Journal of Geographic Information System, **2014, 6, 30-39**
Published Online February 2014 (http://www.scirp.org/journal/jgis)
http://dx.doi.org/10.4236/jgis.2014.61004

Mapping Spatial Data on the Web Using Free and Open-Source Tools: A Prototype Implementation

Sunil Pratap Singh[*], Preetvanti Singh
Department of Physics and Computer Science, Dayalbagh Educational Institute, Agra, India
Email: [*]sunil_pratap@rediffmail.com

Received November 5, 2013; revised December 5, 2013; accepted December 12, 2013

ABSTRACT

There is a growing need for web-based geographic information systems for easy and fast dissemination, sharing, displaying and processing of spatial information. The tremendous growth in the use of web and open-source geospatial resources has sparked development of web-based spatial applications to address multidisciplinary issues with spatial dimensions. This paper presents the integration of open-source geospatial tools and web technology to visualize and interact with spatial data using web browser. The goal of this paper is to implement a prototype system for web-based mapping by providing step-by-step instructions in order to encourage the eager developers and interested readers to publish their maps on the web with no prior technical experience in map servers. The implementation of mapping prototype shows the utilization of open-source geospatial tools which results in a rapid implementation with minimal or no software input cost.

KEYWORDS

Open-Source Technology; Web-Based Mapping; Web Technology; Geographic Information System; Spatial Application

1. Introduction

The geographic information systems (GIS) were initially adopted by users in the government sectors, but are now rapidly growing and have profound implications in a large number of application domains including planning [1], marketing [2], retail [3], transportation [4], traffic [5], tourism [6], natural resources [7], real estate [8], agriculture [9] and healthcare [10]. These systems can be valuable planning tools to assist in the development and administration of scientific programs based on geographically identified needs [11]. Linking location to information is a process that applies to many aspects of business such as choosing a site, targeting a market, planning a distribution network or delivery route, drawing up sales territory and allocating resources [12]. The companies, communities and individuals are realizing that they must have geographical information to understand their world.

The tremendous growth in popularity of the internet and the growing public interests in accessing online geo-spatial information have given a move to develop web mapping applications. The web is playing a huge role in spatial application development mainly because of advantages such as platform independency, reduction in distribution costs and maintenance problems, ease of use and widespread access. The most important power of web-based spatial application is the capability to publish and share geo-spatial information on the web which jointly allows information to be exchanged in a rapid and efficient manner thereby helping individuals make important decision quicker. The geo-spatial information not only includes maps or locations of landmarks, but multiple attribute data, socioeconomic data, aerial photographs, satellite images, etc., which may have static or dynamic characteristics [13].

This paper is aimed to provide a method for building a new web-based spatial visualization prototype by using a

[*]Corresponding author.

combination of open-source geospatial packages and Microsoft .NET platform. The following sections specifically document the development and implementation of a prototype system by adopting web-based client-server computing technology.

2. Materials and Methods

2.1. Open-Source Geospatial Tools

The commercial software are predominantly “closed source” which means the code cannot be accessed or modified and they may, in some cases, come at a relatively high price with ongoing licensing fees to maintain their use. These systems are thus often priced out of the reach of the resource-constrained areas, particularly within developing countries, which lead them into seeking open-source alternatives [14]. The wide spread free of cost availability, freedom to read, redistribution and modification facilities have made the open-source packages as feature rich alternatives to proprietary software packages [15]. These packages can help the implementation of mapping and spatial analysis tools in a large number of private and public firms, especially in those small organizations that cannot afford proprietary software’s cost, complexity, training cost and special requirements. The market of open-source geospatial packages is growing [16] and the quality of these packages is improving with the strong and free collaboration between a large numbers of developers from all around the world.

The open-source project offerings cover the spectrum of tools that interact with geospatial data. The interaction occurs on several fronts: creating, converting, manipulating, analysing or visualizing the geospatial data. These are the common categories of tools or the functional components of open-source geospatial tools. Figure 1 represents the organization of these tools around the geospatial data and some of the well-known geospatial tools are listed in Table 1.

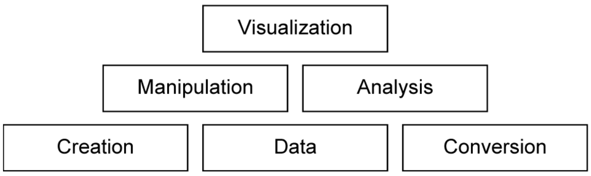


Figure 1. Functional components of open-source tools [17].

Table 1. List of open-source geospatial tools [18].

Category	Open-Source Geospatial Tools
Data Creation	Quantum GIS, OpenEV
Data Conversion	GDAL/OGR, AVCEOO, shp2pgsql (PostGIS)
Data Manipulation	GDAL/OGR, PostGIS, GEOS
Data Analysis	GRASS-GIS, PostGIS
Data Visualization	UMN MapServer, GeoServer, Degree, MapGuide

2.2. Microsoft .NET Platform

A range of development technologies is available to develop web based mapping applications. Microsoft’s .NET is a revolutionary software programming architecture aimed to create web based applications and offers object-oriented support. The use of object orientation in web applications has increased in recent years since applications are well structured, scalable to new changes and offer better operating performance. The Microsoft .NET Platform provides all the tools and technologies that are needed to build web applications. It exposes a language independent and unified programming model. Its main tools are .NET Framework and Visual Studio .NET (Figure 2).

2.2.1. .NET Framework

The .NET Framework is a technology that supports building and running the next generation of web services and applications. It is a managed execution environment that provides a variety of services (memory management, common type system, language interoperability, version compatibility) to its running applications. Its main components include [19]: the common language runtime (CLR), which is the managed execution environment that handles memory allocation, error trapping, and interacting with the operating-system services; the .NET Framework Class Library, which is an extensive collection of programming components and application program interfaces (APIs) for all major areas of application development; ASP.NET, a server side web technology which supports creation of dynamic web applications, websites and web services.

The advantages offered by the .NET Framework include shorter development cycles (code reuse, support for multiple programming languages), easier deployment, fewer data type related bugs due to integral type safety, reduced memory leaks, and, in general more scalable, reliable applications.

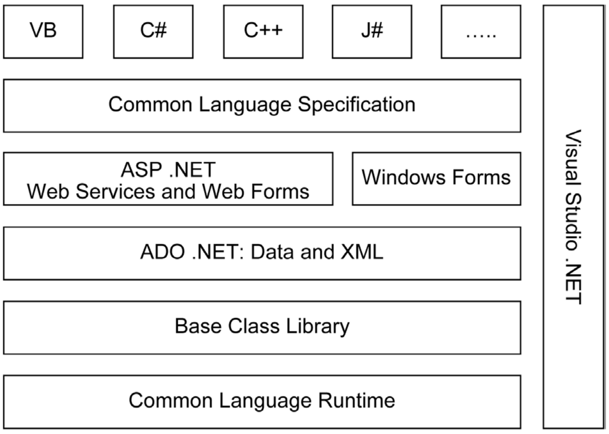


Figure 2. Microsoft .NET platform architecture [19].

2.2.2. Visual Studio .NET

The Microsoft Visual Studio .NET is an integrated development environment (IDE) to develop console and graphical user interface applications along with web applications, web services, websites and windows forms. It supports different programming languages (C#, C++, Visual Basic, J#, etc.) by means of language services, which allow the code editor and debugger to support nearly any programming language, provided a language-specific service exist. The built-in languages include Visual C++, Visual Basic, Visual C# and Visual J#. It also supports XML/XSTL, HTML/XHTML, JavaScript and CSS. It has comprehensive tools (drag-and-drop form designer, IntelliSense, dynamic help) for rapidly building and integrating applications which increases developer's productivity.

The Microsoft Express set of free software applications offer a way around budget limits, as all of the software are free, albeit somewhat restricted. Visual Web Developer 2008 Express is a .NET developer's integrated development environment (IDE) provided for free by Microsoft. It is a lightweight version of Microsoft Visual Studio product line. Figure 3 shows the starting screen of Visual Web Developer 2008 Express. The idea of Express edition is to provide streamlined, easy-to-use and easy-to-learn IDEs for users other than professional software developers. It allows web development with ASP.NET using drag-and-drop user interface designer, enhanced code editor, support for other web technologies (e.g., CSS, JavaScript, XML) and integrated design time validation.

2.3. Prototype Development

2.3.1. Preparation of Geospatial Data

The city of Taj, Agra in India is selected as a pilot area to create the prototype spatial mapping system. The geographic location of the scope lies within 27.11˚ latitude north and 78.20˚ longitude east and is covered in 1:50,000 scaled sheets 54E/16–54I/4 of Survey of India.

The data that is stored in spatial information system can be thought of as a digital representation of real world objects. These objects can be abstracted into either discrete objects such as houses or continuous fields such as elevation and are stored as vector or raster data structures respectively. The raster model is a representation of the world as a surface that is divided into a regular grid of cells. The vector model represents space as a series of discrete entity-defined point, line or polygon units which are geographically referenced. The vector data models are more compact and predominant across the diverse range of spatial information systems [6,7,20,21] and therefore we have focused on vector model for mapping the geographic data in web-based environment.

The geographic objects together with their attribute data constitute spatial data (known locations on the earth with statistical and non-location data associated with spatial entity). The collection of spatial data under vector model is organized in a thematic approach that categorizes data in layers. Usually, each data layer represents only one type of spatial entities, *i.e.*, point, line or polygon.

There are many sources for obtaining spatial data including hard copy maps; aerial photographs; remotely-sensed imagery and existing digital data files. For the scope of this prototype application, a geo-referenced color raster map is obtained in digital format from the Survey of India (government organization). This raster map is processed using Quantum GIS (an open-source desktop mapping engine) to create separate vector layers for different geometry types. Table 2 lists some of the common layers created by extracting geo-objects from the raster map and are used in the prototype application. These layers are saved in ESRI's (Environmental Research System Institute, Redlands, California) shapefile format.

2.3.2. System Components

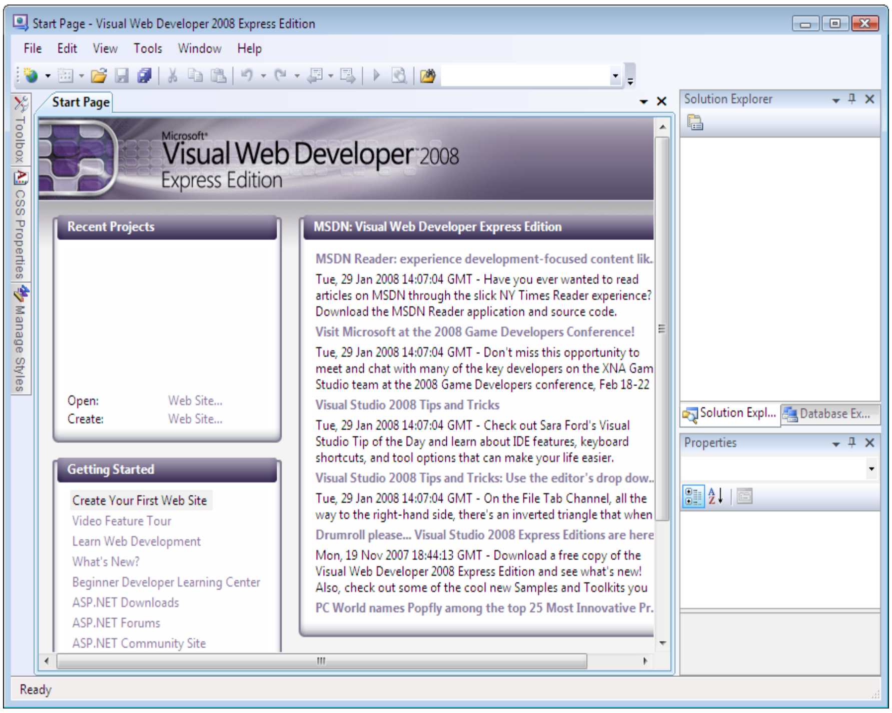
1) Database

The analysis on spatial data can be performed well if the data of geographic objects is stored in relational database. For this reason, all the layers (shapefiles) of the study area are converted to their corresponding tables in a PostgreSQL database. PostgreSQL is one of the most popular object-relational database management system (ORDBMS) on the open-source platform. It is not easy to store the spatial data in a standard RDBMS, thus spatial extensions have been developed and standardized by the OGC (Open Geo Consortium).

PostGIS is an open-source and OGC compliant spatial database extender for PostgreSQL [22]. It adds spatial function and specialty geometry data types to the database. It is an excellent way to bring tabular and spatial data together into a common management environment. The conversion of shapefiles to PostgreSQL database tables is achieved using the shp2pgsql utility included as part of the PostGIS extension. This utility takes a shapefile and outputs a series of SQL statements to create a table in PostgreSQL database. The resulting table contains all the attributes of the shapefile including the

Table 2. Layers extracted from base map (raster map).

Sr. No.	Layer	Geometry Type
1	Roads	Line
2	Administrative Buildings	Polygon
3	Points of Interests	Point
4	Railway Lines	Line
5	Public Services (Amenities)	Point



MSDN: Visual Web Developer Express Edition

MSDN Reader: experience development-focused content lik..

Tue, 29 Jan 2008 14:07:04 GMT - Have you ever wanted to read articles on MSDN through the slick NY Times Reader experience? Download the MSDN Reader application and source code.

Visit Microsoft at the 2008 Game Developers Conference!

Tue, 29 Jan 2008 14:07:04 GMT - Don't miss this opportunity to meet and chat with many of the key developers on the XNA Gam Studio team at the 2008 Game Developers conference, Feb 18-22

Visual Studio 2008 Tips and Tricks

Tue, 29 Jan 2008 14:07:04 GMT - Check out Sara Ford's Visual Studio Tip of the Day and learn about IDE features, keyboard shortcuts, and tool options that can make your life easier.

Visual Studio 2008 Tips and Tricks: Use the editor's drop dow..

Tue, 29 Jan 2008 14:07:04 GMT - On the File Tab Channel, all the way to the right-hand side, there's an inverted triangle that when

Drumroll please... Visual Studio 2008 Express Editions are here

Mon, 19 Nov 2007 18:44:13 GMT - Download a free copy of the Visual Web Developer 2008 Express Edition and see what's new! Also, check out some of the cool new Samples and Toolkits you

PC World names Popfly among the top 25 Most Innovative Pr.

Figure 3. Microsoft Visual Web Developer 2008 Express Edition—start page.

coordinates that define each feature. This process was repeated for each of the vector layers created for the scope of prototype. Figure 4 shows the flow diagram of necessary steps to create the spatial database.

Table 3 represents the open-source geospatial tools used to implement web-based mapping application.

2) Mapping Server

Internet map server applications easily make spatial data accessible through a web browser interface to end users. The map servers usually serve spatial data based on three OGC standards: WMS (Web Mapping Service) for the display of maps as image; WFS (Web Feature Service) for vector data; and WCS (Web Coverage Service) for raster data. The two best-known open-source map servers are Minnesota MapServer and GeoServer. Both of these are viable alternatives to proprietary map servers.

We have used MapServer for development of current prototype, because it has an excellent performance of functionality and speed on processing large volume dataset. It is considered as the world's leading open-source web mapping tool and complies with WMS, WFS and WCS web specifications of OGC [23]. It is a server-side piece of software which renders spatial data source into cartographic map products on-the-fly [24]. The main components of the MapServer are mapfile and CGI program.

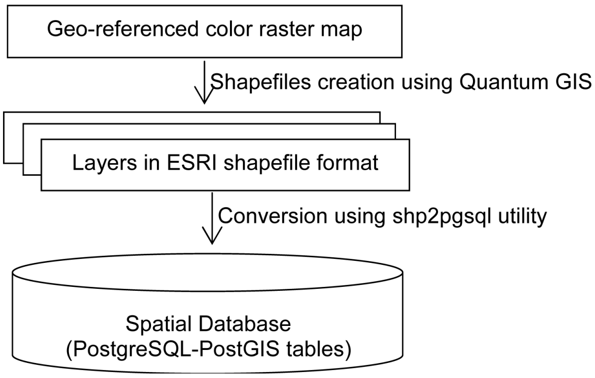

Geo-referenced color raster map
Shapefiles creation using Quantum GIS
Layers in ESRI shapefile format
Conversion using shp2pgsql utility
Spatial Database
(PostgreSQL-PostGIS tables)

Figure 4. Flow diagram of spatial database development.

Table 3. List of open-source geospatial tools used in study.

Category	Open-Source Geospatial Tool
Data Creation	Quantum GIS
Data Conversion	shp2pgsql (PostGIS)
Data Manipulation	PostGIS
Data Analysis	PostGIS
Data Visualization (Internet Mapping)	UMN MapServer

A mapfile defines how data is used and sets the display and query parameters for the map. It needs to set cartographic parameters, cartographic objects, data loading, classification, displaying and querying, and graphic elements definition. It is composed of several objects in which each object starts by keyword OBJECT and finishes by END. Each layer object inside the mapfile starts by keyword LAYER, defines the path and connection type to load the specified data, specifies the style and other parameters, and finishes by END. A mapfile allows direct connections for shapefiles and raster files while more complex connections are required for other data formats. Following code represents the part of a mapfile created for the prototype application with connection to PostGIS layer.

```
………..
LAYER
CONNECTIONTYPE PostGIS
CONNECTION “Host = Localhost DbName = dbDemo
User = postgresql Password = pgAdmin Port = 5432”
DATA “the_geom FROM POI”
NAME “POI”
TYPE POINT
STATUS ON
CLASS
STYLE
COLOR 255 0 0
OUTLINECOLOR 0 0 0
SYMBOL “Circle”
END
END
END
………..
```

A CGI program is the real data processing engine: started up by the web server, it reads the map file settings and returns processed spatial data as maps, cartographic objects, variable values and query results.

MapServer can be used in two ways, with the CGI interface or with the MapScript. MapScript provides a scripting interface for MapServer for the development of web based stand-alone applications. It supports popular scripting languages such as PHP, Java, Perl, C# etc. The prototype application in this study is implemented with MapScript API, accessed with C# MapScript under ASP.NET environment. To gain access to PostgreSQL database server, Npgsql is used which is an open-source ASP.NET data provider. The description on how to configure the MapServer and taking reference of precompiled binaries is provided in Appendix I.

3) Development Environment and User Interface

The user interface design makes the user's interaction simple and efficient in terms of accomplishing user goals. It allows the users to select and input the query criteria in order to view the required data. In the present prototype application, the user interface is made up of “.aspx” files which are populated with the appropriate controls using Visual Web Developer 2008 Express as the .NET development environment. Figure 5 demonstrates the running

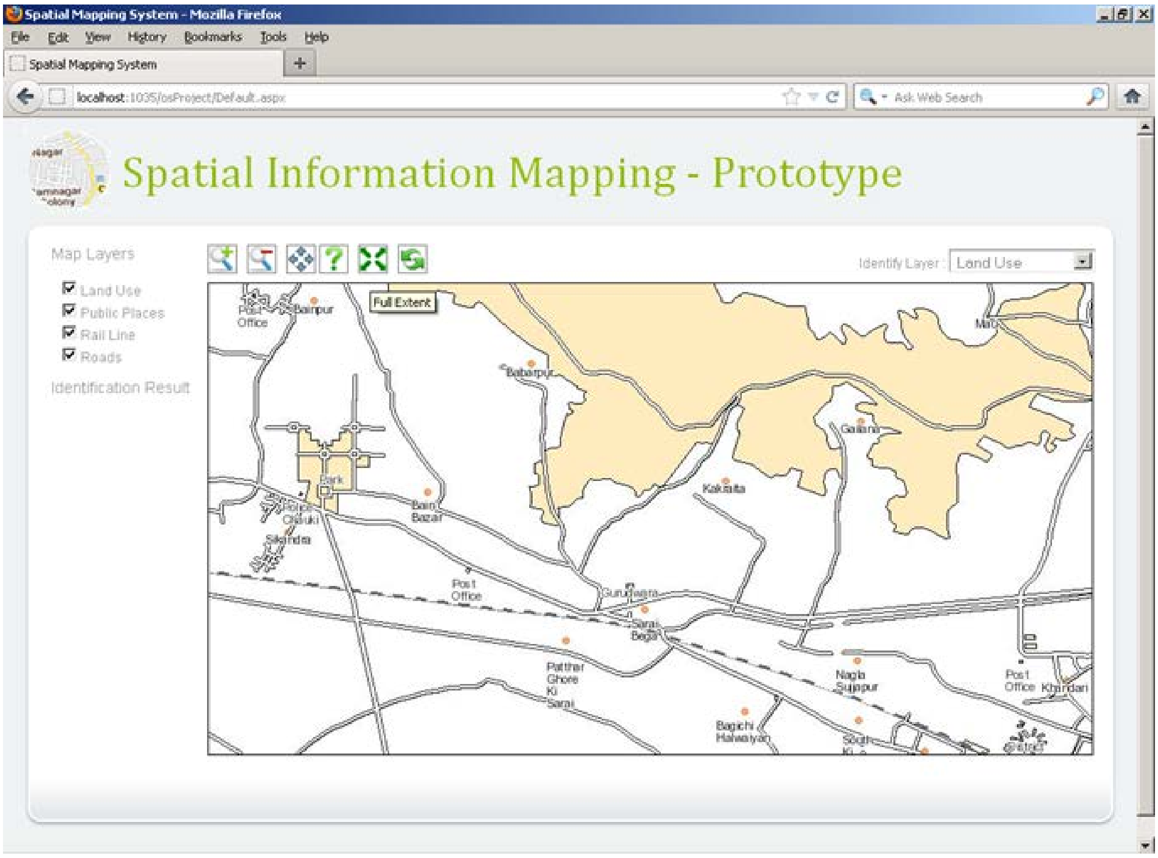


Figure 5. User interface (running view).

view of user interface implemented with ASP.NET by which user interact with the navigation tools such as zoom in, zoom out, zoom to full extent to view the map at different scales and identification of different map features.

2.3.3. System Configuration and Architecture

1) Hardware Platform Configuration

The server machine used for prototype application development has the following specifications:

CPU: Pentium Dual 3.00 GHz with 2 MB Cache
Main Memory: 2048 MB (1024 × 2)
HDD: 80 GB
Operating System: Windows XP (SP3)
Web Server: Internet Information Server (IIS) 6.0
DBMS: PostgreSQL with PostGIS

2) System Architecture

The prototype system adopts client-server architecture where most of the processing takes place on the server side while the client is used for gathering input from the user and displaying map and other associated data based on query parameters. The server-side approach only requires a browser as the spatial mapping and processing; modelling, database and other components will be located on the server. The architecture of the prototype application is shown in the Figure 6, in which the users using any of the popular internet browsers requests geospatial information from the web server.

The web server forwards the request to the ASP.NET application server in which C# Mapscript is called for geospatial data. C# Mapscript forwards the map request to MapServer and the MapServer using the mapfile generates a map based on the geospatial data supplied by the PostGIS. The generated map is sent back to the web server, which forwards it to the web client.

2.4. Explanation and Code

The prototype application contains two web forms named Default.aspx and MapStreaming.aspx by following the flow of [25]. The Default.aspx form contains all the controls required to perform various operations on the map and geospatial data. This form is populated with following controls:

- cblLayers (CheckBoxList) for selection of layers to be displayed
- ddlLayers (DropDownList) for selection of layer to be identified
- litIdentifyResult (Literal) for display the result of identified map features
- ibtnMap (ImageButton) for display the map
- ibtnZoomIn (ImageButton) to perform zoom in
- ibtnZoomOut (ImageButton) to perform zoom out
- ibtnPan (ImageButton) to perform panning
- ibtnIdentify (ImageButton) to identify the features of selected layer
- ibtnFullExtent (ImageButton) to display the full extent map
- ibtnRefresh (ImageButton) to refresh the map
- rblAction (RadioButtonList) to set the map action with item values Zoom In, Zoom Out, Pan and Identify

The MapStreaming.aspx form does not contain any control and is used to send the image stream to the ibtnMap control. The following Figure 7 represents the flow of code among the components of prototype application.

The next section describes the code to implement the essential functions of mapping (detailed code is available on request from authors).

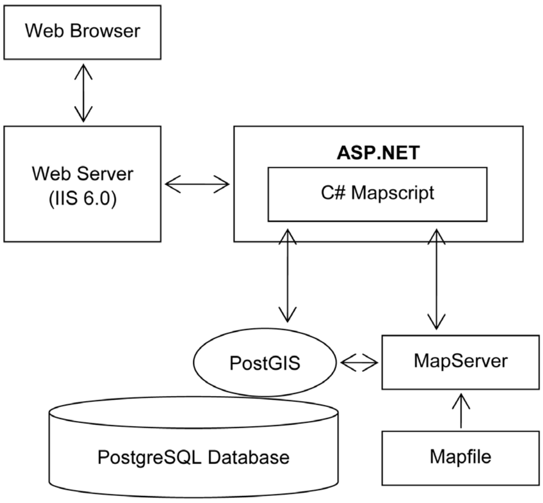


Figure 6. **Architecture of prototype mapping application.**

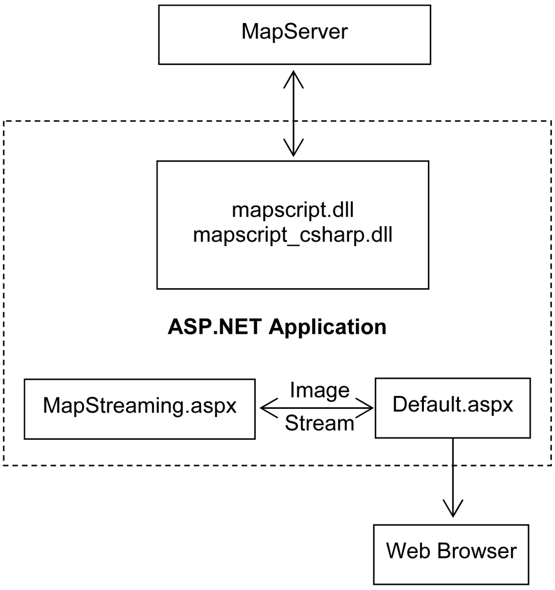


Figure 7. **Flow of code [25].**

2.4.1. C#.NET Mapscript for “Default.aspx”

```
using System;
using System.Web.UI;
using OSGeo.MapServer; //For MapScript
public partial class _Default : System.Web.UI.Page
{
mapObj map;
protected void Page_Load(object sender, EventArgs e)
{
if (!Page.IsPostBack)//First Access to Map
{
//Image Stream from MapServer
ibtnMap.ImageUrl = "MapStreaming.aspx?MAPACTION
=INITMAP";
mapObj map = new mapObj(Server.MapPath("App_Data/
MapFile.map"));
for (int i = 0; i < map.numlayers; i++)
{
layerObj layer = map.getLayer(i);
if (layer.status == (int)mapscript.MS_ON) {
cblLayers.Items[i].Selected = true;
}
}
}
else { //Next Access to Map (By Session)
map = (mapObj)Session["MAP"];
}
}
protected void ibtnMap_Click(object sender, ImageClick
EventArgs e)
{
lblInfo.Text = "";
string mapAction = "";
string activeLayer = ddlLayers.SelectedValue;
//Check and Set Map Action
switch (rblAction.SelectedItem.Text.ToUpper())
{
case "ZOOM IN":
mapAction = "ZOOMIN";
break;
case "ZOOM OUT":
mapAction = "ZOOMOUT";
break;
case "PAN":
mapAction = "PAN";
break;
case "IDENTIFY":
mapAction = "IDENTIFY";
break;
}
//Image Stream from MapServer to ibtnMap according to
required Map Action
ibtnMap.ImageUrl = "MapStreaming.aspx?MAPACTION
=" + mapAction + "&X=" + e.X + "&Y=" + e.Y +
"&ACTIVELAYER=" + activeLayer;
}
//Setting rblAction for different Map Operations
protected void ibtnZoomIn_Click(object sender, Image
ClickEventArgs e) {
rblAction.SelectedValue = "Zoom In";
}
protected void ibtnZoomOut_Click(object sender, Image
ClickEventArgs e) {
rblAction.SelectedValue = "Zoom Out";
}
protected void ibtnPan_Click(object sender, ImageClick
EventArgs e) {
rblAction.SelectedValue = "Pan";
}
protected void ibtnIdentify_Click(object sender, Image
ClickEventArgs e) {
rblAction.SelectedValue = "Identify";
}
//For Full Extent Map
protected void ibtnFullExtent_Click(object sender, Image
ClickEventArgs e) {
ibtnMap.ImageUrl = "MapStreaming.aspx?MAPACTION
=FULLEXTENT";
}
//For Refreshing the Map according to Selected (cblLayers
Checked) Map Layers
protected void ibtnRefresh_Click(object sender, Image
ClickEventArgs e)
{
for (int i = 0; i < cblLayers.Items.Count; i++)
{
layerObj layer = map.getLayerByName(cblLayers.Items[i].
Value);
if (cblLayers.Items[i].Selected) {
layer.status = (int)mapscript.MS_ON;
}
else {
layer.status = (int)mapscript.MS_OFF;
}
}
ibtnMap.ImageUrl = "MapStreaming.aspx?MAPACTION
=REFRESHMAP";
}
}
```

2.4.2. C#.NET Mapscript for “MapStreaming.aspx”

```
using System;
using System.IO;
using System.Web.UI;
using System.Drawing;
using OSGeo.MapServer; //For MapScript
public partial class MapStreaming : System.Web.UI.Page
{
```

```
mapObj map;
rectObj originalExtent;
private enum zoomMode {zoomIn = 0, zoomOut=1} //Enumerator for Zooming
protected void Page_Load(object sender, EventArgs e)
{
//Read Map from Session (if exist), Otherwhise Create new using Mapfile
map = (mapObj)Session["MAP"];
if (map == null)
{
map = new mapObj(Server.MapPath("App_Data/Map File. map"));
originalExtent = new rectObj(map.extent.minx,
map.extent.miny, map.extent.maxx, map.extent.maxy, 0);
Session["ORIGINALEXTENT"] = originalExtent;
}
//Variables to read the required Map Action, (x,y) position and active Layer
String mapAction;
Double x = 0; //Longitude
Double y = 0; //Latitude
String activeLayer;
//Read Map Action from QueryString
mapAction = Request.QueryString["MAPACTION"].ToString().ToUpper();
//Read (x,y) position from QueryString
if (Request.QueryString["X"] != null && Request.Query String["Y"] != null) {
x = Double.Parse(Request.QueryString["X"].ToString());
y = Double.Parse(Request.QueryString["Y"].ToString());
}
//Read active Layer from QueryString
if (Request.QueryString["ACTIVELAYER"] != null) {
activeLayer = Request.QueryString["ACTIVELAYER"].ToString();
}
//Call for different methods based on Map Action
switch (mapAction)
{
case "ZOOMIN":
doZoom(zoomMode.zoomIn, x, y); // Zoom In
break;
case "ZOOMOUT":
doZoom(zoomMode.zoomOut, x, y); // Zoom Out
break;
case "PAN":
doPan(x,y); // Panning
break;
case "IDENTIFY":
doIdentify(x, y, activeLayer); // Feature Identification
break;
case "FULLEXTENT":
doFullExtent(); //Full Extent
break;
case "REFRESHMAP":
break;
}
doRefresh();
Session["MAP"] = map;
}
private void doZoom(zoomMode zMode, Double x, Double y) {    }
private void doPan(Double x, Double y) {    }
private void doIdentify(Double x, Double y, String active Layer) {    }
private void doFullExtent() {    }
private void doRefresh() {    }
```

3. Conclusion

In any project, for the developing countries like India, cost is the primary consideration for the adaptability of spatial technology. Under these circumstances, free and open-source geospatial tools provide all the data storage, analysis and information visualization for free. This paper provides a foundation in web-based mapping using open-source geospatial tools and web technology. It highlights building a prototype mapping application by providing step-by-step instructions which can encourage the eager developers and interested readers to publish their maps on the web with no prior technical experience in map servers. The whole prototype lives in a common web-based application implemented in ASP.NET with C#.NET MapScript exploiting MapServer for mapping functionalities and PostGIS for its connection with the PostgreSQL databse. The techniques and source code of this research work can be applied as the laboratory exercise or a course curriculum and can help a large number of organizations to publish their geographic information on the web, especially those small organizations that cannot afford proprietary software's cost, complexity, training cost and special requirements.

Acknowledgements

The corresponding author is thankful to University Grants Commission, India for providing financial support in the form of research fellowship under RFSMS scheme.

REFERENCES

[1] R. Mari, L. Bottai, C. Busillo, F. Calastrini, B. Gozzini and G. Gualtieri, "A GIS-Based Interactive Web Decision Support System for Planning Wind Farms in Tuscany (Italy)," *Renewable Energy*, Vol. 36, No. 2, 2011, pp. 754-763. http://dx.doi.org/10.1016/j.renene.2010.07.005

[2] A. M. Ozimec, M. Natter and T. Reutterer, "Geographical Information Systems-Based Marketing Decisions: Effects

of Alternative Visualizations on Decision Quality," *Journal of Marketing*, Vol. 74, No. 6, 2010, pp. 94-110. http://dx.doi.org/10.1509/jmkg.74.6.94

[3] A. Okabe and K. Okunuki, "A Computational Method for Estimating the Demand of Retail Stores on a Street Network and Its Implementation in GIS," *Transactions in GIS*, Vol. 5, No. 3, 2001, pp. 209-220. http://dx.doi.org/10.1111/1467-9671.00078

[4] A. K. Ziliaskopoulos and S. T. Waller, "An Internet-Based Geographic Information System That Integrates Data, Models and Users for Transportation Applications," *Transportation Research Part C*, Vol. 8, No. 1-6, 2000, pp. 427-444.

[5] F. Xie, "Design and Implementation of Highway Management System Based WebGIS," *Journal of Networks*, Vol. 5, No. 12, 2010, pp. 1389-1392. http://dx.doi.org/10.4304/jnw.5.12.1389-1392

[6] S. P. Singh, J. Sharma and P. Singh, "A Geo-Referenced Information System for Tourism (GeoRIST)," *International Journal of Geomatics and Geosciences*, Vol. 2, No. 2, 2011, pp. 456-464.

[7] P. S. Singh, D. Chutiya and S. Sudhakar, "Development of a Web Based GIS Application for Spatial Natural Resources Information System Using Effective Open Source Software and Standards," *Journal of Geographic Information System*, Vol. 4, No. 3, 2012, pp. 261-266. http://dx.doi.org/10.4236/jgis.2012.43031

[8] T. Q. Zeng and Q. Zhou, "Optimal Spatial Decision Making Using GIS: A Prototype of a Real Estate Geographical Information System (REGIS)," *International Journal of Geographical Information Science*, Vol. 15, No. 4, 2001, pp. 307-321. http://dx.doi.org/10.1080/136588101300304034

[9] T. Oswari, E. S. Suhendra, E. Haryatmi and F. Agustina, "Prototype Geographic Information Systems Mapping of Crop Products Featured Local," *Journal of Geographic Information System*, Vol. 5, No. 3, 2013, pp. 193-197. http://dx.doi.org/10.4236/jgis.2013.53018

[10] B. Evans and C. E. Sabel, "Open-Source Web-Based Geographical Information System for Health Exposure Assessment," *International Journal of Health Geographics*, Vol. 11, 2012, p. 2.

[11] R. M. Mullner, K. Chung, K. G. Croke and E. K. Menash, "Geographic Information Systems in Public Health and Medicine," *Journal of Medical Systems*, Vol. 28, No. 3, 2004, pp. 215-221. http://dx.doi.org/10.1023/B:JOMS.0000032972.29060.dd

[12] D. Boyles, "GIS Means Business," ESRI, Redlands, 2002.

[13] M. N. K. Boulos and K. Honda, "Web GIS in Practice IV: Publishing Your Health Maps and Connecting to Remote WMS Sources Using the Open Source UMN MapServer and DM Solutions MapLab," *International Journal of Health Geographics*, Vol. 5, 2006, p. 6.

[14] Q. Yi, R. E. Hoskins, E. A. Hillringhouse, S. S. Sorensen, M. W. Oberle, S. S. Fuller and J. C. Wallace, "Integrating Open-Source Technologies to Build Low-Cost Information Systems for Improved Access to Public Health Data," *International Journal of Health Geographics*, Vol. 7, 2008, p. 29.

[15] S. Steiniger and E. Bocher, "An Overview on Current Free and Open Source Desktop GIS Developments," *International Journal of Geographical Information Science*, Vol. 23, No. 10, 2009, pp. 1345-1370. http://dx.doi.org/10.1080/13658810802634956

[16] J. Lowe, "Spatial on a Shoestring: Leveraging free Open Source Software," *Geospatial Solutions*, Vol. 12, No. 6, 2002, pp. 42-45.

[17] T. Mitchell, "Open Source Geo Tools," *O'Reilly Where Where* 2.0 *Conference*, San Francisco, 29-30 June 2005.

[18] T. Mitchell, "An Introduction to Open Source Geospatial Tools," 2005. http://www.oreillynet.com/pub/a/network/2005/06/10/osgeospatial.html

[19] W. M. Lee, "C#. NET Web Developers Guide," Syngress, Rockland, 2002.

[20] A. Mansourian, A. Fasihi and M. Taleai, "A Web-Based Spatial Decision Support System to Enhance Public Participation in Urban Planning Processes," *Journal of Spatial Science*, Vol. 56, No. 2, 2001, pp. 269-282. http://dx.doi.org/10.1080/14498596.2011.623347

[21] A. G. Othman, B. Mohamed, A. Bahauddin, A. P. M. Som and S. I. Omar, "A Geographic Information System Based Approach for Mapping Tourist Accommodations in the East Coast States of Malaysia," *World Applied Sciences Journal*, Vol. 10, 2010, pp. 14-23.

[22] PostGIS. http://postgis.net/

[23] S. Steiniger and A. J. Hunter, "Free and Open Source GIS Software for Building a Spatial Data Infrastructure," In: E. Bocher and M. Neteler, Eds., *Geospatial Free and Open Source Software in the* 21*st Century*, Springer-Verlag Berlin Heidelberg, 2012, pp. 247-261.

[24] P. Ramsey, "The State of Open Source GIS," 2007. http://www.refractions.net/expertise/whitepapers/opensourcesurvey/survey-open-source-2007-12.pdf

[25] P. Corti, "Thinking in GIS," 2006. http://www.paolocorti.net/2006/09/20/mapserver-tutorial-for-c-mapscript-asp-net/

APPENDIX I

Installation and Configuration of MapServer for Windows

The MS4W (MapServer for Windows) package provides quick and easy installer for setting up MapServer for Windows and its accompanying applications. MS4W package is designed to perform a full installation of Map Server CGI, MapScript (C#, java, PHP, etc.) and Apache. This package contains precompiled binaries and can be downloaded by accessing the link http://mapserver.org/download.html Once the package (archive) is downloaded, unzip it in C:\drive of computer.

The references we need in our ASP.NET application are the dll's under C:\ms4w\Apache\cgi-bin\mapscript\csharp and the dll's under C:\ms4w\Apache\cgi-bin. C:\ms4w\Apache\cgi-bin\mapscript\csharp\mapscript_csharp.dll is the only .NET dll and for all other dll's to be accessible in our application, we can either copy all dll's under the bin directory of our application or simply put the path of all these dll's under the PATH environment variable of Windows.