GU-ISS-09-1 ● May 2010 ISSN-1401-5919
Research Reports from
the Department of Swedish
University of Gothenburg
<http://www.svenska.gu.se/publikationer/research-reports-department-of-swedish/>

MapServer at Språkbanken

version 0.1

Dana Dannélls

Språkdata, Institutionen för svenska språket
Göteborgs universitet, Box 200, SE-405 30 Göteborg

CONTENTS

1 Introduction 1
1.1 Mapserver 1
1.2 Mapserver at språkbanken 3
1.3 The GIS dataset 3

2 Litteraturbanken's geographical name entity recognition 4
2.1 Placename database 4
2.2 Coordinates search 5
2.3 User case 6
2.3.1 Static map 6
2.3.2 Interactive map 7

References 7

A Coordinates search 9
A.1 Example 1 9
A.2 Example 2 10

B User case 12
B.1 XML specification 12
B.2 Static map 14
B.3 Interactive map 15

List of Figures

1 The server-client architecture of Mapserver. 2

2 Static map for place names: Norra Bondökallen, Södra Bondökallen and Östan. 14
3 Static map for the place named Kalkskär. 15
4 Interactive map for the place names that are specified in Appendix B.1. 16

1 INTRODUCTION

The combination of place name recognition and geographical information systems that visualise geographical information found in written natural language texts has become a powerful aid to gain valuable knowledge and increase text understanding. With the rise of the Internet and commoditization of hardware, digital maps can today ensure a convenient and efficient rendering of geographical images in which information about place locations are shown in real time. Probably the most known source for providing a catalogue of the contents of tens of millions of computers on the Internet, providing free of charge digital maps for anyone to use and dynamically generate, is the Google Map server.

As the development of digital maps supported by Google is driven by the needs of industry, natural resource managers, etc. whose requirements are ranging from weather maps displayed on weather rapport to driving instructions obtained from GPS, geographical locations which are found in literary texts may no longer be available in this pool of modernised digital maps. Moreover, freely available digital maps are naturally not optimized for all kinds of applications; those maps found on the Internet are often copywrite protected which means they may not officially be edited or published without the creator's approval.

This lack of flexibility and the need to point to geographical locations of places which are recognised in texts that are available at *The bank of literature* (Swe. Litteraturbanken),[1] are two of the main reasons why we decided to investigate in an alternative technology called MapServer (Kropla, 2005).

1.1 Mapserver

Mapserver is an open source Geographic Information System (GIS) development environment for producing maps.[2] It is a tool for rendering geographic

[1] <http://litteraturbanken.se/>
[2] <http://www.mapserver.org/>

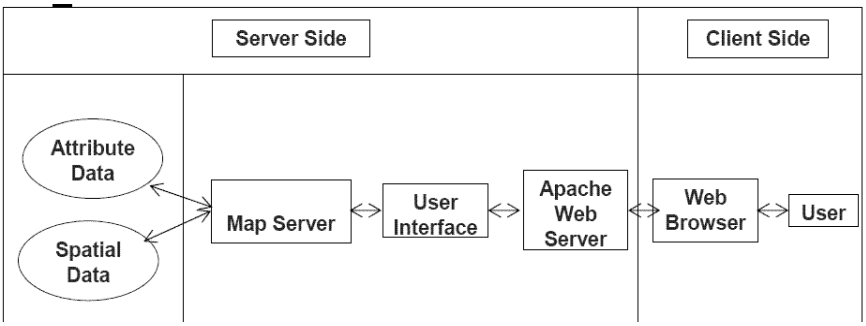

Figure 1: The server-client architecture of Mapserver.

data to the Web that works in various web environments such as Common Gateway Interface (CGI) or as a stand alone application that is available in several programming languages. Map images are created from spatial data stored in digital format. Using such data, geographical features can be described with a matrix (raster data), or points, lines and polygons (vector data). A vector data can come in several formats, one possible format is described in a dokumentation by the Environmental Systems Research Institute (1998). Non-spatial attributes of the geographical features are linked to attribute data that can come in form of spreadsheets, hierarchical or relational files.

As illustrated in Figure 1, a server-client Mapserver application contains: a http server (Apache), a user interface application, Mapserver software and a geographic dataset which involves spatial and attribute data.

Mapserver is executed in response to a web request. First, it reads a configuration file, called *map file* that describes the layer and other components of the map, it than draws and save the map. Next, it reads one or more HTML *template files* that are identified in the mapfile. Each template consists of conventional HTML markup tags and special Mapserver substitution strings that controls the Mapserver application's user interface in the browser window. Substitution strings are used, for example, to specify the paths to the map images that Mapserver has created, to identify which layers are to be rendered, and to specify zone level and direction. Mapserver substitutes the specified values for these strings and then sends the data stream to the web server, which then forwards it to the browser.

1.2 Mapserver at språkbanken

The MapServer web application and all of its supporting libraries, including: FreeType, GD, GDAL, libJPEG, libpng, Proj.4, shapelib and zlib, are currently installed on demo. Below follows a list of the mapserver file locations:[3]

- Mapdata files: MSERV-DATA.
- *OpenStreetMap* data: MSERV-DATA/OpenStreetMap/.
- Image output files: MSERV-HOME/tmp/.
- MapServer fonts: /usr/share/fonts/dejavu/.
- The *DocumentRoot* directory: /MSERV-HOME/svedd/.
- Litteraturbanken's PHP applications: MSERV-HOME/svedd/sb/app.
- Place name program and data files:MSERV-HOME/svedd/sb/data/.
- XML output files: MSERV-HOME/svedd/sb/app/xml/.

The MapServer described in this document operates in MapScript mode, and it is written in PHP programming language.[4] The application was tested with Internet Explorer and Mozilla web browsers.

1.3 The GIS dataset

The geographical data is acquired from Geofabrik.[5] It contains raw data (Open Street Map format) and shape files.[6] The data includes the following file sets: *sweden*, *waterways*, *roads*, *natural* and *railways*. More information about the files and their formats can be found in Haklay and Weber (2008).

[3](1) MSERV-HOME=/export/htdocs, (2) MSERV-DATA=/export/res/lb/mapdata
[4]The API is accessible from: <http://mapserver.org/mapscript/php/index.html>
[5]<http://www.geofabrik.de/data/download.html>
[6]<http://wiki.openstreetmap.org/wiki/Sv:Map_Features>

2 LITTERATURBANKEN'S GEOGRAPHICAL NAME ENTITY RECOGNITION

The research group at Litteraturbanken have recognised two main use cases which we decided to focus on in the first phase of the project: (1) search for coordinates given a set of place names and (2) generate interactive maps given a set coordinates. In this section we describe the developed applications that perform place name search and render maps that point to their geographical locations. [7] [8]

2.1 Placename database

MapServer accesses a MySql database titled *svedd_placenamedb*. All the data in this database is stored in one table, named: *geonamestable*. The geonamestable contains nine fields: geonameid, name, latitude, longitude, feature class, feature code, country code, and modification date.[9]

The database is populated with geographical data about place names in Sweden that are automatically extracted from the GeoNames geographical database.[10] The kind of place names the database contains are names of cities, hotels, gardens, churches, buildings, etc. The database population is an automatic process, it is a Cron job that is scheduled to run every month.

Place disambiguation The Geonames database contains redundant information about place locations which are distinguished with minor longitude and latitude values. Presumably this redundancy appears because several measurements surrounding a particular place that have been taken from different directions have been added to the database. To reduce the list of redundant

[7] Wiki internal pages: <https://dok.spraakdata.gu.se/wiki/lb/index.php/Kartserver_med_Litteraturbankens_platsnamn>

[8] Subversion catalogue:<https://svn.spraakdata.gu.se/repos/lb/trunk/karta>

[9] For complete specification of the geoname table fields and their properties see: <http://demo.spraakdata.gu.se/svedd/sb/dok/geoname-readme.txt>

[10] <http://download.geonames.org/export/dump/SE.zip>

place locations but still preserve ambiguity of place names, i.e. the actual occurrence of several places with the same place name, we use the *Pythagoras' theorem*. Using this theorem, the algorithm counts the distance between places with the same name. If the distance of two or more places with the same place name occur within the space of 3 km, these place occurrences are not added to the database. Only place occurrences which have the latest modification date are added to the databse. If more than one place location have the same modification date, the place with the highest latitude value is kept. Example: the place name 'Berget' has 7 occurrences in the GeoName data file, but only four of these occurrences are added to the place name database.

2.2 Coordinates search

Given a place name, the algorithm that performs coordinate search matches and presents all the place occurrences that are found in the database without applying any heuristics to disambiguate place names. Coordinate search is carried out by calling the function *getcoordinates*. Below follows a specification of the function; two examples of coordinate searches are given in Appendix A.1 and A.2.

getcoordinates.php

File name: *getcoordinates.php*
Output file name: placenamesposition<NUMBER>.xml
Description: this function takes a list of place names from main stream and generates an XML file containing geographic information about the specified place names. The output is a link to an URL that points to the generated XML file.

Comment

The output results could be restricted by performing place name disambiguation, and apply some kind of heuristics that prioritize place names. Disambiguation could be based on the distribution of places appearing in the literary work we generate a map for. Place prioritization can for example follow the population size of places/districts.

2.3 User case

There are two different ways to how maps that point to geographic locations are rendered: (1) static maps (displayed in a png file) and (2) dynamic maps (displayed in a php page). The user can choose the presentation form of the map by using the function *getmap*. Using this function the user can decide whether the output of the given input coordinate set will be specified as an XML or displayed as a dynamic map. These two usage possibilities are illustrated by an example in Appendix B.

getmap.php

File name: *getmmap.php*
Usage: getmap.php[?action=ACTION_TYPE]&[lat[]=LATITUDE&long[]=LONGITUDE]
Output:
(1) http://demo.spraakdata.gu.se/svedd/sb/app/xml/placenamesmaps<NUMBER>.xml;
(2) http://demo.spraakdata.gu.se/svedd/sb/app/getdynamicmap.php?&[lat[]=LATITUDE\&long[]=LONGITUDE]
Description: this function takes as input an action string that specifies the user output preference, i.e. XML, *action=xml* or dynamic *action=dynamic* and a list of place name coordinates. The output is either an URL that points to the XML file (output 1) or to the php page (output 2).

2.3.1 Static map

Appendix B.1 shows an XML file that has been generated for the seven place names: Norra Bondökallen, Södra Bondökallen, Östan, Klackraden, Kalkskär and Nockeby, Geresta. Norra Bondökallen, Södra Bondökallen and Östan appear in one bounding box and have therefore been rendered in the same map. In total the XML specification lists five static maps each of which has a unique id-number and a uri-link pointing to its location. Two of the generated maps are shown in Appendix B.2.

To find geographical locations that appear within the same bounding box (bbox), a spatial query is applied to retrieve a compressed geographic data. The spatial query comprises a radius which is equal to 5 and which is scaled by 1609 miles in one degree of latitude. These measures can of course be increased or reduced depending on the required coverage area.

2.3.2 Interactive map

Given a set of coordinates, the function *getdynamicmap* sends those to mapserver which renders a dynamic map pointing to place's geographical locations. Appendix B.3 provides an example of the rendered dynamic map for the place names that are specified in the XML file (given in Appendix B.1).

getdynamicmap.php

File name: *getdynamicmmap.php*
Map file: *lb_app_map.map*, location: /export/res/lb/mapdata/
Output file name: swedenplacenamesimage<NUMBER>.xml
Description: this function takes one or more coordinates (i.e. latitude and longitude values) that are specified in an URL input. The function generates an XML file with pointers to the rendered static maps and to the coordinates each static map contains.

REFERENCES

Environmental Systems Research Institute 1998. *Esri shapefile technical description*. Environmental Systems Research Institute.

Haklay, Mordechai (Muki) and Patrick Weber 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing* 7: 12–18. http://wiki.openstreetmap.org/wiki/Api.

Kropla, Bill 2005. *Beginning MapServer: Open source GIS development (expert's voice in open source)*. Berkely, CA, USA: Apress.

A COORDINATES SEARCH

A.1 Example 1

Input:

```
> php getcoordinates.php
Gävle
Karlstad
Varberg
Växjö
0
```

Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<geo>
<geoname xml:id="K3dfabcc74bc5bb9f03b3c7e1b493e783">
<name> Växjö</name>
<lat>56.8777</lat>
<long>14.8091</long>
<feat-class>P</feat-class>
<feat-code>PPLA</feat-code>
<country-code>SE</country-code>
<modification-date>2010-03-28</modification-date>
</geoname>
<geoname xml:id="Kfb867951a140821508179e5807404e38">
<name> Gävle</name>
<lat>58.7333</lat>
<long>16.6833</long>
<feat-class>S</feat-class>
<feat-code>FRM</feat-code>
<country-code>SE</country-code>
<modification-date>1994-01-14</modification-date>
</geoname>
<geoname xml:id="K46c09cbe0829cd8dea35fb1e0c3cdca9">
```

```
<name> Karlstad </name>
<lat>59.4447</lat>
<long>13.3375</long>
<feat-class>S</feat-class>
<feat-code>AIRP</feat-code>
<country-code>SE</country-code>
<modification-date>2007-07-17</modification-date>
</geoname>
<geoname xml:id="K4e2868e6cc8fdd8ac4536780b85021d3">
<name> Karlstad</name>
<lat>59.35</lat>
<long>13.4667</long>
<feat-class>S</feat-class>
<feat-code>AIRF</feat-code>
<country-code>SE</country-code>
<modification-date>2001-05-09</modification-date>
</geoname>
<geoname xml:id="K2390e7dcb40686f1255f2d74532c7ef0">
<name> Varberg</name>
<lat>57.1167</lat>
<long>12.2167</long>
<feat-class>S</feat-class>
<feat-code>AIRF</feat-code>
<country-code>SE</country-code>
<modification-date>1994-01-14</modification-date>
</geoname>
</geo>
wrote to URL:
http://demo.spraakdata.gu.se/svedd/sb/app/xml/placenamescoordinates93579.xml
```

A.2 Example 2

Input:

```
> php getcoordinates.php
Berget
0
```

Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<geo>
<geoname xml:id="K461c39d660ecf629a269867cf0eda6c8">
<name> Berget</name>
<lat>59.9333</lat>
```

```
<long>13.2667</long>
<feat-class>T</feat-class>
<feat-code>HLL</feat-code>
<country-code>SE</country-code>
<modification-date>1994-01-14</modification-date>
</geoname>
<geoname xml:id="Kc262b0ee7f9c759d7576ac131f5af9ed">
<name> Berget</name>
<lat>57.8906</lat>
<long>16.4272</long>
<feat-class>P</feat-class>
<feat-code>PPL</feat-code>
<country-code>SE</country-code>
<modification-date>2001-05-11</modification-date>
</geoname>
<geoname xml:id="K5ea48bc17a75dd3d70d8f59553fe0245">
<name> Berget</name>
<lat>57.9</lat>
<long>12.9167</long>
<feat-class>S</feat-class>
<feat-code>FRMS</feat-code>
<country-code>SE</country-code>
<modification-date>1994-01-14</modification-date>
</geoname>
<geoname xml:id="Kc8e86e48d290e36a3e40bcc03363e1de">
<name> Berget</name>
<lat>57.5667</lat>
<long>12.0833</long>
<feat-class>P</feat-class>
<feat-code>PPL</feat-code>
<country-code>SE</country-code>
<modification-date>1994-01-14</modification-date>
</geoname>
</geo>
wrote to URL:
http://demo.spraakdata.gu.se/svedd/sb/app/xml/placenamescoordinates93653.xml
```

B USER CASE

B.1 XML specification

Input:
http://demo.spraakdata.gu.se/svedd/sb/app/getmap.php?action=xml&lat[]=65.1028&long[]=21.8633&lat[]=65.0833&long[]=21.9667&lat[]=58.8558&long[]=17.9367&lat[]=65.15&long[]=21.85&lat[]=59.3296&long[]=17.917&lat[]=62.6264&long[]=17.9589&lat[]=65.1333&long[]=21.8833

Output:

```
<geo>
–
<geomap xml:id="SE">
–
<image xml:id="2b17f2f7a7afb5dfebe63e7d4b9f9843">
<uri>http://demo.spraakdata.gu.se/tmp/mappart36211.png</uri>
<country-code>SE</country-code>
–
<poi>
<name>Norra Bondökallen</name>
<lat>65.15</lat>
<lng>21.85</lng>
</poi>
–
<bbox>
<p1-lat>65.1028</p1-lat>
<p1-lng>21.8633</p1-lng>
<p1-lat>65.1333</p1-lat>
<p1-lng>21.8833</p1-lng>
</bbox>
</image>
</geomap>
–
<geomap xml:id="SE">
```

```
–
<image xml:id="e127bb4976ae80bb9d8307291751ff28">
<uri>http://demo.spraakdata.gu.se/tmp/mappart473580.png</uri>
<country-code>SE</country-code>
–
<poi>
<name>Klackraden</name>
<lat>65.0833</lat>
<lng>21.9667</lng>
</poi>
</image>
</geomap>
–
<geomap xml:id="SE">
–
<image xml:id="7289c1232a3d80c905599e450365d12c">
<uri>http://demo.spraakdata.gu.se/tmp/mappart727540.png</uri>
<country-code>SE</country-code>
–
<poi>
<name>Kalkskär</name>
<lat>58.8558</lat>
<lng>17.9367</lng>
</poi>
</image>
</geomap>
–
<geomap xml:id="SE">
–
<image xml:id="b9680ba2cafe69cedeafc36c2a0cd0ad">
<uri>http://demo.spraakdata.gu.se/tmp/mappart335323.png</uri>
<country-code>SE</country-code>
–
<poi>
<name>Nockeby</name>
<lat>59.3296</lat>
<lng>17.917</lng>
</poi>
</image>
</geomap>
–
<geomap xml:id="SE">
–
<image xml:id="d1e66c54ab114c9e873c5ef400b94450">
<uri>http://demo.spraakdata.gu.se/tmp/mappart151944.png</uri>
<country-code>SE</country-code>
```

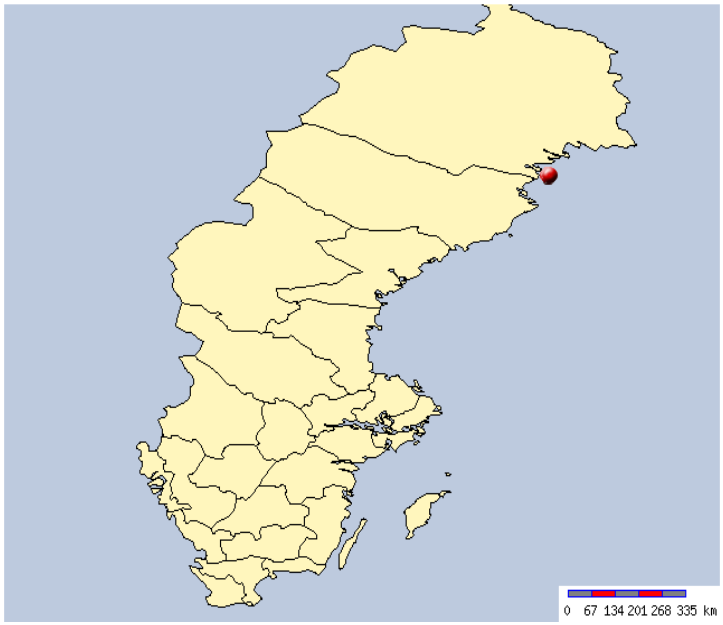

Figure 2: Static map for place names: Norra Bondökallen, Södra Bondökallen and Östan.

```
–
<poi>
<name>Geresta</name>
<lat>62.6264</lat>
<lng>17.9589</lng>
</poi>
</image>
</geomap>
</geo>
```

B.2 Static map

Figure 2 illustrates the static map that has been rendered for the three places: Norra Bondökallen, Södra Bondökallen and Östan.

Figure 3 illustrates the static map that has been rendered for Kalkskär.

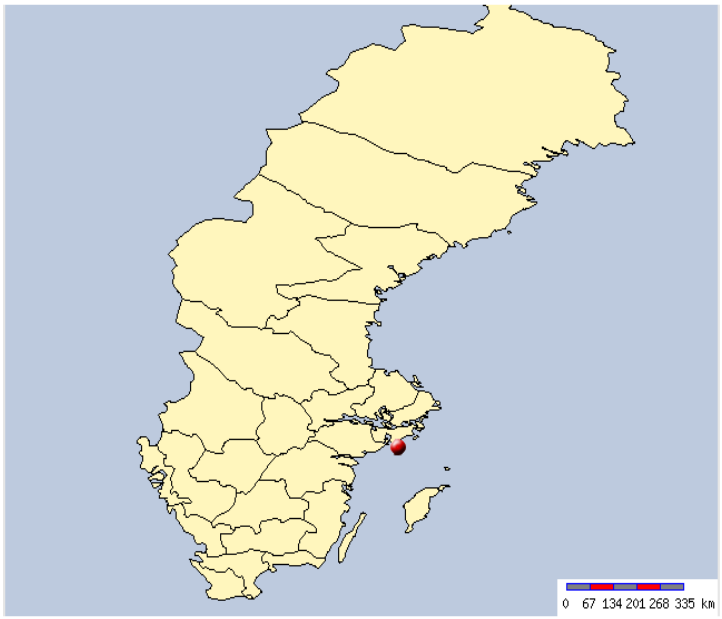

Figure 3: Static map for the place named Kalkskär.

B.3 Interactive map

Input:
http://demo.spraakdata.gu.se/svedd/sb/app/getmap.php?action=dynamic&lat[]=65.1028&long[]=21.8633&lat[]=65.0833&long[]=21.9667&lat[]=58.8558&long[]=17.9367&lat[]=65.15&long[]=21.85&lat[]=59.3296&long[]=17.917&lat[]=62.6264&long[]=17.9589&lat[]=65.1333&long[]=21.8833

Output: See Figure 4.

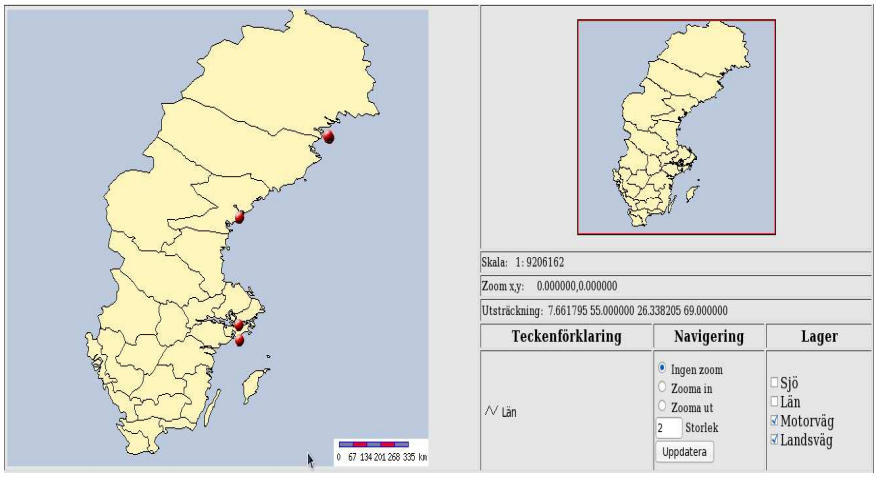

Figure 4: Interactive map for the place names that are specified in Appendix B.1.