garden once a user click on it. This application has shown that attribute data can be used effectively within software based on the developed workflow.

In summary, the model developed has proven to be useful and can be applied to create suitable, cost effective and user-friendly map applications. The three implemented examples also show that this workflow is highly adaptable depending on the circumstances of use.

References/Bibliography

ACER (Africa) Environmental Management Consultants, 2006, *Interdepartmental task team: finding a solution for Dukuduku*

ACER (Africa) Environmental Management Consultants, 2002, *Dukuduku Research Project*

Adams, M, Sibanda, S & Turner, S 1999, *Land tenure reform and rural livelihoods in southern Africa*, Natural resource perspective, Overseas Development Institute (ODI), no 39

Althos 2011, *GPS basics, digital image, Althos*, accessed 7 June 2011, <http://www.althos.com/sample_diagrams/ag_GPS_Operation_New_low_res.jpg>

Apache, 2004, *Apache Software foundation*, retrieved 15 May 2011, <<http://www.apache.org/licenses/LICENSE-2.0>>

Atlas Styler 2011, *Wikisquare, Bonn, Cologne*, viewed 5 April 2011, <<http://en.geopublishing.org/AtlasStyler>>

Barnes, D. L. 2011, *CEO Enhance Strategies*, verbal communication, 1 April, 2011.

BSD, 1998, *University of California, Berkley*, retrieved 8 May 2011, <<http://www.opensource.org/licenses/bsd-3-clause>>

CC-BY-SA, 2011, *Creative-commons*, retrieved 20 April 2011, <<http://creativecommons.org/licenses/by-sa/2.0/legalcode>>

CMC, 2011, *CMC international*, Dallas, Texas, Retrieved 20 May 2011, <<http://www.cmcus.com/productcart/pc/ArcGIS-ArcView-10-Concurrent-User-p314.htm>>

Eklundh, L. 2006. *Introduktion till geografisk informationsbehandling*. In: Eklundh, L. ed. *Geografisk informationsbehandling*. Formas, Stockholm.

Eliris 2011, *Geometric forms in map projections, digital image, Eliris Inc*, accessed 7 June 2011, <<http://hosting.soonet.ca/eliris/gpsgis/LectureImages/projections.gif>>

FOA 2011, Faculty of Engineering, Ubon Ratchathani University, Ubon Ratchathani, viewed 28 August 2011, <http://www.agri.ubu.ac.th/sura/GIS_METEO/Layers.gif>

Hall, B and Leahy, M (eds.) 2008, *Open Source Approaches in Spatial Data Handling*, Springer, Berlin

Harrie, L. (ed.)(2008) *Geografisk informations-behandling*. Fourth edition. Stockholm: Formas.

Geoserver 2011 a, OpenGeo, New York City, New York, viewed 7 April 2011, <<http://docs.geoserver.org/stable/en/user/introduction/history.html>>

Geoserver 2011 b, OpenGeo, New York City, New York, viewed 7 April 2011, <<http://docs.geoserver.org/stable/en/user/introduction/overview.html>>

Geoserver 2011 c, OpenGeo, New York City, New York, viewed 8 April 2011, <<http://docs.geoserver.org/stable/en/user/styling/sld-introduction.html>>

Geoserver 2011 d, OpenGeo, New York City, New York, viewed 7 April 2011, <http://docs.geoserver.org/stable/en/user/security/sec_roles.html>

GIS, 2011, ESRI, Redlands, California, retrieved 10 May 2011, <<http://www.gis.com/content/what-gis>>

Google, 2011, Google Inc, California, retrieved 20 April 2011, <<http://code.google.com/apis/maps/terms.html>>

GPL, 1991, Free software foundation, Boston, retrieved 8 May 2011, <<http://www.opensource.org/licenses/gpl-2.0.php>>

Harrie, L. and Hauska H. 2008. *Koordinatsystem, kartprojektioner och referenssystem*. In: Harrie, L. ed. *Geografisk informationsbehandling*. Formas, Stockholm.

Horemuz, M. 2010. Chapter 11-12 In: KTH, Lantmäteriet, Lund University. *Geodetisk och fotogrammetrisk mättnings- och beräkningsteknik*.

Housing and Land Use Regulatory board 2007, Comparison of the Raster and Vector Models (Table), GIS cookbook, HLURB, Philippines, viewed 5 April 2011 <<http://cookbook.hlurb.gov.ph/>>

Housing and Land Use Regulatory board 2007, Comparison of the Raster and Vector Models, GIS cookbook, HLURB, Philippines, viewed 5 April 2011 <<http://cookbook.hlurb.gov.ph/files/images/RasterVector.preview.jpg>>

IBM 2011, Geodetic coordinates, digital image, IBM, accessed 7 June 2011, <<http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.spatial.doc/latlong0.gif>>.

Jiang, B 2003, 'Beyond Serving Maps: serving GIS Functionality over the Internet', in M.P Peterson (ed.), *Maps and the Internet*, Elsevier Science B.V, pp. 147-158

Keates, J. S. (1996). *Understanding Maps*. Harlow: Addison Wesley Longman.

Lewis, M. Paul (ed.), 2009. *Ethnologue: Languages of the World, Sixteenth edition*. Dallas, Tex.: SIL International. Online version: <http://www.ethnologue.com/>.

LGPL, 2007, *Free software foundation*, Boston, retrieved 20 April 2011, <<http://www.opensource.org/licenses/LGPL-3.0>>

Marianne Breinig 2011, *Cartesian Coordinates*, digital image, The University of Tennessee, Department of Physics and Astronomy, accessed 7 June 2011, <<http://electron9.phys.utk.edu/vectors/images/p20.gif>>.

MIT, 2011, *Open Source Initiative*, retrieved 10 May 2011, <<http://www.opensource.org/licenses/MIT>>

National Forests Act 1998. (c.7), Cape Town:OFFICE OF THE PRESIDENT

Open Geospatial Consortium 2005, Web Feature Service Implementation Specification, OGC 04-094, Open Geospatial Consortium, viewed 18 April 2011, <http://portal.opengeospatial.org/files/?artifact_id=8339>.

Open Geospatial Consortium 2006, OpenGIS Web Map Server Implementation Specification, OGC 06-042, Open Geospatial Consortium, viewed 16 April 2011, <http://portal.opengeospatial.org/files/?artifact_id=14416>.

Open Geospatial Consortium 2007, Styled Layer Description profile of the Web Map Service Implementation Specification, OGC 05-078r4, Open Geospatial Consortium, viewed 8 April 2011, <http://portal.opengeospatial.org/files/?artifact_id=22364>.

Openlayers 2011, Open Source Geospatial Foundation, Vancouver, British Columbia, viewed 7 April 2011, <<http://openlayers.org/>>

Openlayers 2011 b, Open Source Geospatial Foundation, Vancouver, British Columbia, viewed 7 April 2011, <<http://openlayers.org/dev/examples/google-reproject.html>>

Openlayers 2011 c, Open Source Geospatial Foundation, Vancouver, British Columbia, viewed 9 August 2011, <<http://openlayers.org/dev/examples/wfs-protocol-transactions.html>>

Openlayers 2011 d, Open Source Geospatial Foundation, Vancouver, British Columbia, viewed 9 August 2011, <<http://openlayers.org/dev/examples/getfeatureinfo-control.html>>

PostGIS 2011, Refraction research, Victoria, British Columbia, viewed 8 April 2011, <<http://postgis.refrations.net/>>

Samhassan 2011, Differential GPS, digital image, Samhassan, accessed 7 June 2011, <<http://samhassan.com/images/difnav.gif>>

Shah,A 2011 'Poverty around the world', Global Issues, retrieved 1 February 2011 <<http://www.globalissues.org/article/4/poverty-around-the-world>>

UN Habitat 2003, Surveying and land information management for secure land tenure, Retrieved 25 January 2011, <http://www.unhabitat.org/downloads/docs/1523_80316_land_information.pdf>

UNESCO 2011, UNESCO, Paris, France, viewed 14 April 2011, <<http://whc.unesco.org/en/list/914>>

Worboys, M & Duckham, M. (2004). *GIS – A Computing Perspective*. Second edition. Unknown: CRC Press LLC

World Health Organization 2010, *People living in informal settlements*,
Retrieved 1 February 2011,
<<http://www.who.int/ceh/indicators/informalsettlements.pdf>>

Appendix 1: EPSG SRID Example

SOURCE - <http://spatialreference.org/ref/epsg/3785/html/>

```
PROJCS["Popular Visualisation CRS / Mercator",
  GEOGCS["Popular Visualisation CRS",
    DATUM["Popular_Visualisation_Datum",
      SPHEROID["Popular Visualisation
Sphere", 6378137, 0,
        AUTHORITY["EPSG", "7059"]],
      TOWGS84[0, 0, 0, 0, 0, 0, 0],
      AUTHORITY["EPSG", "6055"]],
    PRIMEM["Greenwich", 0,
      AUTHORITY["EPSG", "8901"]],
    UNIT["degree", 0.01745329251994328,
      AUTHORITY["EPSG", "9122"]],
      AUTHORITY["EPSG", "4055"]],
    UNIT["metre", 1,
      AUTHORITY["EPSG", "9001"]],
    PROJECTION["Mercator_1SP"],
    PARAMETER["central_meridian", 0],
    PARAMETER["scale_factor", 1],
    PARAMETER["false_easting", 0],
    PARAMETER["false_northing", 0],
    AUTHORITY["EPSG", "3785"],
    AXIS["X", EAST],
    AXIS["Y", NORTH]
```