Student thesis series INES nr 290

Geovisualization Using HTML5

A case study to improve animations of historical geographic data

Zhiyong Qi

2013
Department of
Physical Geography and Ecosystem Science
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Zhiyong Qi (2013). Geovisualization Using HTML5: A case study to improve animations of historical geographic data
Master degree thesis, 30 credits in *Geomatics*
Department of Physical Geography and Ecosystem Science, Lund University

Geovisualization Using HTML5

A case study to improve animations of historical geographic data

---

Zhiyong Qi

Master thesis, 30 credits, in *Geomatics*

Supervisors:

Associate Professor: Lars Harrie

PhD student: Finn Hedefalk

Department of Physical Geography and Ecosystem Science

Lund University

Abstract

The Scanian Economic-Demographic Database (SEDD) has been assembled by the Centre for Economic Demography (CED), Lund University. It contains information about the demographic and economic conditions of people that have lived in 5 parishes in Scania from the 17th century until the present. The SEDD database has been integrated with geographic data, which are digitized from four independent historical maps. To visualize and analyze these data, a GIS based web mapping application called SEDD Map has been developed and tested.

The previous version of SEDD Map is constructed using Silverlight. As a result, it only can be used on computers which have installed the Silverlight plugin. As Hypertext Markup Languages (HTML) continue to develop, a recent version, HTML5, was published in 2012. It aims to support the latest multimedia formats and reduce the need for plugins. In this study, we use HTML5, Cascading Style Sheets (CSS3), JavaScript and the ArcGIS API for JavaScript to create a new version of SEDD Map to visualize data stored in the SEDD database.

Before we constructed the new version of SEDD Map, a set of web mapping applications and programs were evaluated by the requirements which were needed to create the new version of SEDD Map. From this evaluation and comparison, we found that SEDD Map could be improved in many area, such as improving the animation of historical geographic data.

Animation is a useful tool when presenting historical data. The geographic data in SEDD Map are taken from four independent historical maps. To visualize geographic data as an animation, we need to create a time sense sequential dataset, which is done in a parallel project. In this study, we evaluate techniques for data animation. We used linear interpolation and the four historical maps as start years and end years to simulate 159 maps to visualize the geographic data as animations. The conclusions are as follows:

1) The commonly used web mapping applications for investigating demographic data contain functions, such as interactive visualization, statistical graphics, basic map tools, animations, etc.
2) HTML5 can replace (and improve) the used of Silverlight for web mapping.
3) Animations can be generated (filling in what is missing is to improve the data sets).

Keywords: web mapping, HTML5, animation, ArcGIS for Server, demographic map

Acknowledgements

I would like to express my gratitude to my supervisor **Lars Harrie** for the useful comments, remarks through the working process of this master thesis. Furthermore I would like to thank **Finn Hedefalk** for introducing me to the topic as well as for the support on the way. Also, I would like to thank **my parents** and **all my family members** for encouraging me all the time and supporting me during my study life in Lund University. Finally, I would like to thank all the **staff at the GIS Centre** for giving me help when I needed it and for providing me with a good working area.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

A database called the Scanian Economic-Demographic Database (SEDD) has been developed by the Centre for Economic Demography (CED), Lund University. It contains information about the demographic and economic conditions of the people that have lived in 5 parishes in Scania from the 17th century until now. To visualize and analyze these data, a GIS based web mapping application called SEDD Map has been developed and tested.

Web mapping is a technology to present map information to end users through Internet. It uses databases to store and manage the data, and Hypertext Markup Language (HTML) and scripting languages to show the data on a web browser *(Mitchell, 2005)*. From the browser, a user can view the data anytime and anywhere. For developers and data managers, it is also a convenient tool to create, edit and update data. Management can be done over the Internet. So, data can be kept more easily and up to date.

Hypertext Markup Language (HTML) is a markup language which is used to display web pages and other information through the Internet. For web mapping applications, it is also one of the main languages used to represent the map data *(Mitchell, 2005)*. As the standard of HTML development, a candidate version, HTML5, was released in December 2012 by the World Wide Web Consortium (W3C). It aims to support the latest multimedia formats and reduce plugins. To enable this, it introduces some new elements to embed pictures on a web page, such as the `canvas` element *(Berjon et al., 2012)*.

There are several types of web mapping applications that have been developed to achieve different kinds of tasks or make different types of analyze. They can be generally grouped into two broad categories, static maps and interactive maps *(Kraak and Brown, 2001)*. A static map is displayed as an image on a web site *(Mitchell, 2005)*. It uses one or more digital map pictures, such as scanned paper maps, to show the geographic information of a particular area, and no other data are needed. One example of a static map is to present a historical map on the Internet *(McDermott, 2002)*. Interactive maps are not as easy as static maps. These allow users to access a set of functions to have some interaction with the map, such as zoom out and zoom in *(Mitchell, 2005)*. The functions are embedded directly in the HTML web page. Therefore, the web users do not need to install any affiliated software on their platforms to use them. Interactive maps can be used in many areas, such as visualizing public transit *(Hoar, 2009)*, spatial planning *(Nuojua, 2010)* and analyzing spatio-temporal data *(Chaix et al., 2012)*.

The SEDD Map is an interactive web mapping application. It not only illustrates map information for users, but also contains interactive functions. As a powerful interactive web mapping application, it can make many analyze immediately from the client side, such as clicking on a house or a property to display its geographic information or retrieve the demographic information from the database. To enhance its interactive capability, SEDD Map should be improved in many ways, such as improving its interactive visualization capabilities.

There are several interactive web mapping applications suitable for visualizing and analyzing demographic and geographic data. Examples are NComVA (Norrköping Communicative Visual Analytics) Geovisual Analytics Visualization, which is a web tool used to visualize statistic information (*Jern, 2010*); Social Explorer, which is a web mapping application used to analyze the historical census data of the United States of America (*Bordelon, 2012*); China Geo-Explorer which is a web mapping application used to analyze Mainland China's demographic data (*Xu et al., 2009*). In this study, a couple of web mapping applications are described and investigated to examine their advantages and disadvantages. These web mapping applications are also good examples for suggesting how to improve the functions and the visualization abilities of the SEDD Map in the future.

1.2 Problem Statements

The previous version of SEDD Map is built using the ArcGIS API for Silverlight. As it has been developed with Silverlight, which is an application framework developed by Microsoft for writing and running rich Internet applications, it only supports computer or mobile devices (e.g. Windows Phone) which have installed the Silverlight plugin. Portable devices (e.g. tablet computers and smartphones) are widely used in people's daily lives. Unfortunately, some of the portable devices, such as the iPhone, iPad and Android phones, do not have the ability to install plugins in their browsers. Thus, the previous SEDD Map cannot be accessed from these devices.

Using animation to visualize historical geographic data requires a large quantity of sequential data. The demographic data stored in the SEDD database are continuous (year by year), they can be easily visualized with time changes. However, the geographic data of the SEDD Map are taken from four independent historical maps. The four historical maps are even not in the same series. For each house, it does not contain the exact information about when it was constructed and demolished. So, how to present them with time changes is a formidable task for the previous version SEDD Map.

Therefore, this study will try to answer the questions and solve the problems listed below:

- What functions do commonly used web mapping applications for visualizing and analyzing of demographic data have?
- How do they visualize historical data?
- Can SEDD Map be developed with HTML5, and achieve the basic web mapping functions, required to visualize and analyze the data stored in SEDD database?
- Can the new version of SEDD Map be accessible by both computer and portable devices?
- What advantages and disadvantages will the new version of SEDD Map have?
- Is it possible to use four historical maps to present the geographic data with animation?

1.3 Objectives

This study has three main objectives:

a. Review current web mapping applications which are used to visualize and analyze demographic data.
b. Improve the compatibility of SEDD Map with different platforms and browsers.
c. Improve the visualization for the geographic data which are taken as snapshots from 4 different historical maps during the years 1757 to 1915.

1.4 Methodology

This study is divided into two parts. The first part is a comparative study that compares several web mapping applications and programs which are used to visualize and analyze demographic data. The second part is a case study. In the case study, the compatibility and the animations for visualizing historical geographic data of SEDD Map are improved.

Part I: Comparative study

The general work steps are shown below:

- Set up user requirements of an improved SEDD Map.
- Review some web mapping applications and programs which are used to visualize and analyze the demographic data.
- According to the requirements, investigate different web mapping applications and programs to find out if the requirements could be achieved in these web mapping applications. And also try to find the unique features of each web mapping application.
- Compare these web mapping applications and programs. Illustrate the comparisons in tables.

Part II: Case study

For improving the compatibility of the SEDD Map:

- Use ArcGIS for Server to set up a server to manage the GIS data for the case study.
- HTML5 and Cascading Style Sheets level 3 (CSS3) are used to design the layout of SEDD Map to fit all screens (small or large).
- Using the requirements which are set up in Part I, to develop the new version of SEDD Map. The functions are developed with JavaScript and ArcGIS API for JavaScript.
- Evaluate the compatibility of the new version of SEDD Map with different platforms and browsers.
- Compare the SEDD Map new version with the SEDD Map previous version, using the requirements listed in Part I.

For improving the animation for visualizing historical geographic data of SEDD Map:

- Investigate the data, using linear interpolation, to generate the maps to cover the period from 1750 to 1915 for each year.
- Test if the geographic data could be visualized as an animation with time change.

1.5 Report Structure

The structure of this thesis is shown in *Figure 1.1*. **Chapter 2** describes the Scania Economic-Demographic Database (SEDD) including a description of the data and the data structure as well. This includes where and how the data were collected. Also described is the previous version SEDD Map, and the problems it faced. **Chapter 3** is a comparative study of web mapping applications which are used to visualize and analyze spatio-temporal data. First, it describes the requirements of the improved SEDD Map. Then, it introduces a web mapping program (NComVa Geovisual Analytics Visualization) and three web mapping applications (Social Explorer, China Geo-Explorer and DigDag). In the last part, the web mapping programs and applications are evaluated against the requirements. **Chapter 4** is about the web mapping client-server model. It describes the main components (client side, web server, application server, map server and map data server) from the client side to the server side to show what they do and how they work. **Chapter 5** considers details of the client side of web mapping applications which are constructed with HTML, JavaScript and CSS. This chapter is divided into 3 parts. The first part, gives an overview of HTML, describes how it works and the newest candidate version, HTML5. It also gives a short description of the `Canvas` element. The second part introduces the JavaScript language and the open source modular JavaScript library, DOJO. The last part describes CSS and its newest version, CSS3. **Chapter 6** gives an introduction to the ArcGIS for Server package. It describes its structure, how it works. Then, it gives a short description of the ArcGIS API for JavaScript, which is the tool used to develop the new version of SEDD Map. ArcGIS for Server 10 provides the function used to visualize the time-aware data. So, how it works is also described in this chapter. **Chapter 7** is the case study part. This chapter introduces the study area and the data that the case study used. Then, it shows the implementation for creating the new version of SEDD Map and how to generate the maps from the four historical maps to visualize the geographic data using animation. The result part shows how the new version of SEDD Map works and the functions that were implemented. The final part evaluates the new version of SEDD Map and make a comparison with the previous version of SEDD Map. **Chapter 8** is the discussion part of this thesis. In this portion, it discusses the final results of this thesis from three aspects: web mapping with HTML5, animations and the future of web mapping development. **Chapter 9** shows the final conclusions. All the questions set up in *Chapter 1* are answered in this part. And all the objectives are attained.

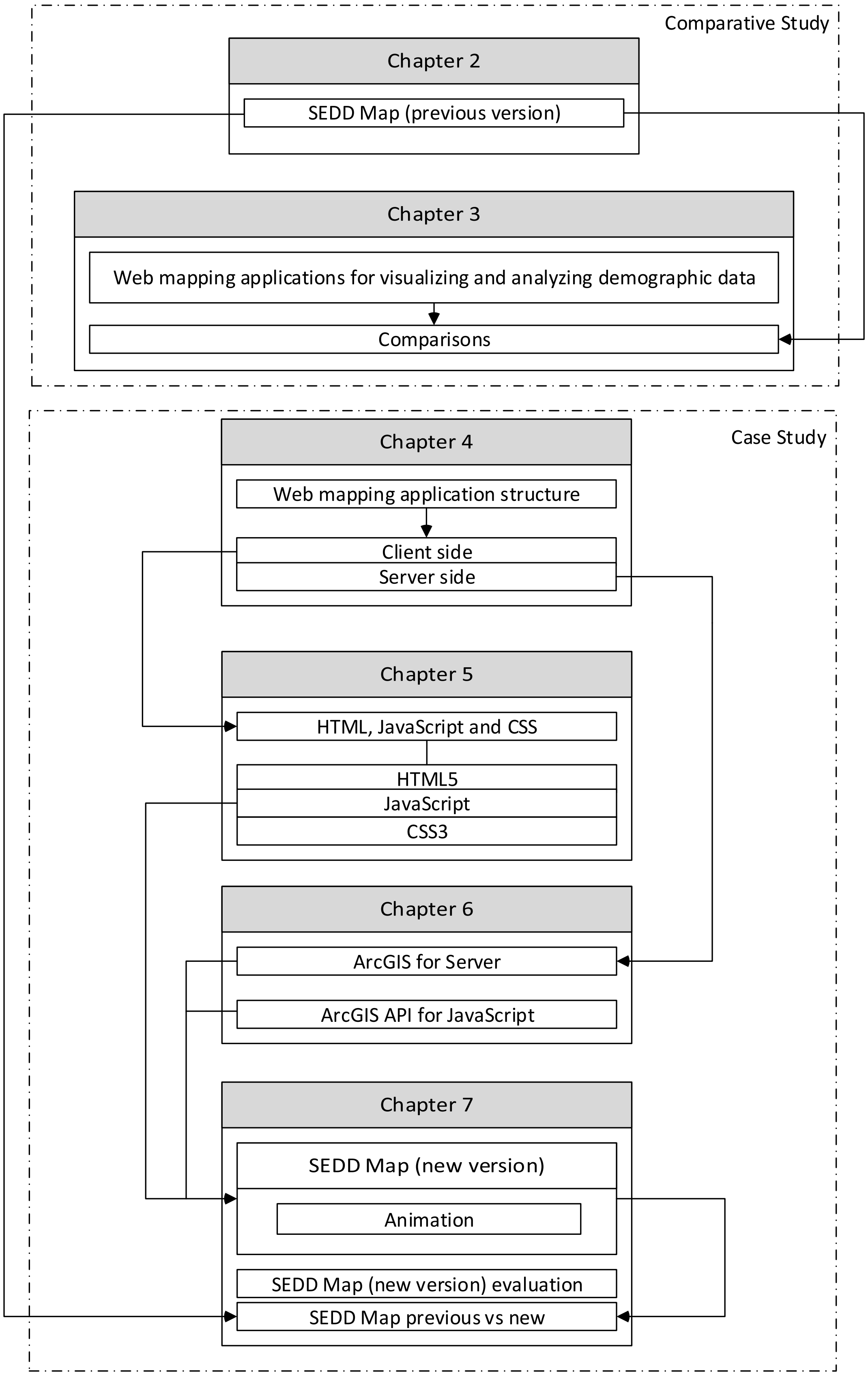


Figure 1.1. Report structure of the thesis.

2 SEDD Map

2.1 Introduction

SEDD is an acronym for the Scanian Economic-Demographic Database. This database is constructed and managed by the Centre for Economic Demography (CED) at Lund University, which handles research in different areas, such as economic history, economics, social medicine, social work and statistics (*Bengtsson, 2012*). At CED, the SEDD database is, for example, used in research about individual behavior, family organization and demographic outcomes during the age when Sweden developed from an agricultural to an industrial society. The Center also focuses on improving the knowledge of contemporary behavior in Sweden. Based on such demographic behavior, it is possible to analyze the role of intergenerational factors and the influence of economic change (*Bengtsson, 2012*).

The SEDD database has been integrated with geographic data, which has enabled visualization and analysis of the data through a GIS based web mapping application (*Hedefalk et al., 2013*). This GIS based web mapping application is called SEDD Map. The figure below (*Figure 2.1*) shows the previous version of the web mapping application, published in 2011. SEDD Map aims to help the researchers to analyze the data stored in the SEDD database. By using some visualization functions on the map, the data can be presented in a more efficient and clear way.

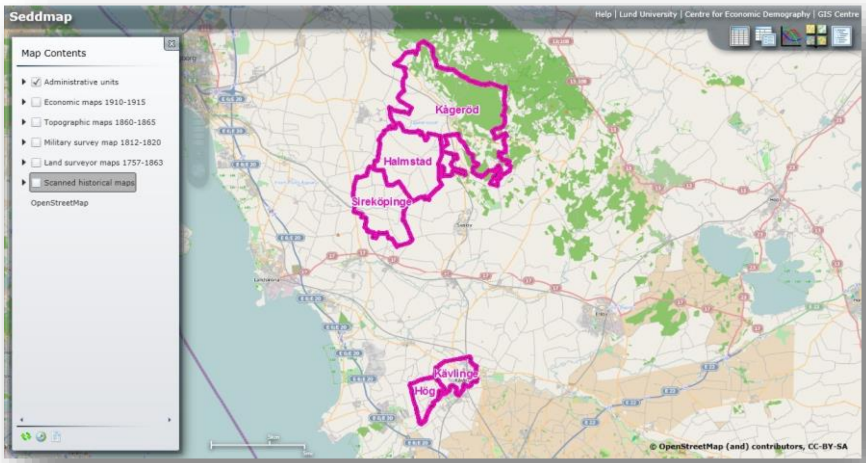

Seddmap
Help | Lund University | Centre for Economic Demography | GIS Centre
Map Contents
Administrative units
Economic maps 1910-1915
Topographic maps 1860-1865
Military survey map 1812-1820
Land surveyor maps 1757-1863
Scanned historical maps
OpenStreetMap
Kågeröd
Halmstad
Sireköpinge
Kävlinge
Hög
© OpenStreetMap (and) contributors, CC-BY-SA

Figure 2.1. SEDD Map application (previous version).

The SEDD Map previous version can achieve the basic web mapping functions. On the left side of the map application there is a map content tool, which is used to display and hide the map layers. The zoom tool is in the upper left corner in the map frame, which is used to choose the zoom level for the map and move the map in different directions. However, for changing the zoom level and moving the map, the users can also use the mouse. A scale bar is located in the bottom left corner to display the scale. In the upper right corner there are some tools for retrieving and visualizing the data. When clicking on the view data tool, a new window opens to access the SEDD database. Users can use this window to search the demographic data. There are three key words used to identify the data: *sex*, *birth day* and *first name*. So when the user needs the information of someone, they can use these three parameters to get the information for a particular person. The download tool is used to export demographic data. A chart tool (*Figure 2.2*) is used to visualize the demographic statistics of *birth*, *death*, *population* and *marriage*. The base map tool is used to change the background maps. SEDD Map can also make smooth interactions with the geographic data. When clicking on some objects, such as a house or property, an information window will appear and display the information of those objects.

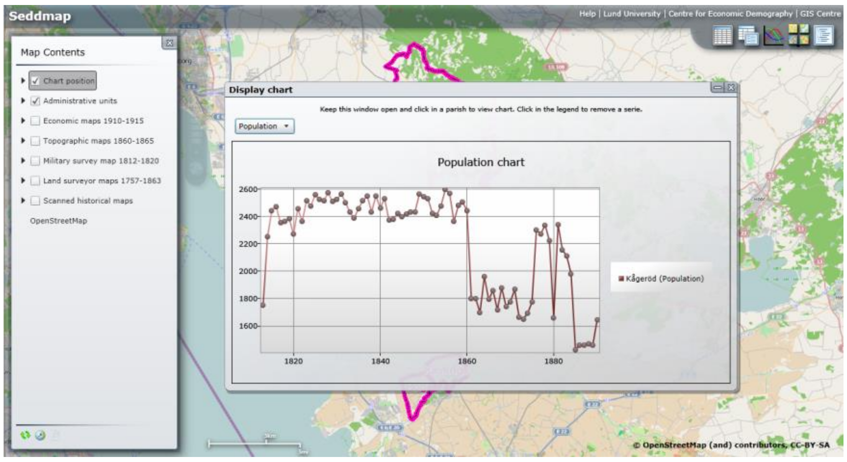

Seddmap
Help | Lund University | Centre for Economic Demography | GIS Centre
Map Contents
Chart position
Administrative units
Economic maps 1910-1915
Topographic maps 1860-1865
Military survey map 1812-1820
Land surveyor maps 1757-1863
Scanned historical maps
OpenStreetMap
Display chart
Keep this window open and click in a parish to view chart. Click in the legend to remove a serie.
Population
Population chart
2600
2400
2200
2000
1800
1600
1820
1840
1860
1880
Kågeröd (Population)
© OpenStreetMap (and) contributors, CC-BY-SA

Figure 2.2. Chart tool shows the statistical information about population during 1750 to 1915.

At this stage, SEDD Map should be improved in two ways: data integration and visualization. A limitation of the SEDD Map application is that the integration between the demographic and the geographic data is only made on an aggregated level. This means one or more administrative units are linked with a group of persons. To improve the connection and to present the data more clearly, we have to create a linkage on individual level (*Hedefalk et al., 2013*). That means each person in the SEDD database would be linked to one or more buildings or properties. As the two sources use different time representations, with the geographic data taken as a snapshot of the historical map for a certain time, while the demographic data is using object lifelines, we have to improve the methodology for making the linkage between the two sources (*Hedefalk et al., 2013*).

This study focuses on improving the visualization aspect. To enable the visualization for different time presentations, we need to model continuous information based on the historical map snapshots. To improve the implementation, a couple of new techniques will be included in the web application, such as HTML5.

2.2 Data

The SEDD Map application uses several types of data, such as economic data, demographic data and geographic data. The database contains information about five parishes (Halmstad, Hög, Kävlinge, Kågeröd and Sireköpinge) in Skåne, which is a county in southern Sweden. The data contain information from late 17th century up to now (*Bengtsson and Dribe, 1997*).

The economic and demographic data are constructed based on information sources recording family reconstitutions, land ownership, migration, etc. The family reconstitution information was collected during the late 17th century to 1895 in five rural parishes and one city recorded by the church on births, deaths and marriages, that was linked together to make up families. Land ownership information was collected from the poll-tax registers. It contains the annual information during the period from 1766 to 1895. It has been linked to the families in five of the parishes (Halmstad, Hög, Kävlinge, Kågeröd and Sireköpinge) to get more information on socio-economic status. Migration and household composition information was collected from the catechetical examination registers in the same five parishes from the year 1829 - 1895. These data, combined with local information about food prices and wages, have been used to analyze the environmental impacts on demographic outcomes. In order to improve the knowledge of social origins, all person registered as married in the five parishes from 1829-1895 have been tracked. The information about parental social status collected from the church records and the poll-tax registers has also been added. In total, there are more than 740,000 observations, covering 210 variables for about 39,000 individuals, in the five parishes in the SEDD database. The data for economic and demographic variables stored in the SEDD database is shown below in *Table 2.1*.

Table 2.1. Data for economic and demographic variables in SEDD database (*Bengtsson, 2011*).

Demographic variables	Births, Deaths, Cause of death, Marriages, In-migrants, Out-migrants, Parity
Socioeconomic variables	Occupation, Land holdings, Type of residence, Property ownership
Household variables	House-hold size, Typology of house-hold members

The geographic data are based on scanned and digitized historical maps from 1750 to 1915 over the five parishes (Halmstad, Hög, Kävlinge, Kågeröd and Sireköpinge). The maps have been transformed into the Swedish national reference system SWEREF99 TM. There are four historical map series that are used in the web mapping application. They are economic maps (EM), topographic maps (TM), land surveyors maps (LSM) and military topographic survey maps (MTSM). All the maps have been converted into both raster and vector format (*Hedefalk et al., 2013*).

Economic maps (EM) cover the period from 1910 to 1915. The raster data of EM are constructed by scanning and geocoding the paper maps. The vector data of EM were derived from the raster maps, which were digitized into different parishes, property units, houses, roads and railroads stored in different layers.

Topographic maps (TM) cover the period from 1860 to 1865. These were also constructed with scanned history map as images stored in the database cover the 5 parishes. These were digitized into houses and roads stored into different layers as vector data.

Land surveyors maps (LSM) cover the period from 1757 to 1863. Two raster files were created from these maps, a clipped map and a complete map. The clipped map only contains the portion where the geographic information was drawn. The complete map contains all the information of the original map. As vector data, it has been separated into properties and houses portions stored in different layers.

Military topographic survey maps (MTSM) cover the period from 1812 to 1820. As raster data, it is only formatted as a scanned map. As vector data, it has been digitized into houses, roads, lakes, streams and wetlands in different layers.

In addition to these maps above, the web application also includes some background maps to enhance the visualization, such as some commercial base map data from ESRI and Bing, free base map data from Open Street Map and further data from National Land Survey of Sweden.

3 Visualization studies of web mapping for spatio-temporal data

3.1 Requirements for the web mapping application

A web mapping application is used to present the data stored in a GIS database (*Li et al., 2011*). As the GIS data are visualized on a web page directly by users, it makes the data represented in a more timely and more understandable fashion. Different map applications are built up for different users and aims. For a particular aim, it must often meet some specific requirements (*Kiehle et al., 2007*). Below we list the requirements for the web mapping applications considered in this study. The requirements are the basic functions or rules that a web mapping application should fulfill. The SEDD Map (new version) is constructed based on these requirements (*see Chapter 7*).

A. **Technical solutions**:

a) No Plugins

This requirement is to avoid the need to use plugins. Instead, only scripting languages supported by modern browsers should be used.

b) Cross-platform support

Cross-platform support means that the web mapping application should be possible to run on different kinds of platforms (*Trelease and Nieder, 2012*). For hardware platforms, it has to support not only computer devices, but also tablet devices and smartphones. For software platforms, it has to run on different operating systems, such as Windows, Linux, Mac OS, iOS, Android, etc.

c) Multiple web browsers support

There are many web browsers on the market. Different web browsers may have different web browser engines (*Williams, 2013*). Multiple web browsers support means that the web mapping application should support the commonly used web browsers (*Boulos et al., 2010*), such as Chrome, Internet Explorer, Firefox, Safari, Opera, etc.

B. Technical implementation details:

a) Interactive visualization

To analyze data using a web application, interactive visualization is also a necessary capability (*Brock et al., 2012*); for example, click on some buildings or properties to get the detailed information about them.

b) Multiple maps based

Multiple maps based means that the web mapping application can visualize more than one map on the web site (*Newswire, 2012*). Using different map layers, such as an aerial photo map and a soil map, will improve the presentation.

c) Animation

Presentation of historical data often needs some analysis between different time periods, such as investigating the data changes over time. So, animation would be a necessary function for the users to comprehend the data (*Shah, 2012*).

d) Statistical graphics

To investigate and analyze data, it is often useful to make some statistical graphics directly on the web mapping application (*Hienert et al., 2011*). Making some charts, such as pie charts and bar charts, would make the data more understandable to many users.

e) Search tool

Search tool means the application has the ability to retrieve data from the client side by users (*Tsai, 2011*). If a user needs some specific information, the search tool will help the user to quickly find that information.

f) Export functions

For analysis, exporting data from the web application is also often a necessary function (*Norambuena et al., 2007*); for example, when they are interested some part of the data, they do not need to export the whole data set, the can just export the selected part.

3.2 GIS based web mapping programs

3.2.1 NComVA Geovisual Analytics Visualization

Geovisual Analytics Visualization (GAV) is a conceptual framework, which is used to visualize and make statistical analyze, developed by NComVA (Norrköping Communicative Visual Analytics) a spin-off company of the National Center for Visual Analytics (NCVA) at Linköping University. There are many studies based on this framework, such as statistical data visualization (OECD), self-organizing mobile networks (Ericsson Research, Sweden) and road emergency analysis (SMHI) (*Ho et al., 2011*). On May 6, 2013 the company NComVA announced that it had been acquired by QlikTech, which is a software company focused on data analysis, to enhance the visualization capability of the software QlikView (*QlikTech, 2013*).

The first generation of GAV was developed using the techniques of `C#`, `DirectX` and `.Net` (*Jern and Franzen, 2007*). In 2008, they changed the development platform used to Adobe Flash/Flex which could be run on most of the computers around the world (*Jern, 2009*). As Adobe announced that Flash would no longer be supported on mobile devices as of December 2011, NComVA decided to change their development platform to HTML5 with JavaScript. This development started in January 2012 (*Ho, 2013*). Now, there are two frameworks provided by NComVA, GAV Flash/ActionScript and GAV HTML5/ JavaScript. *Figure 3.1* shows a dashboard application developed based on GAV HTML5/ JavaScript.

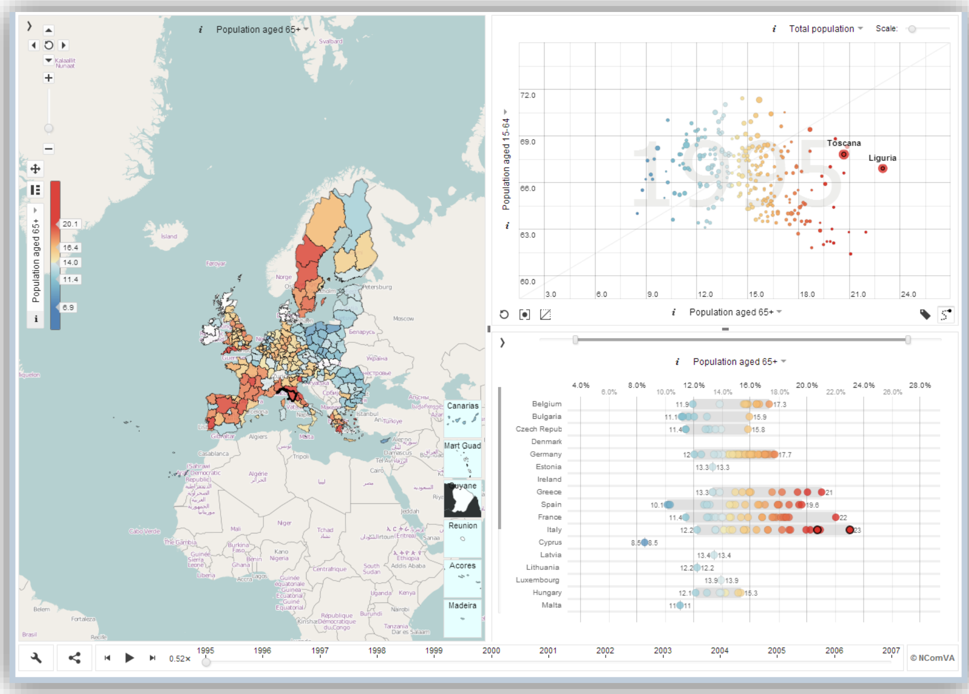


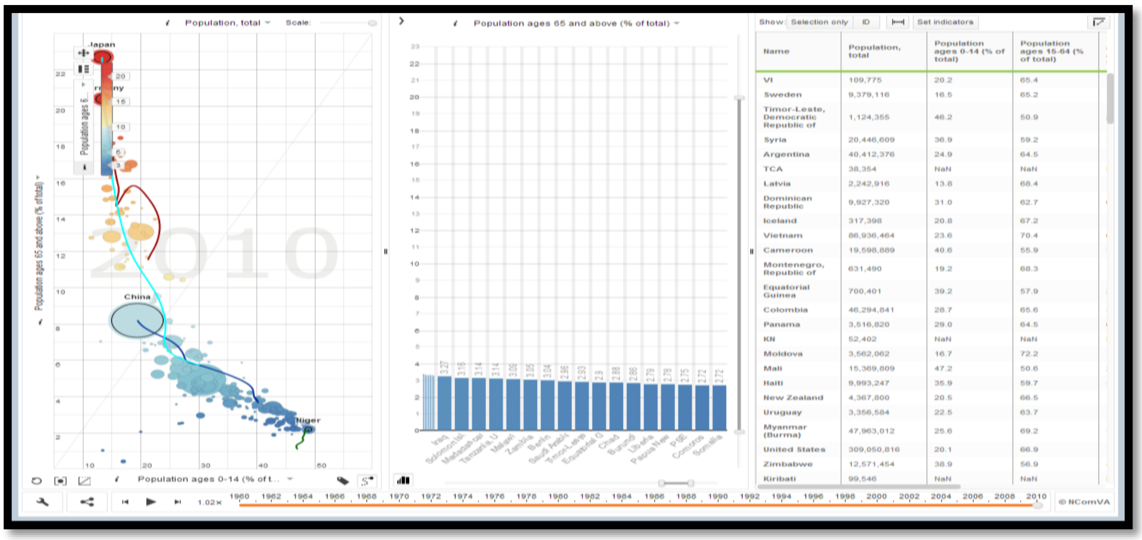
Figure 3.1. Dashboard application based on GAV HTML5/ JavaScript.

GAV helps the user to create customized dynamic linked views with spatial-temporal data. As a web mapping application created with GAV, it has some unique features such as layout define and storytelling (*Lundblad, 2013*).

Users can define the layout style of their application by themselves using the visualization components of the GAV framework. GAV provides a layout attribute that divides the mapping application web site into many rows with brackets and columns with commas. For example, the map application with the layout attribute `[map, timegraph, datatable]` gets a 3-column layout (*Figure 3.2*), while the one with the `[(map, timegraph), datatable]` attribute gets a 2-column map application with 2 rows on the left side (*Figure 3.3*). There are 5 visualization components: `Choropleth Map`, `Scatterplot`, `Barchart`, `Timegraph` and `Datatable`. Even though they are combined together to construct a web application, they are not dependent on each other. So, they can be combined freely according to the user's choice (*Ho et al., 2011*).

Storytelling is one feature of the GAV toolkit when visualizing the data. Storytelling means to tell a story about the data and the related analytics. The function provides a means to create descriptive texts and snapshots to describe the data. So, with the snapshots and the descriptions, the data can be presented more clearly and the user can understand the data more easily. It aims to guide the user to think as an analyst (*Jern, 2010*).

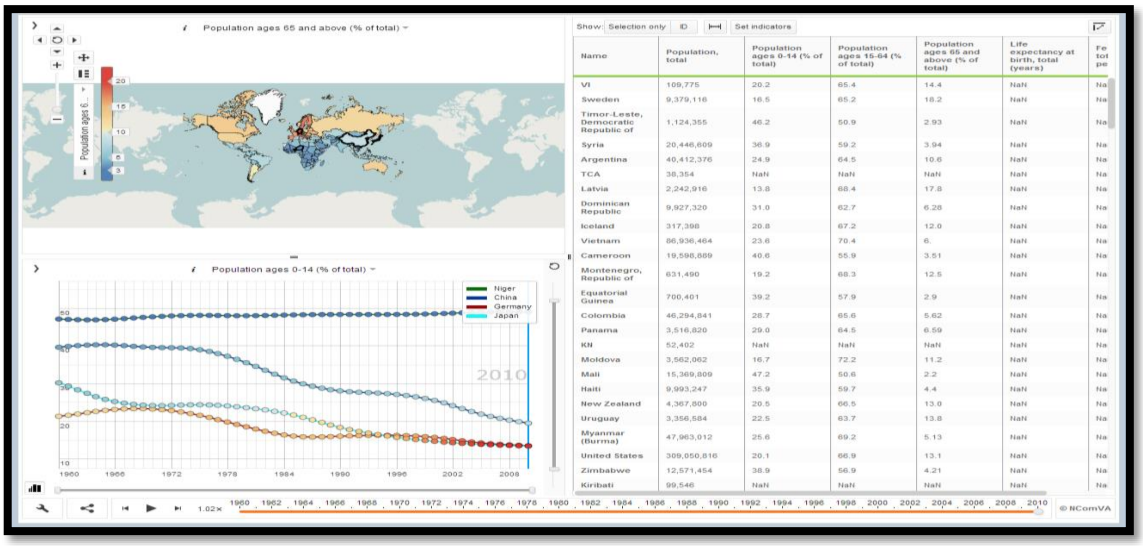
http://www.ncomva.se/html5/dynamic/index.html?layout=[map,timegraph,datatable]



Name	Population, total	Population ages 0-14 (% of total)	Population ages 15-64 (% of total)
VI	109,775	20.2	65.4
Sweden	9,379,116	16.5	65.2
Timor-Leste, Democratic Republic of	1,124,355	46.2	50.9
Syria	20,446,609	36.9	59.2
Argentina	40,412,376	24.9	64.5
TCA	38,354	NaN	NaN
Latvia	2,242,916	13.8	68.4
Dominican Republic	9,927,320	31.0	62.7
Iceland	317,398	20.8	67.2
Vietnam	86,936,464	23.6	70.4
Cameroon	19,598,889	40.6	55.9
Montenegro, Republic of	631,490	19.2	68.3
Equatorial Guinea	700,401	39.2	57.9
Colombia	46,294,841	28.7	65.6
Panama	3,516,820	29.0	64.5
KN	52,402	NaN	NaN
Moldova	3,562,062	16.7	72.2
Mali	15,369,809	47.2	50.6
Haiti	9,993,247	35.9	59.7
New Zealand	4,367,800	20.5	66.5
Uruguay	3,356,584	22.5	63.7
Myanmar (Burma)	47,963,012	25.6	69.2
United States	309,050,816	20.1	66.9
Zimbabwe	12,571,454	38.9	56.9
Kiribati	99,546	NaN	NaN

Figure 3.2. Application constructed using GAV layout=[map,timegraph,datatable].

http://www.ncomva.se/html5/dynamic/index.html?layout=[(map,timegraph),datatable]



Name	Population, total	Population ages 0-14 (% of total)	Population ages 15-64 (% of total)	Population ages 65 and above (% of total)	Life expectancy at birth, total (years)	Fe tot pe
VI	109,775	20.2	65.4	14.4	NaN	Na
Sweden	9,379,116	16.5	65.2	18.2	NaN	Na
Timor-Leste, Democratic Republic of	1,124,355	46.2	50.9	2.93	NaN	Na
Syria	20,446,609	36.9	59.2	3.94	NaN	Na
Argentina	40,412,376	24.9	64.5	10.6	NaN	Na
TCA	38,354	NaN	NaN	NaN	NaN	Na
Latvia	2,242,916	13.8	68.4	17.8	NaN	Na
Dominican Republic	9,927,320	31.0	62.7	6.28	NaN	Na
Iceland	317,398	20.8	67.2	12.0	NaN	Na
Vietnam	86,936,464	23.6	70.4	6.	NaN	Na
Cameroon	19,598,889	40.6	55.9	3.51	NaN	Na
Montenegro, Republic of	631,490	19.2	68.3	12.5	NaN	Na
Equatorial Guinea	700,401	39.2	57.9	2.9	NaN	Na
Colombia	46,294,841	28.7	65.6	5.62	NaN	Na
Panama	3,516,820	29.0	64.5	6.59	NaN	Na
KN	52,402	NaN	NaN	NaN	NaN	Na
Moldova	3,562,062	16.7	72.2	11.2	NaN	Na
Mali	15,369,809	47.2	50.6	2.2	NaN	Na
Haiti	9,993,247	35.9	59.7	4.4	NaN	Na
New Zealand	4,367,800	20.5	66.5	13.0	NaN	Na
Uruguay	3,356,584	22.5	63.7	13.8	NaN	Na
Myanmar (Burma)	47,963,012	25.6	69.2	5.13	NaN	Na
United States	309,050,816	20.1	66.9	13.1	NaN	Na
Zimbabwe	12,571,454	38.9	56.9	4.21	NaN	Na
Kiribati	99,546	NaN	NaN	NaN	NaN	Na

Figure 3.3. Application constructed using GAV layout=[(map,timegraph),datatable].

GAV toolkit has been used to develop many highly interactive statistical web mapping applications which can be used to analyze and visualize statistic data in different areas for different data; for example, to visualize the Parliamentary Elections 2010 (*Figure 3.4*), postcode atlas data (*Figure 3.5*) and statistical data of Sweden (*Figure 3.6*) (*Jern, 2009*).

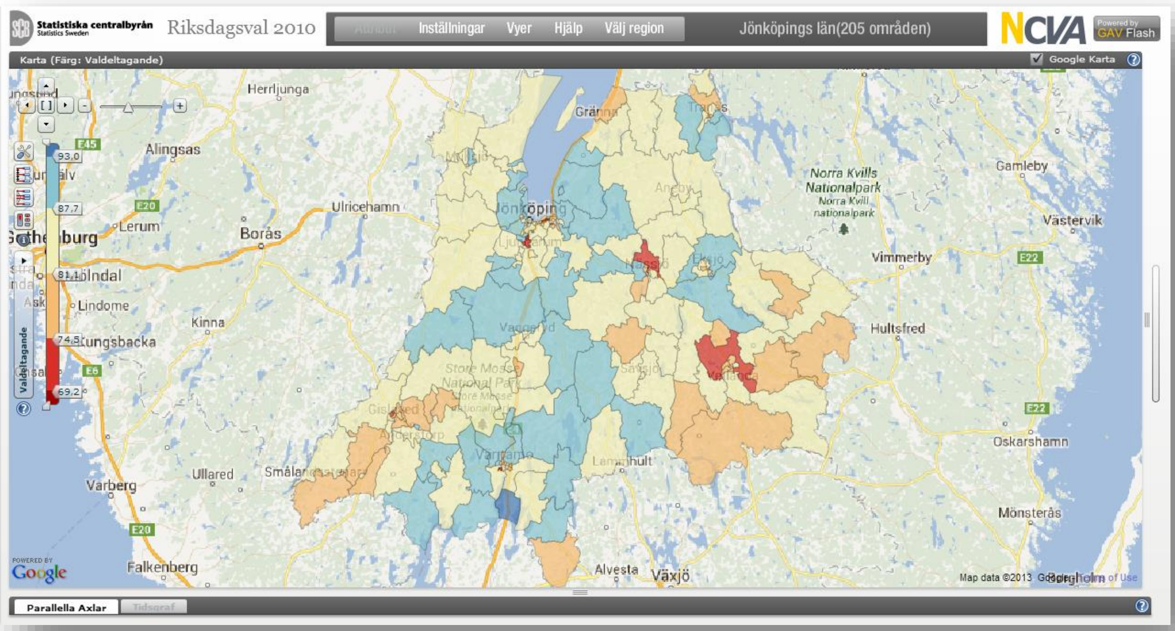


Figure 3.4. Parliamentary Elections 2010 in Sweden (by Statistic Sweden).

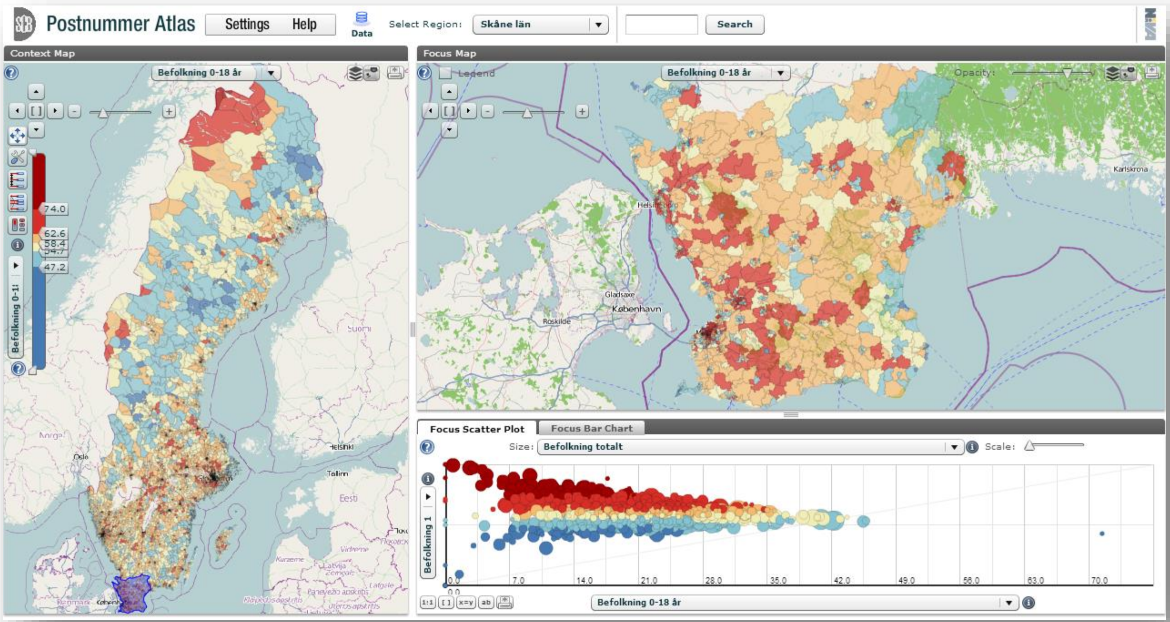
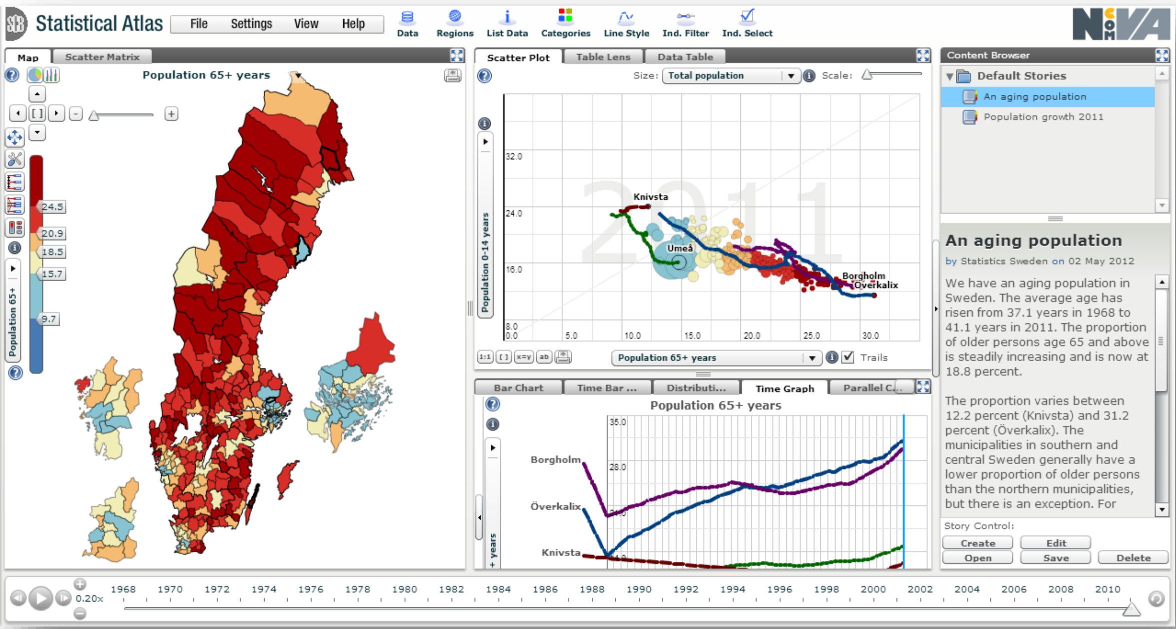


Figure 3.5. Postcode Atlas (by Statistic Sweden).



An aging population

by Statistics Sweden on 02 May 2012

We have an aging population in Sweden. The average age has risen from 37.1 years in 1968 to 41.1 years in 2011. The proportion of older persons age 65 and above is steadily increasing and is now at 18.8 percent.

The proportion varies between 12.2 percent (Knivsta) and 31.2 percent (Överkalix). The municipalities in southern and central Sweden generally have a lower proportion of older persons than the northern municipalities, but there is an exception. For

Figure 3.6. Statistic Atlas (by Statistic Sweden).

Coding languages and plugins

GAV Flex Toolkit is developed with Flash/ActionScript. So, a Flash plugin is needed. However, GAV HTML5/JavaScript is developed with HTML5 and JavaScript. It can be run on all devices without plugins.

Cross-platform support

The web mapping applications which are constructed with the GAV Flex Toolkit cannot work on portable devices. However, GAV HTML5/JavaScript based applications can run on all devices.

Multiple Web browsers support

The GAV Flex Toolkit mapping applications can only run in a browser which has installed the Flash plugin. The GAV HTML5/JavaScript version of these applications can run in all browsers.

Interactive visualization

The mapping application constructed with GAV has all the basic interactive functions. It provides the function to zoom the map to different levels and can easily move the map in different directions. When clicking on the map or the statistical graphics, the user can get information about the data. It also provides a tool to change the color of the map and the statistical graphics.

Multiple maps based

The web mapping applications which are constructed with GAV can embed background maps. For example, *Figure 3.4* contains four Google maps (street map, satellite map, hybrid map and terrain map) as background maps, and *Figure 3.5* contains Open Street Map as its background map.

Animation

GAV provides time-linked animation, which is used to visualize the geographic data and several statistical graphics in the meantime (*Ho et al., 2011*). As a basic web mapping application for historical data analysis, GAV can make powerful statistical analysis figures for visualizing census data with time changes. When the time-linked animation is playing, the statistical graphics and the maps will change together. The animation is controlled by a slider with several buttons. The Start/Stop button is used to control the animation’s start and stop; the Step Left and Step Right buttons are used to control the slider to change to the previous or next year; the Increase Speed and Decrease Speed buttons are used to change the speed of the animation.

Statistical graphics

GAV provides many statistical graphics to enhance the visualization of the data. The statistical graphics, such as distribution plot, bar chart, scatter matrix, histogram, scatter plot, pie chart and time graph, can be organized freely (*Lundblad et al., 2010*).

Search tool

In the postcode atlas mapping application (*Figure 3.5*), there is a search tool above the map window. It can be used to search a particular area. When the user types in the zip code, the mapping application will directly zoom to that place. This would help the user to find their data easily.

Export functions

A mapping application developed with GAV can export data from its database. There are three kinds of export data: map snapshot, demographic data and storytelling data. Take *Figure 3.6* for an example, the data can be exported with the Save Data function in the FILE menu. There are two data formats can be chosen when saving the data. One is to export the data as a text file, another is to save the data using column grid format. The storytelling data can be saved as an XML file.

3.3 GIS based web mapping applications

3.3.1 Social Explorer

Social Explorer (*Figure 3.7*) is an online research application which is designed to present historical census data and demographic information from the United States of America, and let users create reports and maps for visualizing, analyzing and investigating demographic changes. It was first developed in 1999. Four years later, the web site was launched. In 2007 a subscription version was available. It has been cited by numerous media outlets such as New York Times, the Wall Street Journal and Swedish TV4 (*Social Explorer, 2013a*).

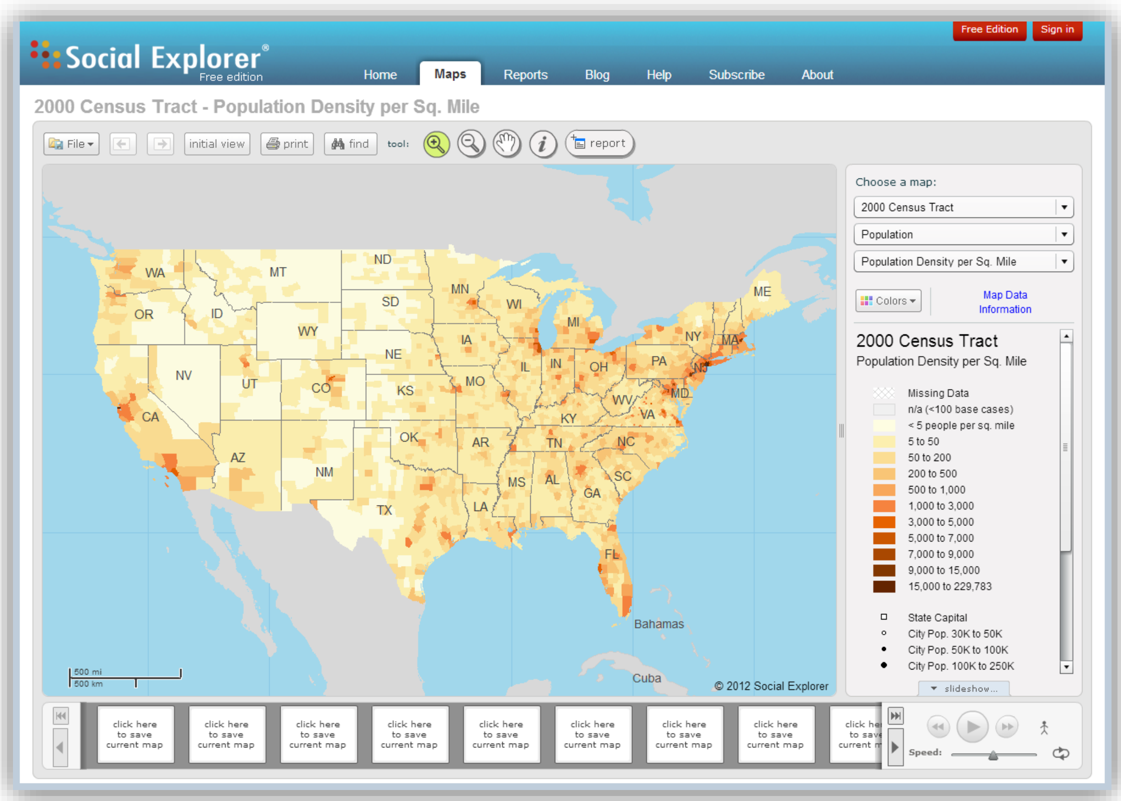


Figure 3.7. Social Explorer.

The database of Social Explorer contains over 18,000 maps, 40 billion data items and around 200,000 variables from 1790 to 2010. The data are annually updated from the American Community Survey. It contains the decadal US Census from 1790 to 2000 and annually updates from the American Community Survey from 2005 to 2010. Religion data are based on the Religious Congregations and Membership Study (RCMS) from 1980 to 2000 (*Social Explorer, 2013a*).

It is easy to create reports and maps with Social Explorer (*Bordelon, 2012*). It supplies an online report making function, creates a report combined with maps and tables automatically. The report can be exported as an Excel document or a CSV file. The search tool is also a useful function, which can be used to select a specific place by searching with words.

The interface of Social Explorer (*Figure 3.7*) can be generally divided into four parts. On the top, it is a tool bar which contains the menu and tools for analyzing the data. In the middle left, there is a map which is used to visualize the data. In the middle right, there is a window which is used to display the map lists and the legend. At the bottom, there is an animation tool called the slideshow tool, used to do dynamic visualization of the data (*Social Explorer, 2013b*).

Coding languages and plugins

Social Explorer is developed with Flex. So, a Flash plugin (Flash version 9 or higher) is needed when using it (*Social Explorer, 2013c*).

Cross-platform support

The web mapping application can be run on computers and on mobile devices which support a Flash plugin.

Multiple Web browsers support

According to its official web site (*Social Explorer, 2013c*), it support the following browsers:

- IE8/Windows
- IE7/Windows
- IE6/Windows (version 6.0.2900.2180 or higher)
- Firefox 3/Windows
- Firefox 2/Windows
- Firefox 1/Windows
- Firefox 2/Macintosh
- Firefox 1/Windows
- Safari 3/Macintosh
- Safari 2/Macintosh

Interactive visualization

The mapping application provides many interactive map tools, such as zoom and pan. Users can also get information from the map directly, by clicking on the map.

Multiple maps based

Social Explorer provides over 18,000 maps for the users to use (*Social Explorer, 2013a*).

Animation

The animation of Social Explorer is created by the users themselves. The function works with the slideshow tool. To create a slideshow, the user first browses a map, then clicks on the layer to add the current map to the slide. When all maps are added, the user can uses the slideshow controller to play the animation. The controller controls the animation speed, turns on/off the animation effect and turns on/off the looping playback.

Statistical graphics

There are not many statistical graphics included in Social Explorer. The statistical information is presented only with choropleth maps.

Search tool

There is a search tool in the tool bar. It can be used to search the data stored in the database. After the result is found, the map view can zoom to the location or center at the location. That depends on the users' choice.

Export functions

Social Explorer provides many export functions. Animation can be saved as a PowerPoint file. Maps can be saved as a picture file. It also provides a function to generate reports, which can be used to export the data as a report.

3.3.2 China Geo-Explorer

China Geo-Explorer (*Figure 3.8*) is an online web mapping application which has been developed by the China Data Center at the University of Michigan, in cooperation with the All China Marketing Research Co. Ltd. (ACMR) and the State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) of Wuhan University. This online web mapping application aims to provide spatial data services, with data integrated from different sources and types, to support spatial analysis with the development of a couple of embedded application, and to provide a more intelligent application to help decision makers. The application provides support in many areas. In business, it can help merchants with site selection, and making business analyses. For research, it can help researchers making spatial models, assessing the environment and doing field surveys. In education, it can help with spatial studies and spatial analysis. For the government, it can help with regional planning and decision making (*China Data Center, 2013*).

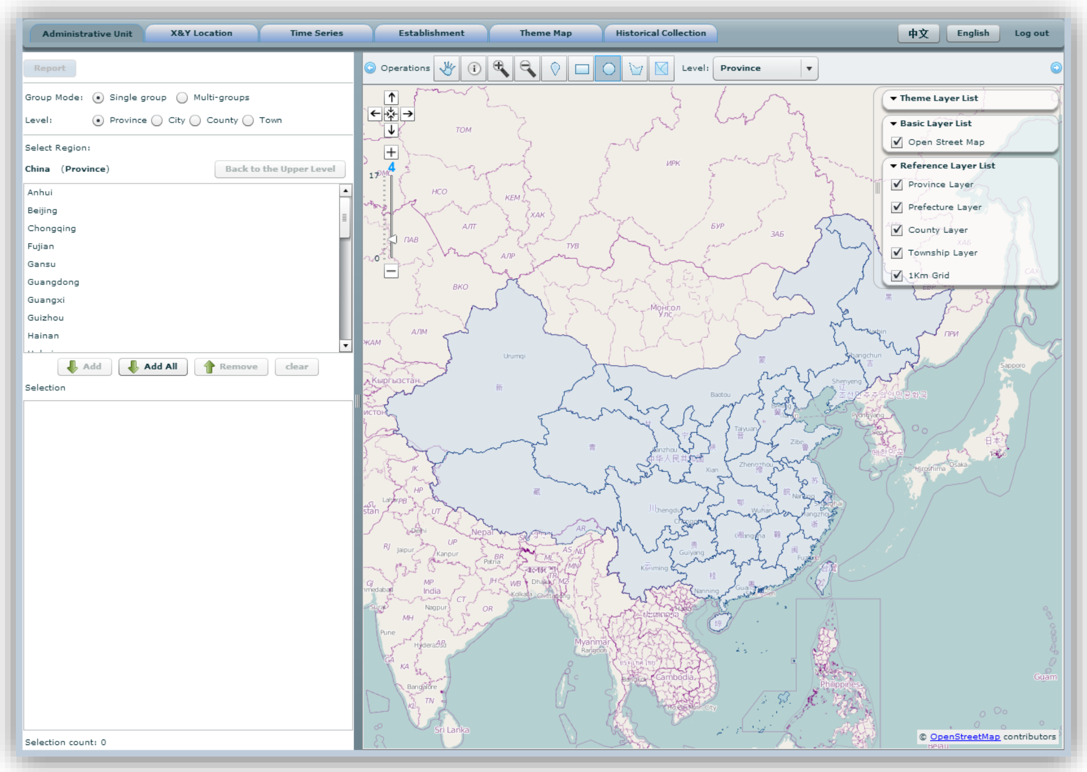


Figure 3.8. China Geo-Explorer.

The database of China Geo-Explorer includes data describing the demography, businesses and geography for all of mainland China. The data covers all 31 provinces, including 345 cities, around 3000 counties and more than 50,000 towns (*China Data Center, 2011*). The data can be generally divided into 5 types: government statistics, population census, economic census, establishments and geography and environment (*Table 3.1*). The government statistics data includes data on three levels: the provincial level, the city level and the county level. It contains information such as price index values, wages and public health, etc. The population census includes all 5 national census data, taken from the years 1953, 1964, 1982, 1990 and 2000. It includes over 2000 variables, such as fertility, marriage and death. The economic census includes the industrial census, basic unit census and economic census. More than 1000 variables are included. Establishment data include the information about different establishments for the entire country. Geography and environment contains geographic information such as land use. It contains maps for different levels such as provincial boundaries, county boundaries and zip codes, etc. (*China Data Center, 2013*).

Table 3.1. Data sources of China Geo-Explorer.

Government Statistics	Provincial statistics City Statistics County Statistics
Population Census	Census 1953 Census 1964 Census 1982 Census 1990 Census 2000
Economic Census	Industrial Census 1995 Basic Unit Census 2001 Economic Census 2004
Establishments	Industrial Census 1995 Basic Unit Census 2001 Economic Census 2004
Geography and Environment	Land Use data Night-Time data Transportation Rivers and Lakes

China Geo-explorer has many functions. It provides fast and flexible selection. The data can be selected by province, city, county or town. Selections can be made by selecting from a list or with a mouse click on the target area. It can calculate and record the coordinates. For example, when the user needs to know the coordinates of a building, just click on it. The application will automatically calculate the coordinates and record them. Spatial statistics are supported. It can make different kinds of patterns or make a trend, such as a time series function, to support the users well understand the data. It can also generate a report for the statistical data. The report contains the data that the user selected and also the patterns and the graphics that the user made. The report can be exported in different formats such as HTML file, PDF file, Word file, Excel file or ESRI Shape file.

Coding languages and plugins

China Geo-Explorer is a web mapping application developed with Flex. So, the Flash plugin is needed to browse the map.

Cross-platform support

The mapping application does not work on mobile devices which do not have the ability to install the Flash plugin.

Multiple Web browsers support

The web mapping application has no official test information, but we can test it in many browsers. It works well in browsers such as Internet Explorer (IE), Chrome and Firefox.

Interactive visualization

The map contains many interactive visualization functions. One of the powerful functions is that it can select and visualize data with a polygon which created by the user.

Multiple maps based

China Geo-Explorer uses an Open Street Map as its background map, and it also contains many maps such as China 2000 Province Population Census Data with GIS Maps, China 2000 County Population Census Data with GIS Maps and China 2000 Township Population Census Data with GIS Maps, etc. (*China Data Center, 2013*).

Animation

It only shows a static graph that is not animation provided.

Statistical graphics

China Geo-Explorer provides many statistical graphics to visualize the statistical information, such as time series chart and choropleth map.

Search tool

There is a search tool embedded in the mapping application. It is easy to retrieve data in this mapping application.

Export functions

Maps can be exported using the map export menu. The format of exported maps can be pdf or jpeg. It also provides for exporting the data in the ESRI shape file format (*China Data Center, 2011*).

3.3.3 DigDag

DigDag map is short for the Digital atlas of the Danish historical-Administrative Geography. It is a research project funded by the Ministry of Science, Technology and Innovation, in cooperation with a wide range of Danish historical research institutions and libraries. The project was run from 2009 to 2012 and aimed at establishing a common database to present the historical geographic data for Danish administrative units; to develop a search engine for the data stored in the database; and to create an application to support the administration historical researches (*DigDag, 2012*).

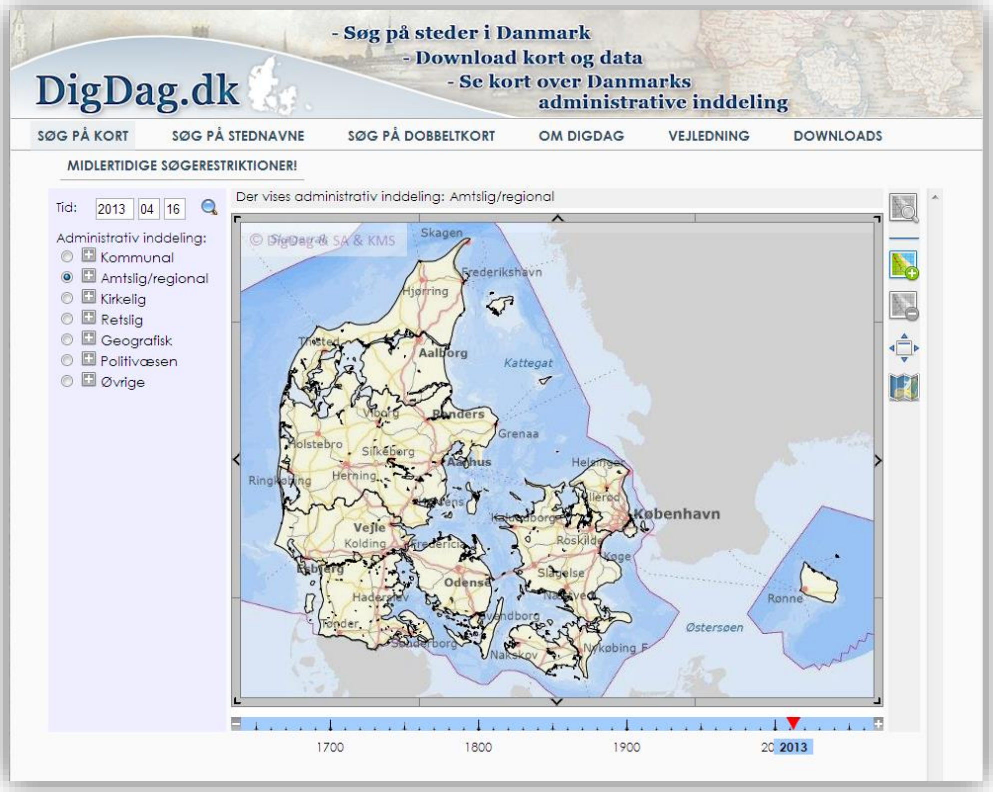


Figure 3.9. DigDag Map

The database is established based on historical information and maps of Denmark from the 16th century up to now. The historical maps are digitized into raster data and vector data, and stored in the database. The database contains information about more than 250,000 places, including the name, origin and meaning, etc. in different time periods. It describes the relationship between counties, municipalities, parishes, dioceses, religious and judicial authorities.

A main feature of the map is that it integrate the time dimension. That can help the user to see both where and when the administrative units change. For an example, when clicking on Aalborg, the table shows the previous names as Alburgis and Olborg, and the time when those names were used (DigDag, 2013a).

Coding languages and plugins

DigDag is developed with XHTML and JavaScript. So, no plugin is needed when browsing this web mapping application.

Cross-platform support

As the web mapping application has been developed without a plugin, it can be run on all kinds of devices, no matter whether they are computer or portable devices.

Multiple Web browsers support

This mapping application works well in Chrome and Firefox, but it is not optimized for Internet Explorer (*DigDag, 2013b*).

Interactive visualization

Interactive visualization is not supported in DigDag.

Multiple maps based

DigDag contains many historical maps for analyzing data, and it also uses Open Street Map as the background map.

Animation

DigDag provides animation with a slider bar to change the time parameter. It works when the user changes the slider manually.

Statistical graphics

DigDag does not contain any statistical graphics to analyze the data. This is because it does not have any demographic data. It only presents geographic data with time changes.

Search tool

DigDag provides a search tool, which can be used to find the data that the user needs.

Export functions

This mapping application cannot export any data, and does not even support the capability to print the map or export an image of the map.

3.4 Summary

When comparing SEDD Map (previous version) with the web mapping applications which were discussed in the previous sections, three comparison tables are provided below to present the differences between the web mapping applications. The comparisons are based on basic information (*Table 3.2*), technical solutions (*Table 3.3*) and technical implementation details (*Table 3.4*).

The basic information about these web mapping programs and applications is presented in *Table 3.2*. From the table we can easily identify that most of the programs or applications are use English as the interface language. The exception is DigDag, which only supports Danish. A large quantity of these map programs or applications started their services around 2008. Social Explorer is the earliest released mapping applications among these, which has been available since 2003. While GAV HTML5/JavaScript is the newest mapping program, released in 2012. All the services provide a free version to browse, while some of them need a commercial license to get full access.

From *Table 3.3* we can see that nearly all of the web mapping applications use plugins. And the most often used plugin is Flash. However, NComVA GAV (HTML5/JavaScript) and DigDag do not require any plugins. This is because they are built with HTML and JavaScript. So, only these two web mapping applications are supported cross-platform. They can be browsed on both computer and mobile devices, such as iOS and Android devices.

Table 3.4 shows the technical implementation details of all the web mapping applications. Except for DigDag, they all provide powerful interactive functions. All the services contain multiple background maps, which enhance the visualization of their services. Animation and visualizing the data with statistical graphics are also very popular functions used in most of the web mapping applications. Data export is also a commonly used function. The exported data can be a picture, an ESRI shape file or a report.

From the comparison tables and the data we have discussed, we find that SEDD Map should be improved in the following ways:

First, SEDD Map has to enhance its compatibility. As the techniques are improving, many kinds of devices have the ability to browse a web site. So, SEDD Map has to fit these kinds of devices. NComVA GAV (HTML5/JavaScript) shows a very good example of how to enhance the compatibility of a web mapping application. Comparing the two GAV toolkits, though they can achieve the same functions, NComVA GAV (HTML5/JavaScript) can be run on more platforms.

Second, SEDD Map has to improve its visualization capability for demographic data. NComVA GAV Toolkit and other mapping applications show many powerful statistical graphics in their products. These graphics have greatly enhanced the ability for the user to understand the data. So, more colorful statistical graphics should be added to enhance the visualization capability of SEDD Map, such as bar charts, pie charts, and plot charts, etc.

Third, SEDD Map has to improve its visualization capability for geographic data. Many web mapping applications can visualize and retrieve data on different levels, such as the provincial level, county level or city level, etc. However, SEDD Map can only retrieve the data on one level. For example, when clicking on a property, the map will show the information of the property. It cannot shows the county level information.

Finally, SEDD Map has to improve visualization of individual data. Compared with other mapping applications used to visualize and analyze demographic data, only SEDD Map has the ability to visualize the data on an individual level. This means one person or house can be presented on the map. Except for China Geo-Explorer, the other mapping applications only visualize the data on an aggregate level, such as visualizing the total population in a particular area. When the SEDD database has been integrated with geographic information, on an individual level we can improve the visualization of SEDD Map. For example, we can use SEDD Map to present a time series for a particular person, e.g. using the migration in/out data to visualize a person's migration track during his/her life.

Table 3.2. Comparison between the web mapping services - Basic information

	Application/Program	Languages	Launch Year	License Type
SEDD Map (previous)	Application	English	2011	Commercial and Free
NComVA GAV (Flash)	Program	English	2008	Commercial and Free
NComVA GAV (HTML5/JavaScript)	Program	English	2012	Commercial and Free
Parliamentary Elections 2010	Application	Swedish	2008	Free
Postcode Atlas	Application	English	2008	Free
Statistic Atlas	Application	English	2008	Free
China Geo-Explorer	Application	English Chinese	2008	Commercial and Free
Social Explorer	Application	English	2003	Commercial and Free
DigDag	Application	Danish	2009	Free

Table 3.3. Comparison between the web mapping services – Technical solutions

	Coding Languages	Plugin	Cross-platfrom Supported	Multiple Browsers Supported
SEDD Map (previous)	Silverlight	Silverlight	No	Yes
NComVA GAV (Flash)	Flex, ActionScript	Flash	No	Yes
NComVA GAV (HTML5/JavaScript)	HTML5, JavaScript	No	Yes	Yes
Parliamentary Elections 2010	Flex, ActionScript	Flash	No	Yes
Postcode Atlas	Flex, ActionScript	Flash	No	Yes
Statistic Atlas	Flex, ActionScript	Flash	No	Yes
China Geo-Explorer	Flex	Flash	No	Yes
Social Explorer	Flex	Flash	No	Yes
DigDag	xHTML, JavaScript	No	Yes	Not for Internet Explorer

Table 3.4. Comparison between the web mapping services – Technical implementation details

	Interactive Visualization	Multiple Map Based	Animation	Statistical Graphics	Search Tools	Export Functions
SEDD Map (previous)	Yes	Yes	No	Yes	Yes	Map
NComVA GAV (Flash)	Yes	Yes	Yes	Yes	No	Report & Map
NComVA GAV (HTML5/JavaScript)	Yes	Yes	Yes	Yes	No	Report & Map
Parliamentary Elections 2010	Yes	Yes	Yes	Yes	No	Report & Map
Postcode Atlas	Yes	Yes	Yes	Yes	Yes	Report & Map
Statistic Atlas	Yes	Yes	Yes	Yes	No	Report & Map
China Geo-Explorer	Yes	Yes	No	Yes	Yes	Report & Map
Social Explorer	Yes	Yes	Yes	Yes	Yes	Report & Map
DigDag	No	Yes	Yes	No	Yes	No

4 Web Mapping Application Architecture

4.1 Introduction

Web mapping applications are usually constructed as a traditional web site. They usually use a client-server model with two main parts, the client part and the server part *(Salemi, 1995, Mitchell, 2005*). The client-server model was developed during the 1970s (*Canaday et al., 1974*), and first introduced to computing in the early 1990s (*Özsu et al., 2011*). Now, it plays an important role in computer networks. It provides services in many areas, such as printer service, mail service, web service and file service, etc. (*Schneider and Gersting, 2007*).

Figure 4.1 shows the structure of a basic client-server model (*2-tier* architecture). The communication begins with the client side sending a request to the server side. Then, the server side receives the request. After analyzing the message, the server side retrieves data from a database and provides the data to the client side. At the client side, the computer displays the final result to the user (*Gallaugher and Ramanathan, 1996*).

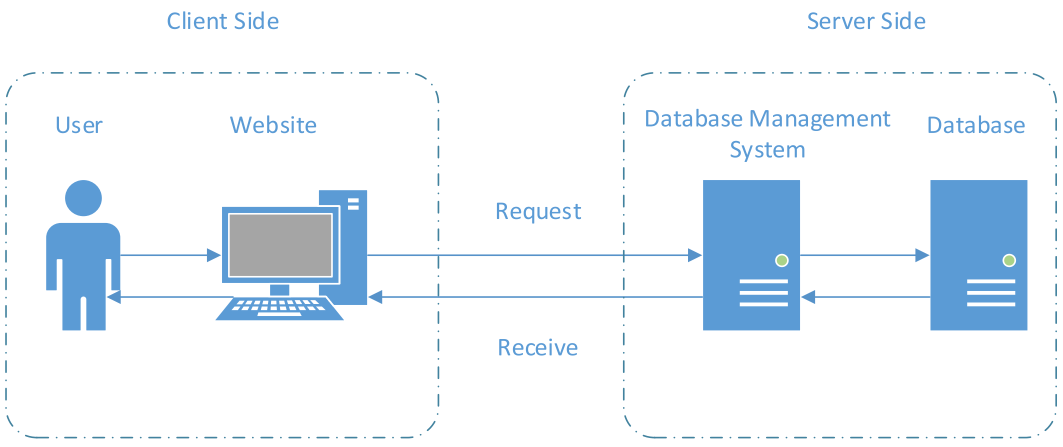


Figure 4.1. Basic client-server model (*Gallaugher and Ramanathan, 1996*).

There are a couple of different types of client-server architectures. The simplest is the *2-tier* architecture shown in *Figure 4.1*. It is called *2-tier* architecture because it is constructed with two components, a client and a server. In practice, a web application is not often that simple (*Ramez and Shamkant, 2010*).

3-tier architecture and *n-tier* architecture are widely used in many web applications (*Miler et al., 2011*). *3-tier* architecture adds one intermediate layer (middle-tier) between the client and database. The middle-tier can be called an application server or web server, depending on which way the web application is used (*Ramez and Shamkant, 2010*). It receives the requests from the client, responds to the requests and sends the requests to an additional server. After receiving the data, it returns the final results to the client side. *n*-tier adds more intermediate layers between the client side and the final data. The middle-tiers could also be application servers to execute the requests or data servers to store the data (*Peng and Tsou, 2003*). So, with the middle-tiers, the *n-tier* architecture increases system flexibility (*Gallaugher and Ramanathan, 1996*) and also the security of web applications (*Ramez and Shamkant, 2010*).

Figure 4.2 shows a commonly used web mapping application, constructed using a *3-tier* architecture. Though different web mapping applications have different structures, the basic structure would be similar to this one. When browsing the web mapping application, the user sends a request from the web site to the web server. The web server analyzes the request and passes the request to the web mapping server to create a map. Then, the mapping server analyzes the request and retrieves data from the map database. The map database then sends the map data (e.g. roads data and soil data) to the web mapping server. After the web mapping server receives the data, it combines the data to create a map. When the map is ready, the web server delivers the map to the mapping web site which displays the map to the user (*Mitchell, 2005*).

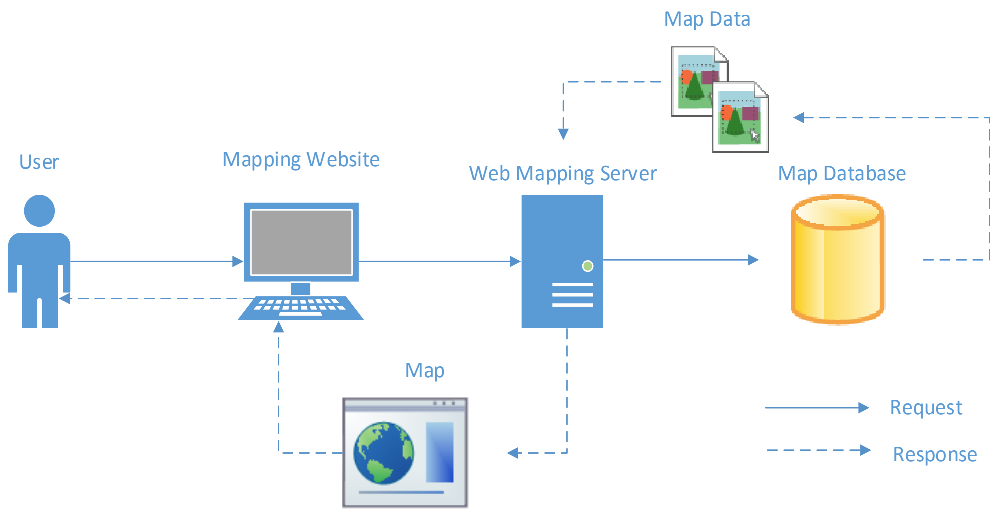


Figure 4.2. A commonly used web mapping application architecture (*Mitchell, 2005*).

4.2 Client Side

The client side refers to everything that happens in the web browser for a web mapping application (*Deborah, 2002*). It plays a significant role in web mapping applications. It works on the local machine of the user and connects to the map server. It visualizes the map data for the user and allows the user to interact with maps. In a normal web mapping application, it visualizes not only the final result but also the map's tools, such as zoom-out and zoom-in. With these tools, users can do analyses and browse web maps easily (*Peng and Tsou, 2003*).

There are several client types in client-server architecture. The two typical client types are thin client and thick client. A thin client only contains the interface to communicate with the server and visualize the results. It does not contain any complicated functions. Instead, all the operations are executed on the server side. When the server side finishes the execution, it returns the final result to the client. A thick client (or fat client) is based on the thin client and contains more functionalities and tools embedded in the client side. With these tools, some operations will not be sent to the server side. They will be executed on the client side (*Alesheikh et al., 2002*). For example, when a user wants to zoom out a part of the map: with a thin client, a new map picture will be sent to the client after the map server executes the function; with a thick client, the client could do the zoom out function by itself with the same map picture. So, the server side will not need to do any executions in this case.

Client-side scripting and/or plugins are needed to develop a powerful thick client for a web mapping application (*Alesheikh et al., 2002*). The term client-side scripting refers to a computer program which embedded directly into a web page to perform functions on the client side, such as JavaScript, VBScript and other scripting languages and systems (*Raggett, 1997*). A plugin is a small computer program which is run in a web browser (*Peng and Tsou, 2003*). For web mapping applications, there are several commonly used plugins, such as Java applets, Flash and Silverlight (*Lammarsch et al., 2008*). The difference between scripting and plugins is that scripts can run directly in the web browser, while plugins must run on the client which has installed the plugin program.

Though plugins are helpful, they have several drawbacks. First, they must be installed on the client machine. This can be a problem for the person who has no permission to install any programs on a computer. Second, it is not easy for the developer to test their web site. A plugin is normally not written by the web site developer, it is an external program. And the same plugins always have different versions for different platforms. If a web site produces some errors, it is not easy for the developer to find the real problem. Third, plugins can affect the running speed of a web page. Plugins waste some internal storage on a computer, and if it is too big or using too many resources to work, the speed of the whole program will slow down.

4.3 Web Server and Application Server

The client side and the server side are two independent parts of the client-server architecture. They have different programs running on each side. They are different programs using different programming languages, there are some communication barriers between the two sides. In order to let the client side communicate with the server side, an intermediate layer called a web server is used (*Peng and Tsou, 2003*). A web server, also called HTTP server, uses the TCP/IP transport protocol to communicate between the client side and the server side (*Pai et al., 1999*). In the simplest case, it works as the user sends a request from a browser over the Internet. Then it parses the request to analyze which file is needed. After analyzing the request it reads the files from the server side and sends it back to the browser over the Internet (*Yeager, 1996*).

There are several ways for a web server to respond the request from the server side. One method is to respond to the request by sending an existing HTML file containing the map result. This method is used in a static map application. The user requests a map just like requesting another web page. The web server can also respond to the request by sending Java applets or ActiveX controls to the web client. This method is the process that occurs in an interactive map application. The result can be executed by using plugins on the client side. Another method is to send the request to other programs to execute the result. If the request needs to be executed in another program, then an application server is needed (*Peng and Tsou, 2003*).

An application server is also an intermediate layer. It connects several software components with the web server (*Shekhar and Xiong, 2008*), and presents itself as a translator between the web server and the map server. It can establish, maintain and terminate the connection between the web server and the map server. It can also interpret and manage requests from the client side and send them to the map server (*Peng and Tsou, 2003*).

CGI application can be used in the application server to enhance functionality (*Peng and Tsou, 2003*). CGI stands for Common Gateway Interface. It is a standard method for an application server to run executable applications. It could be written in any programming language or scripting language (*Shekhar and Xiong, 2008*). As an example, *MapServer* (which is an open source program, different from the term Map Server in *4.4*) is a CGI program which could be embedded in the application server to display dynamic spatial maps (*MapServer, 2013*). When a user sends a request from the browser to the *MapServer*, it uses the information to generate a requested map. As a CGI program, the requested map not only contains a picture of the map but also includes the images for legend and scale bar, etc.

4.4 Map Server

A map server is an important component of a web mapping application. It is used to perform spatial queries, make spatial analyses, generate maps and deliver the final map to the users *(Peng and Tsou, 2003*). For example, ArcGIS Server, which is developed by ESRI (Environmental Systems Research Institute), is a kind of map server. It provides the traditional GIS functions such as data extraction and spatial analysis, etc. It can also analyze requests from the user and generate a map for map users.

The output of a map server can be of two kinds. One kind is a simple map image, such as raster files in the formats JPEG, BMP and PNG. Another kind is to generate a map composition which combines several layers. Each layer contains one or several elements or features, such as color, legend and scale bar and so on (*Peng and Tsou, 2003*).

Generally speaking, a map server has several basic functions. First, it can produce a symbolized map with the user's request. Second, it can respond basic queries about the content and attributes of the map. Third, it can extract data from the map database which the user requests. Fourth, it can make spatial analyses such as buffering and overlay. Fifth, it can communicate with other programs to know what kind of service and data the program can query. Sixth, it can provide an interface for the user to publish the data. Last, it can pass the request to other map servers to make other analyses (*Peng and Tsou, 2003*).

4.5 Map Data Server

A map data server is the component that stores and manages the data for web mapping applications. It contains not only the database but also the database management system (*Peng and Tsou, 2003*). A database is a container that stores the data and the relationship between the data (*Beighley, 2007*). A database management system is a software application that is used to create, organize, query and update the database. There are mainly three types of database management systems are available for GIS users. They are relational database management systems (RDBMS), object database management systems (ODBMS) and object-relational database management systems (ORDBMS) (*Longley, 2005*).

The client side or the web server acquires data from the data server by using SQL (Structured Query Language). Therefore, a data server is often called an SQL server. As different server vendors may use different versions of SQL, there is often database middleware used to access different databases. The three commonly used database middleware are Open Database Connectivity (ODBC), Java Database Connectivity (JDBC) and Object Linking and Embedding, Database (OLE DB) (*Peng and Tsou, 2003*).

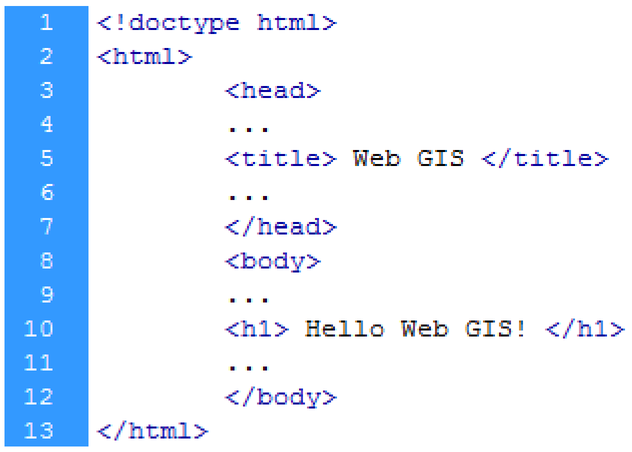
5 HTML5, JavaScript & CSS3

5.1 HTML Introduction

Hypertext Markup Language (HTML) is a markup language which is used to construct a web page and present the information of a web page in a web browser (*Raggett, 1998*). HTML is developed by World Wide Web Consortium (W3C), which is an international standard organization, and the Web Hypertext Application Technology Working Group (WHATWG), which is organized by the people who are interested in evolving HTML and related techniques (*Lawson and Sharp, 2011*). Another HTML version is the Extensible Hypertext Markup Language (XHTML) which writes the HTML code as Extensible Markup Language (XML) developed by W3C (*W3Schools, 2013*).

Several versions of HTML have been developed since the first document called "HTML tags" was mentioned by Tim Berners-Lee, who is also considered as the inventor of the World Wide Web, in the year 1991. In 1995, HTML 2.0 was introduced to developers. Two years later, in 1997, a new version called HTML 3.2 was published by W3C. In 1997, the fourth version of HTML (HTML 4.0) was published at the recommendation of W3C (*Raggett, 1998*). In 1999, HTML 4.01 was published. In 2012, W3C published the candidate version of the HTML standard, HTML5 (*Robbins, 2013*).

HTML is written with a set of HTML elements. An HTML element is composed of a start tag, an end tag and everything that is in between (*Robbins, 2013*). An HTML tag is written with angle brackets. In an HTML web page, tags always appear in pairs, with a start tag and an end tag. A start tag is written in angle brackets with the name of the element, while the end tag is written using the same name as the start tag, but with a slash in front of the element's name, such as *<html>…</html>*, which is used to distinguish from the start tag. *Figure 5.1* shows some HTML elements written in an HTML file.



```
<!doctype html>
<html>
        <head>
        ...
        <title> Web GIS </title>
        ...
        </head>
        <body>
        ...
        <h1> Hello Web GIS! </h1>
        ...
        </body>
</html>
```

Figure 5.1. Structure of an HTML file.

HTML elements are the most important part of an HTML file. They define the structure of a web page, the word style that displayed on the screen and can embed pictures or videos in a web page, etc. (*Robbins, 2013*). An HTML file has a certain structure that starts with an `html` tag and ends with a `/html` tag. It usually contains two parts: the `head` element and the `body` element.

In the `head` element, the main information is stored, such as the metadata, title and some JavaScript programs, etc. The file is read from the beginning to the end. So some other elements which also should be loaded first are stored in the `head` element, such as Cascading Style Sheets (CSS).

After the `head` element is the `body` element. The `body` element stores all the contents of an HTML file, such as text, tables and lists, etc. In the `body` element, a set of elements could be embedded. For editing the text, the `p` element is used to define the paragraphs. The `h` element is used to define the section headings. The section headings could also be set in different levels. To do this, just adds a number behind the `h` tag, such as `h1` or `h2`. When editing a media source, such as embedding a picture in a web page, there is an `img` tag that is used. So, with the help of the different HTML elements, a well-designed web page could be rendered in a web browser to present to web users.

There are several characteristics of Hypertext Markup Language. First, it is a simple and easy language. It is very easy to use and understand by the developer. Second, graphics can be easily added in the web page. So, this may enhance the presentations and make them more vivid and interesting. Third, it can run on all platforms, such as Mac OS, Windows OS and Linux OS. So, users can view web pages with any computer or mobile phone.

5.1.1 HTML5

HTML5 is the 5th version of Hypertext Markup Language standard developed by both W3C and WHATWG. It is aimed at replacing the previous HTML standard, HTML 4.01, which was published in 1999 and has not been updated since 2000. The new standard was first established by the Web Hypertext Application Technology Working Group (WHATWG) in 2004. In 2012, the candidate version was published. W3C plans to release the stable HTML5 specification by the end of 2014 (*Pilgrim, 2103*).

The core aim of HTML5 is to improve the ability to support the latest multimedia formats and reduce the needs for plugins (e.g. Flash, Microsoft Silverlight and Oracle JavaFX) to keep web pages easy to read by both humans and computers.

There are many functions for drawing in HTML, such as Canvas, SVG and VML. Canvas element are used to draw graphics directly on a web page based on bitmap graphs. It does not require the installation of any plugins in the browser to draw and render graphics on the web page (*Fulton and Fulton, 2011*). SVG stands for Scalable Vector Graphics. It is an XML based mark-up language. It is also the best method for drawing vector graphics on a web page. In the HTML5 specification, inline SVG is supported. This means SVG could be directly embedded in an HTML web page (*Dailey et al., 2012*). VML is the abbreviation for Vector Markup Language. It is a mark-up language based on XML. Only Internet Explorer support this language, so it is not commonly used by developers.

HTML5 has added many new elements to enhance its ability to render web pages for users. These include `audio`, `video` and `canvas` (*Pilgrim, 2103*). The `audio` element and the `video` element are two elements which are used to embed audio and video in a web page. In previous version of HTML, in order to run an audio or video clip, a plugin need to be installed. For instance, an audio file always runs in Windows Media Player Plugin for Windows users, and for video files, Flash plugin is most often chosen. When the new elements are added, most of the plugins could be dropped from web browsers.

5.1.2 Canvas Element

The `canvas` element, which is a new element added in HTML5, is used to draw graphics directly on a web page. It was first introduced by Apple in the Mac OS X WebKit component in 2004. And now, most of browsers, such as Internet Explorer, Firefox, Opera, Chrome, and Safari, support this element.

The `canvas` element is only a graphics container. That means it cannot draw any graphics by itself, and it needs the help of JavaScript (*Fulton and Fulton, 2011*). The `canvas` element only defines the area where the graphics are placed by defining width and the height and using CSS to define its position. After the `canvas` is set up, graphics could be drawn onto it with a set of JavaScript programs. With the help of JavaScript, dynamic graphic can be rendered in the browser. And several functions can be achieved, such as drawing a line, a circle and filling in colors for a rectangle or circle. Text could be drawn in the `canvas` element and also pictures can be added in it. The `getContext()` method returns an object that provides methods and properties for drawing on the canvas.

Canvas 2D

HTML Canvas 2D Context is a JavaScript API which is used to draw 2D graphics on the `Canvas` element. It is a separate specification from HTML. It works with the command `getContext("2d")` *(Fulton and Fulton, 2011*).

The following figure (*Figure 5.2*) shows an HTML clip which is used to draw a rectangle on a `canvas` element. The first line gives the `canvas` element the name `myCanvas`. Then, uses JavaScript to draw a rectangle on the canvas starting from line 2. Before drawing the rectangle we have to let the computer know where to put this graphic. So, line 3 gives the information: draw the graphic on `myCanvas`. The 4th line uses `getContext('2d')` to define what kind of graphic should be drawn, in this case it uses *2d* to draw a 2D graphic. Line five uses `fillstyle` to fill the color of the rectangle. In line six, a rectangle is drawn with the upper-left corner's position at (0, 0), with width 80 and height 100. The running result of the code is presented in *Figure 5.3*.

```
<canvas id="myCanvas"></canvas>
<script>
    var c=document.getElementById('myCanvas');
    var ctx=c.getContext('2d');
    ctx.fillStyle='#000000';
    ctx.fillRect(0,0,80,100);
</script>
```

Figure 5.2. Canvas draw a 2D rectangle

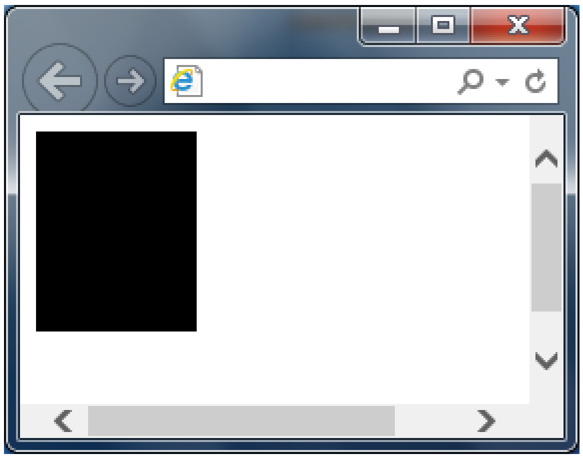


Figure 5.3. The result of running the code in *Figure 5.2.*

Canvas 3D

At this stage, there is no specification developed by W3C or WAHTWG to draw 3D objects on the `Canvas` element. However, the `Canvas` element could use WebGL, which is a JavaScript API for rendering interactive 3D graphics (*Danchilla, 2012*), to present 3D contents. It will work using `getContext("webgl")` to define the object (*Pfeiffer, 2011*).

A couple of JavaScript libraries have also been developed to help developers to use WebGL to draw 3D figures. These include Curve3D, O3D, Three.js and X3DOM. These JavaScript libraries are used to reduce the complexity and increase the speed when coding with WebGL. Because WebGL works directly with the GPU, it is hard to code (*The Khronos Group, 2011*).

5.2 JavaScript

JavaScript was first introduced and developed by Netscape, which is a US computer services company. In the beginning, Netscape developed a language named LiveScript which was used to add some basic scripting capabilities to their web browser products. When they added some support for Java applets on their later product (called Navigator 2), they changed the name from LiveScript to JavaScript (*Mastering JavaScript 1997*).

JavaScript is a script-based programing language that is used by web developers to enhance the functionality of web applications. It can be used on both the client side and the server side. On the client side, it is used to develop some dynamic web pages which can be directly executed by web browsers. On the server side, it can be used to create programs to process information delivered from the client side and then return the result to display on the user's browser (*Mastering JavaScript 1997*).

JavaScript can either be embedded directly into an HTML file or stored as a separate file. When programing with JavaScript it is always presented in an HTML file in the `script` element. Within the start tag, the language attribute is set to JavaScript to identify the script language. When JavaScript is stored as a single file, it uses a filename extension *.js*. When using *.js* files, the `script` tag is required to import the functions to the HTML file. A src attribute is needed to tell the HTML file where the *js* file is located (*Mastering JavaScript 1997*).

Figure 5.4 shows an example where JavaScript is directly embedded in an HTML file. The HTML file is called *Test1.html*. When the file is executed, a sentence (Hello GIS!), will be displayed on the browser. *Figure 5.5* shows a *.js* file which contains a JavaScript program embedded in an HTML file. The HTML file is called *Test2.html*, while the *.js* file is named *write.js*. When *Test2.html* file is executed, the *write.js* file will be loaded. This will print the sentence on the user's browser.

Test1.html

```
<html>
    <body>

        <script type="text/javascript">
        document.write(" Hello GIS! ");
        </script>

    </body>
</html>
```

Figure 5.4. JavaScript embedded in an HTML file.

Test2.html

```
<html>
    <body>

        <script src="write.js">
        </script>

    </body>
</html>
```

write.js

```
document.write(" Hello GIS! ");
```

Figure 5.5. JS file embedded in an HTML file.

Actually, *Test1.html* and *Test2.html* return the same result to the user. But in practice, *Test2.html* with the *write.js* file is normally preferred. This is because using a *.js* file avoids redundancy and increases efficiency. For example, if there are ten HTML files that want to use the same function, the developer will not need to write the same code in each file. Instead, just write a *.js* file and import it when needed. This not only saves coding time but also saves space for storing the code.

5.2.1 Dojo Toolkit

Dojo Toolkit is an open source modular JavaScript library which can be used to help developers to improve efficiency in developing web applications and web pages (*Mounts, 2009*). It has been used in many web applications and JavaScript APIs, such as ArcGIS API for JavaScript.

Dojo Toolkit can be separated into five parts: *Base*, *Core*, *Dijit*, *Dojox* and *Util* (*Mounts*, *2009*).

- *Base* is the kernel of Dojo.
- *Core* constructs on Base and supply additional facilities.
- *Dijit* is short for Dojo widget. It is a library that contains many JavaScript files, ready for the user to use.
- *Dojox* is the extension part of Dojo, which contains the modules that are not stable for use in Dojo or *Dijit*.
- *Util* contains the build tools, such as optimization, documentation, style-checking, and testing tools.

5.3 CSS

CSS is short for Cascading Style Sheets. It is a style sheet language used to describe the visualization of a web page, including colors, layout, and fonts. It allows web pages to adapt the presentation to different types of devices, such as large screens, small screens, or printers. It is usually used in an HTML file. However, it can also be used to present some XML based documents such as SVG (*McFarland, 2012*).

The first version of the CSS specification was published in 1996. At that time, no browser could support all its functions. It applies functions such as setting font properties, defining text color, setting alignment of text and defining the position of the elements, etc. (*Blansit, 2008*).

The CSS level 2 specification was developed by W3C and published in 1998. It brought some properties such as using absolute, relative and fixed to define the position of the elements. It also added a z-index and more properties in this specification. CSS level 2 vesion1 also called CSS2.1 was lately released a Candidate Recommendation in 2004, but it was reverted to a Working Draft in 2005. Finally became a W3C Recommendation in 2011 (Johnson, 2013).

5.3.1 CSS3

CSS level 3 was first published as a draft in 1999. Unlike previous CSS versions, CSS3 is divided into several separate documents. Each document is called a module. Four of these modules have been made formal recommendations by the end of May, 2013. They are Media Queries, Namespaces, Selectors Level 3 and Color (Johnson, 2013).

There are some new features added in CSS3. These include *Rounded Corners*, *Background Decoration*, *Colors*, *Text Effects* and *Web Fonts*. *Rounded Corners* is used to change the four angled corner of a rectangle to rounded corners. *Background Decoration* can be used to define the background of the content. *Colors* imported some new color schemes such as HSL, HSLA and RGBA. *Text Effects* added some properties for visualizing text, such as text shadow. *Web Fonts* introduced many new collections of web fonts (*Grotophorst, 2012*).

Some powerful functions are also added in CSS3, such as 2D and 3D animation, which are used to aid HTML5 to replace plugins such as Flash and reduce the use of JavaScript. 2D and 3D animation can use CSS to present many animation effect for both 2D and 3D graphics, such as rotate and fade in/out. However, CSS3 animation is not supported by all browsers. Only the newest browsers, such as Internet Explorer 10, can render these effects. So, at this stage, using these functions are limited (*Grotophorst, 2012*).

6 ArcGIS for Server

6.1 Introduction

ArcGIS for Server is a GIS software package developed by ESRI (Environmental Systems Research Institute), which is an international supplier of GIS software products. ArcGIS for Server is one of the core products of the ArcGIS Software products family. It provides the platform for sharing GIS resources, such as spatial data and maps with users through the Internet. It works on a server computer, by receiving and processing the requests from devices such as computers, tablets or mobile phones, sending back the results to the requesting devices (*ESRI, 2013j*).

The architecture of a web mapping application which is built with ArcGIS for Server is shown in *Figure 6.1*. From the figure, we can see that web mapping applications contain several components, such as web clients, a web server, a web adapter, a GIS server, a data server, a publisher that publishes the service, and a GIS server administrator who manages the server side of the whole application (*ESRI, 2013e*).

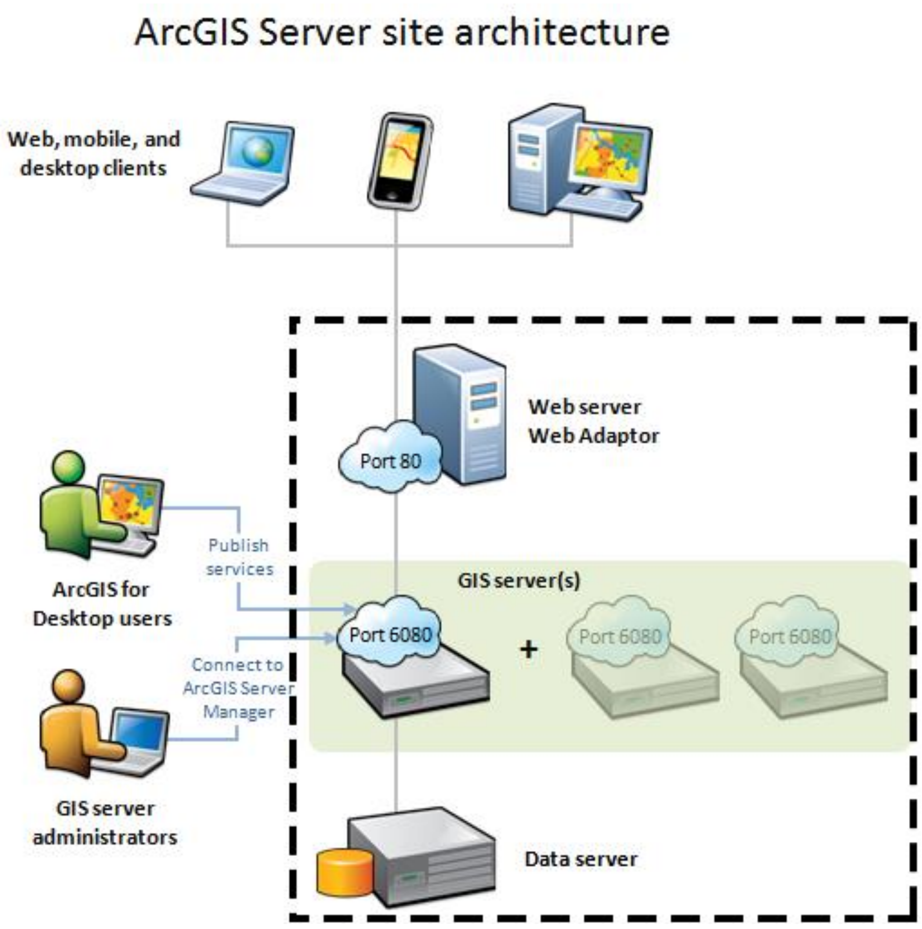

ArcGIS Server site architecture
Web, mobile, and desktop clients
Web server
Web Adaptor
Port 80
ArcGIS for Desktop users
Publish services
GIS server(s)
Port 6080
+
Port 6080
Port 6080
Connect to ArcGIS Server Manager
GIS server administrators
Data server

Figure 6.1. The architecture of a web site constructed with ArcGIS for Server (*ESRI, 2013e*).

An ArcGIS Server site refers to an installation of ArcGIS for Server (*Law, 2013*). The purpose of an ArcGIS Server site is to receive and execute the requests sent from the client side, and return results to users (*ESRI, 2013e*). *Figure 6.1* in the dashed box shows the components that constitute an ArcGIS Server site: a GIS server, a web adaptor, a web server and a data server. A GIS server is the main part of an ArcGIS Server site. In a single ArcGIS Server site, there should be at least one GIS server. The GIS server is used to fulfill the requests to the web mapping application. It can serve many functions, such as drawing maps, making spatial analyses, query data, etc. Every request sent from the client side is handled by the GIS server (*ESRI, 2013e*). A web adaptor is used to integrate the ArcGIS Server with the customers' enterprise web servers, such as Microsoft IIS and Oracle WebLogic (*Law, 2013*). It receives requests from a common URL, and passes the requests to different GIS server machines in the same ArcGIS Server site (*ESRI, 2013e*). An ArcGIS Server site may contain many web adaptors, but a web adaptor can only be configured for one ArcGIS Server site (*Law, 2013*). A web server can be used to host web applications, such as ArcGIS Viewer for Flex and ArcGIS Viewer for Silverlight (*Law, 2013*). A data server is used to store GIS data resources. The data can be stored directly on the GIS server or using a central data repository (e.g. a network folder or an ArcSDE geodatabase). The data stored in the data server can be any kind of data, including maps, globes and geodatabases (*ESRI, 2013e*).

An ArcGIS Server site can be set up with a single machine or a multiple machine architecture, and it can be run in both Windows and Linux operating systems (*Law, 2013*). A single machine architecture is constructed with only one machine which has the ArcGIS for Server application installed. With a single GIS server, the ArcGIS Server site can be used for finishing simple tasks, such as sharing resources and solving easy problems. But when working with large data sets and different types of data, a multiple machine architecture is always the best choice. Multiple machine architecture means that the ArcGIS for Server software is installed on several server machines, and these machines are combined together to set up the ArcGIS Server site. In practice, there are two types of multiple machine architectures: One type manages the GIS servers independently, another type combines the GIS servers into clusters (*ESRI, 2013b*).

A typical multiple machine ArcGIS Server site is constructed as *Figure 6.2* shows. In this example, there are 3 GIS servers working in this ArcGIS Server site. Each GIS server has installed the ArcGIS for Server software and works independently, but they share the same web adapter. The web adapter is the single point which passes the requests from the client side to the GIS servers. When the requests are sent to the web adaptor, the web adaptor will automatically choose one GIS server to fulfill the request. Then the GIS server sends the result back to the user through the web adaptor. The advantage of using this type of architecture is that more GIS servers make the service more stable. If one GIS server is not working for some reason, the other redundant GIS servers will continue to function. So, the whole service would not be effected (*Law, 2013*).

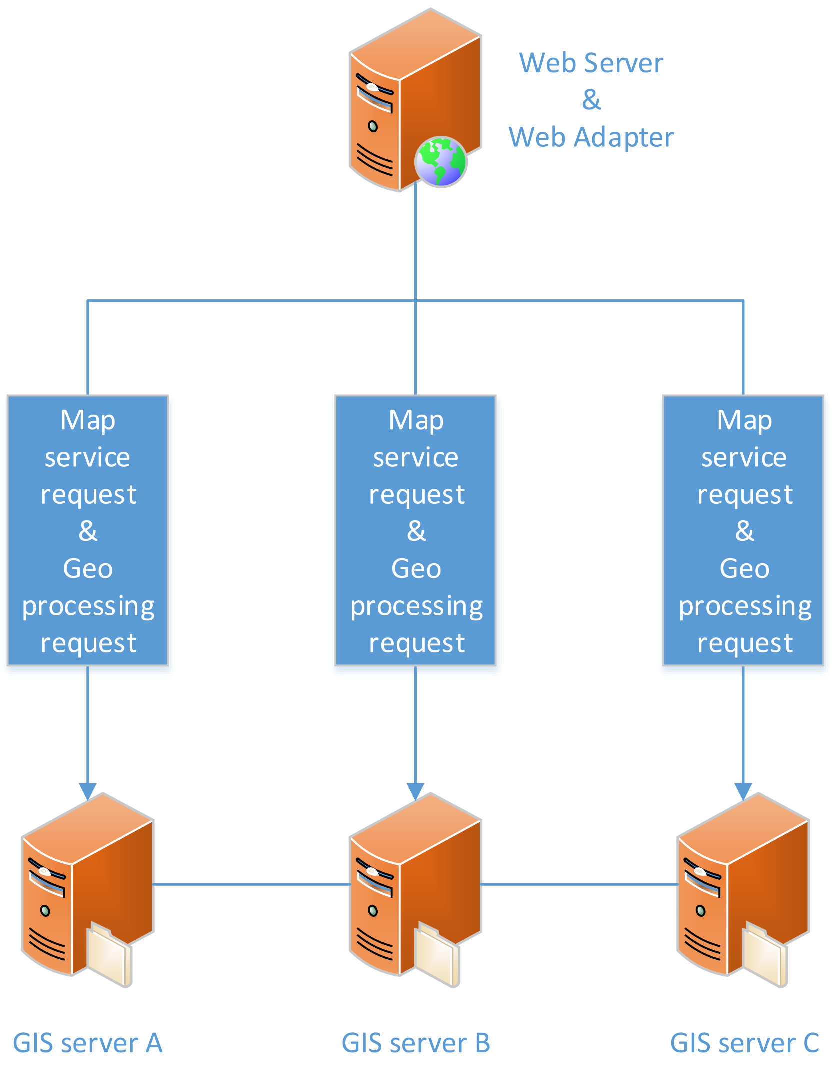

Web Server
&
Web Adapter
Map service request & Geo processing request
Map service request & Geo processing request
Map service request & Geo processing request
GIS server A
GIS server B
GIS server C

Figure 6.2. Multiple machine ArcGIS Server site architecture (*Law, 2013*)

GIS servers could also be grouped into different clusters in a multiple machine ArcGIS Server site (*ESRI, 2013b*). A cluster refers to a logic grouping of machines, which are configured with the same hardware specification (*Law, 2013*). An ArcGIS Server site must contain at least one cluster. When first installing ArcGIS for Server, it will automatically set up one default cluster. Each cluster can be used to manage a particular service. For example, *Figure 6.3* shows an ArcGIS Server site which is constructed with three GIS servers. They are grouped into two clusters, cluster A and cluster B. In this case, cluster A is used to run all the map services, while cluster B is used to run the groprocessing services. So, when the web adaptor receives a request, it will analyze the request and identify the appropriate cluster to solve the problem (*ESRI, 2013b*).

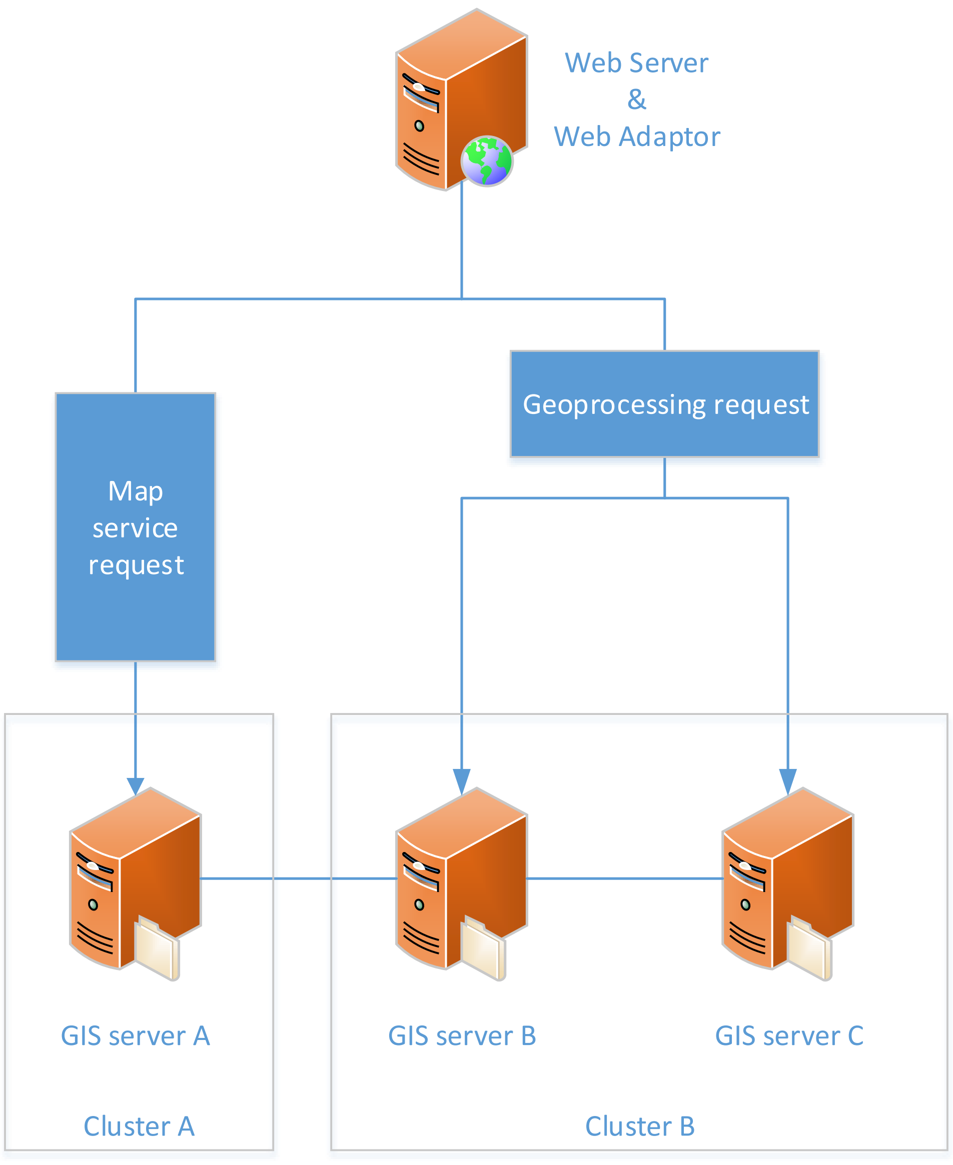
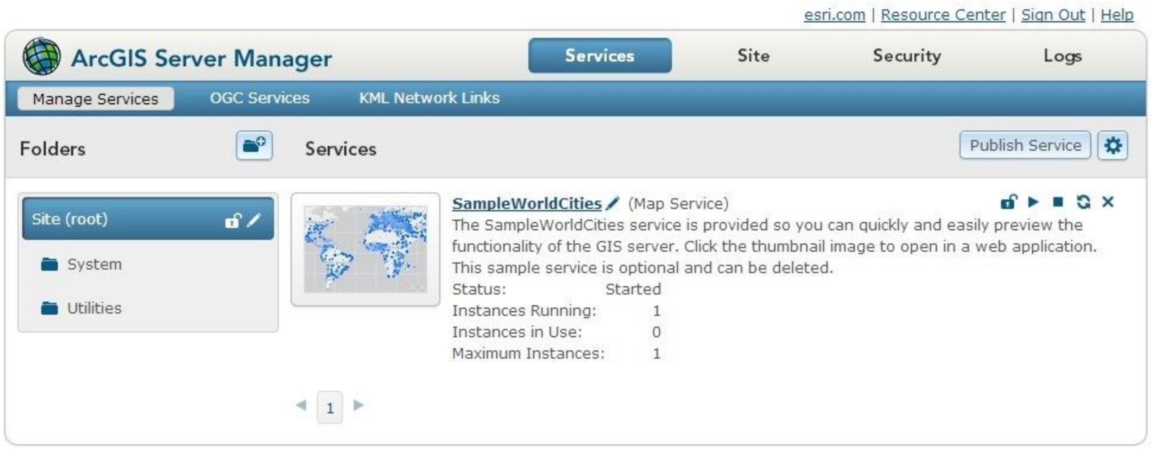


Figure 6.3. Multiple machine ArcGIS Server site constructed with 2 cluster (*ESRI, 2013b*)

Personnel are also important components in an ArcGIS Server site (*ESRI, 2013e*). After the software and hardware are set up, the service cannot function without the people who manage the service, publish the data and maintain the service. To perform this work, ArcGIS Server site administrators, ArcGIS for Desktop content authors and publishers and application developers are needed.

The ArcGIS Server site administrator is the person who installs the software and configures the web application (*ESRI, 2013e*). When installing ArcGIS for Server software, a single operating level account is needed. The default name is set to ArcGIS Server account. The ArcGIS Server site can be accessed from three points by the administrator: ArcGIS Server Manager (*Figure 6.4*), the Services Directory (*Figure 6.5*), and the ArcGIS Server Administrator Directory (*Figure 6.6*) (*Law, 2013*).


esri.com | Resource Center | Sign Out | Help

ArcGIS Server Manager

Services | Site | Security | Logs

Manage Services | OGC Services | KML Network Links

Folders

Services

Publish Service

Site (root)

System

Utilities

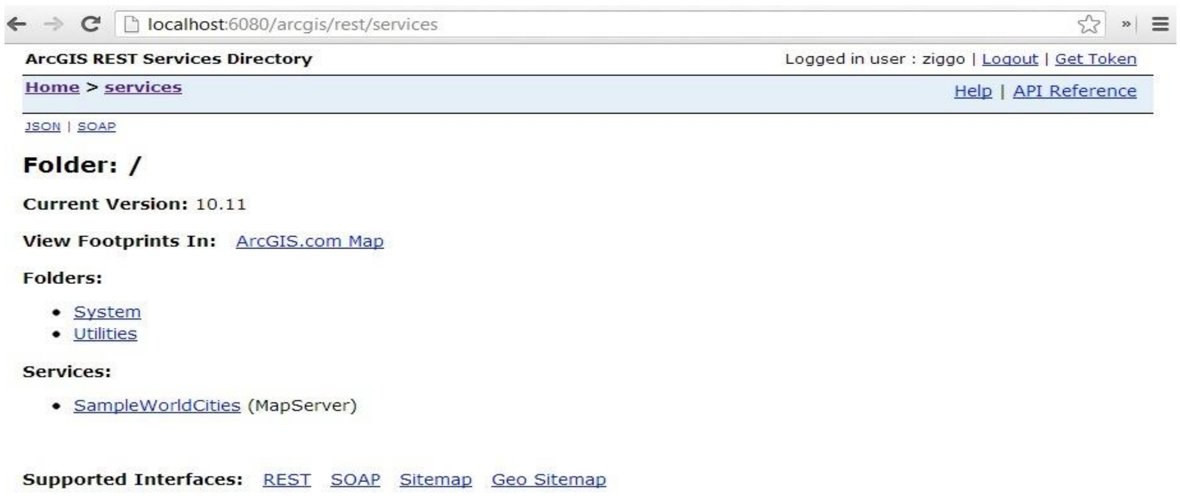
SampleWorldCities (Map Service)

The SampleWorldCities service is provided so you can quickly and easily preview the functionality of the GIS server. Click the thumbnail image to open in a web application. This sample service is optional and can be deleted.

Status:	Started
Instances Running:	1
Instances in Use:	0
Maximum Instances:	1

1

Figure 6.4. ArcGIS Server Manager is a web browser based application which is used to manage the web service, configure the properties of the site, as well as provide access to query and view logs.


localhost:6080/arcgis/rest/services

ArcGIS REST Services Directory

Logged in user : ziggo | Logout | Get Token

Home > services

Help | API Reference

JSON | SOAP

Folder: /

Current Version: 10.11

View Footprints In: ArcGIS.com Map

Folders:

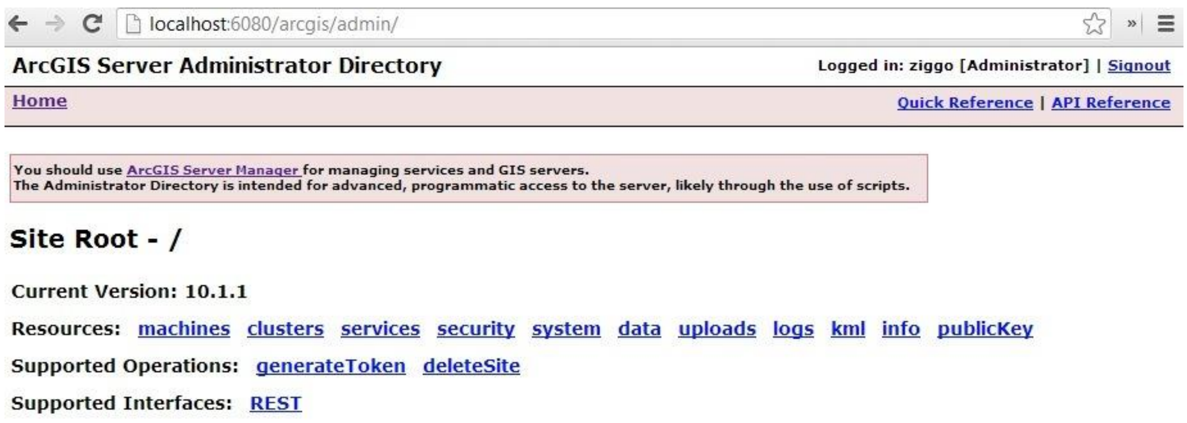
- System
- Utilities

Services:

- SampleWorldCities (MapServer)

Supported Interfaces: REST SOAP Sitemap Geo Sitemap

Figure 6.5. The Services Directory is a web browser-based web page which used to view all the GIS resources that are available from the site.


localhost:6080/arcgis/admin/

ArcGIS Server Administrator Directory

Logged in: ziggo [Administrator] | Signout

Home

Quick Reference | API Reference

You should use ArcGIS Server Manager for managing services and GIS servers.
The Administrator Directory is intended for advanced, programmatic access to the server, likely through the use of scripts.

Site Root - /

Current Version: 10.1.1

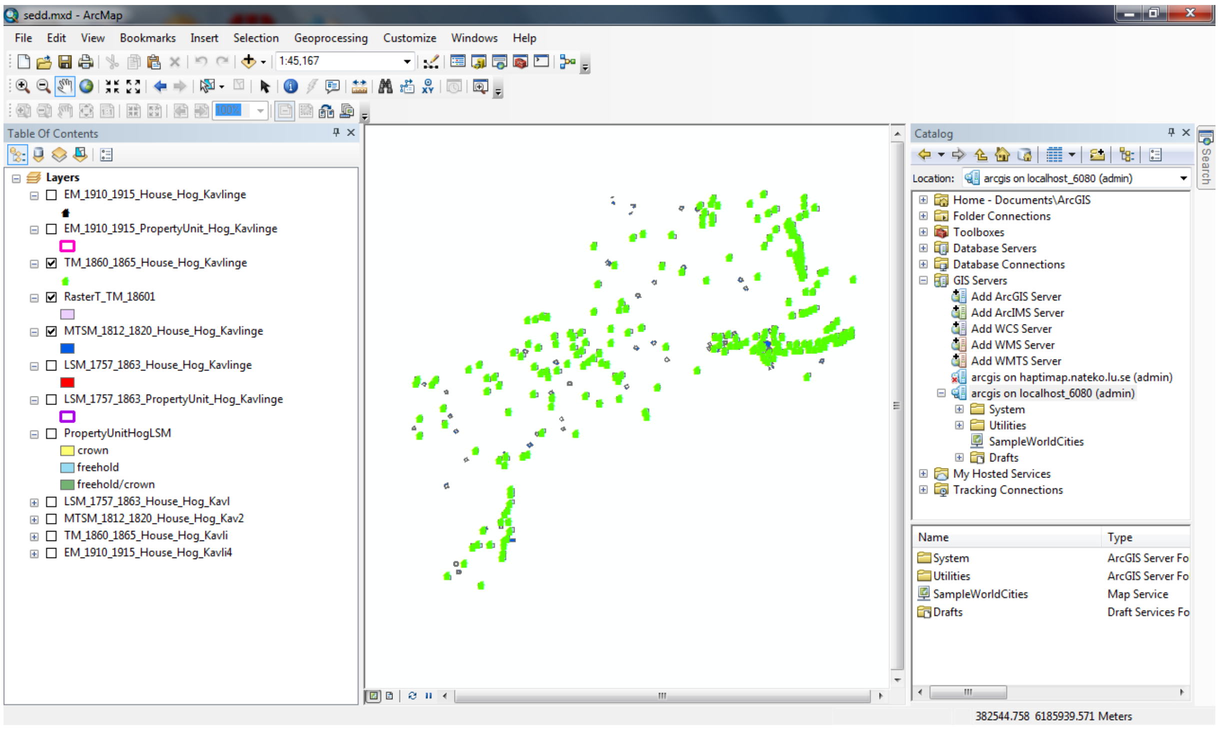
Resources: machines clusters services security system data uploads logs kml info publicKey

Supported Operations: generateToken deleteSite

Supported Interfaces: REST

Figure 6.6. The ArcGIS Server Administrator Directory is usually called the REST admin API, and is used to view the settings of the ArcGIS Server site's configuration.

ArcGIS for Desktop content authors and publishers are the persons who publish the GIS resources such as maps and geodatabases to the site. The authors use ArcGIS for Desktop (e.g. ArcMap, ArcCatalog, and ArcGlobe) to edit and update GIS resources. After the resources are ready, the authors could use the ArcGIS for Desktop software to directly publish the data to the server (*ESRI*, *2013e*). *Figure 6.7* shows ArcMap which can be used to edit and publish the data to the site.



sedd.mxd - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:45,167
Table Of Contents
Layers
EM_1910_1915_House_Hog_Kavlinge
EM_1910_1915_PropertyUnit_Hog_Kavlinge
TM_1860_1865_House_Hog_Kavlinge
RasterT_TM_18601
MTSM_1812_1820_House_Hog_Kavlinge
LSM_1757_1863_House_Hog_Kavlinge
LSM_1757_1863_PropertyUnit_Hog_Kavlinge
PropertyUnitHogLSM
crown
freehold
freehold/crown
LSM_1757_1863_House_Hog_Kavl
MTSM_1812_1820_House_Hog_Kav2
TM_1860_1865_House_Hog_Kavli
EM_1910_1915_House_Hog_Kavli4
Catalog
Location: arcgis on localhost_6080 (admin)
Home - Documents\ArcGIS
Folder Connections
Toolboxes
Database Servers
Database Connections
GIS Servers
Add ArcGIS Server
Add ArcIMS Server
Add WCS Server
Add WMS Server
Add WMTS Server
arcgis on haptimap.nateko.lu.se (admin)
arcgis on localhost_6080 (admin)
System
Utilities
SampleWorldCities
Drafts
My Hosted Services
Tracking Connections
Search
382544.758 6185939.571 Meters

Name	Type
System	ArcGIS Server Fo
Utilities	ArcGIS Server Fo
SampleWorldCities	Map Service
Drafts	Draft Services Fo

Figure 6.7. GIS resources are edited and published from ArcMap.

Application developers develop the web application, which helps the users easily access data which is stored on the ArcGIS Server site. The client side can be a desktop, web or mobile device. So, a number of APIs are available for the developers to choose from for supporting different clients, such as ArcGIS Web Mapping APIs (*ESRI*, *2013e*).

6.2 ArcGIS Web Mapping APIs

API is the acronym for *application programming interface*. It is a set of data or instructions for improving the interactions between programs or applications (*Kail, 2013*). A Web API is the connector between the client and the web service. Using a Web API, a web service can be directly manipulated by the client (*Masse, 2011*). *Figure 6.8* shows the workflow of a Web API working in a web application. When the client requests information, it directly talks to the Web API. Then, the Web API retrieves the results from the web service and returns the final results to the client. Web APIs provide an easy way to use a web service. For example, any kind of web application can access the Google Maps service by using the Google Maps API. The developers do not need to know how the service works, but they can access the service freely.

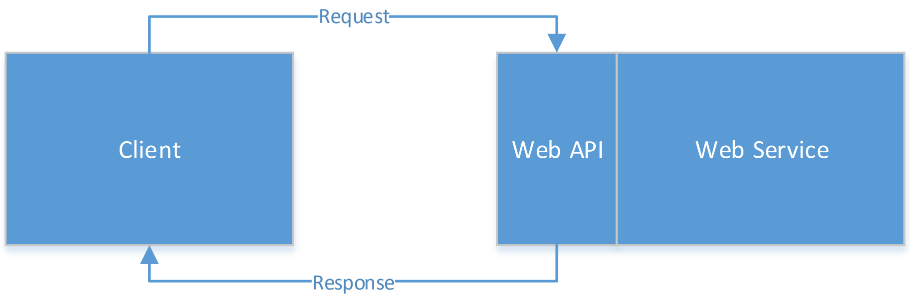


Figure 6.8. How a Web API works in a web application (*Masse, 2011*).

ESRI offers several Web APIs for developers to construct their web mapping applications, such as ArcGIS API for JavaScript, ArcGIS API for Silverlight and ArcGIS API for Flex. ArcGIS Web Mapping APIs are built for developers to enhance the capabilities of their web mapping applications to access map services. The three APIs are for different web mapping applications to use. ArcGIS API for JavaScript is for applications which are built with HTML and JavaScript. ArcGIS API for Silverlight is for applications which are built with Silverlight. ArcGIS API for Flex works with the Flex framework (*ESRI, 2013i*).

ArcGIS Web Mapping APIs provides many tools and widgets. The users can use the API's tools to access some powerful functions. These include query functions, which are used by a user to query data from an ArcGIS service; and geoprocessing, which is used to manipulate with spatial data. It also provides some widgets, such as *basemap*, which contains many kinds of background maps; map components, including the zoom bar, legend and scale bar, etc.; and edit widget, which can be used to edit and update the data directly from client application (*ESRI, 2013a*).

6.3 ArcGIS API for JavaScript

The ArcGIS API for JavaScript is a development tool used to embed GIS resources, such as maps and spatial data, in a web application. It was first introduced in ArcGIS Server 9.3 (*ESRI, 2013c*). With ArcGIS API for JavaScript, a web application can directly interact with the user's action. For instance, when the user clicks on a house, it will automatically select the house and highlight the object. The client side does not need to wait for the response of the GIS server. It works immediately on the client side (*ESRI, 2013c*).

The ArcGIS API for JavaScript contains four kinds of resources: maps, graphics, tasks and accessing to the dojo toolkits or other libraries (*ESRI, 2013c*). The maps resources include two types of maps, dynamic maps and tiled maps, and they can be displayed in any projections. The graphics resources means a JavaScript API that supports users, so they can draw graphics by themselves. Users can draw any kinds of graphics directly on the web application, such as points, lines and polygons. For task resources, the JavaScript API supports common GIS tasks, such as querying and finding a location, etc. For enhancing the functionality, ArcGIS API for JavaScript also supports some powerful JavaScript libraries. These libraries can be used directly in the web applications (*ESRI, 2013c*).

The ArcGIS API for JavaScript is supported by the REST API; it is able to retrieve information from the ArcGIS Server (*ESRI, 2013f*). REST stands for Representational State Transfer. It is a commonly used web architectural style (*Masse, 2011*). The ArcGIS REST API provides Uniform Resource Locators (URLs) to make the service accessible by a web application. For example, when a service is published, it can be accessed from a unique URL (e.g. http://...xxx.../argis/rest). Thus, if a web application needs to access the service, this URL will simply direct the service to the web application (*ESRI, 2013d*). *Figure 6.9* shows the workflow of a web mapping application developed by ArcGIS API for JavaScript.

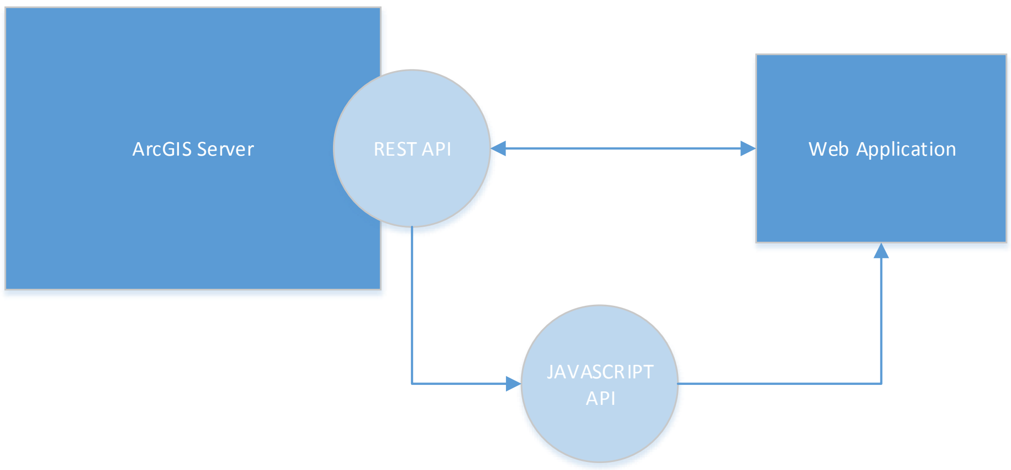


Figure 6.9. ArcGIS API for JavaScript workflow (*ESRI, 2012*).

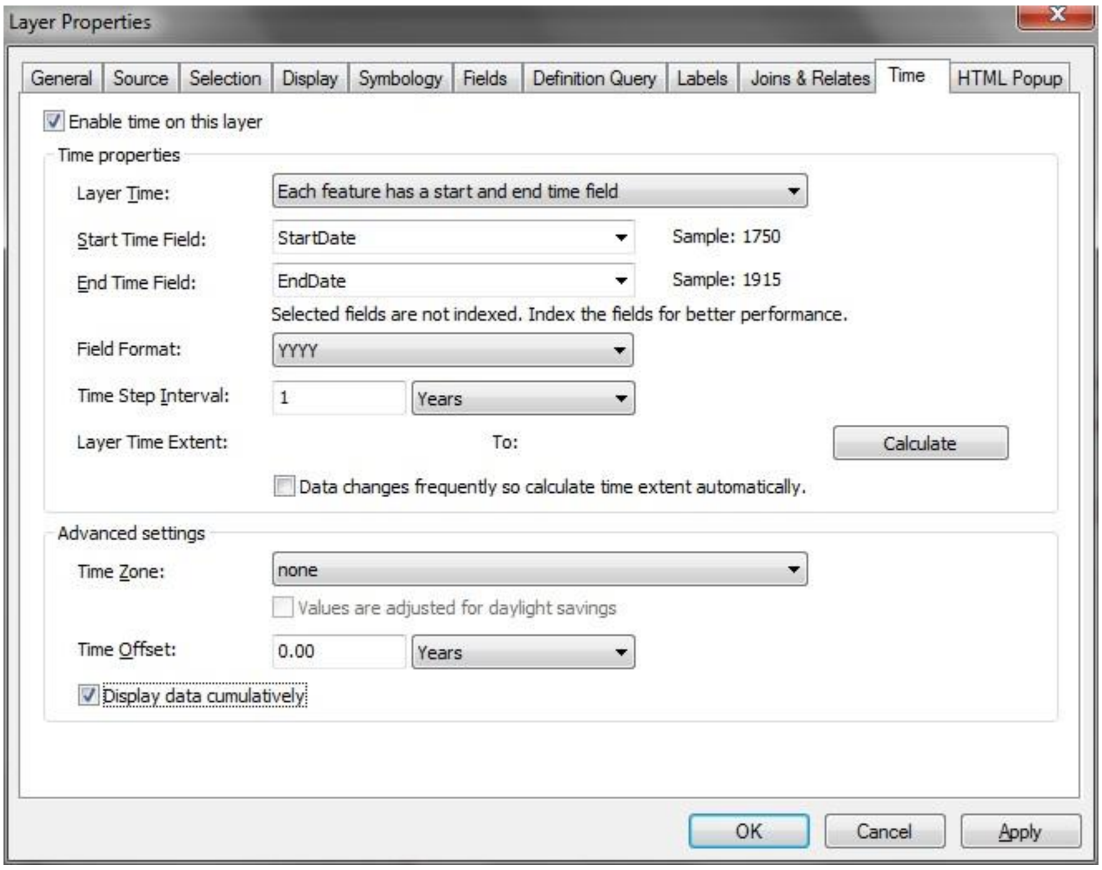
The ArcGIS API for JavaScript is built based on the Dojo Toolkit (*ESRI, 2013f*). The Dojo Toolkit is directly embedded in the ArcGIS API for JavaScript. When using the ArcGIS API for JavaScript, users do not need to download or host any other external Dojo libraries. Different versions of the ArcGIS JavaScript API may contain different Dojo libraries. For example, ArcGIS API for JavaScript 1.0 uses Dojo 1.1.0, while ArcGIS API for JavaScript 1.6 uses Dojo 1.4.1. In general, the user should use the Dojo toolkit which is integrated in the API. But the API also supports the user using an external Dojo libraries, in order to keep the Dojo library up to date (*ESRI, 2009*).

With the robust Dojo Toolkit, a developer could simply develop some reusable tools or widgets which could be used by anyone, anywhere (*ESRI, 2008*). For example, the *basemap* in the ArcGIS API for JavaScript is a single widget which is used to change the background map for a web mapping application. When the widget is first developed by a developer, the widget is set as a reusable source and placed in the *dijit* folder in Dojo toolkit. Then next time, when a user develops a new web application, they do not need to write the code again, they just require the widget directly from Dojo Toolkit. This reduces the development time for users and makes the JavaScript API more powerful.

6.4 ArcGIS visualization using a time-aware layer

ArcGIS 10 provides a time-enabled layer which can store spatio-temporal information (*ESRI, 2013g*). The time information can be set in the property of the layer. When the time period is set, the user can use a time slider to view the change of data during the defined time period. The time slider is contained in ArcMap and ArcGIS Online. And it also can be added to web applications developed with JavaScript, Silverlight and Flex.

The time period information can be added in ArcMap if the layer does not contain the information (*ESRI, 2013h*). *Figure 6.10* shows how to enable time information in a layer. The function can be found in the layer's properties. When opening the properties, there are some tabs, such as *General*, *Source* and *Selection*, etc. When clicking on the *Time* tab, check the box enabling time on this layer. Then, this layer is time enabled. There are many properties that can be set, such as defining the start time and the end time, defining the time format and time interval, and setting the time zone of the layer.



☑ Enable time on this layer

Time properties

Layer Time: Each feature has a start and end time field

Start Time Field: StartDate Sample: 1750

End Time Field: EndDate Sample: 1915

Selected fields are not indexed. Index the fields for better performance.

Field Format: YYYY

Time Step Interval: 1 Years

Layer Time Extent: To:

☐ Data changes frequently so calculate time extent automatically.

Advanced settings

Time Zone: none

☐ Values are adjusted for daylight savings

Time Offset: 0.00 Years

☑ Display data cumulatively

Figure 6.10. Enable time on a layer.

After the time property is set, the layer can be presented as animation in both ArcMap and the web applications which are developed with ArcGIS API. So, that would be an easy way to visualize the data as an animation.

7 Case Study

The SEDD Map previous version is a powerful interactive web mapping application. It is used to visualize and analyze data stored in the SEDD database. However, the previous version of SEDD Map has some limitations. First, it was developed using Silverlight. Therefore, it could not be accessed by portable devices and browsers which do not support the Silverlight Plugin. Second, the geographic data of the SEDD Map are taken from four independent historical maps. It contains too little time information to visualize as animation. This case study aims to solve these two problems.

7.1 Aim

The aims of this case study are:

a. Fulfill the requirements which are set up in section *3.1* to develop an improved version of SEDD Map.
b. Improve the compatibility of SEDD Map for different platforms and browsers.
c. Improve the visualization of geographic data which are taken as snapshots from four historical map series during the period from 1757 to 1915.

7.2 Study area

The case study area is located near the parishes of Kävlinge and Hög (*Figure 7.1*), which are two parishes belonging to Kävlinge municipality, located in Scania in southern Sweden. This case study only uses parts of the SEDD data derived from these two parishes.

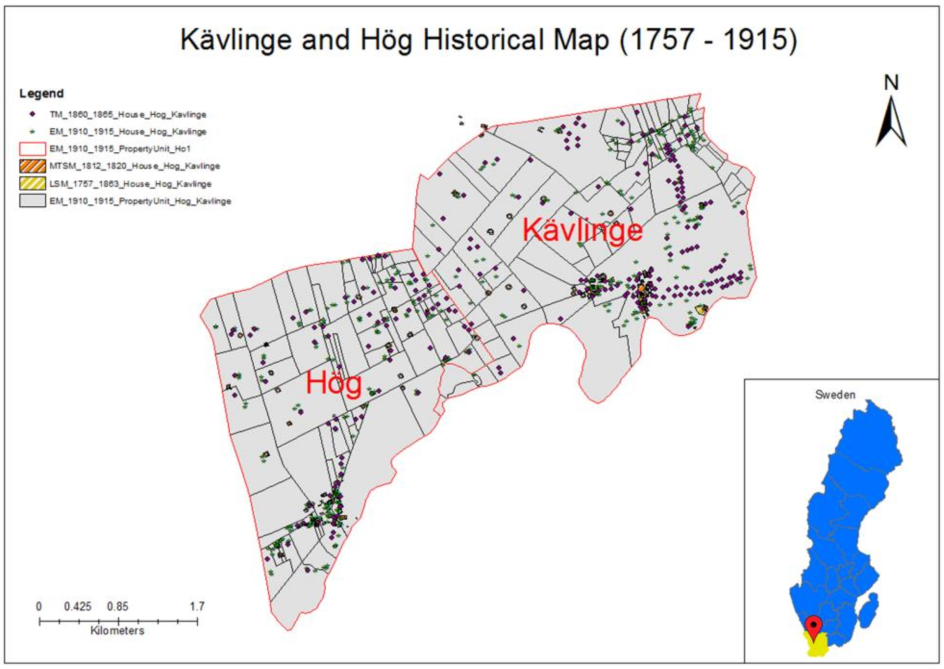

Kävlinge and Hög Historical Map (1757 - 1915)
Legend
TM_1860_1865_House_Hog_Kavlinge
EM_1910_1915_House_Hog_Kavlinge
EM_1910_1915_PropertyUnit_Ho1
MTSM_1812_1820_House_Hog_Kavlinge
LSM_1757_1863_House_Hog_Kavlinge
EM_1910_1915_PropertyUnit_Hog_Kavlinge
N
Kävlinge
Hög
0 0.425 0.85 1.7
Kilometers
Sweden

Figure 7.1. Study area of this case study.

7.3 Data

The geographic data was digitized by others from historical raster maps, and stored as vector data in ESRI shape files in an earlier SEDD Map project. The files contain information mapping houses and properties for the two parishes at four times. The spatial reference system of the geographic data is SWEREF 99 TM. Below we list the layers of the vector data. Descriptions of the historical maps are given in *Section 2.2*.

EM_1910_1915_House_Hog_Kavlinge layer and *EM_1910_1915_PropertyUnit_Hog_Kavlinge* layer cover the period 1910 to 1915. They were digitized into vector data from several raster economic maps. *EM_1910_1915_House_Hog_Kavlinge* layer contains the house information. The houses were digitized to points and contain the information such as from which map a feature was digitized and to which parish they belong. *EM_1910_1915_PropertyUnit_Hog_Kavlinge* contains the property unit information. The property units were digitized into polygons. The attribute of the layer contains the same information as the house layer. But two more columns are used to store the address of the properties.

TM_1860_1865_House_Hog_Kavlinge layer contains the houses information from the year 1860 to 1865. The layer was digitized from a series of topographic raster maps. The houses are presented as points in the layer. The information contains such as the origin of map's name, year and the name of the parish.

MTSM_1812_1820_House_Hog_Kavlinge layer was digitized from military topographic survey maps made during the years 1812 to 1820. The layer contains the houses information during that period. The houses were digitized into polygons. The attributes of the shape file shows from which map the house was digitized and also in which parish the house belongs.

LSM_1757_1863_House_Hog_Kavlinge layer and *LSM_1757_1863_PropertyUnit_Hog_Kavlinge* layer were digitized from land surveyor maps, and cover the period from 1757 to 1863. They were all digitized into polygons. The *LSM_1757_1863_House_Hog_Kavlinge* layer contains the house information while the *LSM_1757_1863_PropertyUnit_Hog_Kavlinge* layer contains the property information. The house layer contains the information such as from which map it was digitized, the address of the house, how many people lived there, etc. The property layer contains information almost the same as the house layer. Moreover, it also contains the area information for each property.

7.4 Method

Animation is a useful tool to visualize spatio-temporal data. Most of the web mapping applications which are used to visualize and analyze spatio-temporal data contain an animation function (see *Table 3.4*). In this case study, we provide a way to enable animation in the SEDD Map for presenting historical geographic data.

The geographic data in the SEDD Map are taken from four independent historical maps. *Figure 7.2* shows the four maps (see details in section *2.2*) in a time line. These maps are taken from the map series: Land Surveyors Maps (LSM), Military Topographic Survey Maps (MTSM), Topographic Maps (TM), and Economic Maps (EM).

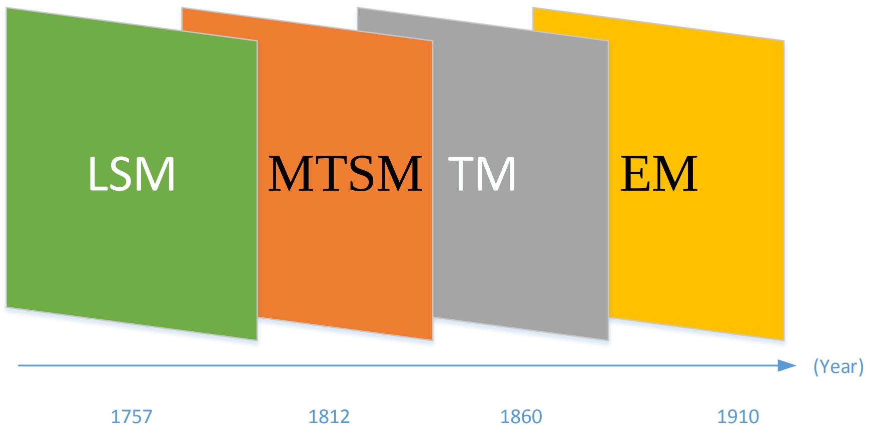


Figure 7.2. Geographic data in the SEDD Map are collected in four independent historical maps.

For presenting the geographic data as animation, we need large quantity of sequential data. However, the data in this case study are limited. So, we have to find a method to generate intermediate data. For example, if we want to visualize the data from 1757 to 1812, we have to approximate the situation in between them (*Figure 7.3*). So, to generate these maps requires a new strategy.

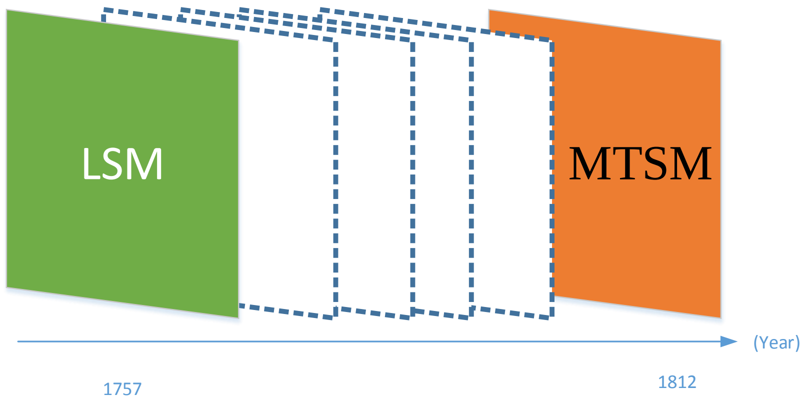


Figure 7.3. Generate maps between 1757 and 1812.

The in between maps could be generated by using the register data. For instance, the geographic data have been digitized from the four historical maps. Then, we can use the identity information of the geographic data (e.g. name or address) to retrieve the time information from the demographic data in the SEDD database. Finally, the time information can be registered on the geographic data. So, when we present the map for a certain year, we just present the geographic data which contains the time information of that year. This is an in-progress work for integrating the geographic data with demographic data (*Hedefalk et al., 2013*). The method in this case study can be used when the data are ready.

The historical maps and the demographic information are not collected at the same time. Therefore, the geographic data and the demographic data are not always completely matched. There are three cases that can occur when generating the maps. Here we take house layer as an example to show these three cases. In *Figure 7.4*, each house presents one case.

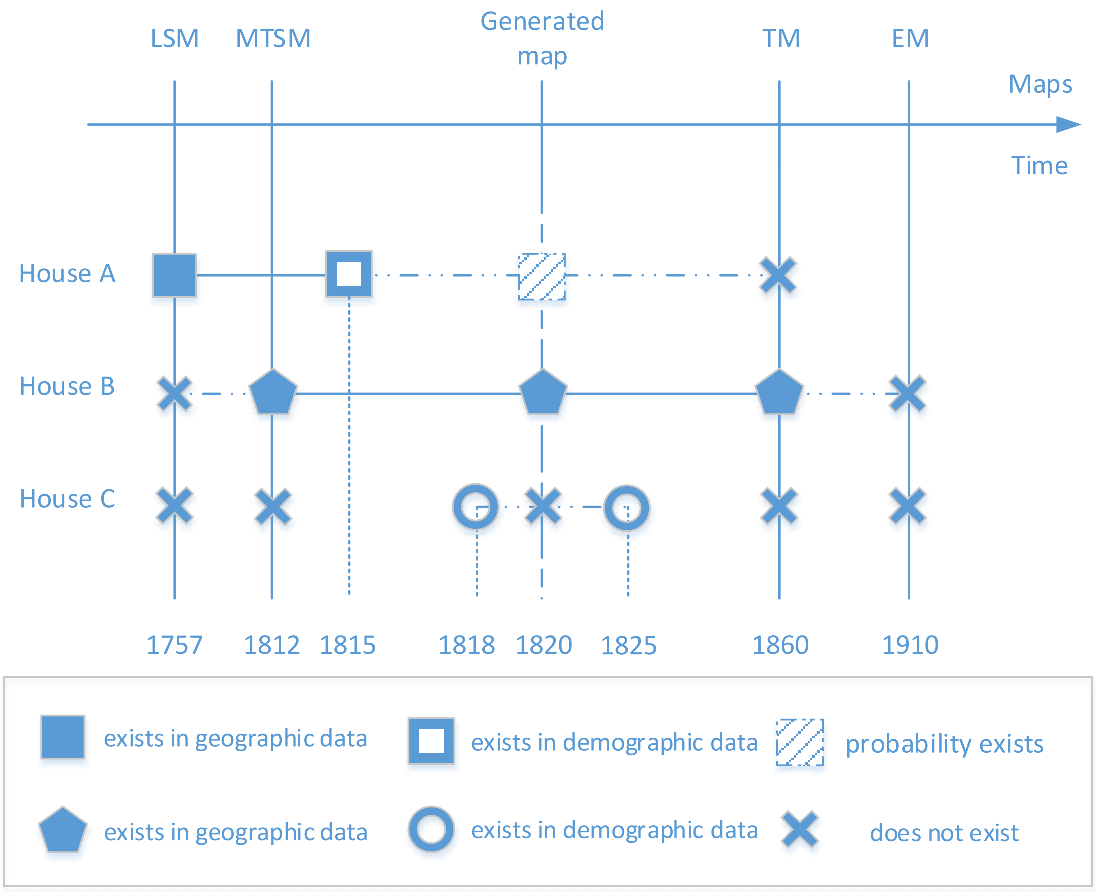


Figure 7.4. Three cases when generate maps.

The first case is that the house can be found in the map and also can be found in the demographic database, such as House A in *Figure 7.4*. House A is a house that exists on the LSM map in 1757.

In the demographic database, 1815 is the last year the House A is found. And it does not exist in the TM map in 1860. So, we can register the time information of House A as existing from 1757 to 1815. However, this is not exactly true. This is because we do not have the enough information to confirm that House A was destroyed in 1815. So we have to find a way to simulate further information for House A.

The second case is that the house can be found on the map, but it cannot be found in the demographic database, such as House B in *Figure 7.4.* House B is a house that exists in the MTSM map in 1812 and also exists in the TM map in 1860. But we cannot find the demographic information of House B in the demographic database. For this case, we have to present House B when we generate maps from 1812 to 1860. While for the period from 1757 to 1812 and the period from 1860 to 1910, we still have to find a solution to simulate the unknown situation of House B in these maps.

The third case is that we can find the demographic information of a house in the demographic database, but we cannot find the house on any of the historical maps, such as House C in *Figure 7.4.* House C is a house that only exists in the demographic database, we cannot find its geographic information on any of the maps. For this case, we cannot present it on the map. To solve this problem, we have to find more maps to get its geographic information, and change its situation to one of the first two cases.

This work gives a solution to solve the first two cases. For example, if we want to generate the map for 1820: For House A, we can give it a probability value to estimate if it can be presented on the map in 1820. For House B, we can directly present it on the map in 1820. But for the third case, we cannot find any geographic information about the House. Hence, even if it existed in that year, we cannot present it on the map. Finally, for the year 1820, we generate a map with House A (with a probability value) and House B shown, but without House C.

ArcGIS 10 provides the animation function to visualize time-aware data (section *6.4*). But it only visualizes data which contains exact time information, such as House A from 1757-1815 in *Figure 7.4*. From the visualization aspect, in this case study, the time information is a period. For instance, if we visualize the data from 1757 to 1812 using ArcGIS 10, it works as in *Figure 7.5*. It visualizes the LSM map from 1757 to 1811, then changes to the MTSM map to visualize another period. Though it can visualize the data with time changing, it still represents the data as four individual maps for the whole period. So, we cannot see any gradual changes in the data.

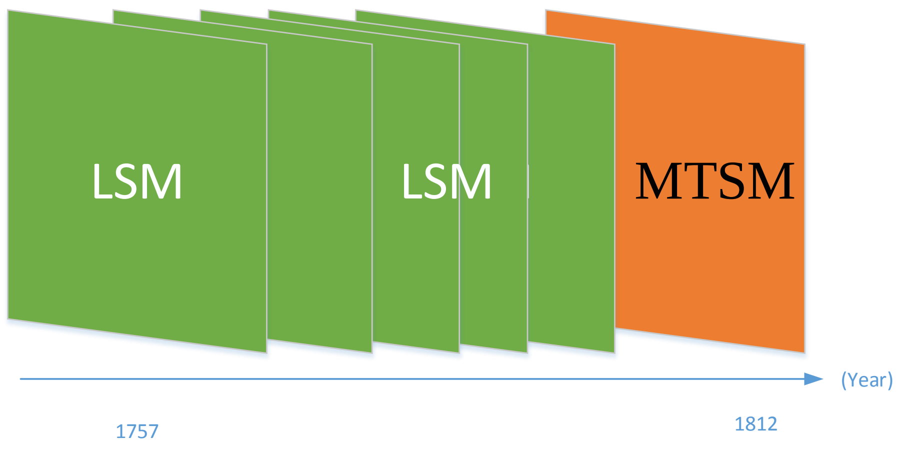


Figure 7.5. Visualize animation using ArcGIS time-aware layer.

Interpolation is used to estimate the value of an unknown point by other known points (*Harrie, 2006*). Linear interpolation is a simple form of interpolation (*Figure 7.6*). If we know at time t_0 the likely value is P_0, while at time t_1 the likely value is P_1, we can connect the two points as a straight line and get use Equation *7.1*. Then using Equation *7.1*, we can calculate the likely value of time *t* which is between time t_0 and t_1.

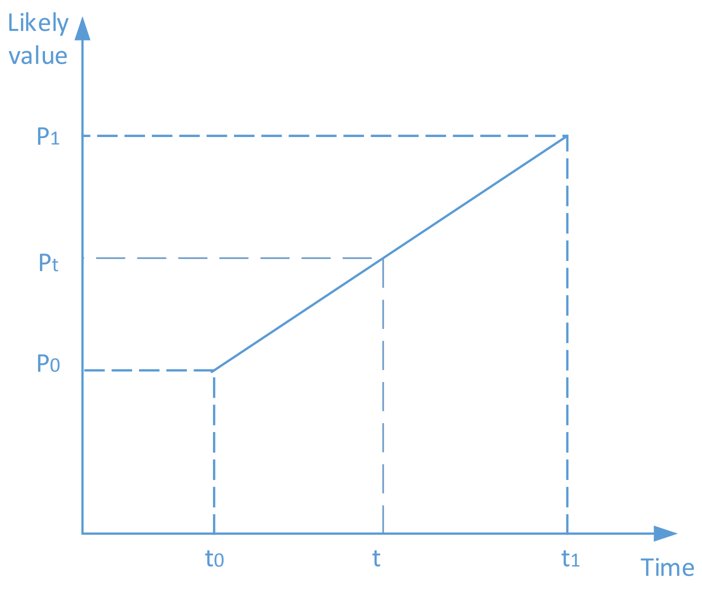


Figure 7.6. Using linear interpolation to estimate the likely value (*Phillips, 2003*).

$$P_t = P_0 + (P_1 - P_0)\frac{t-t_0}{t_1-t_0} \tag{7.1}$$

To improve the time animation, we can use linear interpolation to simulate 159 maps from 1757 to 1915. Each map represents a particular year. Each house is associated with a value to control the opacity to show the estimated probability of the house exists in the map for a certain year.

This approach is combined with the first two cases in *Figure 7.4.* The method to generate maps can be grouped into three situations.

The first situation is shown in *Figure 7.7*. In this situation, the house exist in both t_0 and t_1. So the house would exist in the map in the year *t*. And the P value for this situation is a constant value 1. Thus, we use *Equation 7.2* for presenting this situation.

$$P_t = 1 \tag{7.2}$$

The second situation works as below (*Figure 7.8*): There is a house in the year t_0, we set the value with 1 (P_0 = 1, in *Equation 7.1*). With time passing, it disappeared in the year t_1, we set the value with 0 (P_1 = 0, in *Equation 7.1*). So, to simulate the probability of the house existing in the year *t*, we can use the *Equation 7.3* to calculate the value.

$$P_t = \frac{t_1 - t}{t_1 - t_0} \tag{7.3}$$

The third situation is shown in *Figure 7.9*. For this situation, there is no house exist in the year t_0. So we set the probability value to 0 (P_0 = 0, in *Equation 7.1*). And for the year t_1 there is a house. Then we set the value with 1 (P_1 = 1, in *Equation 7.1*). Therefore, we can calculate the *P* value for time *t*.

$$P_t = \frac{t - t_0}{t_1 - t_0} \tag{7.4}$$

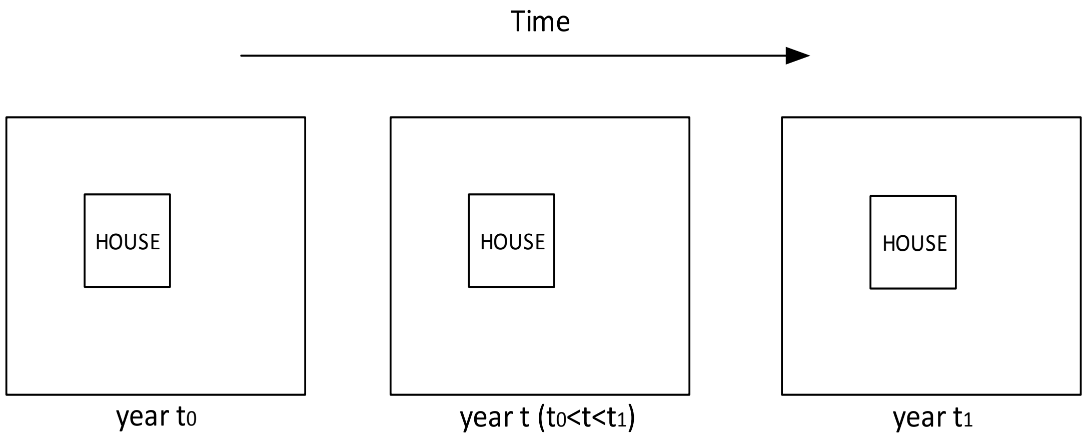


Figure 7.7. Method to generate maps from year t_0 to t_1 - situation 1.

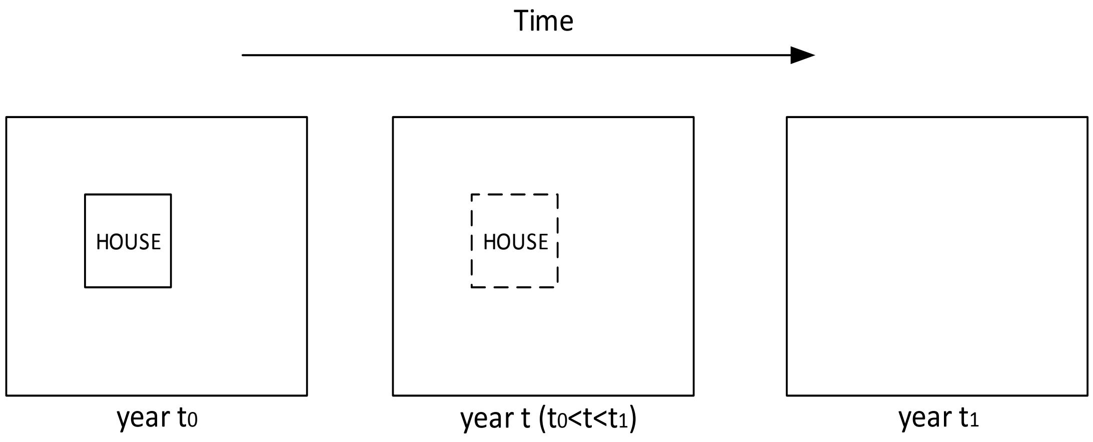


Figure 7.8. Method to generate maps from year t_0 to t_1 - situation 2.

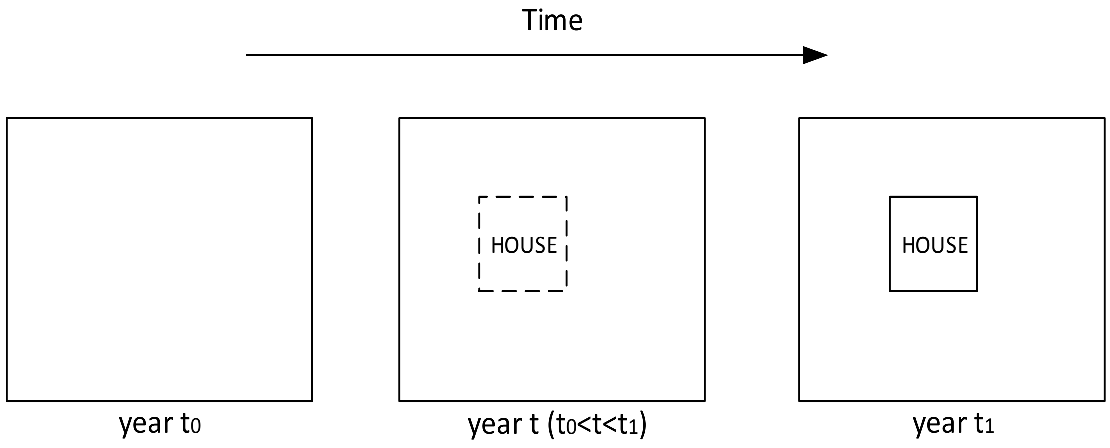


Figure 7.9. Method to generate maps from year *t_0* to t_1 - situation 3.

7.5 Implementation

Figure 7.10 shows the architecture of the new version of SEDD Map. The web mapping application is constructed using the client-server model (see *Chapter 4*). The client side is developed using HTML5, CSS3, JavaScript (see *Chapter 5*) and ArcGIS API for JavaScript (see section *6.3*). A web server is used to receive and send requests between the client side and the server side. The ArcGIS Server site (section *6.1*) uses a one-machine architecture. The data are stored in the data server of the ArcGIS Server site and managed using ArcGIS for Desktop by the data manger and Publisher.

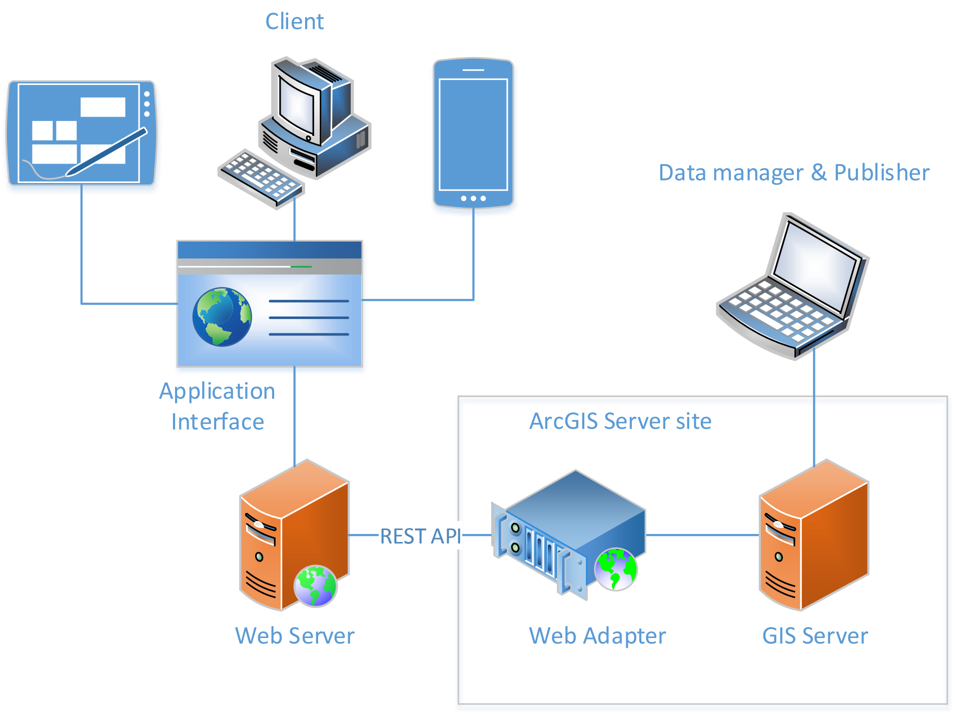


Figure 7.10. The architecture of the SEDD Map (new version).

7.5.1 Development of the web mapping application

The server side in this case study uses ArcGIS for Server 10.1 SP1. All the settings were set to default. After the server side was constructed, ArcGIS for Desktop 10.1 SP1 was used to manage the data and upload the data to the GIS server. *Figure 7.11* shows the data managed in ArcGIS for Desktop. On the left side is the table of contents window, which contains information about the layers, such as the layer’s name and display status. In the middle is the main working area, where the layers are edited and updated. On the right side, the catalog window is located. In the catalog window, we can connect to the ArcGIS Server site in the GIS servers menu. After the data are updated, we can send the data directly to the GIS server.

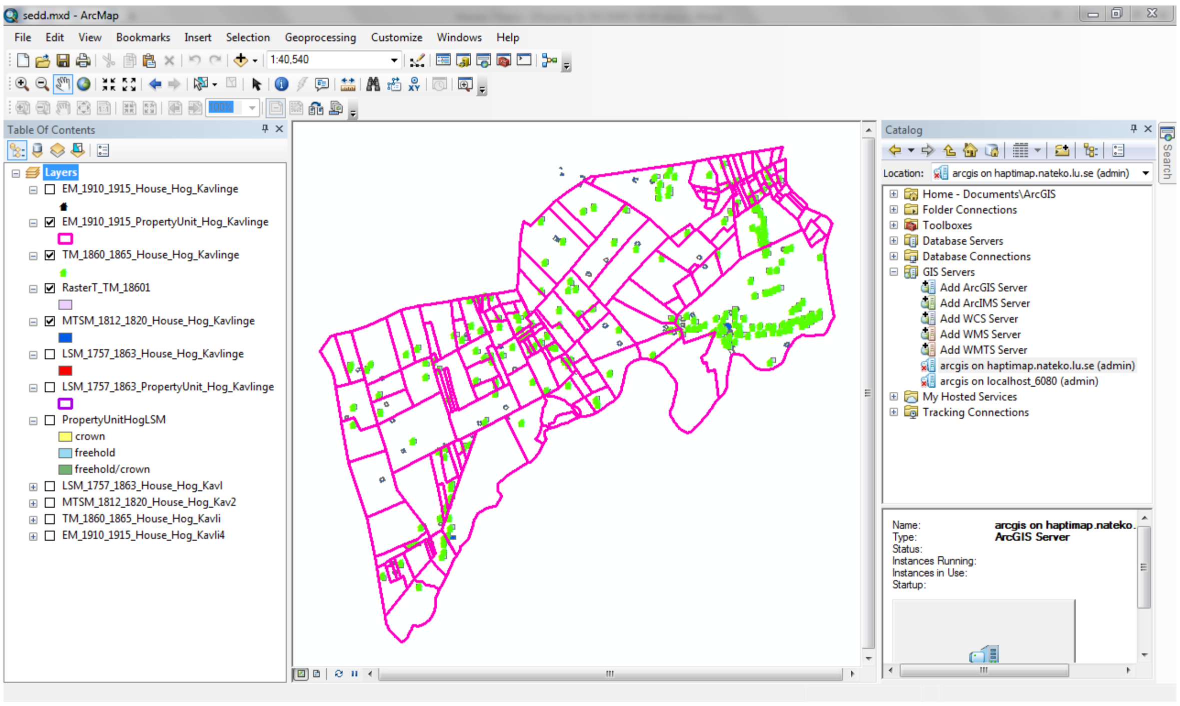
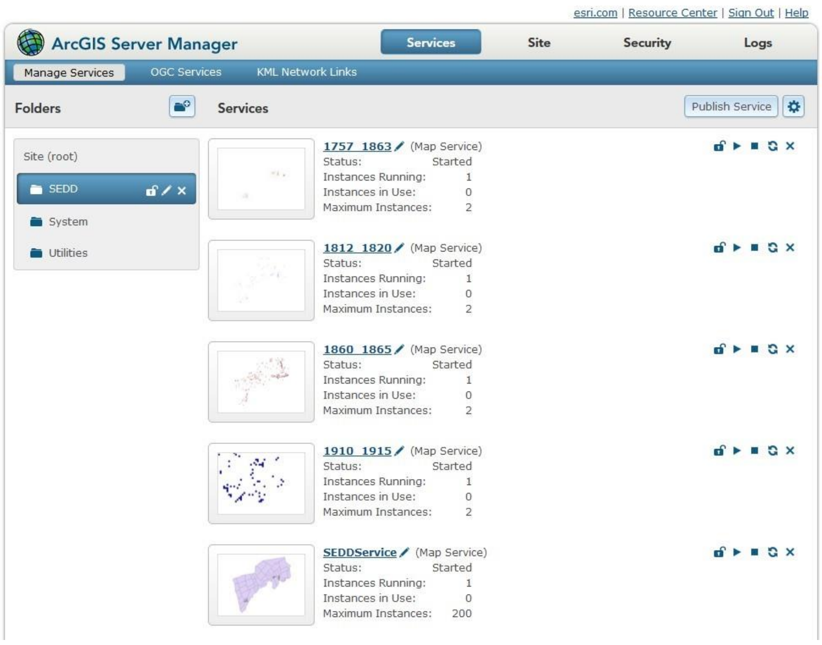

sedd.mxd - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:40,540
Table Of Contents
Layers
EM_1910_1915_House_Hog_Kavlinge
EM_1910_1915_PropertyUnit_Hog_Kavlinge
TM_1860_1865_House_Hog_Kavlinge
RasterT_TM_18601
MTSM_1812_1820_House_Hog_Kavlinge
LSM_1757_1863_House_Hog_Kavlinge
LSM_1757_1863_PropertyUnit_Hog_Kavlinge
PropertyUnitHogLSM
crown
freehold
freehold/crown
LSM_1757_1863_House_Hog_Kavl
MTSM_1812_1820_House_Hog_Kav2
TM_1860_1865_House_Hog_Kavli
EM_1910_1915_House_Hog_Kavli4
Catalog
Location: arcgis on haptimap.nateko.lu.se (admin)
Home - Documents\ArcGIS
Folder Connections
Toolboxes
Database Servers
Database Connections
GIS Servers
Add ArcGIS Server
Add ArcIMS Server
Add WCS Server
Add WMS Server
Add WMTS Server
arcgis on haptimap.nateko.lu.se (admin)
arcgis on localhost_6080 (admin)
My Hosted Services
Tracking Connections
Name: arcgis on haptimap.nateko.
Type: ArcGIS Server
Status:
Instances Running:
Instances in Use:
Startup:
Search

Figure 7.11. Editing and uploading the data to ArcGIS server site.

After the data were sent to the ArcGIS server site, we used the ArcGIS Server Manager to manage the services. *Figure 7.12* illustrates the ArcGIS Server Manger managing the services which are running on the ArcGIS Server site. Using ArcGIS Server Manager, we can easily manage a service, for example restarting the SEDD Map service, or stopping the geometry service. We can also use the ArcGIS Manger to configure the properties of the site.



1757_1863 (Map Service)
Status: Started
Instances Running: 1
Instances in Use: 0
Maximum Instances: 2

1812_1820 (Map Service)
Status: Started
Instances Running: 1
Instances in Use: 0
Maximum Instances: 2

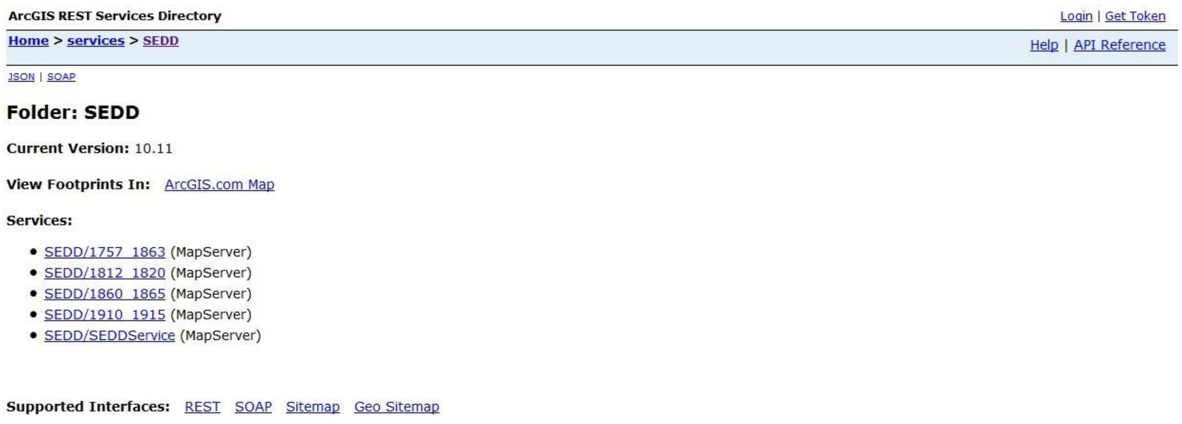
1860_1865 (Map Service)
Status: Started
Instances Running: 1
Instances in Use: 0
Maximum Instances: 2

1910_1915 (Map Service)
Status: Started
Instances Running: 1
Instances in Use: 0
Maximum Instances: 2

SEDDService (Map Service)
Status: Started
Instances Running: 1
Instances in Use: 0
Maximum Instances: 200

Figure 7.12. ArcGIS Server Manager.

After the service was running, we used the ArcGIS REST Services Directory (*Figure 7.13*) to explore the data stored on the GIS server. From the ArcGIS REST Services Directory we can find detailed information about the service, such as which layers are contained in the service and the spatial reference of the service, etc. The ArcGIS REST Services Directory can be opened from an URL which is also the place where the REST API uses for retrieving data from the web mapping client side.



ArcGIS REST Services Directory

Login | Get Token

Home > services > SEDD

Help | API Reference

JSON | SOAP

Folder: SEDD

Current Version: 10.11

View Footprints In: ArcGIS.com Map

Services:

- SEDD/1757_1863 (MapServer)
- SEDD/1812_1820 (MapServer)
- SEDD/1860_1865 (MapServer)
- SEDD/1910_1915 (MapServer)
- SEDD/SEDDService (MapServer)

Supported Interfaces: REST SOAP Sitemap Geo Sitemap

Figure 7.13. ArcGIS REST Services Directory.

When the data are ready, the next step is to use CSS3 to define the layout style of the new version of the SEDD Map application (*Appendix A*). All the buttons were set into the four corners to fit small screens devices, such as smartphones. The menu bar was designed using CSS3 transparency effect and CSS3 animation which are used to enhance the visualization of the mapping application.

The next step is to add the functions to the styled HTML file (*Appendix B*). The functions were developed using JavaScript and the ArcGIS API for JavaScript. There are several functions has been added in this map application to fulfil the requirements, such as changing the background maps, controlling the display layers, search tool, animation, etc. A `Canvas` element is used to draw a pie graph e.g. to show the percentage of the property types for Hög in 1804.

7.5.2 Animation

The data for the animation is created using linear interpolation (as described in *Section 7.4*) to generate the annual maps for the period 1757 to 1915. This work focuses on the house layers.

The first step in generating the annual maps is to symbolize all the point houses with a square. There are four ESRI shape files containing the house information. Two layers represent the houses as points, while the other two represent the houses as polygons (*Table 7.1*). For enhancing the visualization effect, all the houses stored as points were symbolized with squares.

Table 7.1. House shape in different layers.

ArcGIS Layer	House shape
EM_1910_1915_House_Hog_Kavlinge	Point
TM_1860_1865_House_Hog_Kavlinge	Point
MTSM_1812_1820_House_Hog_Kavlinge	Polygon
LSM_1757_1863_House_Hog_Kavlinge	Polygon

After the house shapes were symbolized, we have four independent maps as shown in *Figure 7.14* with the time period stated on *Figure 7.15*. After the map covering period was defined, we started the simulation portion.

At this stage, the same house on each map are not linked. So, in this case study, we used *Equation 7.3* to simulate the maps. For the first three periods, take the period from 1757 to 1811 as an example, using Map A to simulate the annual maps between the years from 1757 to 1812. We defined the probability of the houses existing in 1757 to 1, and the probability of the houses existing in 1811 to 0.

For the last period, from 1910 to 1915, we do not need to make any changes for each year. This is because we do not know the future condition. For this period, it is treated as if the houses will exist at all times in the future. So, the value of each year during the last six years is set to 1.

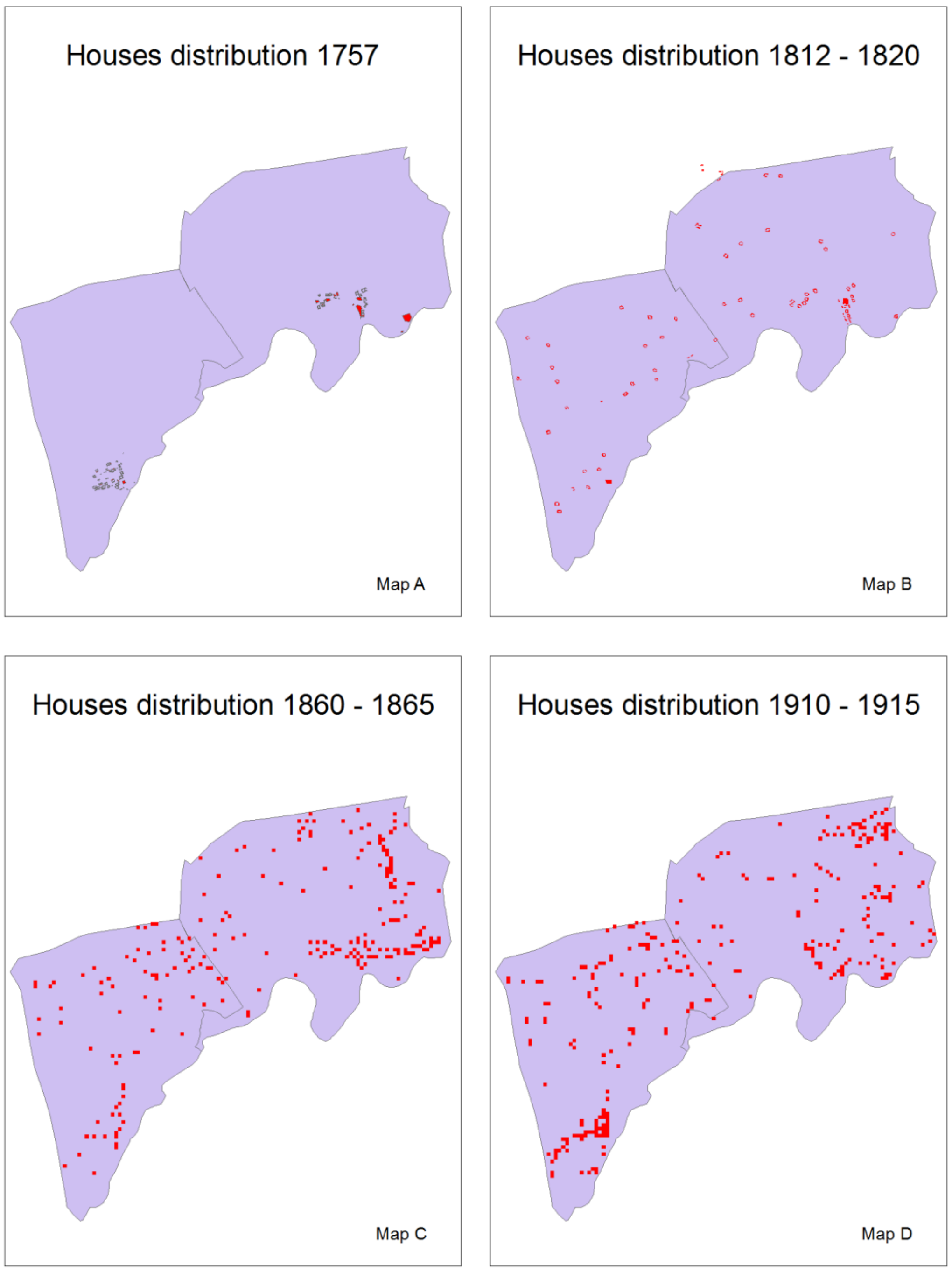

Houses distribution 1757
Map A
Houses distribution 1812 - 1820
Map B
Houses distribution 1860 - 1865
Map C
Houses distribution 1910 - 1915
Map D

Figure 7.14. Houses distribution during the four period from 1757 to 1915.

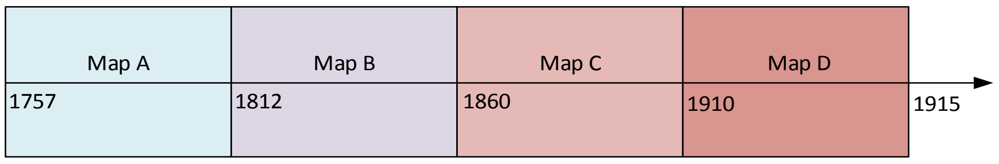
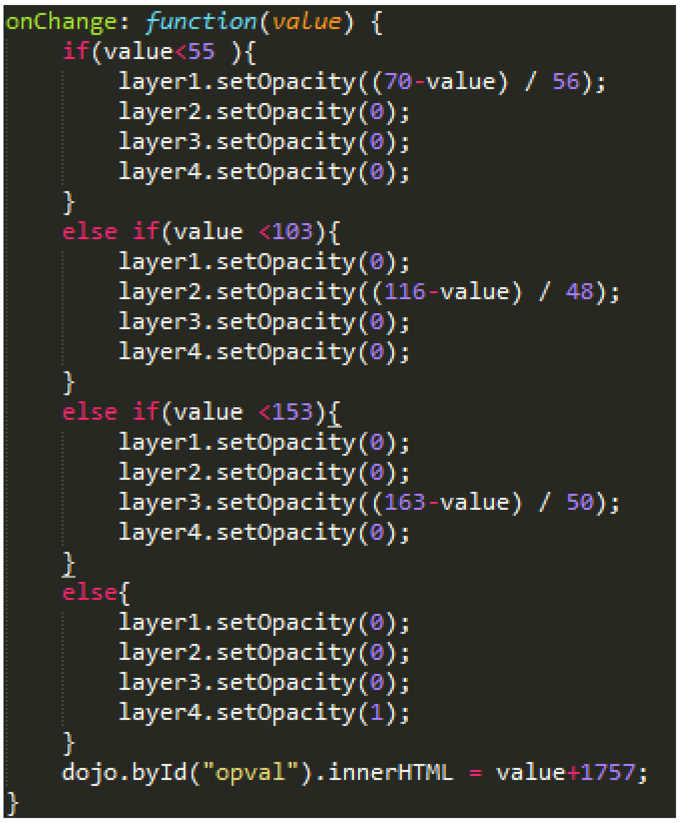


Figure 7.15. Final map covers period.

In the end, the final function to simulate the map for the year *t* is shown in *Equation 7.5*:

$$f(t) = \begin{cases} \frac{1811-t}{1811-1757} * Map\ A & (1757 \leq t \leq 1811) \\ \frac{1859-t}{1859-1812} * Map\ B & (1812 \leq t \leq 1859) \\ \frac{1909-t}{1909-1860} * Map\ C & (1860 \leq t \leq 1909) \\ 1 * Map\ D & (1910 \leq t \leq 1915) \end{cases} \quad (7.5)$$

Where *t* presents in which year to generate the map. So, it is an integral number. Map A (or B, C and D) presents which map that was used to generate the map. And the code to perform the equation is shown in *Figure 7.16*. In practice, the portability of the houses in the end year for each interval were set to 0.4 (instead of 0) to get a good visualization result.



```
onChange: function(value) {
    if(value<55 ){
        layer1.setOpacity((70-value) / 56);
        layer2.setOpacity(0);
        layer3.setOpacity(0);
        layer4.setOpacity(0);
    }
    else if(value <103){
        layer1.setOpacity(0);
        layer2.setOpacity((116-value) / 48);
        layer3.setOpacity(0);
        layer4.setOpacity(0);
    }
    else if(value <153){
        layer1.setOpacity(0);
        layer2.setOpacity(0);
        layer3.setOpacity((163-value) / 50);
        layer4.setOpacity(0);
    }
    else{
        layer1.setOpacity(0);
        layer2.setOpacity(0);
        layer3.setOpacity(0);
        layer4.setOpacity(1);
    }
    dojo.byId("opval").innerHTML = value+1757;
}
```

Figure 7.16. The code to perform *Equation 7.5*.

7.6 Results

Figure 7.17 and *Figure 7.18* show the web mapping application SEDD Map (new version) running on a computer and a portable device respectively. The functions of the mapping application are arranged in the four corners, so the position of the function will not change when changing from large screen to small screen.

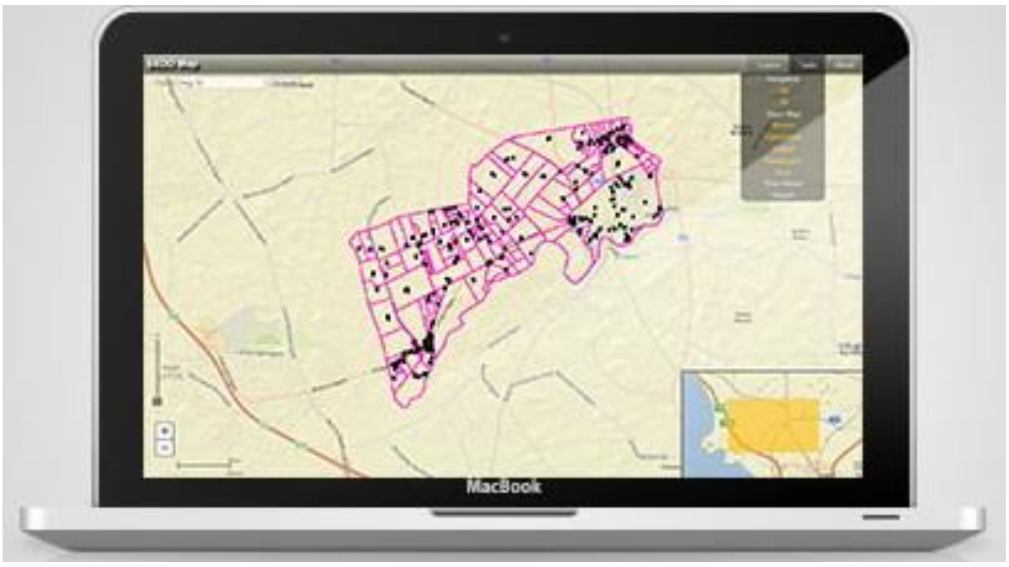


Figure 7.17. The SEDD Map (new version) running on a computer.

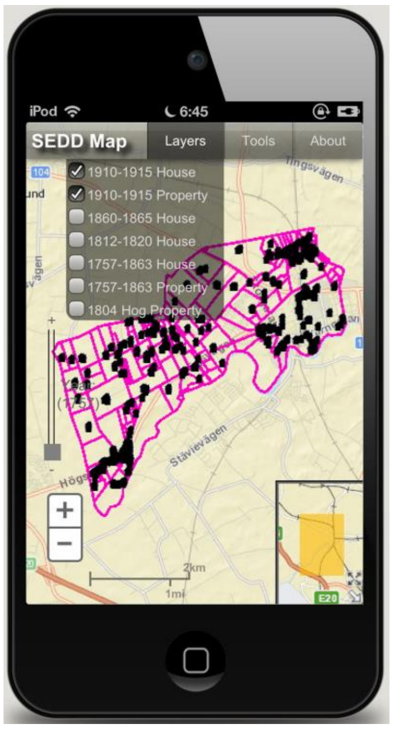


Figure 7.18. The SEDD Map (new version) running on a portable device.

7.6.1 Evaluate against the requirements

Coding languages and plugins

The SEDD Map (new version) was developed using HTML5, CSS3, JavaScript and ArcGIS API for JavaScript. Modern browsers have native support for all of these technologies; there is no need to install any plugins to run this application.

Cross-platform support

The SEDD Map (new version) has been tested on both computer and portable devices (e.g. iOS devices and Android devices) with good results.

Multiple Web browsers support

Table 7.2 shows the browsers that SEDD Map (new version) supported. From the table we can see that all five main browsers can run this application.

Table 7.2. The browsers which SEDD Map (new version) supports.

	IE	**FireFox**	**Chrome**	**Safari**	**Opera**
Windows	YES	YES	YES	YES	YES
Mac OS	N/A	YES	YES	YES	YES
Linux	N/A	YES	YES	N/A	N/A
Chrome OS	N/A	N/A	YES	N/A	N/A
iOS	N/A	YES	YES	YES	YES
Android	N/A	YES	YES	N/A	YES

Interactive visualization

SEDD Map provides interactive visualization; when the user clicks on a property, SEDD Map retrieves its information. A Navigation tool is used to get the location of the device. This function is achieved by the Geolocation API. It is a real time location service to enhance the functionality of portable devices. There are also some tools for improving the interaction capabilities. These include the Zoom out/in button in the lower left quarter and the overview map in the lower right quarter (*Figure 7.17*).

Multiple maps based

In the upper right corner, there is a *Layers* menu (*Figure 7.17*). This menu contains the layers that the application uses. The user can use this menu to change the display layers. There is also a *Base Map* function in the *Tools* menu which is used to change the background map of the mapping application.

Animation

Animation is used to visualize the houses' distribution over time. The animation is presented with a slider. The slider is used to change the time to different years. As the time passes, we can easily identify the changes on the geographic data. *Figure 7.19* shows the changes of a house when the animation is playing from 1812 to 1859. In the figure, on the left part, it presents the probability of the house would exist in the year 1812, while on the right part, it illustrate the probability of the same house existing in the year 1859. From the picture, we can say that the house has more possibility of existing in 1812 than in 1859.

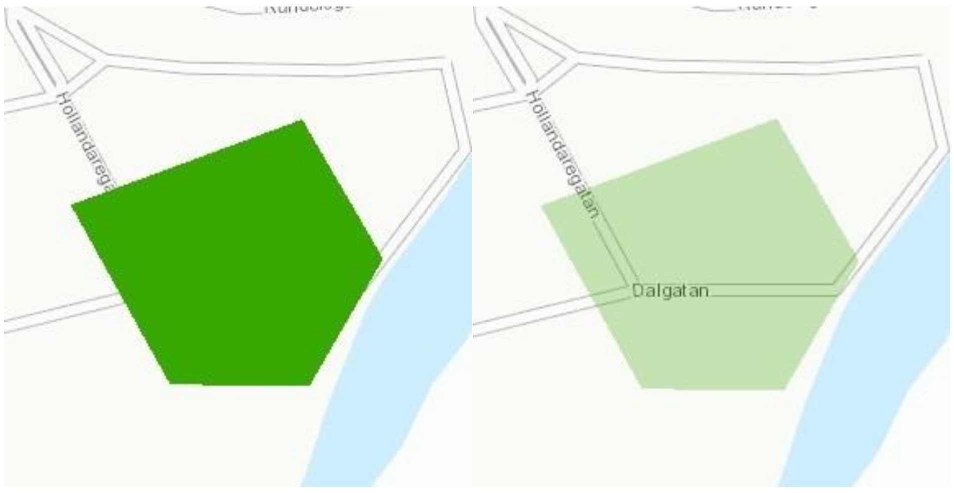


Figure 7.19. A house generated in the animation, shown in the year 1812 (left) and 1859 (right).

Statistical graphics

Figure 7.20 shows a pie chart which is drawn by the HTML5 `Canvas` element to present the percentage of the area for each property. It uses JavaScript to draw the static pie chart on the `Canvas` element.

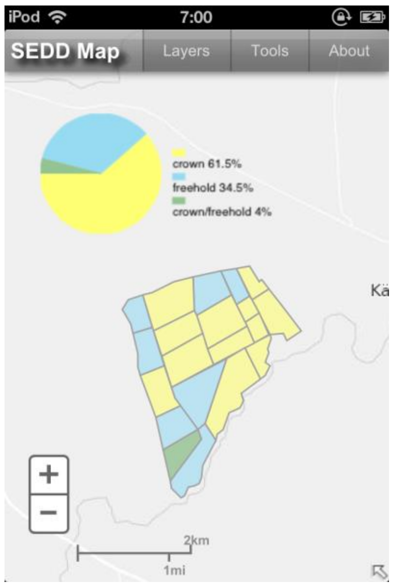
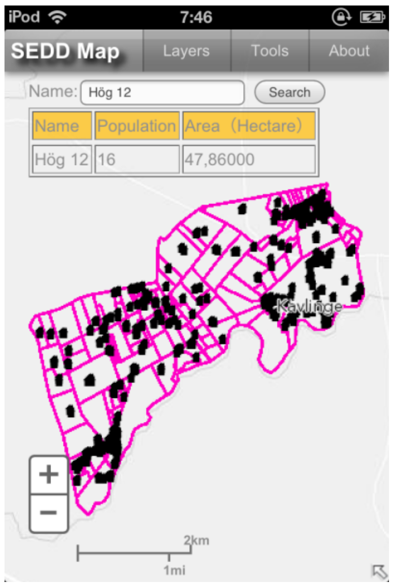


Figure 7.20. Using Canvas to draw statistical graphics.

Search tool

In the *Tools* menu there is a search tool (*Figure 7.21*). The search tool is used to retrieve information from the database. For this study, it used to query the population and area of a property unit.



Name: Hög 12

Name	Population	Area (Hectare)
Hög 12	16	47,86000

Figure 7.21. Seach Tool in the SEDD Map (new version).

Export functions

SEDD Map HTML5 version could export maps as pictures, and save pictures in pdf format or JPEG format.

7.6.2 Comparisons between the SEDD Map previous version and the new version

Table 7.3 and *Table 7.4* show a comparison between the SEDD Map previous version and the SEDD Map new version.

Table 7.3 shows the different technical solutions used in the two web mapping applications. From the table we can see that the two web mapping applications are developed with two different languages. The previous SEDD Map was developed using Silverlight, while the new version of SEDD Map was developed using HTML5 and JavaScript. Though they both can be accessed by multiple browsers, the new version of SEDD Map is more flexible. It does not require the installation of any plugins and can be run on both computer and portable devices.

Table 7.4 shows the technical implementation detail differences between the SEDD Map previous version and SEDD Map new version. From the tables we can identify that both of the web mapping applications can achieve the basic functions that a web mapping application needs for visualizing and analyzing historical data. However, the new version of SEDD Map has a more powerful function when dealing with animation. This enhances its ability to visualize historical data.

Table 7.3. Comparison between the two versions of SEDD Map – Technical solutions.

	Coding Languages	Plugin	Cross-platfrom Supported	Multiple Browsers Supported
SEDD Map (previous)	Silverlight	Silverlight	No	Yes
SEDD Map (new)	HTML5 and JavaScript	No	Yes	Yes

Table 7.4. Comparison between the two versions of SEDD Map – Technical implementation details.

	Interactive Visualization	Multiple Map Based	Animation	Statistical Graphics	Search Tools	Export Functions
SEDD Map (previous)	Yes	Yes	No	Yes	Yes	Map
SEDD Map (new)	Yes	Yes	Yes	Yes	Yes	Map

8 Discussions

8.1 Web mapping with HTML5

Web mapping applications developed using HTML5, CSS and JavaScript could replace most of web mapping applications built with plugins in the future. At least, in this case study, the new version of SEDD Map has the ability to replace the previous version of SEDD Map.

Compared with the web mapping applications built with plugins, HTML5 web mapping applications have many advantages. First, they have cross-platform support. This means that the developer only builds one application that can run on all devices, though different devices may have different user interfaces. This can save a lot of time and money for web mapping providers. Second, the file size of an HTML5 application is relatively small. This reduces the loading time for web mapping users. Third, to develop HTML5 web mapping applications, developers do not need to use any unique software. They can be coded in any text editor. To develop Flash version or Silverlight based web mapping applications, the commercial software are needed, such as Adobe Flash and Visual Studio.

There are also some disadvantages to HTML5 web mapping application. From a security aspect, the source code of the HTML web mapping application is easily accessed by users. All modern browsers have a debug function. By using this function, it is easy to see the origin code of web mapping applications from the client side. So, the source code of web mapping applications can be easily gained by others. Another problem is that the standard of HTML5 is only a candidate standard, and some functions are not supported in some browsers. So, this could be a problem for the developers. When developers develop web mapping applications, they have to test all functions in every browser.

CSS3 is used to design the layout and enhance the visualization for web mapping applications. In this work, CSS3 transition and CSS3 animation are used to develop the menu bar. It works well in most of browsers, but in some early version browsers some functions (e.g. CSS3 animation) are not supported. So, for developing a good web mapping application, the developer has to fully understand its features.

The HTML5 `Canvas` element gives a new way to draw graphics on the web sites. That is, it can be performed well without rendering using any plugins. It presents graphics as raster. So, it needs to be carefully treated when designing web mapping application for fitting all screen sizes (small screens and large screens). The `canvas` element is placed on the very top level of the browser. So, other elements cannot be displayed in the same place.

8.2 Animation

This work only used a very basic approach to simulating the maps for each year to make animations. For each year, the houses in the map are given a value to show the probability that they existed. This value is used to set transparency of the houses to enhance the geographic data visualization when the time changes.

In other words, it also means giving the house layer (geographic data) a time value which could be used to visualize the geographic data, in combination with the demographic data, which is stored in the SEDD database. Though the time information is not directly stored in the geographic database, we can use a slider as an intermediate to link the geographic data with the demographic data. For example, when clicking a house on the map, the ID or name of the house will be used to retrieve the data in the demographic database. So, all data about the house will be listed. Then we can use the slider to get a year value. When the slider changes to a year, the slider will return a value. The value can be used to retrieve the data from the returned list which is created by the house. So all the information for the house for that year would be selected. We can use the selected demographic data to draw a statistical graphic about that year, such as a bar chart. At the same time the map application simulates a map of that year. So we can use one slider to control both the geographic data changes and the demographic data changes. This enhances the time series presentation.

The method to simulate houses can use for two kinds of houses. *Figure 8.1* shows the two kinds of houses. We use House A and House B to present these two types. Each house presents one type. The first type is that the house can be found in the historical maps, which used to digitize the houses. And it can also be found in the demographic database (SEDD database). In the figure, we use House A to present this type. The second type is that the house can only be found in the historical maps. And we cannot find its demographic information in the database. In the figure, we use House B to present this type.

For House A, to calculate its probability that exists in each year during the whole period, we can separate the whole period to two parts. Then, use linear interpolation to calculate its probability value for each year. For example, House A exists in the LSM (Land Surveyors Maps) map in the year 1757. The last year it occurs in the demographic database is 1815. And we cannot find its further information in the TM (Topographic Maps) map in 1860. So, we can separate the whole period from 1757 to 1860 to two parts. The first part is from the year 1757 to 1815. We set the probability value to 1 to the house for each year during this period. The second part is from 1815 to 1860. For this period, we set the probability value of the house to 1 in the year 1815, and we set the probability value of the house to 0 in the year 1860. So, the probability of the house exists in each year during the year 1815 to 1860 can be calculate using linear interpolation.

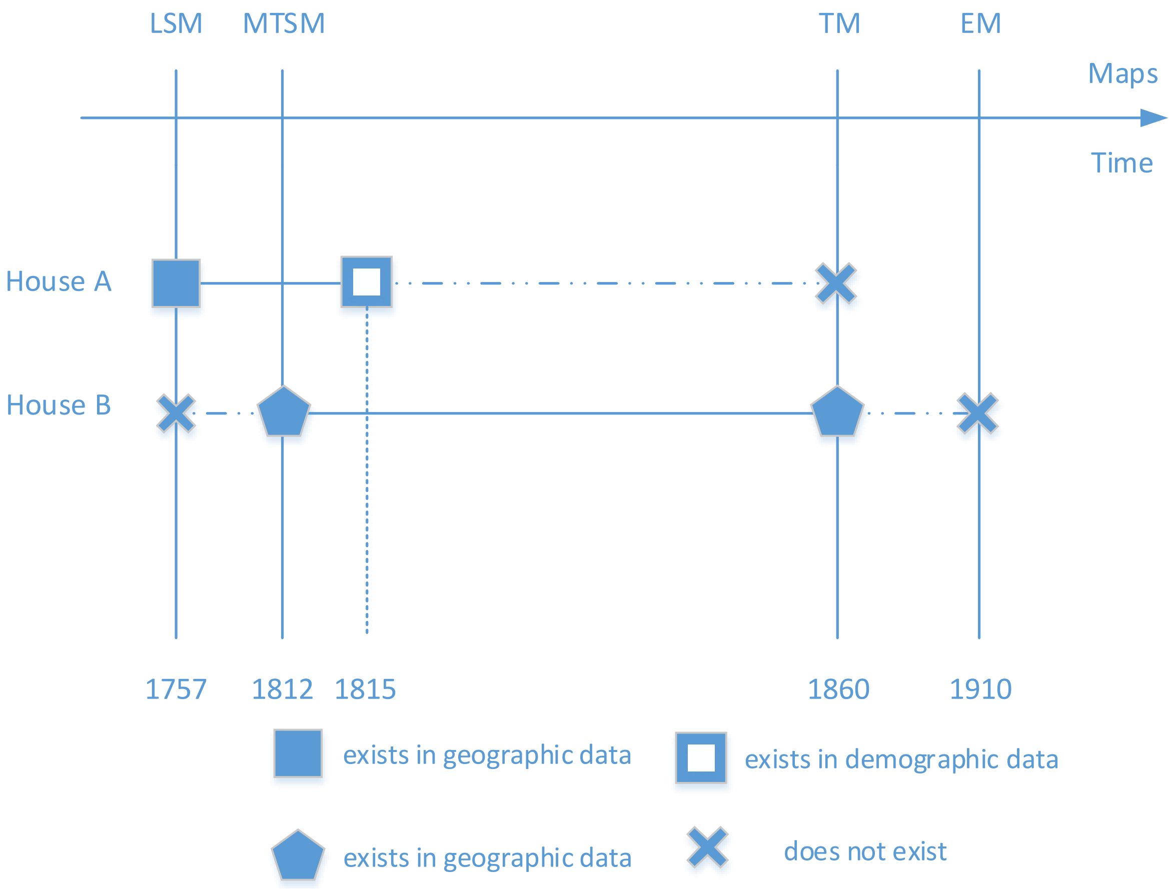

LSM
MTSM
TM
EM
Maps
Time
House A
House B
1757
1812
1815
1860
1910
exists in geographic data
exists in demographic data
exists in geographic data
does not exist

Figure 8.1 Two cases that can be solved.

For House B, to calculate its probability that exists in each year during the whole period, we can divide the whole period into three parts. Thereafter, we can use linear interpolation to calculate its probability value for each year. For example, House B exists in the MTSM (Military Topographic Survey Maps) map in the year 1812 and exists in the TM map in the year 1860. We cannot find its information neither in the LSM map in the year 1757 or in the EM (Economic Maps) map in the year 1910. So we can divide the whole period from 1757 to 1910 to three parts. The first part is from 1757 to 1812, the second part is from 1812 to 1860 and the last part is from 1860 to 1910. For the first part, we can set the probability value of the house to 0 in the year 1757, and set the probability value to 1 in the year 1812. Then, the probability of the house exists in each year during 1757 to 1812 can be calculated. For the second part, we can confirm that the house exists during this period. Thus, we set the probability of the house exists in each year during this period to 1. For the last part, we set the probability value of the house to 1 in the year 1860 and the probability value of the house to 0 in the year 1910. Then, the probability of the house exists in this period can be calculated.

Finally, combine the three parts, we get the probability of the house exists for each year during the year 1757 to 1910.

In the future, we can use the probability value to confirm whether a house exists in a certain year. From the equations, we calculate a probability that the house will exist in the year *t*. We can use this value combine with the demographic data to estimate if the house will exist in the year *t*. To solve this we can use fuzzy logic.

For example, there are four houses that have been estimated in the year *t*. The probabilities of the House A, House B, House C and House D are 0.3, 0.4, 0.6 and 0.7 respectively. House A and House C are found in the demographic database while House B and House D are not. So, we set the value 1 for House A and House C, while we set the value 0 for House B and House D. Then, we can use *Fuzzy And* to calculate the final value for the four houses. Finally, we set a condition, i.e. if the fuzzy value is larger than 0.5, it will be displayed in the map for the year *t*. Then, we can delete house B from the map, because its value is 0.4.

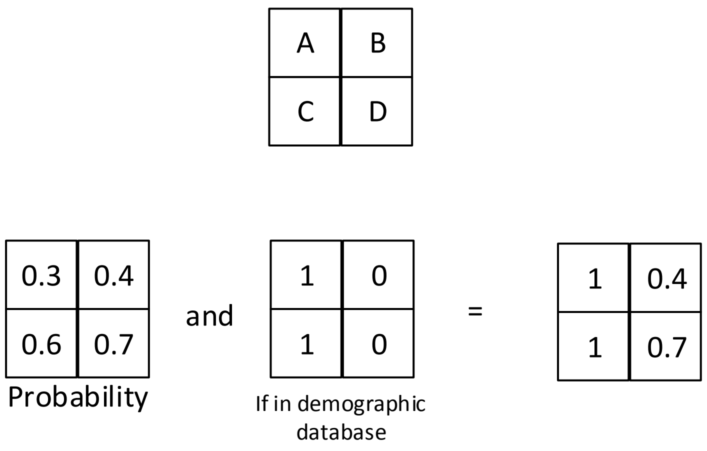


Figure 8.2. Using fuzzy logic to display the house on a map.

In general, when the maps are generated, we have to make an evaluation to test if the linear interpolation is good enough to use. For example, after the maps were generated, we have to use some sample data, such as some houses which existed in this period, to test the generated map to test the accuracy of the linear interpolation. However, the evaluation cannot be implemented, because, we don’t have any spare ground truth data to use to test the maps. We only have the four historical maps as the available data. So, this method cannot be used as a scientific way to analyze the results of this approach. However, from the visualization aspect, we can use it to simulate the animation. And it works well.

The method to symbolize the houses as squares to react the real geographic information should be improved. In this work, only a very basic method was used to change the style of the houses. Some of the point houses which were changed to squares are combined together (*Figure 8.3*). It is hard to identify the houses without the origin layer. Therefore, we have to find a way to improve this symbolization method.

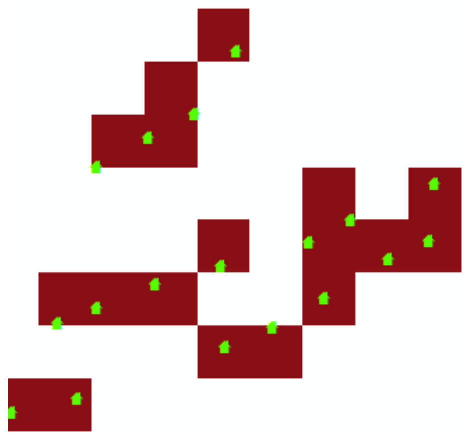


Figure 8.3. House points are presented as squares.

8.3 Future

The methods used to generate the maps in the new version of SEDD Map can be further studied, because there are so many conditions that can be improved, such as the interpolation method and the use of fuzzy logic. The interpolation method can be more complex; we can improve it to spline interpolation or bilinear interpolation, etc. The fuzzy logic can also be set freely, in this case we just used a very basic way to simulate the final maps from the existing maps.

HTML5 is designed to fit all screens, from small screen to large screens. With the hardware technology continually developing, wearable devices, such as Google Glass, are getting more popular. Some smaller screen devices will also appear, e.g. functional watch and glass equipment. These devices would be a new kind of platform to access web mapping applications in the future. So, how to present maps on smaller screens and how to facilitate interactions when browsing geographic data on these kinds of devices would be a new challenge for GIS developers. So, in the future web mapping applications should try to fit these kinds of devices.

9 Conclusions

This study shows a general comparison between the commonly used web mapping programs and applications which are used to visualize and analyze historical demographic data. It also shows the method to improve the compatibility of the SEDD Map and a method to generate animation using four independent historical maps.

The first aim in this study was to review the current web mapping programs and applications. This review revealed that the functions that a commonly used web mapping application for visualizing and analyzing historical demographic data required are: Interactive visualization; Presentation of demographic data with statistical graphics; Presentation of both geographic data and demographic data with animation; Basic mapping tools such as changing the background maps, searching tools, zoom out/in functions, export maps from the mapping application, etc. Most mapping applications present historical data as animation. For example, using statistical graphics to visualize demographic data, at the same time, presenting the geographic data on the screen. The demographic data and geographic data are change together. So, users can easily identify the changes with time varying.

The second aim was to improve the compatibility of SEDD Map. To enable that, a new version of SEDD Map was implemented using HTML5. The new version of SEDD Map can achieve the basic functions required to visualize and analyze the data stored in the SEDD database. The SEDD Map new version can be accessed by both computer and portable devices using most browsers. The browsers which can be used to run the web mapping application are shown in *Table 7.2*.

The last aim was to improve the visualization for presenting the four historical maps as animation, this work used the four historical maps as start years and end years to simulate the maps in between. The simulated maps are created by linear interpolation. The value that calculated by linear interpolation was used to set the probability that a house would exist for a certain year. In the animation, the value was presented by transparency of the house. Compared with the ArcGIS 10 time-aware layer approach, we can visualize the annually changed data. However, this method is only used to visualize the data, it cannot be used to analyze the data in a scientific way. This is because the method to generate the maps cannot be tested, as we do not have the ground truth data to test if a house present in a certain year did exist in reality. However, there is a parallel project that aims at this; when this project is finished, my method could be used with much better input data.

References

ALESHEIKH, A., HELALI, H. & BEHROZ, A. 2002. Web GIS: Technologies and its applications. *In:* Y.C.LEE, C. A. (ed.) *Geospatial Theory, Processing and Applications.* Ottawa, Canada: ISPRS Commission.

BEIGHLEY, L. 2007. *Head first SQL,* Beijing ; Sebastopol, CA, O'Reilly Media.

BENGTSSON, T. 2011. *SEDD - the Scanian Economic Demographic Database* [Online]. Available: http://www.ed.lu.se/EN/databases/sdd.asp.

BENGTSSON, T. 2012. *Research Areas* [Online]. Available: http://www.ed.lu.se/EN/research/default.asp.

BENGTSSON, T. & DRIBE, M. 1997. *Economy and demography in Western Scania, Sweden, 1650-1900,* Kyoto,.

BERJON, R., LEITHEAD, T., NAVARA, E. D., O'CONNOR, E., PFEIFFER, S. & HICKSON, I. 2012. *HTML 5 A vocabulary and associated APIs for HTML and XHTML W3C Candidate Recommendation 17 December 2012* [Online]. Available: http://www.w3.org/TR/2012/CR-html5-20121217/.

BLANSIT, B. D. 2008. An Introduction to Cascading Style Sheets (CSS). *Journal of Electronic Resources in Medical Libraries,* 5**,** 395-409.

BORDELON, B. 2012. Social Explorer. *Journal of Business & Finance Librarianship,* 17**,** 183-187.

BOULOS, M. N., WARREN, J., GONG, J. & YUE, P. 2010. Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping. *Int J Health Geogr,* 9**,** 14.

BROCK, A., TRUILLET, P., ORIOLA, B., PICARD, D. & JOUFFRAIS, C. 2012. Design and User Satisfaction of Interactive Maps for Visually Impaired People.

CANADAY, R. H., HARRISON, R. D., IVIE, E. L., RYDER, J. L., WEHR, L. A. & MORGAN, H. 1974. A back-end computer for data base management. *Communications of the ACM,* 17**,** 575-582.

CHAIX, B., KESTENS, Y., PERCHOUX, C., KARUSISI, N., MERLO, J. & LABADI, K. 2012. An interactive mapping tool to assess individual mobility patterns in neighborhood studies. *Am J Prev Med,* 43**,** 440-50.

CHINA DATA CENTER 2011. China Geo-Explorer User's Guide.

CHINA DATA CENTER 2013. An introduction to China Geo-Explorer.

DAILEY, D., FROST, J., STRAZZULLO, D. & MICROSOFT CORPORATION. 2012. *Building web applications with SVG,* Sebatopool, Calif., Published with the authorization of Microsoft Corp. by O'Reilly Media.

DANCHILLA, B. 2012. *Beginning WebGL for HTML5,* Berkeley, Calif.
New York, Apress ;
Distributed to the book trade worldwide by Springer Science+Business Media.

DEBORAH, K. 2002. *Doing Web development: client-side techniques*.

DIGDAG. 2012. *About DigDag* [Online]. Available: http://digdag.dk/index.php/om-digdag/digdag-er.

DIGDAG. 2013a. *DigDag mapping Denmark through time* [Online]. Available: http://hum.ku.dk/videnonline/histark/digdag/.

DIGDAG 2013b. TEMPORARY SEARCH RESTRICTIONS.

ESRI. 2008. *What is Dojo and why is it important to ArcGIS users?* [Online]. Available: http://blogs.esri.com/esri/arcgis/2008/07/10/what-is-dojo-and-why-is-it-important-to-arcgis-users/.

ESRI. 2009. *Working with Dojo* [Online]. Available: http://resources.esri.com/help/9.3/arcgisserver/apis/javascript/arcgis/help/jshelp_start.htm#jshelp/inside_dojo.htm.

ESRI. 2012. *Getting Started with ArcGIS Web Mapping APIs* [Online]. Available: http://proceedings.esri.com/library/userconf/cahinvrug12/papers/webmapping_apis.pdf.

ESRI. 2013a. *About developing with ArcGIS Web APIs* [Online]. Available: http://resources.arcgis.com/en/help/arcgisonline/index.html#/About_developing_with_ArcGIS_Web_APIs/010q000000mt000000/.

ESRI. 2013b. *About GIS server clusters* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/index.html#//015400000418000000.

ESRI. 2013c. *ArcGIS JavaScript API Overview* [Online]. Available: http://resources.esri.com/help/9.3/arcgisserver/apis/javascript/arcgis/help/jshelp_start.htm#jshelp/new_v16.htm.

ESRI. 2013d. *ArcGIS Server REST API - Overview* [Online]. Available: http://resources.esri.com/help/9.3/arcgisserver/apis/rest/.

ESRI. 2013e. *Components of ArcGIS for Server* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/#/Components_of_ArcGIS_for_Server/01540000035p000000/.

ESRI. 2013f. *Creating web applications with the ArcGIS API for JavaScript* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/index.html#//01540000040n000000.

ESRI. 2013g. *Serving time-aware layers* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/index.html#//01540000028q000000.

ESRI. 2013h. *Setting the time properties on data* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/index.html#//005z0000000q000000.

ESRI. 2013i. *Web APIs* [Online]. Available: http://resources.arcgis.com/content/arcgisserver/web-apis.

ESRI. 2013j. *What is ArcGIS for Server?* [Online]. Available: http://resources.arcgis.com/en/help/main/10.1/index.html#/What_is_ArcGIS_for_Server/01540000037p000000/.

FULTON, S. & FULTON, J. 2011. HTML5 Canvas. Sebastopol: O'Reilly Media, Inc.,.

GALLAUGHER, J. M. & RAMANATHAN, S. C. 1996. Choosing a client/server architecture: a comparison of two- and three-tier systems. *Information Systems Management,* 13**,** 7-13.

GROTOPHORST, C. W. 2012. The book of CSS3: a developer's guide to the future of Web design. *Choice: Current Reviews for Academic Libraries,* 49**,** 1099-1099.

HARRIE, L. 2006. Statistical aspects of spatial interpolation. GIS centre, Lund University.

HEDEFALK, F., HARRIE, L. & SVENSSON, P. 2013. Integrating and distributing longitudinal historical demographic and geographic micro-data.

HIENERT, D., ZAPILKO, B., SCHAER, P. & MATHIAK, B. 2011. Web-Based Multi-View Visualizations for Aggregated Statistics.

HO, Q. 2013. *Architecture and Applications of a Geovisual Analytics Framework.* Linköping University Electronic Press.

HO, Q., LUNDBLAD, P., ÅSTRÖM, T. & JERN, M. A Web-Enabled Visualization Toolkit for Geovisual Analytics Visualisation and Data Analysis. Visualization and Data Analysis, 2011.

HOAR, R. 2009. Visualizing Transit Through a Web Based Geographic Information System. *International Journal of Humanities & Social Sciences,* 3**,** 320-325.

JERN, M. 2009. Collaborative Web-Enabled GeoAnalytics Applied to OECD Regional Data. *In:* LUO, Y. (ed.) *Cooperative Design, Visualization, and Engineering.* Springer Berlin Heidelberg.

JERN, M. 2010. Explore, Collaborate and Publish Official Statistics for Measuring Regional Progress. *In:* LUO, Y. (ed.) *Cooperative Design, Visualization, and Engineering.* Springer Berlin Heidelberg.

JERN, M. & FRANZEN, J. Integrating InfoVis and GeoVis Components. Information Visualization, 2007. IV '07. 11th International Conference, 4-6 July 2007 2007. 511-520.

JOHNSON, G. 2013. *Training Guide: Programming in HTML5 with JavaScript and CSS3,* Microsoft Press.

KAIL, C. 2013. Web Service APIs and Libraries. *Library Journal,* 138**,** 120-120.

KIEHLE, C., GREVE, K. & HEIER, C. 2007. Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoprocessing and Web Service Orchestration. *Transactions in GIS,* 11**,** 819-834.

KRAAK, M. J. & BROWN, A. 2001. *Web cartography : developments and prospects,* London, Taylor & Francis.

LAMMARSCH, T., AIGNER, W., BERTONE, A., MIKSCH, S., TURIC, T. & GARTNER, J. A Comparison of Programming Platforms for Interactive Visualization in Web Browser Based Applications. Information Visualisation, 2008. IV '08. 12th International Conference, 9-11 July 2008 2008. 194-199.

LAW, D. 2013. ArcGIS for Server 101. Understanding architecture, deployment, and workflows. . Available: http://www.esri.com/esri-news/arcuser/spring-2013/arcgis-for-server-101.

LAWSON, B. & SHARP, R. 2011. *Introducing HTML5,* Berkeley, CA, New Riders.

LI, S., DRAGIĆEVIĆ, S. & VEENENDAAL, B. 2011. *Advances in web-based GIS, mapping services and applications [Elektronisk resurs] / editors Songnian Li, Suzana Dragicevic, Bert Veenendaal*, Boca Raton, Fla. : CRC Press/Balkema, c2011.

LONGLEY, P. 2005. *Geographical information systems and science,* Chichester ; Hoboken, NJ, Wiley.

LUNDBLAD, P. 2013. *Applied Geovisual Analytics and Storytelling / Patrik Lundblad,* Norrköping : Department of Science and Technology, Linköping University, 2013.

LUNDBLAD, P., THOURSIE, J. & JERN, M. Swedish Road Weather Visualization. Information Visualisation (IV), 2010 14th International Conference, 26-29 July 2010 2010. 313-321.

MAPSERVER. 2013. *About MapServer* [Online]. Available: http://www.mapserver.org/about.html.

MASSE, M. 2011. *REST API Design Rulebook*, O'Reilly Media.

MCDERMOTT, I. E. 2002. Where Was I?: Maps on the Web. *Searcher*, 10**,** 75.

MCFARLAND, D. S. 2012. CSS3: The Missing Manual, 3rd Edition.

MILER, M., MEDAK, D. & ODOBAŠIĆ, D. 2011. Two-Tier Architecture for Web Mapping with NoSQL Database CouchDB.

MITCHELL, T. 2005. *Web mapping illustrated,* Sebastopol, CA, O'Reilly.

MOUNTS, M. 2009. Dojo: the definitive guide. *Choice: Current Reviews for Academic Libraries,* 46**,** 944-944.

NEWSWIRE, P. R. 2012. Esri Releases National Geographic World Basemap. *CA-Esri-Basemap.* Y.

NORAMBUENA, T., MALIG, R. & MELO, F. 2007. SAGExplore: a web server for unambiguous tag mapping in serial analysis of gene expression oriented to gene discovery and annotation. *NUCLEIC ACIDS RESEARCH,* 35**,** W163-W168.

NUOJUA, J. 2010. WebMapMedia: a map-based Web application for facilitating participation in spatial planning. *Multimedia Systems,* 16**,** 3-21.

ÖZSU, M. T., VALDURIEZ, P. & SPRINGERLINK (ONLINE SERVICE) 2011. Principles of Distributed Database Systems, Third Edition. New York, NY: Springer New York,.

PAI, V. S., DRUSCHEL, P. & ZWAENEPOEL, W. Flash: An Efficient and Portable Web Server. Proceedings of the USENIX 1999 Annual Technical Conference, 1999.

PENG, Z.-R. & TSOU, M.-H. 2003. *Internet GIS : distributed geographic information services for the internet and wireless networks,* Hoboken, N.J., Wiley.

PFEIFFER, S. 2011. HTML5 Media and Canvas. *The Definitive Guide to HTML5 Video.* Apress.

PHILLIPS, G. M. 2003. *Interpolation and approximation by polynomials [Elektronisk resurs] George M. Phillips*, New York : Springer, 2003.

PILGRIM, M. 2103. *DIVE INTO HTML5* [Online]. Available: http://diveintohtml5.info/.

QLIKTECH. 2013. *QlikTech Bolsters Advanced Visualization With Acquisition of NComVA* [Online]. Available: http://www.qlikview.com/us/company/press-room/press-releases/2013/en/0506-qliktech-bolsters-advanced-visualization-with-acquisition-of-ncomva.

RAGGETT, D. 1997. *Client-side Scripting and HTML* [Online]. W3C. Available: http://www.w3.org/TR/WD-script-970314.

RAGGETT, D. 1998. *Raggett on HTML 4,* Harlow, England ; Reading, Mass., Addison-Wesley.

RAMEZ, E. & SHAMKANT, N. 2010. *Fundamentals of Database Systems*.

ROBBINS, J. N. 2013. *HTML5 Pocket Reference, 5th Edition*.

SALEMI, J. 1995. *Guide to client/server databases,* Emeryville, Calif., Ziff-Davis Press.

SCHNEIDER, G. M. & GERSTING, J. L. 2007. *Invitation to computer science,* Boston, Mass., Thomson.

SHAH, M. 2012. Fuzzy based trend mapping and forecasting for time series data. *Expert Systems with Applications,* 39**,** 6351-6358.

SHEKHAR, S. & XIONG, H. 2008. *Encyclopedia of GIS,* New York, Springer.

SOCIAL EXPLORER. 2013a. *About Social Explorer* [Online]. Available: http://www.socialexplorer.com/pub/aboutus/home.aspx.

SOCIAL EXPLORER. 2013b. *Site Overview* [Online]. Available: http://www.socialexplorer.com/pub/help/index.php/site-overview.

SOCIAL EXPLORER. 2013c. *With which browsers is Social Explorer compatible?* [Online]. Available: http://www.socialexplorer.com/pub/help/index.php/faq#Technical.

THE KHRONOS GROUP 2011. Khronos Releases Final WebGL 1.0 Specification to Bring Accelerated 3D Graphics to the Web without Plug-ins.

TRELEASE, R. B. & NIEDER, G. L. 2012. Transforming clinical imaging and 3D data for virtual reality learning objects: HTML5 and mobile devices implementation. *Anat Sci Educ*.

TSAI, F. S. 2011. Web-based geographic search engine for location-aware search in Singapore. *Expert Systems with Applications,* 38**,** 1011-1016.

W3SCHOOLS. 2013. *HTML - XHTML* [Online]. Available: http://www.w3schools.com/html/html_xhtml.asp.

WILLIAMS, M. 2013. Mozilla, Google, Opera introduce new Web browser engines. *CIO (13284045)***,** 23-23.

XU, Z., XINYAN, Z., BING, S. & SHUMING, B. 2009. The spatial data integration and analysis with China Geo-Explorer. *2009 17th International Conference on Geoinformatics*, 1.

YEAGER, N. J. 1996. *Web Server Technology: The Advanced Guide for World Wide Web Information Providers*, Morgan Kaufmann.

Appendix

Appendix A - CSS file of the new version of SEDD Map

```css
html, body {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
    overflow:hidden;
    font-family: Arial,Helvetica,sans-serif;
    font-size: 14px;
    line-height: 20px;
    color: #666;
    background-color: #fff;
}

.map {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
}

#pie{
    position: fixed;
    top: 70px;
    left: 30px;
}

#scale{
    position: absolute;
    height:30px;
    width:20px;
    left:60px;
    bottom:10px;
    margin:0;
    opacity:0.7;
}

#search{
    position: absolute;
    height:30px;
    left:20px;
    top:40px;
    margin:0;
    opacity:0.7;
}

#overview{
    position: absolute;
    top:40px;
}

.sedd_title{
    position: absolute;
    font-weight: bold;
    font-size:large;
    top: 7px;
    left: 5px;
    color: white;
    text-shadow: 5px 5px 5px #000;
    z-index:2;
}

.sedd_title a{
    display: block;
    color: white;
    text-decoration: none;
}
#header {
    position: fixed;
    width: 100%;
    height: 35px;
    top: 0;
    left:0;
    right:0;
    margin:0;
    padding:0;
    border-bottom: 0px solid rgba(0,0,0,0.3);
    background: rgba(88,88,88,0.5);
    box-shadow: 0 2px 6px rgba(0,0,0,0.3),
                inset 0 1px rgba(255,255,255,0.3),
                inset 0 8px rgba(255,255,255,0.2),
                inset 0 8px 20px rgba(255,255,255,0.25),
                inset 0 -15px 30px rgba(0,0,0,0.3);
   -o-box-shadow: 0 2px 6px rgba(0,0,0,0.5),
                  inset 0 1px rgba(255,255,255,0.3),
                  inset 0 8px rgba(255,255,255,0.2),
                  inset 0 8px 20px rgba(255,255,255,0.25),
                  inset 0 -15px 30px rgba(0,0,0,0.3);
   -webkit-box-shadow: 0 2px 6px rgba(0,0,0,0.5),
                       inset 0 1px rgba(255,255,255,0.3),
                       inset 0 8px rgba(255,255,255,0.2),
                       inset 0 8px 20px rgba(255,255,255,0.25),
                       inset 0 -15px 30px rgba(0,0,0,0.3);
   -moz-box-shadow: 0 2px 6px rgba(0,0,0,0.5),
                    inset 0 1px rgba(255,255,255,0.3),
                    inset 0 8px rgba(255,255,255,0.2),
                    inset 0 8px 20px rgba(255,255,255,0.25),
                    inset 0 -15px 30px rgba(0,0,0,0.3);
    z-index:1;
}


.right{
    float : right;
    right:0;
    margin:0;
    padding:0;
}

.menu{

    margin: 0;
    padding: 0;
    border:0;
    width:auto;
    outline: 0;

}

.menu li {
    position:relative;
    float: left;
    list-style: none;
    display: block;
    font-size: 13px;
    padding: 8.5px 16px 7.5px 16px;
    border-left:1px solid rgba(0,0,0,0.3);
    background: rgba(0,0,0,0.1);
}

.menu li:hover {
    background: rgba(0,0,0,0.5);
}

.menu li a {
    display: block;
    color: #dddddd;
    text-decoration: none;
}

.menu li a:hover {
    color: #ffffff;
}

.menu ul {
    margin: 0;
    padding: 0;
    border:0;
    border-radius:0 0 10px 10px;
    position: absolute;
    top: 35px;
    right: 0;
    opacity: 0;
    background:rgba(0,0,0,0.3);

    -webkit-transition: opacity .3s ease .1s;
    -moz-transition: opacity .3s ease .1s;
    -o-transition: opacity .3s ease .1s;
    -ms-transition: opacity .3s ease .1s;
    transition: opacity .3s ease .1s;
}

.menu li:hover > ul { opacity: 1; }

.menu ul li:last-child {
    border-radius:0 0 10px 10px;
}

.menu ul li {
    height: 0;
    overflow: hidden;
    padding: 0;
    border:0;
    background:rgba(0,0,0,0.3);

    -webkit-transition: height .3s ease .1s;
    -moz-transition: height .3s ease .1s;
    -o-transition: height .3s ease .1s;
    -ms-transition: height .3s ease .1s;
    transition: height .3s ease .1s;
}

.menu li:hover > ul li {
    height: 22px;
    overflow: visible;
    padding: 0;
}

.menu ul li a {
    width: 150px;
    margin: 0;
    border: none;
    text-align:center;
}

#box a{
    width: 150px;
    margin: 0;
    border: none;
    text-align:left;
}

.menu span{
    color:#ffbb00;
}

.menu span:hover{
    color:#ffffff;
}

#Verticalslider{
    position: absolute;
    height:30px;
    width:5px;
    left:20px;
    bottom:120px;
    margin:0;
    text-align:center;
}
#button{
    position: absolute;
    height:30px;
    width:5px;
    left:30px;
    bottom:150px;
    margin:0;
    text-align:center;
}
```

Appendix B - HTML file of the new version of SEDD Map

```
<!DOCTYPE html>
<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=7,IE=9">
        <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <meta name="apple-mobile-web-app-status-bar-style" content="translucent-black">

        <title>SEDD Map</title>

        <link href="/lu.ico" rel="icon" type="image/x-icon"/>
        <link rel="stylesheet" href="http://serverapi.arcgisonline.com/jsapi/arcgis/3.4/js/esri↙
    .css"/>
        <link rel="stylesheet" href="http://serverapi.arcgisonline.com/jsapi/arcgis/3.4/js/dojo/dijit/  ↙
    themes/tundra/tundra.css"/>
        <link rel="stylesheet" href="/seddstyle.css"/>

        <script src="http://serverapi.arcgisonline.com/jsapi/arcgis/3.4compact/"></script>
        <script>var dojoConfig = { parseOnLoad: true };</script>

        <script>
            dojo.require("esri.map");
            dojo.require("dojo.ready")
            dojo.require("dojo.parser");
            dojo.require("esri.layers.FeatureLayer");
            dojo.require("esri.dijit.Scalebar");
            dojo.require("esri.dijit.OverviewMap");
            dojo.require("dijit.form.VerticalSlider");
            dojo.require("dijit.form.Button");
            dojo.require("esri.tasks.find");

            var map;
            var watchId;
            var graphic;
            var attributes;
            var infoTemplate;
            var visible = [];
            var layer, layer1, layer2, layer3, layer4;
            var overview;
            var scalebar;
            var featureLayer;
            var find;
            var params;

            // create the map, scale bar, overview map
            function init() {

                map = new esri.Map(
                    "SEDD",
                    {
                        logo: false,
                        center: [13.077, 55.782],
                        zoom: 13,
                        basemap: "streets",
                        sliderPosition: "bottom-left",
                        showAttribution: false,
                    }
                );

                layer = new esri.layers.ArcGISDynamicMapServiceLayer("http://haptimap.gis.lu.se/arcgis/ ↙
    rest/services/SEDD/SEDDService/MapServer/");

                map.addLayer(layer);

                layer1 = new esri.layers.ArcGISDynamicMapServiceLayer("http://haptimap.gis.lu.se/arcgis/↙
    rest/services/SEDD/1757_1863/MapServer/", {
                    opacity: 0,
                });
                map.addLayer(layer1);
                layer2 = new esri.layers.ArcGISDynamicMapServiceLayer("http://haptimap.gis.lu.se/arcgis/↙
    rest/services/SEDD/1812_1820/MapServer/", {
                    opacity: 0,
                });
                map.addLayer(layer2);

                layer3 = new esri.layers.ArcGISDynamicMapServiceLayer("http://haptimap.gis.lu.se/arcgis/↙
    rest/services/SEDD/1860_1865/MapServer/", {
                    opacity: 0,
                });
                map.addLayer(layer3);

                layer4 = new esri.layers.ArcGISDynamicMapServiceLayer("http://haptimap.gis.lu.se/arcgis/↙
    rest/services/SEDD/1910_1915/MapServer/", {
                    opacity: 0,
                });
                map.addLayer(layer4);

                find = new esri.tasks.FindTask("http://haptimap.gis.lu.se/arcgis/rest/services/SEDD/     ↙
    SEDDService/MapServer/");

                params = new esri.tasks.FindParameters();
                    params.layerIds = [6];
                    params.searchFields = ["objectTitl","population"];

                dojo.connect(map, "onLoad", function Scale() {
                    scalebar = new esri.dijit.Scalebar(
                                    {
                                       map: map,
                                       scalebarUnit: "dual",
                                    },
                                       dojo.byId("scale")
                                    );
                });

                dojo.connect(map, "onLoad", function OverViewMap() {
                    overview = new esri.dijit.OverviewMap(
                                    {
                                        map:map,
                                        maximizeButton:true,
                                        attachTo:"bottom-right",
                                        color:"#ffbb00",
                                        opacity:0.6,
                                    }
                                );
                    overview.startup();
                });

                var Verticalslider = new dijit.form.VerticalSlider({
                    name: "Verticalslider",
                    value: 1,
                    minimum: 0,
                    maximum: 158,
                    intermediateChanges: true,
                    discreteValues: 159,
                    showButtons: true,
                    style: "height:159px;",
                    onChange: function(value) {
                        if(value<55 ){
                            layer1.setOpacity((70-value) / 56);
                            layer2.setOpacity(0);
                            layer3.setOpacity(0);
                            layer4.setOpacity(0);
                        }
                        else if(value <103){
                            layer1.setOpacity(0);
                            layer2.setOpacity((116-value) / 48);
                            layer3.setOpacity(0);
                            layer4.setOpacity(0);
                        }
                        else if(value <153){
                            layer1.setOpacity(0);
                            layer2.setOpacity(0);
                            layer3.setOpacity((163-value) / 50);
                            layer4.setOpacity(0);
                        }
                        else{
                            layer1.setOpacity(0);
                            layer2.setOpacity(0);
                            layer3.setOpacity(0);
                            layer4.setOpacity(1);
                        }
                        dojo.byId("opval").innerHTML = value+1757;
                    }
                }, "Verticalslider");

                map.on("load", initFeatureLayer);

            }

            function initFeatureLayer() {

                var infoTemplate = new esri.InfoTemplate(
                        "${objectTitl}",
                        "Population: ${population} <br/> SizeMantal: ${sizeMantal} <br/>SizeHectar: $   ↙
    {sizeHectar} <br/> Property Right: ${prptyRight}"
                    );

                featureLayer = new esri.layers.FeatureLayer("http://haptimap.gis.lu.se/arcgis/rest/     ↙
    services/SEDD/SEDDService/MapServer/6/",{
                    mode: esri.layers.FeatureLayer.MODE_ONDEMAND,
                    outFields: ["*"],
                    infoTemplate: infoTemplate,
                    visible: false,
                    opacity:0.6
                    }
                );

                map.addLayer(featureLayer);
                map.infoWindow.resize(190, 100);
            }

            function dosearch() {
                params.searchText = dojo.byId("searchText").value;
                find.execute(params, showResults);
            }

            function showResults(results) {
                var result, attribs;
                var s = ["<table border=\"1\"><thead><tr style=\"background-color:#ffbb00;\"><td>Name</ ↙
    td><td>Population</td><td>Area Hectare </td></tr></thead><tbody id=\"states\">"];
                dojo.forEach(results,function(result){
                    attribs = result.feature.attributes;
                    s.push("<tr><td>" + attribs.objectTitl + "</td><td>" + attribs.population + "</td>  ↙
    <td>" + attribs.sizeHectar + "</td></tr>");
                });
                s.push("</tbody></table>");
                dojo.byId("tbl").innerHTML = s.join("");
            }

            // geolocation function
            function showGeolocation() {

                dojo.connect(window, 'resize', map,map.resize);
                if(navigator.geolocation){
                    navigator.geolocation.getCurrentPosition(zoomToLocation, errorHandler);
                    watchId = navigator.geolocation.watchPosition(showLocation, errorHandler);
                }
                else{
                    alert("Geolocation not support!");
                }

            }

            function zoomToLocation(position) {
                var pt = new esri.geometry.Point(position.coords.longitude, position.coords.latitude);
                addGraphic(pt);
                map.centerAndZoom(pt, 16);

            }

            function showLocation(position) {

                var pt = new esri.geometry.Point(position.coords.longitude, position.coords.latitude);
                if (!graphic) {
                    addGraphic(pt);
                }
                else {
                    graphic.setGeometry(pt);
                }
                map.center(pt, 16);

            }

            function errorHandler(err) {

                if(err.code == 1) {
                    alert("Error: Access is denied!");
                }else if( err.code == 2) {
                    alert("Error: Position is unavailable!");
                }

            }

            // add the point symbol for the geolocation point
            function addGraphic(pt){

                var PositionSymbol = new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol. ↙
    STYLE_CIRCLE, 12,
                                        new esri.symbol.SimpleLineSymbol(esri.symbol.SimpleLineSymbol.  ↙
    STYLE_SOLID,
                                        new dojo.Color([255, 187, 0, 0.3]), 8),
                                        new dojo.Color([255, 187, 0, 0.9])
                                     );
                attributes = {"lat":pt.y.toFixed(4),"lon":pt.x.toFixed(4)};
                infoTemplate = new esri.InfoTemplate(
                    "I am here!",
                    "Latitude: ${lat} <br/> Longitude: ${lon}"
                );
                graphic = new esri.Graphic(pt,PositionSymbol,attributes,infoTemplate);
                map.graphics.add(graphic);

            }

            function clearGeolocationGraphics() {

                navigator.geolocation.clearWatch(watchId);
                map.graphics.clear();

            }

            // control the visible layers
            function Visibility() {

                var inputs = dojo.query(".list"), input;
                visible = [];
                for (var i=0, il=inputs.length; i<il; i++) {
                    if (inputs[i].checked) {
                        visible.push(inputs[i].id);
                    }
                }
                if(visible.length === 0){
                    visible.push(-1);
                }
                layer.setVisibleLayers(visible);
            }

            function FLVisibility() {
                (featureLayer.visible) ? featureLayer.hide() : featureLayer.show();

            }

            function showHide() {
                var div = document.getElementById("PI");
                if (div.style.display == 'none') {
                    div.style.display = '';
                }else {
                    div.style.display = 'none';
                }
            }

            function TSDIV(){

                var tsdiv = document.getElementById("Verticalslider");

                if(tsdiv.style.display == 'none'){
                    tsdiv.style.display='';
                    layer1.show();
                    layer2.show();
                    layer3.show();
                    layer4.show();
                }else{
                    tsdiv.style.display = 'none';
                    layer1.hide();
                    layer2.hide();
                    layer3.hide();
                    layer4.hide();
                }

            }

            function SDIV(){

                var sdiv = document.getElementById("search");

                if(sdiv.style.display == 'none'){
                    sdiv.style.display='';
                }else{
                    sdiv.style.display = 'none';
                }

            }

            dojo.ready(function(){
                var button = new dijit.form.Button({
                    label: "Play",
                    onClick: function(){
                        var val = 1;
                        var one = function () {
                            dijit.byId('Verticalslider').set("value", val);
                            setTimeout(function() {
                                val = val + 1;
                                dijit.byId('Verticalslider').set("value", val);
                                if( val < 160 ){
                                    one();
                                }
                            }, 100);
                        };
                        one();
                    }
                }, "progButtonNode");

            });

            dojo.addOnLoad(init);
        </script>

    </head>

    <body>
        <div id = "SEDD"></div>
        <div id = "scale"></div>
        <div id = "Verticalslider"> Year:<br>(<span id="opval">1757</span>)</div>
        <div id = "button"><button id="progButtonNode" type="button"></button></div>
        <div style = "display: none;" id = "search">
            Name:<input type="text" id="searchText" size="20" value="Hög 12"/>
            <input type="button" value="Search" onclick="dosearch()"/>
            <div id="tbl"></div>
        </div>

        <div class = "sedd_title"><a href="http://gisweden.com/sedd.html">SEDD Map<a></div>

        <div id = "header" >
            <ul class = "menu right">
                <li><a href = "#">Layers</a>
                    <ul id="box">
                        <form name="layer_form">
                            <li><a><input type='checkbox' checked="checked" class='list' id='0' onclick=↙
    'Visibility();'/><label for="0">1910-1915 House</label></a></li>
                            <li><a><input type='checkbox' checked="checked" class='list' id='1' onclick=↙
    'Visibility();'/><label for="1">1910-1915 Property</label></a></li>
                            <li><a><input type='checkbox' class='list' id='2' onclick='Visibility();'/> ↙
    <label for="2">1860-1865 House</label></a></li>
                            <li><a><input type='checkbox' class='list' id='3' onclick='Visibility();'/> ↙
    <label for="3">1812-1820 House</label></a></li>
                            <li><a><input type='checkbox' class='list' id='4' onclick='Visibility();'/> ↙
    <label for="4">1757-1863 House</label></a></li>
                            <li><a><input type='checkbox' class='list' id='5' onclick='Visibility();'/> ↙
    <label for="5">1757-1863 Property</label></a></li>
                            <li><a><input type='checkbox'  id = '6' onclick='FLVisibility();showHide(); ↙
    '/><label for="6">1804 Hog Property</label></a></li>
                        </form>
                    </ul>
                </li>

                <li><a href = "#">Tools</a>

                    <ul>
                        <li><a href = "#">Navigation</a></li>

                        <li onclick="showGeolocation();"><a href = "#"><span>On</span></a></li>
                        <li onclick="clearGeolocationGraphics();"><a href = "#"><span>Off</span></a></  ↙
    li>



                        <li><a href = "#">Base Map</a></li>

                        <li onclick="map.setBasemap('streets');"><a href = "#"><span>Streets</a></li>
                        <li onclick="map.setBasemap('osm');"><a href = "#"><span>OpenStreet</a></li>
                        <li onclick="map.setBasemap('hybrid');"><a href = "#"><span>Hybrid</a></li>
                        <li onclick="map.setBasemap('topo');"><a href = "#"><span>Topographic</a></li>
                        <li onclick="map.setBasemap('gray');"><a href = "#"><span>Gray</a></li>

                        <li><a href = "#" onclick = "TSDIV();">Animation</a></li>
                        <li><a href = "#" onclick = "SDIV();">Search</a></li>

                    </ul>

                </li>

                <li><a href = "#">About</a>
                    <ul>
                        <li><a href = "http://www.lu.se/" target="_blank">Lund University</a></li>
                        <li><a href = "http://www.ed.lu.se/" target="_blank">CED</a></li>
                        <li><a href = "http://www.gis.lu.se/english/index.htm" target="_blank">GIS      ↙
    Center</a></li>
                    </ul>
                </li>

            <ul/>
        </div>

        <div style = "display : none;" id ="PI">
        <canvas id="pie" width="200" height="100"></canvas>
        </div>

        <script type="text/javascript">
            var data = [61.5, 34.5,4];
            var color = ["#ffff73","#97dbf2","#94c494"];
            // draw the pie chart and legend
            function PieChart(){
                var canvas = document.getElementById("pie");
                var ctx = canvas.getContext("2d");
                var startPoint = Math.PI;
                var Rstart = 30;

                for(var i=0;i<data.length;i++){
                    ctx.fillStyle = color[i];
                    ctx.strokeStyle = color[i];
                    ctx.beginPath();
                    ctx.moveTo(50,50);
                    ctx.arc(50,50,50,startPoint,startPoint-Math.PI*2*(data[i]/100),true);
                    ctx.rect(110,Rstart+10*i,10,5);
                    ctx.fill();
                    ctx.stroke();
                    startPoint -= Math.PI*2*(data[i]/100);
                    Rstart += 10;
                }
                ctx.fillStyle = "#000000";
                    ctx.fillText("crown 61.5%",110,45);
                    ctx.fillText("freehold 34.5%",110,65);
                    ctx.fillText("crown/freehold 4%",110,85);
            }

            PieChart();
        </script>

    </body>

</html>
```

Institutionen för naturgeografi och ekosystemvetenskap, Lunds Universitet.

Student examensarbete (Seminarieuppsatser). Uppsatserna finns tillgängliga på institutionens geobibliotek, Sölvegatan 12, 223 62 LUND. Serien startade 1985. Hela listan och själva uppsatserna är även tillgängliga på LUP student papers (www.nateko.lu.se/masterthesis) och via Geobiblioteket (www.geobib.lu.se)

The student thesis reports are available at the Geo-Library, Department of Physical Geography and Ecosystem Science, University of Lund, Sölvegatan 12, S-223 62 Lund, Sweden. Report series started 1985. The complete list and electronic versions are also electronic available at the LUP student papers (www.nateko.lu.se/masterthesis) and through the Geo-library (www.geobib.lu.se)

230 Cléber Domingos Arruda (2011) Developing a Pedestrian Route Network Service (PRNS)
231 Nitin Chaudhary (2011) Evaluation of RCA & RCA GUESS and estimation of vegetation-climate feedbacks over India for present climate
232 Bjarne Munk Lyshede (2012) Diurnal variations in methane flux in a low-arctic fen in Southwest Greenland
233 Zhendong Wu (2012) Dissolved methane dynamics in a subarctic peatland
234 Lars Johansson (2012) Modelling near ground wind speed in urban environments using high-resolution digital surface models and statistical methods
235 Sanna Dufbäck (2012) Lokal dagvattenhantering med grönytefaktorn
236 Arash Amiri (2012)Automatic Geospatial Web Service Composition for Developing a Routing System
237 Emma Li Johansson (2012) The Melting Himalayas: Examples of Water Harvesting Techniques
238 Adelina Osmani (2012) Forests as carbon sinks - A comparison between the boreal forest and the tropical forest
239 Uta Klönne (2012) Drought in the Sahel – global and local driving forces and their impact on vegetation in the 20th and 21st century
240 Max van Meeningen (2012) Metanutsläpp från det smältande Arktis
241 Joakim Lindberg (2012) Analys av tillväxt för enskilda träd efter gallring i ett blandbestånd av gran och tall, Sverige
242 Caroline Jonsson (2012) The relationship between climate change and grazing by herbivores; their impact on the carbon cycle in Arctic environments
243 Carolina Emanuelsson and Elna Rasmusson (2012) The effects of soil erosion on nutrient content in smallholding tea lands in Matara district, Sri Lanka
244 John Bengtsson and Eric Torkelsson (2012) The Potential Impact of Changing Vegetation on Thawing Permafrost: Effects of manipulated vegetation on summer ground temperatures and soil moisture in Abisko, Sweden
245 Linnea Jonsson (2012). Impacts of climate change on Pedunculate oak and Phytophthora activity in north and central Europe
246 Ulrika Belsing (2012) Arktis och Antarktis föränderliga havsistäcken
247 Anna Lindstein (2012) Riskområden för erosion och näringsläckage i Segeåns avrinningsområde
248 Bodil Englund (2012) Klimatanpassningsarbete kring stigande havsnivåer i Kalmar läns kustkommuner
249 Alexandra Dicander (2012) GIS-baserad översvämningskartering i Segeåns

avrinningsområde
250 Johannes Jonsson (2012) Defining phenology events with digital repeat photography
251 Joel Lilljebjörn (2012) Flygbildsbaserad skyddszonsinventering vid Segeå
252 Camilla Persson (2012) Beräkning av glaciärers massbalans – En metodanalys med fjärranalys och jämviktslinjehöjd över Storglaciären
253 Rebecka Nilsson (2012) Torkan i Australien 2002-2010 Analys av möjliga orsaker och effekter
254 Ning Zhang (2012) Automated plane detection and extraction from airborne laser scanning data of dense urban areas
255 Bawar Tahir (2012) Comparison of the water balance of two forest stands using the BROOK90 model
256 Shubhangi Lamba (2012) Estimating contemporary methane emissions from tropical wetlands using multiple modelling approaches
257 Mohammed S. Alwesabi (2012) MODIS NDVI satellite data for assessing drought in Somalia during the period 2000-2011
258 Christine Walsh (2012) Aerosol light absorption measurement techniques: A comparison of methods from field data and laboratory experimentation
259 Jole Forsmoo (2012) Desertification in China, causes and preventive actions in modern time
260 Min Wang (2012) Seasonal and inter-annual variability of soil respiration at Skyttorp, a Swedish boreal forest
261 Erica Perming (2012) Nitrogen Footprint vs. Life Cycle Impact Assessment methods – A comparison of the methods in a case study.
262 Sarah Loudin (2012) The response of European forests to the change in summer temperatures: a comparison between normal and warm years, from 1996 to 2006
263 Peng Wang (2012) Web-based public participation GIS application – a case study on flood emergency management
264 Minyi Pan (2012) Uncertainty and Sensitivity Analysis in Soil Strata Model Generation for Ground Settlement Risk Evaluation
265 Mohamed Ahmed (2012) Significance of soil moisture on vegetation greenness in the African Sahel from 1982 to 2008
266 Iurii Shendryk (2013) Integration of LiDAR data and satellite imagery for biomass estimation in conifer-dominated forest
267 Kristian Morin (2013) Mapping moth induced birch forest damage in northern Sweden, with MODIS satellite data
268 Ylva Persson (2013) Refining fuel loads in LPJ-GUESS-SPITFIRE for wet-dry areas - with an emphasis on Kruger National Park in South Africa
269 Md. Ahsan Mozaffar (2013) Biogenic volatile organic compound emissions from Willow trees
270 Lingrui Qi (2013) Urban land expansion model based on SLEUTH, a case study in Dongguan City, China
271 Hasan Mohammed Hameed (2013) Water harvesting in Erbil Governorate, Kurdistan region, Iraq - Detection of suitable sites by using Geographic Information System and Remote Sensing
272 Fredrik Alström (2013) Effekter av en havsnivåhöjning kring Falsterbohalvön.
273 Lovisa Dahlquist (2013) Miljöeffekter av jordbruksinvesteringar i Etiopien
274 Sebastian Andersson Hylander (2013) Ekosystemtjänster i svenska agroforestrysystem

275 Vlad Pirvulescu (2013) Application of the eddy-covariance method under the canopy at a boreal forest site in central Sweden

276 Malin Broberg (2013) Emissions of biogenic volatile organic compounds in a Salix biofuel plantation – field study in Grästorp (Sweden)

277 Linn Renström (2013) Flygbildsbaserad förändringsstudie inom skyddszoner längs vattendrag

278 Josefin Methi Sundell (2013) Skötseleffekter av miljöersättningen för natur- och kulturmiljöer i odlingslandskapets småbiotoper

279 Kristín Agustsdottír (2013) Fishing from Space: Mackerel fishing in Icelandic waters and correlation with satellite variables

280 Cristián Escobar Avaria (2013) Simulating current regional pattern and composition of Chilean native forests using a dynamic ecosystem model

281 Martin Nilsson (2013) Comparison of MODIS-Algorithms for Estimating Gross Primary Production from Satellite Data in semi-arid Africa

282 Victor Strevens Bolmgren (2013) The Road to Happiness – A Spatial Study of Accessibility and Well-Being in Hambantota, Sri Lanka

283 Amelie Lindgren (2013) Spatiotemporal variations of net methane emissions and its causes across an ombrotrophic peatland - A site study from Southern Sweden

284 Elisabeth Vogel (2013) The temporal and spatial variability of soil respiration in boreal forests - A case study of Norunda forest, Central Sweden

285 Cansu Karsili (2013) Calculation of past and present water availability in the Mediterranean region and future estimates according to the Thornthwaite water-balance model

286 Elise Palm (2013) Finding a method for simplified biomass measurements on Sahelian grasslands

287 Manon Marcon (2013) Analysis of biodiversity spatial patterns across multiple taxa, in Sweden

288 Emma Li Johansson (2013) A multi-scale analysis of biofuel-related land acquisitions in Tanzania - with focus on Sweden as an investor

289 Dipa Paul Chowdhury (2013) Centennial and Millennial climate-carbon cycle feedback analysis for future anthropogenic climate change

290 Zhiyong Qi (2013) Geovisualization using HTML5 - A case study to improve animations of historical geographic data

291 Boyi Jiang (2013) GIS-based time series study of soil erosion risk using the Revised Universal Soil Loss Equation (RUSLE) model in a micro-catchment on Mount Elgon, Uganda