

DEVELOPMENT OF A WEB GEOSERVICES PLATFORM FOR SCHOOL OF ENVIRONMENTAL SCIENCES, MAHATMA GANDHI UNIVERSITY, KERALA, INDIA

Sumith Satheendran S.¹, John C. M.¹, Fasalul Faseeh K.¹ and Aanisa K. M.^{1,2}

¹Dr. R. Satheesh Centre for Remote Sensing and GIS, School of Environmental Sciences, Mahatma Gandhi University, Kottayam, Kerala, India – 686560 rsgismgu@gmail.com

²Tata Institute of Fundamental Research, Mumbai kmaanisa@gmail.com

KEY WORDS: Web GIS, Map server, Data sharing, Interoperability, Geoservices, Internet

ABSTRACT

Web geoservices is the obvious graduation of Geographic Information System in a distributed environment through a simple browser. It enables organizations to share domain-specific rich and dynamic spatial information over the web. The present study attempted to design and develop a web enabled GIS application for the School of Environmental Sciences, Mahatma Gandhi University, Kottayam, Kerala, India to publish various geographical databases to the public through its website. The development of this project is based upon the open source tools and techniques. The output portal site is platform independent. The premier webgis frame work 'Geomoose' is utilized. Apache server is used as the Web Server and the UMN Map Server is used as the map server for this project. It provides various customised tools to query the geographical database in different ways and search for various facilities in the geographical area like banks, attractive places, hospitals, hotels etc. The portal site was tested with the output geographical database of 2 projects of the School such as 1) the Tourism Information System for the Malabar region of Kerala State consisting of 5 northern districts 2) the geoenvironmental appraisal of the Athirappilly Hydroelectric Project covering the entire Chalakkudy river basin.

1. INTRODUCTION

Web GIS is the obvious graduation of GIS on a distributed environment through a simple browser. It enables organizations to share domain-specific rich and dynamic spatial information over the web. The web GIS is an extension and application of client/server computing, where the geospatial data is accessible in a shareable environment (Karnatak et al., 2007). During the past decade or so, GIS has been widely used in various types of business, government and university projects. Large volumes of geographic data, which are valuable to various organizations, have been accumulated but in the dispersed and heterogeneous manner. In order to fully utilize the available spatial data efficiently and effectively, GIS has to play a critical role, not just in disseminating the raw data, but also in providing information and offering value-added services to potential users. GIS-enabled environments have to be available to the public. It has been widely recognized that future developments in GIS will centre on web-based GIS, accessing geospatial data and conducting geospatial analyses on the Web. While WebGIS is gaining in popularity, an important issue should be considered: share and interoperate the heterogeneous spatial data. The emergence of Web Services provides a new WebGIS framework for dealing with this issue. Spatial information plays a more important role in the form of services in socio-economics (Fang and Min, 2009).

In essence, Web Services can package data, message and behaviour based on unified standards, without considering the specific application environment. Different systems can achieve seamless communication, data sharing and interoperation across platforms and languages based on Web Services framework. In fact, the ultimate goal of WebGIS is to realize a real sense of sharing and Interoperability in GIS fields and guide GIS into the service stage from sharing stage. Web Services provide a very good solution in its completely open standard. OGC has also provided a series of standardized interface specifications to support geospatial data sharing and

interoperability among web-based GIS systems (Fang and Min, 2009).

Web GIS finds applications in a wide-range of internet based applications such as environmental sustainability indicators, global population predictions, water sustainability, hazard vulnerabilities, climate and disaster monitoring, education, banking, armed forces etc. There are many applications that use online GIS like car navigation, tourism, house and planning management. Internet is a great asset in helping to disseminate knowledge derived from the GIS via the web-based GIS.

This project is an attempt to provide an overview of the current status of Web Based GIS by looking into the different types of systems that have been implemented, its techniques and the trends in the industry

1.1 MapServer for Windows

MapServer is a popular Open Source project whose purpose is to display dynamic spatial maps over the Internet. Some of its major features include:

- Support for display and querying of hundreds of raster, vector, and database formats
- Ability to run on various operating systems (Windows, Linux, Mac OS X, etc.)
- Support for popular scripting languages and development environments (PHP, Python, Perl, Ruby, Java, .NET)
- On-the-fly projections
- High quality rendering
- Fully customizable application output
- Many ready-to-use Open Source application environments

In its most basic form, MapServer is a Common Gateway Interface (CGI) program that sits inactive on your Web server. When a request is sent to MapServer, it uses information passed in the request URL and the Mapfile to create an image of the requested map. The request may also return images for legends, scale bars, reference maps, and values passed as CGI variables (MapServer, 2014) (Figure. 1)

1.2 GeoMOOSE

GeoMOOSE is a Web Client JavaScript Framework for displaying distributed cartographic data. GeoMOOSE has a number of strengths including modularity, configurability, and delivers a number of core functionalities in its packages. GeoMOOSE is also very light weight for servers making it easy to handle a large number of users, with a large number of layers, and a large number of services without stressing a server.

The GeoMOOSE core is written using JavaScript and HTML. It is entirely possible to run GeoMOOSE with nothing more than a basic web server (Nginx, Apache, and IIS). But besides the basic client core, GeoMOOSE also comes pre-packaged with a number of built in services written in PHP. These services add the ability to perform drill-down identify operations, selection operations, and search data sets. If you have existing scripts that perform similar functions, GeoMOOSE can be tuned to work with those services, no matter which language they were written (Little, 2014).

2. MAIN WORK DESCRIPTION

This section explains what all things what we have done in the project. Mainly we have used two components.

- MapServer for windows with Apache web server.
- GeoMOOSE is a framework for displaying distributed cartographic data.

2.1 Map Server for Windows (MS4W)

MapServer is a popular Open Source project whose purpose is to display dynamic spatial maps over the Internet.

2.1.1 Map file: The .map file is the basic configuration file for data access and styling for MapServer. The file is an ASCII text file, and is made up of different objects. Each object has a variety of parameters available for it.

2.2 Installation

MS4W package is simply a zip archive containing MapServer, Apache, and all their dependencies. Add-on packages can later be downloaded and unzipped over existing MS4W folders.

Now, goto services in administrative tools in control panel and click on restart apache. Then type localhost in a web browser, it will display the mapserver page. Then type localhost/phpinfo.php in a web browser, then it will display php version information. This shows that mapserver and php is working properly.

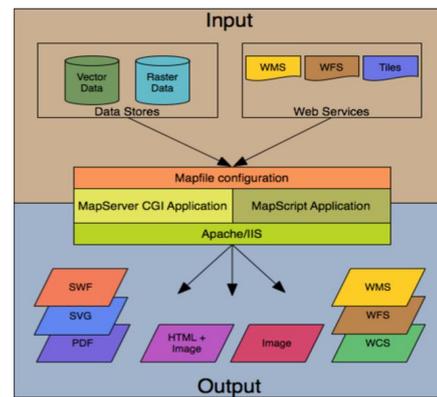


Figure.1 Architecture of MapServer

2.3 Why GeoMOOSE?

GeoMOOSE excels at creating a useful web-based GIS environment for those who need something that works from the first download. It can render, investigate, and even edit layers without the need to write a single line of code.

GeoMOOSE allows users to compare local data seamlessly with popular web mapping services such as Google, Bing, and MapQuest. These are some of the most common tasks performed with MapServer and GeoMOOSE includes real, working, and practical examples.

Beyond rendering, gaining information on a dataset is an important everyday use of a Web GIS. GeoMOOSE includes all of the basic tools for traditional selection, identification, and searching on a dataset. However, GeoMOOSE's functionality is not limited by those services. To power that functionality we have developed a power service-based architecture that users can use to create their own custom scripts in any language.

Finally, GeoMOOSE is designed with lightweight-extensibility in mind. There are a myriad of custom configuration options, a highly flexible services architecture, and a powerful user-extension model. GeoMOOSE can create powerful web GIS applications for those with no programming knowledge and provides an astounding platform for those willing to write some code (Little, 2014).

3. PROJECT DESIGN

The present project was designed to gain better understanding of the actual entity to be built. The following flowcharts show the design of the entire project with detailed idea of each component tools.

4. FLOWCHARTS

The specific flowcharts of each customized tools are given below. The Figure 2 is represent the flow diagram of identify tool. This tool is used to identify the particular features in the Malabar region. Figure 3 represents the flow diagram of select tool. This tool is used to select the particular features from the Malabar geodatabase. and finally the figure 4 represents the flow diagram of search tool. This tool helps to search a particular facility in the Malabar region.

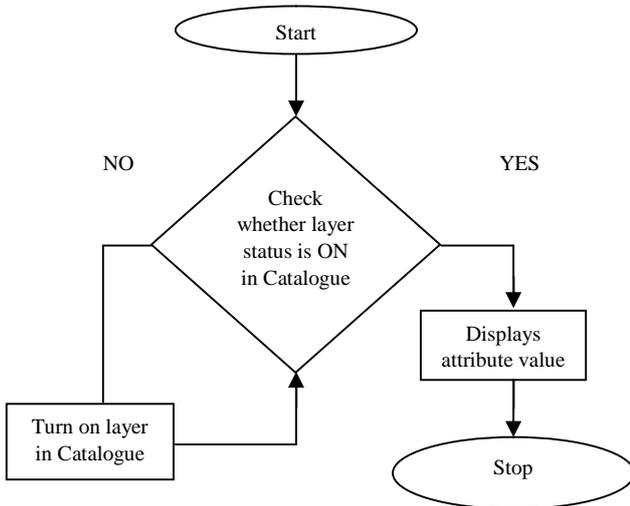


Figure2. Flow chart of identify tool

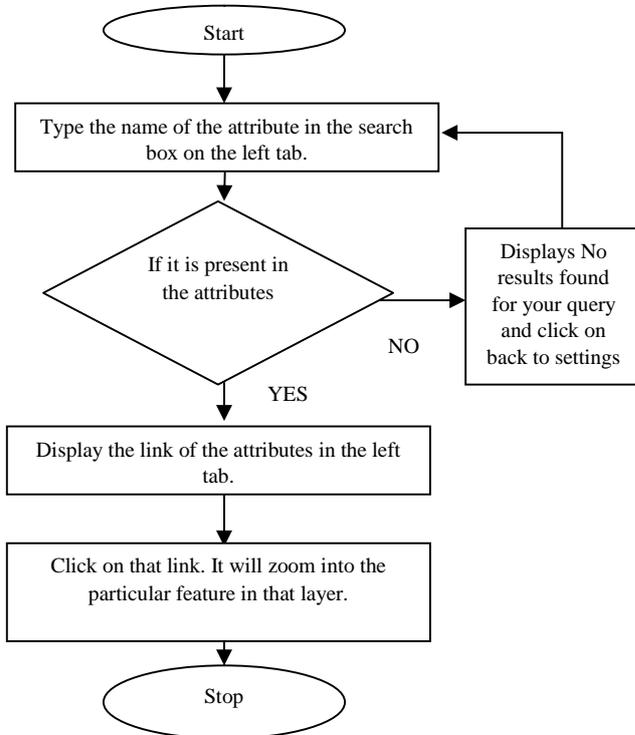


Figure3. Flowchart of select tool

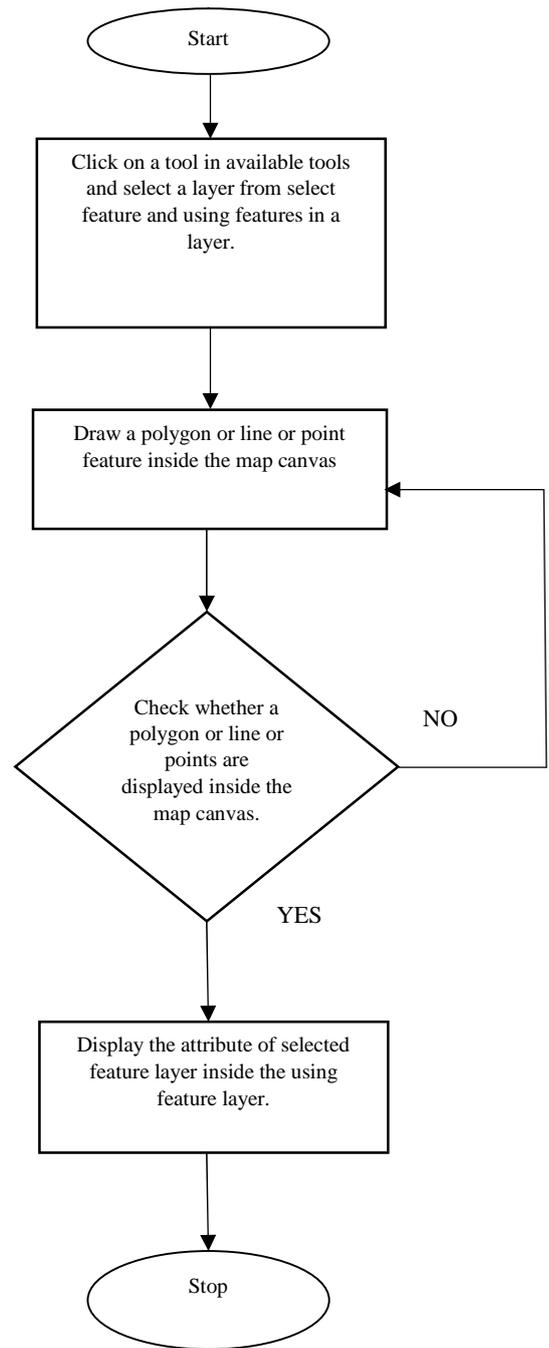


Figure4. Flowchart of search tool

5. CODING

The goal of coding is to translate the design of the system into code in a given programming language. The programs should not be constructed so that they are easy to write, but so that they are easy to read and understand. Examples of coding are given below:

5.1 Mapbook.xml

The <mapbook> tag contains 6 child elements as discussed in above chapter. The code for the above is described below.

5.2 Configuration

The configuration tag is used to specify user interface startup settings and other application settings. This is done within the <configuration> ... </configuration> section of the mapbook.xml. The configuration tag has <param> children. <param> elements have a simple format, a "name" attribute which specifies the interface setting to change and then CDATA that specifies the values.

5.3 Map-source

The map-source tag is used to specify a single or collection of layers in GeoMOOSE. These are all in the <map-source> </map-source> section and later referenced in the <catalog></catalog> section.

All map-source types support two children:

<param> – The 'param' child will add or change parameters sent over the URL to a <map-source>.

<layer> (required) – At least one layer child is required for each map-source.

5.4 Mapserver

This type of layer is meant to communicate with the default mapserver as specified in config.js. The basic additional parameters reference to use a Mapserver layer are:

5.4.1 Layer – The name of layers to be referenced by the map source. This name must be referenced in the mapserver file. The layer name could also be "all" and all layers within the mapserver file will be used.

5.4.2 File – Layers of type 'mapserver' require a value <file> child specifying the location of the mapfile on disk. Mapfiles can be specified with relative paths in <file> tags.

5.5 Service

The service element is the key to GeoMOOSE's interoperability. All functions are a type of service (identify, select, Birdseye view, etc.). Services have attributes to define some of its behaviour:

5.5.1 Name - This is the internal name of the service.

5.5.2 Display - Sets whether the service will be displayed while "launching." It is useful to set this to false if there is no text-entry required for the service.

5.5.3 Title - This title will be displayed automatically in the input tab.

5.5.4 Target - If set then this attribute will display the results in a different window.

5.6 Layer-controls

As of GeoMoose the only layer control available are popups.

5.7 Catalog

The GeoMOOSE Catalog, or Catalog, is the layers listing found in the table of contents on the user interface. The catalog is represented by the <catalog> child inside of the mapbook.

5.8 Toolbar

GeoMOOSE now has a better defined toolbar. Just as the catalog and layer definitions have become more separated, so too has the toolbar and service definitions. The toolbar is specified using the <toolbar> child.

6. RESULTS AND DISCUSSION

Development of Web Enabled Geographical Information System performed all of its features. No system could be useful if it does not produce the required output in the specific format. The output generated or displayed by the system under consideration is observed as the users about the format required. The figures 5,6and7 show the screen shots of the successful testing of Map server, PHP and GeoMoose respectively. The Figure 8 shows that the screenshots of identify tool. The identify tool will allow you to find out more information about a geographical feature. Click anywhere on the map and the identify tool will display the location, the section you selected. The Figure 9 shows that the Screenshots of Select tool. If you would like to find out information about more than one feature at a time, you may use the select tool. The select tool will allow you to select many features by drawing a line, a polygon, or a box. Furthermore, you find out what features are within a certain distance of your drawing by entering in a value in "buffer selection." Basic information about all the selected features will be displayed. The Figure 10 shows that the screenshots of Search tool. Search tool will allow the user to search by various attributes on a map. The Figure11 shows that Screenshot for displaying Layers in Impact Study of Athirapilly Hydro Electric Project

Here the outputs actually bothered or tensed about are

1. The layer will be displayed inside the map canvas.
2. Customized tools are tested and work properly.

All the above required results obtained correctly and functioned successfully.

MS4W - MapServer 4 Windows - version 3.0.6



Introduction

Welcome to MS4W v3.0.6, the MapServer package for Windows, developed by Gateway Geomatics. MS4W is designed to get you up and running with MapServer with very little configuration.

There are some important notes about MS4W:

- Since it is not a true installer, it will not install to any location. It **MUST** be installed at the root of one of the drives on your system (e.g. C: or D:).
- Pre-configured add-on packages for MS4W can be downloaded from the [MS4W Downloads](#) page.
- The MS4W add-on packages must be extracted at the same root as the base installer, and files must be overwritten.

Documentation

Please view your local [README](#) file for full documentation about your MS4W package.

Features

MS4W contains default installations of Apache, PHP, MapServ, MapScript and some sample applications. It is structured in such a way as to allow upgrading individual components without affecting the install. The base installer package is pre-configured with the following software:

- Apache 2.2.22 (with OpenSSL 0.9.8r)
- PHP 5.4.3
- GDAL 1.9.1
- MapServer CGI 6.0.3 at [cgi-bin/mapproxy.exe](#)
- MapScript 6.0.3 flavours:
 - php_mapscrip.dll at [ms4w\Apache\php\ext](#)
 - CSharp mapscrip at [ms4w\Apache\cgi-bin\mapscripcsharp](#)
 - Java mapscrip at [ms4w\Apache\cgi-bin\mapscripjava](#)
 - Python mapscrip at [ms4w\Apache\cgi-bin\mapscrippython](#)

Figure5. Screenshot for Testing MapServer

System	Windows NT FASALULFASEEH 6.2 build 9200 (Unknown Windows version Home Premium Edition) i586
Build Date	May 14 2012 15:14:00
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscrip\nologo configure.js "--enable-cli" "--with-pgsql" "--with-curl" "--with-openssl" "--with-mysql" "--with-mysqli" "--enable-pdo" "--with-pdo-pgsql" "--with-pdo-odbc" "--with-pdo-sqlite" "--with-sqlite3" "--enable-dbase" "--with-gettext=D:\build\mapserver-buildkit-2008\gettext-runtime-0.17-source\MSVC9\libintl_static\Release" "--enable-mbstring" "--enable-mbregex" "--without-t1lib" "--enable-soap" "--with-mcrypt" "--enable-xml" "--with-ldap"
Server API	CGI/FastCGI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\ms4w\Apache\cgi-bin\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini	(none)

Figure6. Screenshot for testing php

MS4W contains default installations of Apache, PHP, MapServ, MapScript and pre-configured with the following software:

- Apache 2.2.22 (with OpenSSL 0.9.8r)
- PHP 5.4.3
- GDAL 1.9.1
- MapServer CGI 6.0.3 at [cgi-bin/mapproxy.exe](#)
- MapScript 6.0.3 flavours:
 - php_mapscrip.dll at [ms4w\Apache\php\ext](#)
 - CSharp mapscrip at [ms4w\Apache\cgi-bin\mapscripcsharp](#)
 - Java mapscrip at [ms4w\Apache\cgi-bin\mapscripjava](#)
 - Python mapscrip at [ms4w\Apache\cgi-bin\mapscrippython](#)
- Commandline Utilities
 - execute [ms4w\setenv.bat](#) to use the utilities
 - [MapServer Utilities](#)
 - physically located at [ms4w\tools\mapserv\](#)
 - [GDAL/OGR Utilities](#)
 - physically located at [ms4w\tools\gdal-ogr\](#)
 - [PROJ.4 Utilities](#)
 - physically located at [ms4w\proj\bin\](#)
 - [Shapelib Utilities](#)
 - physically located at [ms4w\tools\shapelib\](#)
 - Shpdiff Utility
 - physically located at [ms4w\tools\shpdiff\](#)
 - Shp2tile Utility
 - physically located at [ms4w\tools\shp2tile\](#)
 - [AVCE00 Utility](#)
 - physically located at [ms4w\tools\slavce00\](#)
- OWTChart utility:
 - [introduction](#)
 - [owtchart.exe](#)
 - [gifsamples.exe](#)
- OGR/PHP Extension ([documentation](#)) ([phpinfo](#))

Applications

There are currently 2 MS4W add-on packages installed.

GeoMOOSE Demo Applications, Version 2.6

- [GeoMOOSE 2.6 Demo Application](#)
- [GeoMOOSE Web Site](#)

Figure7. Screenshot for testing GeoMoose

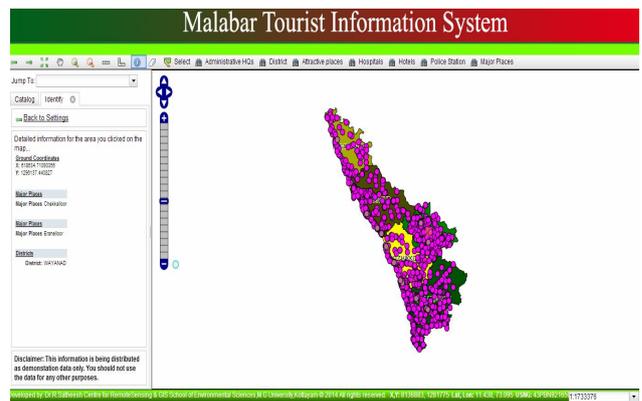


Figure 8. Screenshot for testing Identify tool in Malabar Tourist Information System

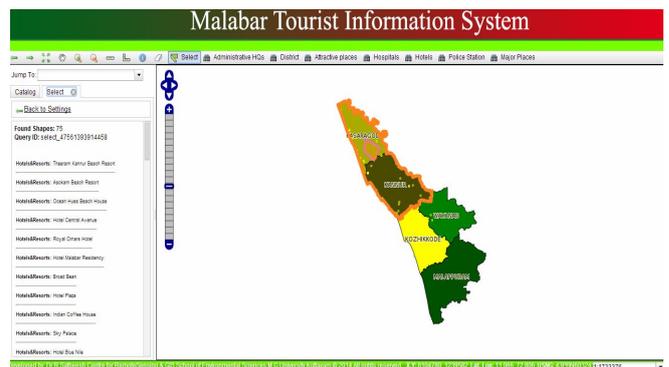


Figure9. Screenshot for testing Select tool in Malabar Tourist Information System

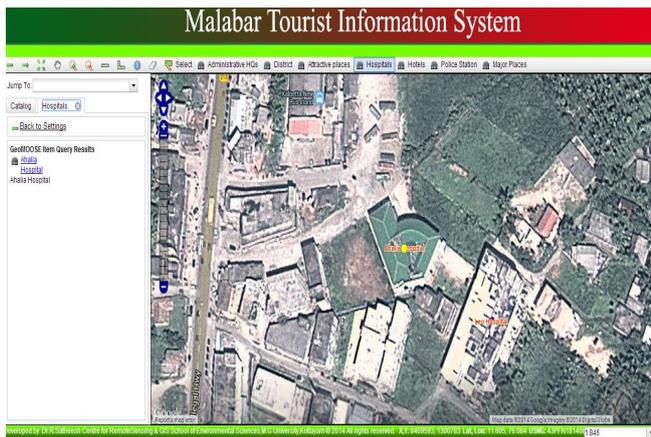


Figure10. Screenshot for testing Search tool in Malabar Tourist Information System

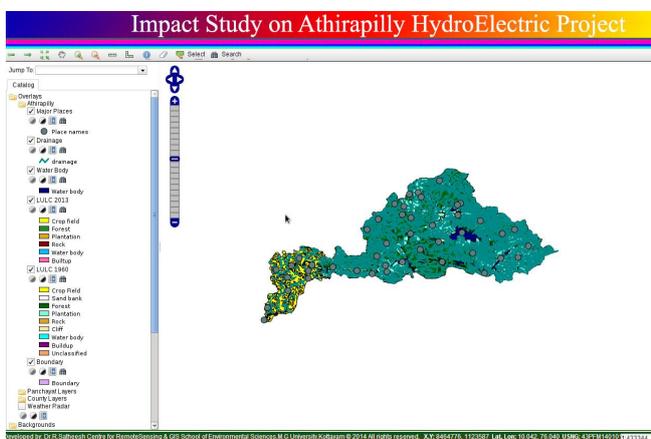


Figure11. Screenshot for displaying Layers in Impact Study of Athirapilly Hydro Electric Project

7. CONCLUSION

The system has been developed using the GeoMOOSE java script builtin framework for distributed cartographic data and MapServer for windows. All the modules are tested separately and put together to form the main system. Finally the system is tested with the following student projects:

1. Impact Study of Athirapilly Hydroelectric Project
2. Malabar Tourist Information System

Thus the system has fulfilled the entire objective identified. Every effort has been made to produce the new website as a secured and user friendly GIS website. We tried our best to make the system exception free. Many trial runs of the system have been conducted and we found excellent results. We look forward for upgrading our project one that provides WMS service and the user can download the spatial as free of cost. The feature was not added due to time limitation.

8. FUTURE SCOPE

The website of the school can have the following enhancement in future.

8.1 WFS Support

The Open Geospatial Consortium Web Feature Service Interface Standard (WFS) provides an interface allowing requests for geographical features across the web using platform-independent calls. One can think of geographical features as the "source code" behind a map, whereas the WMS interface of this project return only an image, which end-users cannot edit or spatially analyze.

8.2 Spatial Warehousing

Map Server can use a wide variety of sources of data input. One of the solutions growing in popularity is to use spatially enabled databases to store data, and to use them directly to draw maps for the web.

8.3 Database Connectivity

This project can be easily upgrade using PostGIS adds support for geographic objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS).

8.4 Visualization Techniques

The JavaScript library called d3js can be used to visualize the spatial data in various modes.

ACKNOWLEDGEMENTS

The Director, Students and all other staffs of School of Environmental Sciences M. G. University, Kottayam, Kerala, South India

REFERENCES

- Fang YIN and Min Feng., 2009. A WebGIS Framework for Vector Geospatial Data Sharing Based on Open Source Projects. In Nanchang, P. R., Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA '09). Academy Publisher: pp. 124-127.
- Karnatak H. C., Sameer, Saran., Karamjit, Bhatia and P.S. Roy., 2007. Multicriteria Spatial Decision Analysis in Web GIS Environment. *Journal of Geoinformatica*. 2007 (11): pp. 407-429.
- MapServer. 2014. MapSever 6.4.0. - Documentation on mapserver. The MapServer Team. <http://www.maptools.org/ms4w/index.phtml?page=downloads.html> (10 February, 2014)
- Little. 2014. GeoMOOSE Documentation Release 2.6.1. Dan "Ducky" Little. <http://docs.geomoose.org/2.6/GeoMOOSE.pdf> (24 January, 2014)