

sm

Final Project Report Group 3

IKT 407
Autumn 2006



Author(s): Santhakumar Chanrasekaram Ni Chen Mats Oustad	Supervisor(s): Morten Goodwin Olsen
Version: 1.0 Status: FINAL	Pages: 18 (including this page) Modified date: 2006-11-27
Keywords: Geographical Information System (GIS), Flash, PHP, DIY Map, EIAO, MapServer	

Abstract:

In this project we have developed a GIS system based on Flash that can show the accessibility measures of different countries according to the scorecards. The report describes the two solutions we have done in building the GIS system. Firstly, we started working with using MapServer and making some progress, but we found that the MapServer maybe not a perfect way of the project, because it is time-consuming and immature. However, it was a good experience for us to be familiar with working on a GIS system. We noticed the shortages of using MapServer and found that using Flash is a better way. The application called DIY MAP which is written in Flash is fairly easy for us to work with. Using DIY MAP we can simply carry out the functions which meet EIAO's needs. The implementation is discussed in the report.

Besides MapServer and Flash, the technologies we use here are quite mature, for example PHP, XML, JavaScript and etc. Therefore there is no big problem when we are using these technologies. The report also describes how we use these technologies to work on the project.

Licence:

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Version Control

Version¹	Status²	Date³	Change⁴	Author⁵
0.01	DRAFT	2006-11-18	Introduction, design, implementation	Mats
0.02	DRAFT	2006-11-18	Background, problem description	Santhan
0.03	DRAFT	2006-11-19	Requirements specification	Chen
0.04	DRAFT	2006-11-19	Putting the report together	Santhan
0.05	DRAFT	2006-11-19	Evaluation and testing	Mats
0.06	DRAFT	2006-11-21	Discussion	Mats, Santhan
0.07	DRAFT	2006-11-22	Conclusion	Mats
0.08	DRAFT	2006-11-23	Executive summary	Santhan
0.09	DRAFT	2006-11-23	Abstract	Chen
0.10	DRAFT	2006-11-23	Putting the report together	Santhan
1.0	FINAL	2006-11-26	Fixed some errors	Santhan

1 Version indicates the version number starting at 0.1 for the first draft and 1.0 for the first review version.

2 Status is DRAFT, REVIEW or FINAL

3 Date is given in ISO format: yyyy-mm-dd

4 Change describes the changes carried out since the previous version

5 Author is the one who did the change

Table of Contents

<u>1 EXECUTIVE SUMMARY.....</u>	<u>5</u>
<u>2 INTRODUCTION.....</u>	<u>5</u>
<u>3 BACKGROUND.....</u>	<u>6</u>
<u>3.1 TOOLS.....</u>	<u>7</u>
<u>4 PROBLEM DESCRIPTION.....</u>	<u>8</u>
<u>5 REQUIREMENTS SPECIFICATION.....</u>	<u>9</u>
<u>6 DESIGN.....</u>	<u>10</u>
<u>6.1 DEVELOPMENT SOFTWARE.....</u>	<u>11</u>
<u>7 IMPLEMENTATION.....</u>	<u>12</u>
<u>8 EVALUATION AND TESTING.....</u>	<u>15</u>
<u>9 DISCUSSION.....</u>	<u>16</u>
<u>9.1 PROJECT OUTCOMES.....</u>	<u>16</u>
<u>9.2 EVALUATION OF THE OVERALL RESULTS.....</u>	<u>16</u>
<u>10 CONCLUSION.....</u>	<u>17</u>
APPENDIX.....	18
A0 GLOSSARY AND ABBREVIATIONS.....	18
A0 REFERENCES.....	18

1 Executive summary

The goal of the project was to create a Geographical Information System based on an Open Source GIS module that could present the accessibility measures as colour coded scorecards for the different levels of EU NUTS.

To solve this task we researched two kinds of solutions, one with MapServer and the other with DIY Map. MapServer is a traditional GIS system and DIY Map is a Flash GIS solution. For both of these solutions we needed a map with NUTS support, but it was impossible to get hold on, so we used a regular Europe map.

DIY Maps gives us a good solution to make a colour coded map of Europe. Development time, is a mere fraction in comparison to the time it would take in a big GIS system and the system renders the maps, zooms in and out substantially faster than MapServer does.

The best way to build a GIS today would be to make it as a flash application, then you get client side calls, everything is loaded at once, and given you get a good implementation of a map you got a lot more opportunities with Flash than with for example MapServer.

2 Introduction

They say that a picture says more than a thousand words and that is also the case with a GIS, geographical information system. It gives an opportunity to display otherwise “boring and vast” statistical information into a graphic interface. Instead of browsing through endless data the user can easily get an overview of the situation by looking at the image. GIS is used widely by companies, organizations and others around the world to show information about themselves, for instance locations of branches around the world. It's a good way to lighten up a website with some graphics that actually has a purpose.

The most common way to draw maps is to interpret shapes files. This format is developed by ESRI, which is the leading company within GIS and GIS research. These files are a very powerful way to display geospatial data. Most big GIS solutions is supporting this way of representing a map. It's generally expensive and difficult to get a hold of good shape files.

Accessibility measures like those we are getting from the EIAO Data Warehouse Database consist of data that is very detailed and difficult to read properly in plain tables and text. This goes especially for users that don't have much prior knowledge in the field and is just looking for a quick way to check for instance their countries accessibility ratings. Then the tables can be behind for the user who wants to drill deeper down into the data.

3 Background

The goal of this project is to create a Geographical Information System based on an Open Source GIS module that can present accessibility measures as colour coded scorecards. These scorecards are defined in Unified Web Evaluation Methodology (UWEM) ^[17] for the EU NUTS regions ^[18]. Data that will be presented will be extracted from the EIAO RDF repository, or possibly the EIAO data warehouse.

Geographical Information System is a system that we use to register, model, save, get, manipulate, analyze and present the geographical referred data. This type of system has been used by governments, organizations and companies for about twenty years. The use of GIS has increased during the last decade.

UWEM is based on the W3C Web Content Accessibility Guidelines 1.0 [WCAG10]. In these guidelines we also find definitions for Nomenclature of Territorial Units for Statistics (NUTS). We have three levels of NUTS, NUTS level 1, level 2 and level 3. NUTS level 1 area is a territory with 3 to 7 million in population. NUTS level 2 from 800 000 to 3 million and level 3 from 150 000 to 800 000 in population. Present results of statically data with NUTS makes sense since we split countries into territories with population rate, which makes it easy to understand and we get a better overview.

There are many GIS systems out there in the market. One of them is ArcGIS which is one of the most popular one, but costs a lot to use. When it comes to open source GIS systems, we have MapServer and GRASS GIS. MapServer is the most popular one between them and supports many platforms and programming languages.

In addition to the traditional GIS systems, we also have Flash GIS solutions, which are touted to be next big thing within GIS systems. ^[8] DIY Map is such Flash GIS solution.

We have two possible solutions to present accessibility measures, one with MapServer and the other one with DIY Map. Both solutions might have some challenges with presenting results on NUTS levels, since it would be difficult to get hold of map with NUTS support. Maps with NUTS support are not cheap.

3.1 Tools

MapServer ^[1]

MapServer was developed by the University of Minnesota ForNet project in cooperation with NASA and the Minnesota Department of Natural Resources. Today the MapServer project is hosted by the TerraSIP project, a NASA sponsored project between the UMN and consortium of land management interests and maintained by a growing number of developers around the world.

MapServer is an Open Source development environment for building spatially-enabled internet applications. MapServer is not a full-featured GIS system but has a lot of features as advanced cartographic output and supports for popular scripting and development environments as PHP, Python, Perl, Ruby, Java and C#.

MapServer has been used in many projects. It has been used to show chronological layers of vulcanology of Tenerife Island. ^[12] United Nations Environment Programme (UNEP) uses MapServer in their GEO data portal, which has national, regional and global statistics covering themes like freshwater, population, forests, emissions, climate, disasters, health and GDP. ^[13]

DIY Map ^[4]

DIY Map is a clickable zooming map written in Flash. And this map is coloured by data from an external file, which is an xml file. The external data file makes it easy to configure, customize and update colours, add points and we can also use the same Flash file many times in the same Web page with different data sets.

DIY Map is free for personal, educational, and non-commercial use. For-profit corporations, government agencies, and religious organizations must purchase a license for \$20 per domain.

The United Nations Association of Norway uses DIY Map on their educational Web site Globalis, to show countries in NATO, population statistics, map of global conflicts and etc. ^[14] DIY has been used by an activist group to show, from where the Iraq casualties came from in the US. ^[15] Used to show website traffic analysis to display recent traffic geographically. ^[16]

PHP ^[2]

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into. This language is specially suited for creating dynamic and interactive websites.

The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems.

Apache HTTP server ^[3]

Apache HTTP server is the most popular web server on internet today. According to Netcraft Web Server Survey more than 70% of the web sites on the internet were using Apache in November 2005. It is easy to install and configure the Apache for personal use. Apache also has support for PHP, which scripting language we are using in our project.

4 Problem description

The goal of this project is to research and make a prototype of a Geographical Information System based on an Open Source GIS module that can present the accessibility measures as colour scorecards. The accessibility scores and corresponding colour scorecards are defined in UWEM and may be applied for the EU NUTS regions.

First thing we have to do is to get the accessibility measures from the EIAO's PostgreSQL database and then present the data we get with MapServer or DIY Map.

We have two types of solution as we mentioned above, one with MapServer and the other with DIY Map. With MapServer solution, we have to work with map files, which contain map data such as coordinates, colours and etc. We have to write zoom functions in PHP and HTML that interacts with the map to zoom into countries and NUTS levels. We also will use JavaScript to interact with the map.

With DIY Map solution, we have to work with XML file, which contains the map data such as colour range, links and etc. We have to make a script in PHP that gets the data from PostgreSQL and writes the changes to the XML file. Then the DIY Map shows the results. The maps of Europe and zoom functions are already hard coded into the Flash.

We will research to find out which of the mentioned GIS solution is fully usable for EIAO or not, and will come up with a recommendation.

5 Requirements specification

Requirement 1: Map

Description: Implementation of a map of Europe, which can be shown on the web

Testing: Implement and see if it shows in a browser

Requirement 2: Database

Description: Get the data of the scores from the EIAO's database

Testing: Use a PHP script that can get the data

Requirement 3: Grades of accessibility

Description: Different scores are divided into different grades from 1-5, each grade is shown in different colour on the map.

Testing: The PHP script parses the data into integer scores from 1-5.

Requirement 4: Colour

Description: Showing different colours on the map, according to different grades of the countries

Testing: An XML file is loaded into the flash application, stores the data of grades and modifies the map.

Requirement 5: Drop down

Description: This function makes it easy for the users to find out the countries and cities they are interested in.

Testing: Use a JavaScript to carry out the function

The DIY Map which we used for implementing is free for personal, educational, and non-commercial use. Besides, the Apache, XML, PHP and JavaScript are all open sources, so that there is no concerning matter.

6 Design

The design of this GIS system is quite simple, we query what we need from the database in a PHP file. The PHP script reads the data and parses it into scores, and changes the tags in the XML file which is used to load the data into the flash application. In the XML one can add points with links and hover texts, change colours on the different countries and so on. The script that queries the database and updates the XML file is not connected directly to the rest of the application, since the database is only updated once a month, it's sufficient to run the script when this is done. Seeing the database is so huge it takes several minutes to do the queries, the separation of these two parts was a necessity.

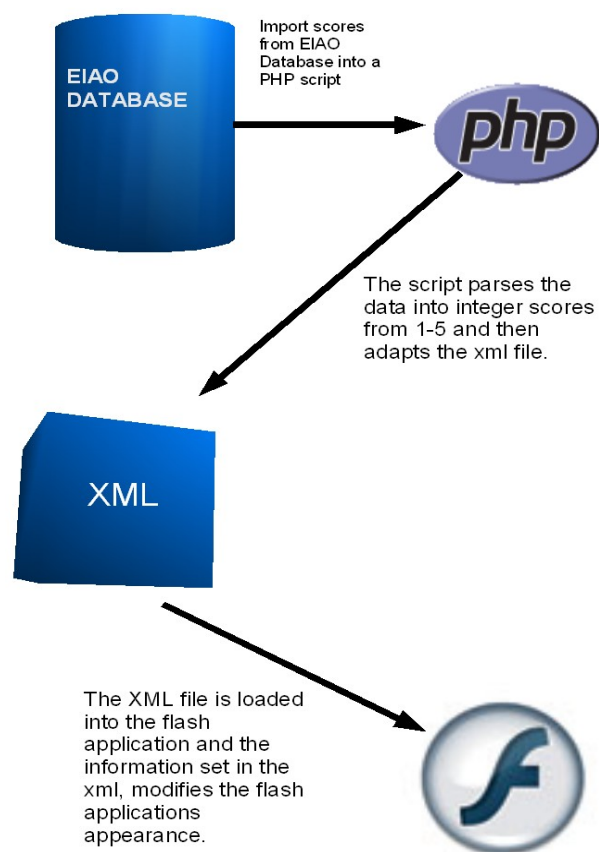


Figure 1:

The sketch of the solution with DIY Map

6.1 Development Software

HTML-KIT ^[10]

HTML-Kit is simple, free and very good tool to write HTML/XML/CSS files in, very nice coding environment with highlighting of the tags and so on.

ZEND STUDIO 5 ^[11]

We used Zend Studio 5 to make and edit the PHP parts of the project. This software is made by some of the key contributors to the PHP engine and is one of the best PHP development platforms on the market.

7 Implementation

The implementation of this prototype was fairly uncomplicated. This can easily be seen in the design part as well. We used PHP to dynamically change the XML-file, this created some implementational challenges such as having to use the DOM functions from PHP libraries to get it done, but maybe the biggest advantage of PHP is its on-line documentation and example base. This makes it easy to find literature about how things are done. The key thing happening in the scripts, is traversing the hierarchy of the XML file to find the grade value of each country, we then send this node into a function. This function removes the current grade and replaces it with a new one which has been fetched from the database.

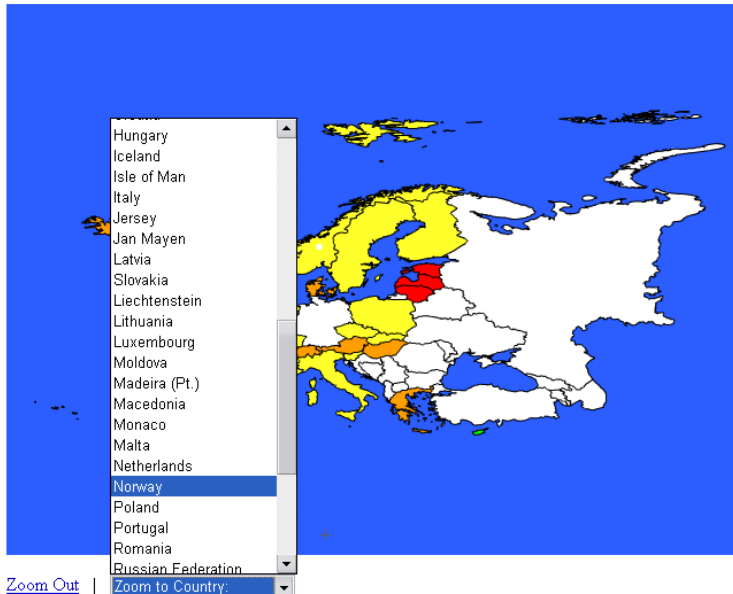


Figure 2:

Europe map with accessibility measure values shown as scorecards.

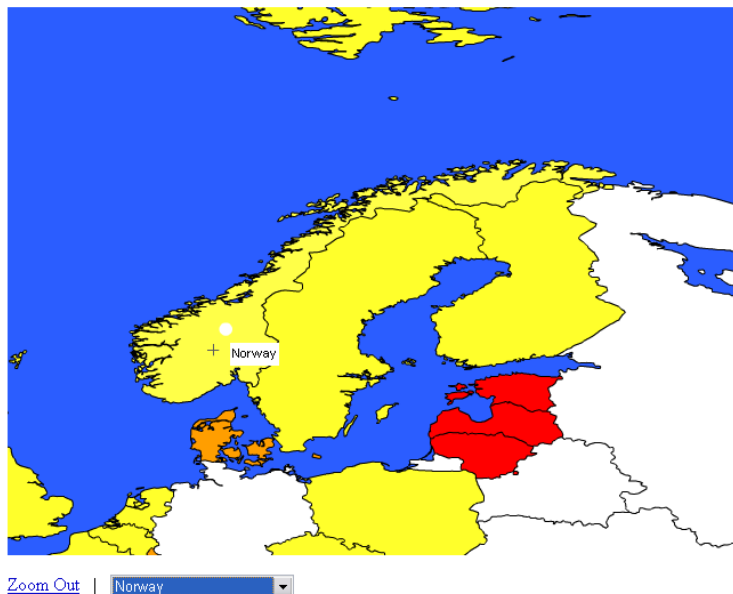


Figure 3:

To zoom into countries, we can click on the map or chose from the drop down box.

```

0.435992335753781002025001903508022005264625899469494902165700281965323812379057326069386293479040691839022819842340247644812872934619
0.548574883982489885337085278274572294474634808803132178312386952312351952368923415995762651071771211054670418243975512976197893571030
0.482529717272329023164522015167745532283373763202025473885789516683037649117188581940852302916411093373939151688720138040847782749067
0.481863335071421727071400115639945472492274978960918563292377392420633580691165470697056552769332871122287542675621807575225825078125
0.390489805037941142209614427637031606672342528628789718017897456756847961007004353363073007482869615551661951476480772229228162668607
0.360197667268564806010965453819571659031637904871713823296375719136146398680137058559608763627356298875285453915569265479538487946756
AU ----> DONE
BE ----> DONE
SZ ----> DONE
CY ----> DONE
EZ ----> DONE
DA ----> DONE
DONE
EN ----> DONE
SP ----> DONE
DONE
FI ----> DONE
FR ----> DONE
GR ----> DONE
HU ----> DONE
EI ----> DONE
IC ----> DONE
IT ----> DONE
LS ----> DONE
LH ----> DONE
LU ----> DONE
LG ----> DONE
MT ----> DONE
NL ----> DONE
NO ----> DONE
DONE
DONE

```

Figure 4:

This is the output we get when we run the PHP script and use it to control if everything is imported to the DIY map.

```

function update($Xml, $node, $newgrade)
{
    //replaces current grade with new grade from database...
    $NestedNode = &$node->childNodes;
    foreach ( $NestedNode as $nested )
    {
        $node->removeChild($nested);
    }

    $node->appendChild($Xml->createTextNode($newgrade));

    echo "DONE <br />";
} //end function

```

The function gets the XML document, the node we want to alter and the new grade in as parameters. Inside the function we find the child nodes of the node and remove this, and then we append a new child node containing the new grade. This function makes it easy to maintain the grade and thus colours of the map without having to manually set each one of them.

It's also added points on the map that can be clicked to for instance get a more detail text based overview of the country the point is located in. These points are located on the map by setting coordinates.

The EIAO Data Warehouse Database uses the ISO country codes to identify the countries, while the flash application uses the First FIPS to identify the countries on the map. This we solved by adding the FIPS codes into an array in the same order as they would have been listed in ISO country codes. With the Flash solution we chose, the implementation didn't bring us to many obstacles to pass.

8 Evaluation and testing

The testing phase was on this project like the implementation phase a quite simple task, but as always it's important to do thorough tests no matter how intricate the solution may or may not be.

We've been testing during the entire development process. We developed the different parts for themselves and put them together at the end. The first thing we made was to get the map up and running in a PHP web page together with the JavaScript modifications. Once this was tested and working we started working on the database and the XML script. The XML script simply removes the old value and replaces it with a new one, this can simply be tested by running it and either look at the map or in the XML file to see if it has changed. The database script was time consuming to test due to the time it took to query the database, but the script itself is quite simple and is working properly. When we got all working, we put it together and did some tests to see if they worked together. Once the scripts were integrated with each other and tested, we had fulfilled the functional requirements we had set.

We've reached all the goals set in the requirements and things are tested and working like they should. So with this solution we get a better and more functional GIS in a fraction of the time we would have used to do the same on MapServer.

9 Discussion

9.1 *Project outcomes*

The focus of the project changed during the project period to less implementation and more research. This was because it was impossible to get hold of a free map with NUTS support.

To start with we focused on MapServer solution and found out that it was hard way to solve the task. It had some lacks that made our programming hard and a lot of unnecessary code. We also found out that it was impossible to get hold of a map with NUTS support. MapServer was stiff and took some time to render picture as we zoomed in and out in the map. So it was meaningless to make a prototype with this when we had a similar solution with DIY Map. With DIY Map rendering went smoothly and quick and looked much better. One downside with this solution is that DIY Map is not a “real” GIS system compared to MapServer. But the Flash GIS solution is the easiest and best solution with most of the functionality that EIAO wants.

In future path of development the map should be connected to scorecards, so that every NUTS level value could be visible in a table beside the map. EIAO should also consider buying a map with NUTS support, to show the scorecards graphically on the map. It may be a lot of money spent for a small gain.

9.2 *Evaluation of the overall results*

DIY maps gives us a good solution to make a colour coded map of Europe. Development time, is a mere fraction in comparison to the time it would take in a big GIS system and the system renders the maps, zooms in and out substantially faster than what for example MapServer does. This is a result of client side calls, the entire flash file is loaded to the client, when the page is first loaded in his browser, so that all the other calls to the flash file can be done without contacting the server. The relevance towards EIAO using it will of course be a bit halted due to the lack of NUTS support for the maps. If you measure the effort put into development against the outcome, it's very clear that the DIY maps gives you the most for the time you put in.

For EIAO it's very important that the content they have on their websites yields good accessibility for all that are using the sites, since the point of the project is put focus on web accessibility in EU. With DIY maps this is an issue, since it's hard to adapt the information in the flash map to suit for example blind people.

With choosing the DIY maps solution the only methods we needed to develop was to port the Database data into the XML file that the flash uses to load the information. If we had gone with MapServer this would have been a much bigger task and since MapServer needs to contact the server every time the map renders, it would have been a very slow application, both with the rendering and especially because of the database calls. Since we didn't have access to EU NUTS maps neither on MapServer nor DIY, the obvious choice to make the prototype in this scenario was DIY maps, since it really outperforms MapServer in most areas.

EIAO as an organization use open source software. Thus we focused on making the project on open source solutions. There are several open source platforms to develop web solutions on, such as Java, Zope, PHP, RubyOnRails. With the project starting with MapServer we found that PHP seemed to be the best supported and document language out of those mentioned. When we later went over to the DIY Maps and flash, we decided to keep PHP as our scripting language, much due to the large database of documentation on-line. When using a web server to run the application on the obvious choice in most cases is Apache, it's a very good web server, which is easily configurable and thus very adaptable.

10 Conclusion

In this project we have found out a lot about the capabilities of different GIS systems, pros and cons the different platforms available presents. We've learned that though MapServer is a big and popular it might not be the best framework out there. The DIY that is a flash application shows a lot of potential, but if EIAO should use it, it might be best to write their own flash application from scratch. That way, they can tailor it to suit EIAOs needs specifically.

In the past more traditional GIS, like MapServer or ArcGIS has been the only way to build a GIS system, but now we can see that Flash also is beginning to establish itself as a viable solution to build a good GIS. The use of Flash in general has been in small scale due to its need of bandwidth to work smoothly, but with the broadband «boom» it is now more and more common to use Flash on websites. So it's highly likely we will see more GIS solutions in Flash in the future.

Maps are an expensive commodity, if the goal is to make a completely free GIS system you can't expect to get a hold of very detailed or special maps for free, if you're not prepared to build them from scratch yourself. So even though the development tools etcetera is free, one must most probably expect to pay to get a good map to use as the basis for the GIS.

To conclude, we will say that the best way to build a GIS today would be to make it as a flash application, then you get client side calls, everything is loaded at once, and given you get a good implementation of a map you got a lot more opportunities with Flash than with for example MapServer. Though the accessibility issue, would probably be the biggest issue for EIAO to overcome if they go for this, because an organization who measure others accessibility should show a good example and be totally accessible on their websites. So if this can't be solved sufficiently, they will probably have to go for a more traditional GIS that is based on generated photos which is easier to add for instance help text that can be read by screen readers and so on.

Appendix

A0Glossary and abbreviations

Term	Explanation	URL
W3C	World Wide Web Consortium	http://w3.org
Apache	Web server application	www.apache.org/
EIAO	European Internet Accessibility Observatory	http://eiaonet.net/
NUTS	Nomenclature of Territorial Units for Statistics	http://europa.eu/scadplus/leg/en/lvb/g24218.htm
UWEM	Unified Web Evaluation Methodology	http://www.wabcluster.org/uwem1/

A0References

- [1] <http://mapserver.gis.umn.edu/> (2006-09-05)
- [2] <http://www.php.net/> , <http://www.w3schools.com/php/default.asp> (2006-11-17)
- [3] <http://httpd.apache.org/> (2006-11-13)
- [4] <http://backspace.com/mapapp/> (2006-10-26)
- [5] http://www.w3schools.com/flash/flash_intro.asp,
http://en.wikipedia.org/wiki/Adobe_Flash (2006-11-16)
- [6] http://www.w3schools.com/xml/xml_what_is.asp (2006-11-17)
- [7] <http://www.postgresql.org/> (2006-11-18)
- [8] http://www.directionsmag.com/article.php?article_id=208&trv=1 (2006-10-28)
- [9] http://www.w3schools.com/js/js_intro.asp (2006-11-17)
- [10] <http://www.chami.com/html-kit/> (2006-10-30)
- [11] <http://www.zend.com/> (2006-10-30)
- [12] <http://earth.jscc.ru/tenerif/?lang=en> (2006-11-20)
- [13] <http://geodata.grid.unep.ch>,
<http://mapserver.gis.umn.edu/gallery/GEO%20Data%20Portal> (2006-11-20)
- [14] <http://www.globalis.no/land/nato> (2006-11-21)
- [15] <http://www.icasualties.org/flashmap/> (2006-11-21)
- [16] <http://www.sitestats.com/home/examples.php> (2006-11-21)
- [17] <http://www.wabcluster.org/uwem1/> (2006-11-21)
- [18] <http://europa.eu/scadplus/leg/en/lvb/g24218.htm> (2006-11-21)