Implementation of Web Map Server Test-bed and Development of Training Material for Advancing FOSS4G Solutions

Venkatesh Raghavan*, Phisan Santitamnont**,
Shinji Masumoto*** and Tatsuya Nemoto***

* Graduate School of Creative Cities, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan, email raghavan@media.osaka-cu.ac.jp
** Faculty of Engineering. Department of Survey Engineering, Chulalongkorn University, Patumwan, Bangkok, Thailand 10330, email Phisan.S@eng.chula.ac.th
*** Department for Geosciences, Graduate School of Science, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan, email masumoto@sci.osaka-cu.ac.jp

1 Introduction

With increasing availability of spatial data, there is an urgent need to develop way and means for providing easy accessibility and promote spatial data sharing. A wide range software tools for spatial data sharing are now readily available. Proprietary solutions often adopt a black-box approach providing little insight about the underlying technology and are also often too expensive for users with limited budget. Free/Libre and Open Source Software (FOSS) not only provide a far more economical alternative but also more amenable to easy customization and maintenance. However, in using FOSS, novice users often feel intimated with an apparently steep learning curve and with a bleak notion of being faced with the lack of institutionalized support. The results presented in this paper address some of these issues and provide a simple, viable and low-cost alternative to spatial data sharing and Web Map Server (WMS) implementation using existing FOSS tools.

Our research aims to fulfill the following objectives;

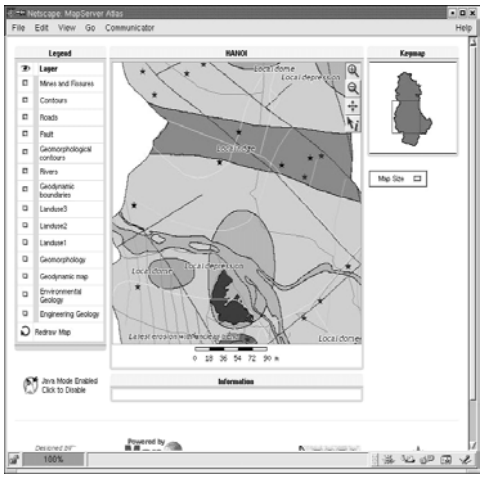
- Implementation of a WMS test-bed using FOSS (Linux OS, Apache Web server, Minnesota MapServer)
- Implementation of thin and thick client interface and development of application examples using existing spatial datasets (e.g. DGCM III Working Group)
- Evaluation of WMS test-bed in a distributed environment
- Spatial data generation using FOSS GIS GRASS
- Simplified software packaging for novice users
- Development of training material to promote hands-on and self-paced learning.

2 Standardization & Interoperability in Spatial Data Delivery

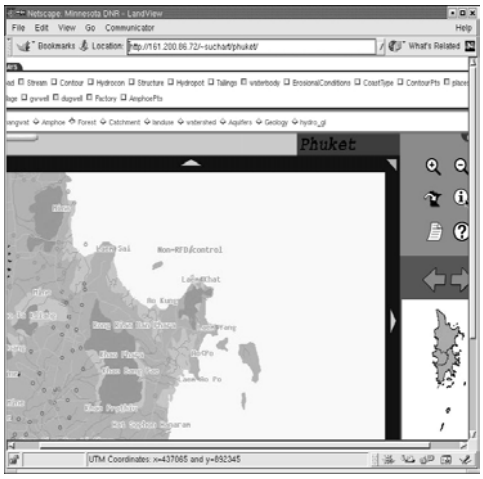
Apparently, one of the main limitations of the prototype systems developed in our previous research (e.g. [3], [5]) is the interoperability and non-compliance with widely accepted international standards. Implementing such systems (e.g. [3], [5]) requires the information to be stored in the GRASS GIS format and the GRASS GIS needs to be installed on the server. The advantage in having a full-fledged backend GIS running on the server would be the ability to implement online systems with spatial analytical capabilities rather than providing visualization or portrayal capabilities alone (as are commonly available in other Web GIS applications). Such online systems for 3D online geological modeling have been demonstrated in our previous research [2]. However, in most general situation of providing easy access to spatial data, the advantages of adopting approach wherein issues such as interoperability and compliance to international standards far outweigh any other consideration. Specifications have emerged within the Geoinformatics user community for

the design of interoperable systems for spatial data sharing amongst users with only map reading skills.

Considering the obvious advantages of providing standardized data access and portrayal services, we are now developing spatial database system using the MapServer toolkit (http://mapserver.gis.umn.edu/). In this regard, we have developed several applications using existing spatial data that are available through regional initiatives (e.g. [1]). Some examples of the MapServer implementation using the data from Phuket, Thailand (Figure 1a) and Hanoi City, Vietnam (Figure 1b) have already been developed. Multi-media rich attributes can be easily integrated in a Mapserver Application as shown in Figure 2). We have also set up WMS compliant test-bed (Figure 3; http://wgrass.media.osaka-cu.ac.jp/ms-demos.html) for serving information in a distributed environment.



(a)



(b)

Figure 1: MapServer applications for (a) Phuket Island, Thailand (b) Hanoi City, Vietnam

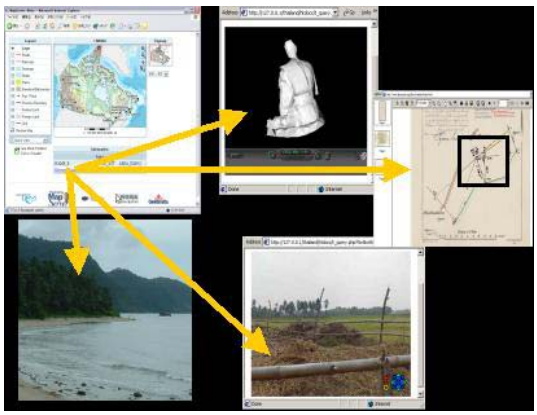


Figure 2: Multi-media rich application.

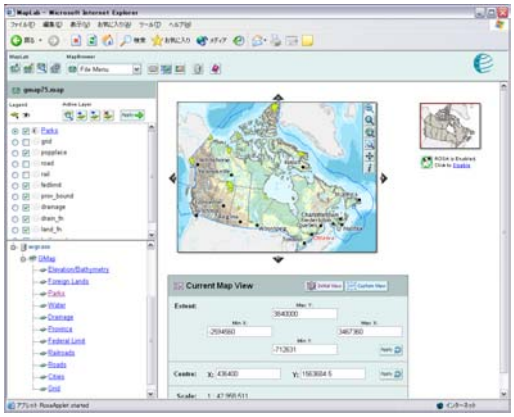


Figure 3: Distributed Access using WMS

2.1 Implementing WMS Test-bed

WMS allows for use of data from several distributed servers, and enables for the creation of a network of spatial data repository from which clients can build customized maps. WMS servers interact with their clients via the HTTP protocol. In most cases, a WMS server is a CGI program. The WMS specification defines a number of request types that are transmitted using the HTTP protocol. For each of the request, a set of query parameters and associated behaviours are stipulated. A WMS-compliant server must be able to handle at least the following 3types of WMS requests:

- *GetCapabilities:* return an XML document with metadata of the information available on a Web Map Server.
- *GetMap*: return an image of a map according to the user's needs.
- *GetFeatureInfo*: return info about feature(s) at a query (mouse click) location.

The basic framework of the WMS test-bed is shown in Figure 4. The WMS capability for the test-bed is provided by using the MapServer CGI program. The MapLab (http://www.dmsolutions.ca/techserv/maplab.html) Rapid Application Development (RAD) tool greatly simplifies process of developing web-mapping applications. MapLab supports every stage of publishing spatial data and facilitates the discovery and integration of local and remote data sets, the cartographic representation of map layers and the management of maps and applications on local and remote web servers.

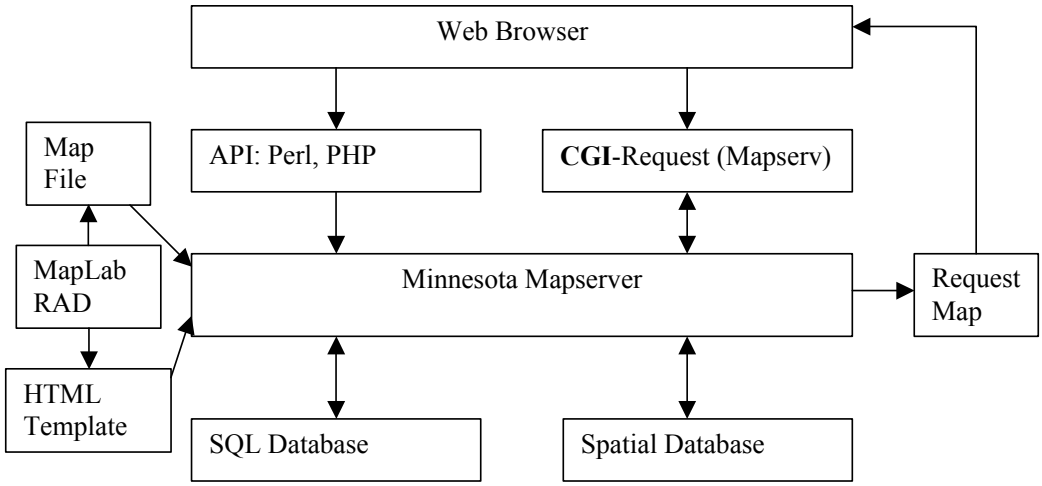


Figure 4: Framework of the WMS Test-bed.

2.2 WMS test-bed, software installation and training material

In order to implement the WMS test-bed, several packages need to be installed. RPM (http://www.rpm.org) packages of all required (Table 1) software were prepared. Mandrake Version 9.1 Linux distribution was used for implementing the WMS test-bed. Most of the additional libraries needed to implement the WMS (e.g. libproj, libgdal, libwww etc.) are also made available as RPM binaries and can be easily installed using automated installation script. PHP (PHP Hypertext Processor) is configured as a CGI in order to facilitate PHP/MapScript Scripting access to the MapServer C API using PHP. This required some additional configuration steps for the Web Server. GRASS GIS Version 5.0.3 was also packaged along with other FOSS tools. GRASS GIS allows preparation of raster data in GeoTIFF and other formats compatible with the Mapserver. Moreover, Mapserver can also read GRASS raster files using the "libgrass" library. GRASS GIS also supports digitization of vector maps and export of vector data as "shape" file. The above features of GRASS GIS enable the user to prepare their own dataset that could be incorporated into the WMS test-bed or in other Web applications.

Two methods for linking multi-media attribute information to shape file layer were devised. The first method is fairly simple to implement and requires editing of the .dbf file associated with the shape file layer. This can be done using a general-purpose spreadsheet software. The second method allows automatic searching to multi-media attribute information associated with the shape layer. This approach is more flexible but requires additional PHP coding. Viewing of multi-media attributes depend on availability of appropriate plug-in on the clients browser. Still pictures (e.g. JPEG, PNG), motion picture (e.g. MPEG), 3D Virtual Reality Models (e.g. VRML) and compressed image format (e.g. DJVU) files were linked to shape file layer and tested.

Table 1: FOSS tools used in the project

FOSS Project	**Geoinformation Service**
Mandrake Linux Version 9.1	Operating System
Apache Version 2.0	Web Server
PHP Version 4.3.2	Web Application Development
GRASS Version 5.0.3	GIS
MapServer Version 3.6.5	Web Mapping
libgrass Version 5.1	GRASS 5 I/O Library
GDAL Version 1.1.8	Data translation
PROJ4 Version 4.4.5	Coordinate transformation
MapLab Version 2.0	Rapid Application Development Tool
Djvu 3.5.1	Image Compression

MapLab suite of Web-based, open source tools that simplifies the process of deploying MapServer Web mapping applications is also include in the package. To deploy Web Mapping applications using Mapserver, it is necessary to create two configuration files, namely the MapFile and Template file. The MapFile defines the data to be used in an application and display and query parameters; you can think of it as the application's configuration file. The MapFile also includes information about how to draw the map, the legend, and maps resulting from a query. MapFile normally have a .map extension.

The Template file controls how the maps and legends output from MapServer will appear on the web browser; it operates like any other HTML file, except that certain fields can be modified by the MapServer CGI. The Template file allows the author to place the map and legend on the page, and determine what ways the user is allowed to interact with the mapserver application (i.e browse, query, zoom, etc.). Mapserver uses the Template file and replaces keywords in the template file with information on its current state or the GIS dataset, to produce the html file that is sent to the browser. Since the Template file will be used to create an html file, it is usually stored with an .html extension. Template files can also uses PHP scripts (as in the case of gmap demo) in which case the .phtml or .php3 extension is used.

The Maplab RAD online tool allows the user to create, edit and manage the Mapfile using the MapEdit application. Further, the GmapFactory application of Maplab allows the user to create the Template file which controls the appearance of the Web Mapping application. The GmapFactory uses the “gmap” template and Rosa Java applet to display map layers defined in the Mapfile. Maplab MapBrowser application is a spatial data discovery tool that is compliant with the OGC Web Mapping specification. Using MapBrowser, application developers can browse and integrate map layers from local or WMS compliant remote servers. List of parameters and metadata items that are usually optional with MapServer, but are required (or strongly recommended) for a WMS configuration are listed below

At the map level:

- Map NAME
- Map PROJECTION
- Map Metadata (in the WEB Object):
- wms_title
- wms_onlineresource
- wms_srs (unless PROJECTION object is defined using "init=epsg:...")

And for each layer:

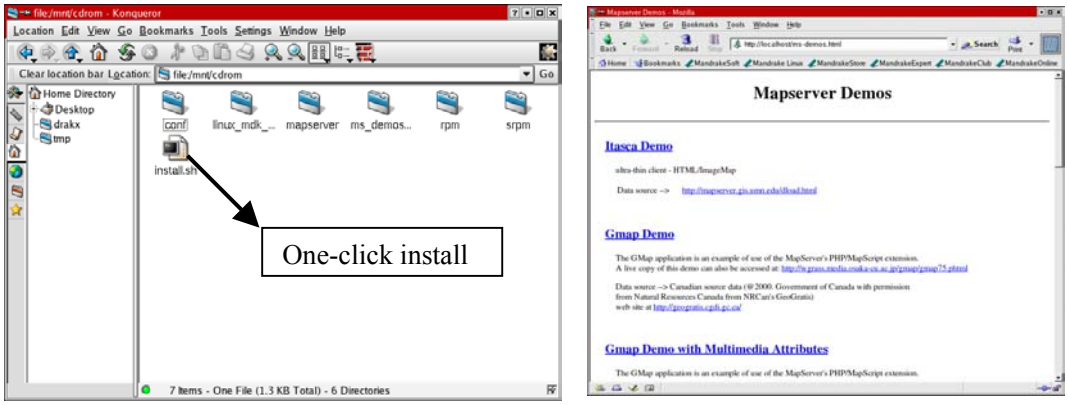
- Layer NAME
- Layer PROJECTION

- Layer Metadata
- wms_title
- wms_srs (optional since the layers inherit the map's SRS value)

WMS test-bed was successfully tested by appropriately configuring the Mapserver Mapfiles and by using the Maplab Mapbrowser application for linking and interacting with local and remote WMS servers.

2.3 Software packaging

Implementing MapServer and WMS test-bed is a daunting task for novice users and developers since several software packages and libraries need to be installed and configured. Further, to enable self-paced learning, appropriate demonstration applications and related datasets must also be made available. As a part of this project, complete suite of software tools have been packaged using the RPM technology and a simple shell script (Figure 5a) for facilitating easy installation of software, demo applications (Figure 5b) and datasets has been prepared and tested. CD-ROM containing the necessary software for implementation of MapServer and deployment of WMS compliant Web Mapping application has been prepared for distribution along with the training document. The CD-ROM contains demo application and datasets and electronic version of the "Training Notes on Spatial Data Sharing using Open Source and Free Software" [4] enabling the users to learn by examples.



(a) (b)

Figure 5: (a) One-click installation (b) Demos and WMS test-bed at Osaka City University

2.4 Development of training material

Training material [4] was that facilitates hands-on exercise and self-paced learning for novice users were developed. The training document accompanied with 4 CD-ROM that provides users with a complete set of tools to develop, customize and implement their own applications. The training material includes instruction for installing of the WMS and configuring the Web server and MapServer toolkit. It would also contain instructions for interacting with the demonstration applications and a step-by-step introduction to the MapLab Web mapping toolkit. Details for deploying and testing WMS compliant Web mapping applications are also included in the training material. The training material have been used in several international and national training programs and were found the material to be very useful for getting an overview of the technology and developing their own applications with minimal efforts. The training document has been released under the terms and conditions of the GNU Free Document License Version 1.1 (http://www.gnu.org/licenses/ fdl.html) or any later version released by the Free Software Foundation. Japanese language translation has also been developed. Updating of training material and translation to other languages is also in progress

3 Conclusions

The WMS test-bed as a part of this research affords easy, rapid, standardized dissemination of spatial information. Since system has been developed using FOSS, it would be greatly beneficial to organizations that might neither have the financial resources nor the inclination to choose proprietary solutions. The system provides necessary framework for implementing WMS compliant servers in a distributed environment and facilitates sharing of spatial data. Further, the training material prepared as a part of this project could foster capacity building in Web mapping technologies and thereby facilitate the development spatial data infrastructure. Such efforts will help coordinate better strategies for sustainable development of our natural and social environment. As a future work, it is needed to incorporate OGC Web Feature Server (WFS) functionality and metadata search capabilities in order to provide an efficient mechanism for the creation, maintenance and delivery value-added spatial information.

Acknowlegement

The authors are thankful to the Remote Sensing Technology Center (RESTEC) of Japan, for supporting this project. This study is partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, Grant-in-Aid for Scientific Research (c)(2) (14540430), 2004.

References

[1] DCGM III Working Group. Urban Geoscientific Data of East and Southeast Asia – GIS data sets of 11 cities, Second Edition, Digital Geoscience Map G-8. *Geological Survey of Japan*, CD-ROM. , 2001.

[2] Nemoto T., Raghavan V., Masumoto S., and Shiono S. Development of SISGeM - An Online System for 3D Geological Modelling. *Geoinformatics* 11(2), pages 211-223. , 2003.

[3] Raghavan, V., Santitamnont, P., Masumoto, S., and Honda, K.. Implementing Web-GIS and Developing Spatial Data Infrastructure Using Open Source Software. *Proc. GIS-IDEAS Symp., Hanoi*, pages 207-212., 2002.

[4] Raghavan, V., Santitamnont, P., Masumoto, S., and Nemoto, T.. Training Notes on Spatial Data Sharing using Free and Open Source Software. *GRASS/Mapserver i18n Project, Japan*, pp. 193., 2004.

[5] Yoshida. D., Raghavan, V., Kuwahara, K., and Yao, A.. Implementing Spatially Enabled Bibliographic Database using Open Source Software. Proc. The Regional Conference on Digital GNS, Bangkok, pages 225-229., 2003.