GIS - Open Source initiatives

Abstract

GIS awareness and requirements have increased exponentially, especially in the Government Sector. There is a need to communicate and share spatial and associated non-spatial information quickly amongst the users. The convenience and luxury of sharing information over the web through a simple user-friendly browser is the hot topic.

The scope of this paper is to highlight the power of Open Source Tools in GIS. The paper presents a "Proof of Concept" designed and developed for an end to end GIS solution using Open Source tools. The model could be extremely useful, especially for developing countries like India.

Internet GIS Solution experimented at National Informatics Centre, Tamil Nadu State Unit, Chennai, using free Open Source includes the following salient features:

1. Using a free GIS webserver that talks to GIS data in any form that is, Shape file/Tab file or accessing them through any RDBMS like MySQL, PostGreSQL/PostGIS, thus allowing interoperability.

2. An interface that talks to any form of non-spatial attribute data and linking the spatial features.

3. Selection of non-spatial attributes based on a spatial query.

4. Selection of spatial data based on query on non-spatial attributes.

5. Ability to classify spatial data as thematic maps.

6. Flexibility in user-selectable layers on/off, colours customization, etc.

The above initiatives have been implemented at NIC - Chennai.

GIS - Open Source initiatives

Introduction

Worldwide, the growth of Open Source software has been remarkable. The availability of the source code and the right to modify enables the unlimited tuning and improvement of software. The right to redistribute these modifications and improvements to the code, and to reuse other open source code, adds to all the advantages to be shared by large communities.

At **National Informatics Centre, Chennai**, we have experimented Open Source GIS Solutions for Internet based applications. One such website that has already been launched is http://www.animaldiseaseinfo.tn.nic.in. Here, the Animal Census population is being shown as dynamic thematic maps. Also, user can customize the map (for the ranges, colours, legends and titles) and get the output map in the required format.

Design Methodology

The design of the application has been split into different modules to enable easy development and better maintenance. Also, the model developed is platform independent. This would enable hosting of the website on a Microsoft Windows Platform or on a Linux Platform. The tools that have been used are available for both the platforms.

The overall design is presented in figure 1.

Open Source GIS and RDBMS Tools/Softwares Used

GIS Server	–	MapServer
Server Side Scripting	–	PHP
Client Side Scripting	–	JavaScript
Database Server		
Non-Spatial Data	–	PostGreSQL
Spatial Data	–	PostGIS Extn of PostGreSQL

Design Diagram

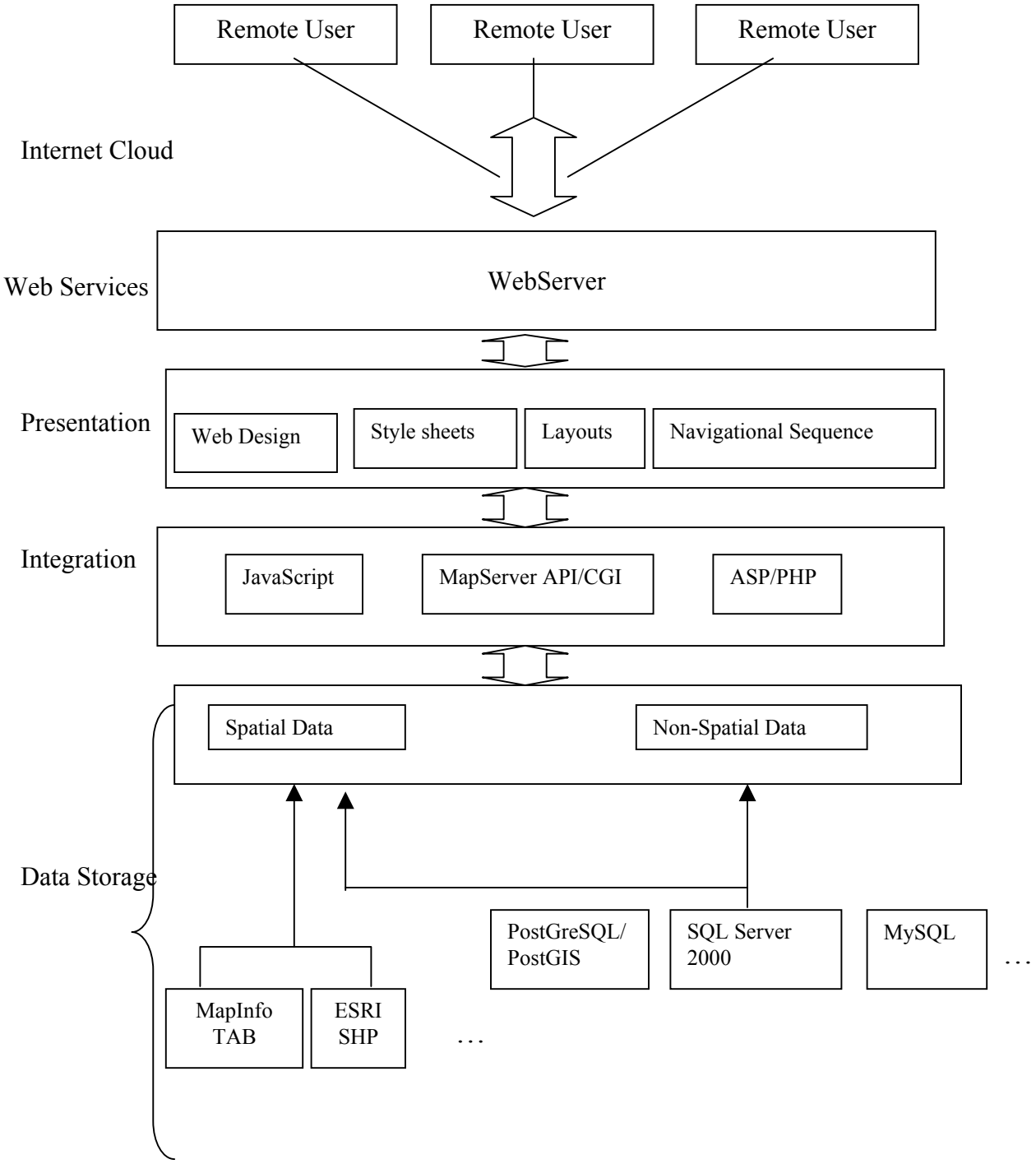

Figure 1 Design Diagram

Modules

The design aspects are discussed for the following modules:

- **MODULE 1: Making data Internet ready**. Data acquisition. Associating non-spatial data for the spatial features.
- **MODULE 2: Area or Region Selection.** A model which would select the geographical entity(s) that need to be displayed (State / District / Taluk or Block map)
- **MODULE 3: Spatial Query.** An interface that would enable user to select the spatial features and query the associated non-spatial features.
- **MODULE 4: Non-Spatial Query.** A flexible user interface where the users would make a query on non-spatial attributes and display the associated spatial features which satisfy the query.
- **MODULE 5: Thematic/Choropleth Mapping.** A flexible user interface where the users would select the theme or parameter, classify the range for the parameter, etc and create a dynamic thematic/choropleth map.
- **MODULE 6: Simple Map Browsing.** A user interface where the users would be able to browse the map by zooming in/out, querying etc.

The modular design diagram is shown below figure 2

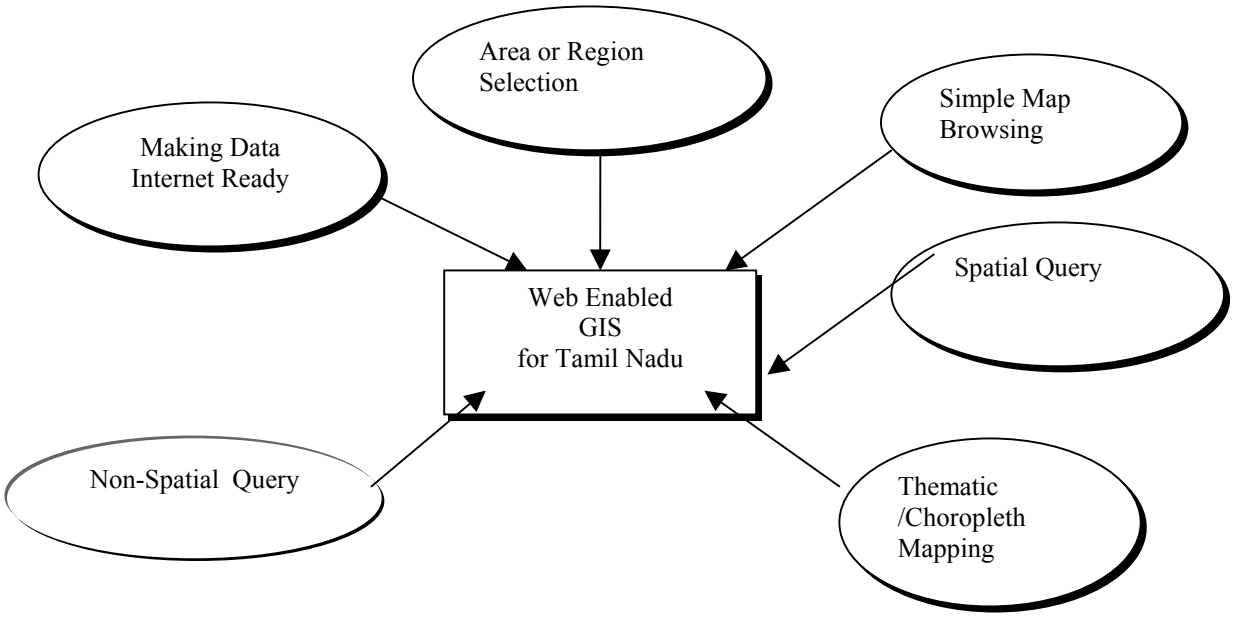

Figure 2 Modular Design Diagram

MODULE 1: Making data Internet ready

Open GIS offers interoperability so that the map could be in any OGR Standard formats for Internet applications.

The following steps were carried out for making spatial and non-spatial data Internet ready.

- **Non Spatial Data:** The Census 2001 data has been used for query shells. Population and Amenities model data have been used.
- **Spatial Data:** Spatial data was prepared in the form of shape files. Each individual layer of the map was initially converted to shape files. When a layer is converted to shape file, five or less number of flat files are generated with extensions .dbf, .shx, .shp, .sbn an .sbx. The Census codes for each feature have been added to the layer itself, apart from the automatically generated default shape-id, for easy linking across spatial and non-spatial data. Refer Figure 3 for a pictorial representation. The shape files could be later ported to PostGreSQL/PostGIS Server or MySQL for easy retrieval, maintenance and deployment. The spatial data once imported is in the form of tables within the database.

Shape_id	State_code	District_code	Taluk_code
0	30	14	0001
1	30	14	0002

Population table

Population_male	Population_female	State_code	District_code	Taluk_code
186063	186058	30	14	0001
82039	86038	30	14	0002

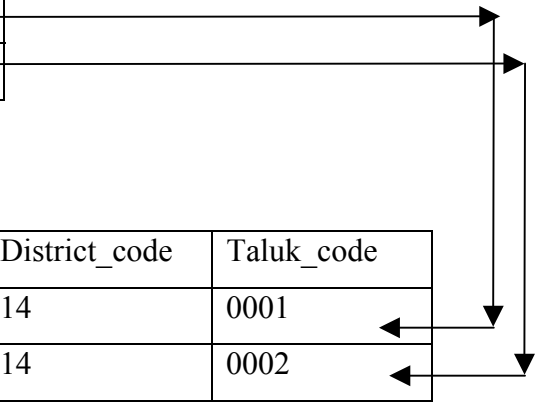

Figure 3 Shape file

- **Catalogue:** Catalogue of the available administrative boundaries/features need to be made for the available spatial data level wise. For District Level maps, a table District_Maps (Figure 4) has been designed so that it would capture all the features/layers available for each district along with extents, shape-path/table id,

reference image. This data would help to draw the layers dynamically on selection. Similar tables were created for Taluks, Blocks etc.

District Maps
State Code
District Code
District Name
Shape Path/Table Name
Reference Image
Extents
District Boundary Layer
Taluk Layer
Block Layer
Panchayat Village Layer
Revenue Village Layer
Rivers
Transport
Urban

Figure 4 District_Maps

- **QueryTab:** For each level of region State/District etc. Census data that could be queried upon are identified and added as entries in querytab table. For instance, to enable population query at district level, we would add an entry for the same by referring table name, fields that could be queried and fields that could be used for thematic maps at district level fields. For the same population data at taluk level, similar values need to be keyed in at taluk level fields and so on. Dynamically these tables would be used for Module 3, 4 and 5 discussed below. Id and Id1 would be used for classifying the parameters. Figure 5 overleaf represents the table structure for Querytab. For example, Id=1 would be for demographic query, Id=2 would be for Town Amenities, Id=3 for Village Amentities and so on. We can add as many parameters that we want to base the query on for the specified region. Figure 6 overleaf represents the Data Flow Diagram.

Querytab
Id
Id1
Parameter Name
District Parameter Table
District Theme Parameters List
District Level Fields List for Query
Taluk Parameter Table
Taluk Theme Parameters List
TalukLevel Fields List for Query
Town Parameter Table
Town Theme Parameters List
Village Parameter Table
Village Theme Parameters List
Village Level Fields List for Query

Figure 5 Querytab

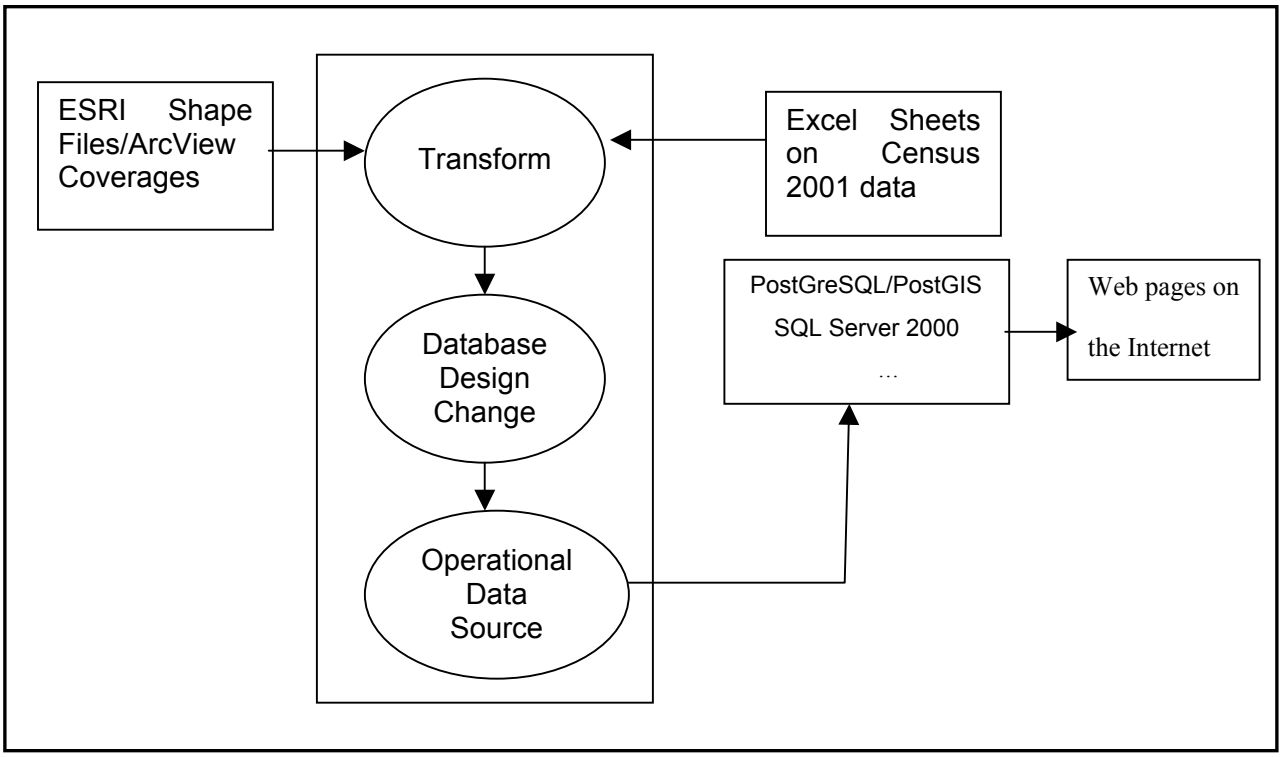

Figure 6 Data Flow Diagram for Spatial and Non Spatial Data Base

MODULE 2: Area or Region Selection.

Designed and developed a user interface to select the map of choice to be displayed viz State / District / Taluk or Block map with the help of combo boxes. Based on the level and region of interest, dynamically the features available for the region viz. Rivers, Transport, etc. are identified from the catalogue created during Step 3 of Module I mentioned above. A dynamic map file is generated which is later manipulated for customization.

MODULE 3: Spatial Query.

Based on entries in **querytab** tables created during Module I – Step 4, dropdown boxes are dynamically generated so that user can choose his parameter of interest. When user queries on the map, the corresponding fields from the respective table are queried and the result would be displayed in the Details View. The map can be queried at various levels such as District, Taluk, Village, etc. areas as only appropriate dropdown boxes would be made selectable.

MODULE 4: Non-Spatial Query.

Similar to Modules 3 discussed above, **querytab** table is used to generate columns of various tables for the region of interest, that can be used for querying and mapping the resultant features. Based on the user selection of the column, the minimum and maximum values are displayed. User can feed the criteria in the Query Builder form, give his own values for selection criteria. Finally, for the query presented by the user, only those features that match the criteria are displayed dynamically.

MODULE 5: **Thematic/Choropleth Mapping.**

Similar to Modules 3 and 4 discussed above, **querytab** table is used to generate columns of various tables for the region of interest that can be used for dynamically creating a choropleth/thematic mapping. Based on the user selection of the column, the minimum and maximum values are displayed. User can select the number of ranges, feed

the ranges in the form, choose his own colour, legend and titles for the thematic map. Finally, for the ranges and colours presented by the user, thematic/choropleth map is displayed dynamically.

MODULE 6: Simple Map Browsing.

The maps are presented in an applet form so that the user can perform the zooming operations (zoom in, zoom out, recenter) on the map. These operations allow the user to see the map at his/her own magnification scale. The Maps in this website are generated dynamically based on the geographic selection made by the user. The map file is a file in which the layer definitions are done for a map. Once the user makes the geographic selection, then the map file is generated on the fly with the layers available for that geographic entity. This file is consulted for the layers available, order in which the layers are to be displayed and so on. The layer details for any geographic entity are stored in the table and it is verified before generating the map file. This design would be effective especially when additional layers are added to a specific entity since it would be enough if it is added to the table.

Customizations that are possible include the following:

Legend

The Legend portion of the map displays all the layers of the map, its visibility status and the colour used to represent it. The user is provided with the option of turning a layer on/off. The user can at any time change the colour of a layer from the palette and this change will be reflected in the map once it is redrawn.

Reference Map

This is the reference map and it is useful to indicate at any point of time where the user is in the map. This can also be turned off by checking the Key map Check box so that the user can see the map alone.

Map size

The size of the map can be changed dynamically. This helps the user to view the map under different resolutions.

Zooming Options

The zooming options include zoom in, zoom out and centering the map. These options allow the user to view the map under different magnification scales.

Query

With the query option the user is allowed to retrieve information, based on any Census Parameter and about a specific geographic location in the map.

GIS Server and Tools Used

Mapserver

MapServer is an Open Source development environment for building spatially enabled Internet applications. It was developed by University Of Minnesota(UMN). The software builds upon other popular Open Source or freeware systems like Shapelib, FreeType and others. MapServer is known to compile on most UNIX systems and will run under all flavours of Windows.

MapServer system supports MapScript which allows popular scripting languages such as Perl, Python, PHP, Guile and even Java to access the MapServer C API. MapScript provides a rich environment for developing applications that integrate disparate data. If the data has a spatial component then via a scripting environment it can be mapped. For example, with PHP/Mapscript module we could accomplish the above task.

Parts of a MapServer Application

MapServer generally runs as a CGI application from http server. This holds true unless a more advanced application is built with MapScript which accesses the MapServer API directly.

MapServer CGI applications use the following resources:

- An http server like Apache or Internet Information Server.
- MapServer software.
- An initialization file that triggers the first view of the MapServer application.
- A Mapfile that controls what MapServer does with the data,
- A template file that controls the MapServer application's user interface in the browser window.
- A GIS dataset.

PHP/Mapscript

The PHP/MapScript module is a PHP dynamically loadable module that makes MapServer's MapScript functions and classes available in a PHP3/PHP4 environment. With this, the we could accomplish any of the Mapserver's functionalities as suited to our application.

The functional flow can be given as follows:

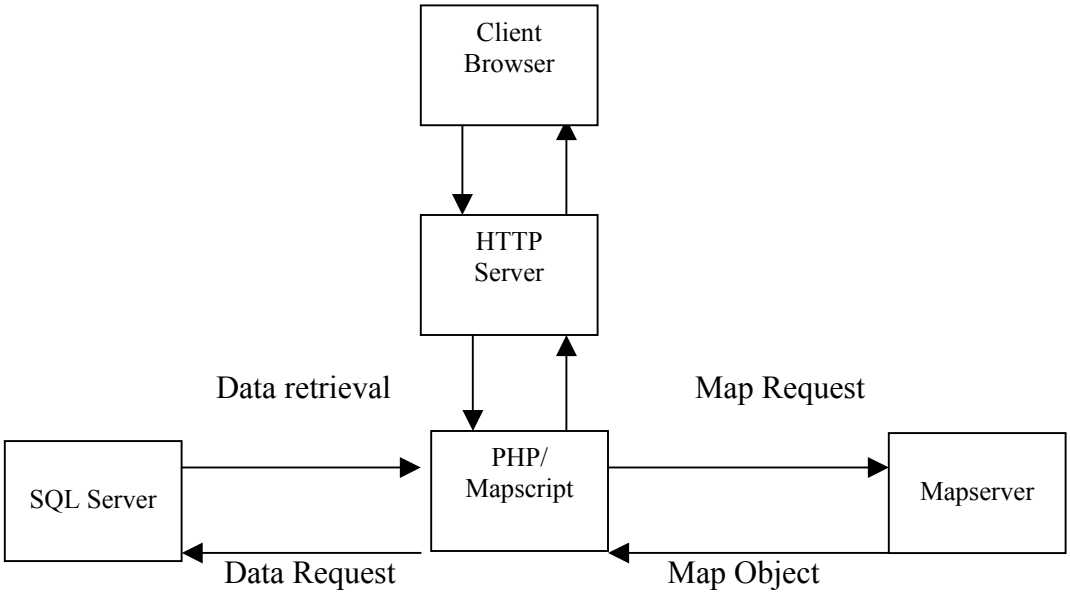

Figure 7 Functional Flow

Conclusion

The model presented above tries to leverage the power of GIS open source tools. It can be extended to any geographical extents and made more flexible to talk to heterogeneous database platforms. The model can be enhanced to create a GIS web service which would dynamically create maps of regions of interest on any theme by pulling data dynamically.

Bibliography

- **Spatial Analysis and GIS**

  By Stewart Fotheringham, Peter Rogerson
- **Beginning Databases with PostgreSQL**

  By Richard Stones and Neil Matthew
- **Linux Database Bible**

  By Michele Petrovsky, Stephen Wysham, Mojo Nichols

Web References

- **MapServer and MapScript**

  http://mapserver.gis.umn.edu
  http://www.dmsolutions.on.ca

- **Open Source Tools on GIS**

  http://www.freegis.org/index.en.html