

INTERFACE FOR DISSEMINATION OF GIS APPLICATION ON INTERNET

M. Koehl *, J. Joseph

INSA de Strasbourg, Graduate School of Science and Technology, Photogrammetry and Geomatics Group,
MAP-PAGE UMR 694, 24 Boulevard de la Victoire, F-67084 Strasbourg, France –
mathieu.koehl@insa-strasbourg.fr

Commission IV, WG IV/5

KEY WORDS: Internet GIS, Data Dissemination, Computer Vision, Spatial Analysis, Geography.

ABSTRACT:

Many cartographic applications on Internet offer very efficient toolsets which allow to perform basic tasks, i.e. data up- and download, access to data related attributes, geo-referenced data representation. Other applications provide the possibility to add geometric features and even to perform searches and access to local and distributed geospatial catalogues. However it is difficult to find GIS applications on Internet that allow to perform complex queries interactively. The main objective of the project described in this paper is to study and develop an Internet GIS application which allows the users to work interactively on the composition of a map that contains layers built from complex queries.

RESUME:

Les applications cartographiques sur Internet comportent des outils de base très performants comme, par exemple, le téléchargement et le transfert de données, l'accès à des attributs associés, la représentation des données géolocalisées. D'autres applications fournissent des outils permettant de rajouter des données géométriques, d'accéder à des données stockées localement ou à des catalogues de données distribuées. Néanmoins, il n'existe pas encore beaucoup d'applications de type SIG sur Internet permettant notamment d'effectuer des requêtes complexes de manière interactive. Le principal objectif de ce projet est l'étude et le développement d'une application de type SIG sur Internet permettant à l'utilisateur d'accéder de manière simple et interactive à des outils de composition de cartes comportant notamment des couches résultant de requêtes complexes.

1. INTRODUCTION

1.1 Problem

With the development of the computing technologies, geographic information system (GIS) has been taken an increasing importance in the day-to-day life. GIS can now be considered as an essential tool for analysis and assistance in decision process. It is very powerful and makes different users dependent to such tools.

The globalization comes to confirm this GIS dependence. The world looks smaller with the very fast dissemination of the knowledge and the information through the development of the Web. To make this stream of geographical information accessible to the most users, it is necessary to offer to private users basic tools for distant consulting, modifying and editing of geographical data. Following their needs, such tools should be available through the Web.

To achieve this purpose, it is essential to have adapted equipments, qualified users as well as sophisticated software. Commercial tools are often very complete, but still expensive for a basic use. In this point of view, the large development of Open Source and free GIS offers interesting alternative. All users should have access to GIS tools for publications, management of the geographical data. Although the acquisition of the Open Source software doesn't cost anything, on the other hand a qualified users and adapted equipments are requested.

1.2 Objectives

There exist various possibilities for the conception of a Web GIS application. In this case, the choice of a server-side application has been made. This application works in Open Source environment by using more particularly the cartographic server *MapServer* and the spatial *PostgreSQL / PostGIS* DBMS.

More generally, three fundamental objectives have been followed:

- Studies of different available GIS applications proposed by the Open Source community on Internet. client-side, server-side and hybrid applications were compared.
- Study of *PostgreSQL / PostGIS* DBMS and possibility of performing interactive queries on a geographic map.
- Understanding and installation of a GIS application coupled with data managed by *PostgreSQL / PostGIS* and allowing SQL queries with spatial data.

To take this problem as a whole, it has been focused on the objectives of creating a GIS application by using the *Apache* server, *MapServer* and *PostgreSQL / PostGIS* DBMS. This efficient combination offers the possibility to create and perform spatial queries on geographic maps from a distant client.

1.3 Methodology

The project has been organized in four parts:

The first part introduces the project and defines the objectives. The second part is mainly centered on a bibliographic study. It seems to be important to give some definitions of all different existing concepts of GIS applications on Internet. Different accomplished studies in this field give interesting indications and comparisons of available tools.

The third part is subdivided into two groups: the first dedicated to the configuration and installation of the server *Mapserver* and database *PostgreSQL / PostGIS* DBMS package, a second concerning mainly understanding and development of application.

And finally the fourth part gives the conclusion of the project as well as some recommendations to improve the developed application.

2. STATE OF ART AND ANALYSIS OF EXISTENT SOLUTIONS

2.1 Definitions

With the intention of a better understanding of the objective, it is important to analyze some existing well known solutions proposed in the Web GIS field. The definition of the main concepts is necessary and given in the following paragraphs extracted from (Hächler, T., 2003).

2.1.1 Web map

Basically, every map available on the Internet can be called a web map. This includes the simplest version of a scanned paper map that is integrated in a website. Web maps can also include some classical GIS functionality (like querying attribute data by clicking the map), which expands the classical understanding of cartography as a process unrelated to data handling.

2.1.2 Web mapping

The creation, distribution and use of web maps are called Web Mapping and imply the process of creation rather than the product itself or the application that has been used. Visualization aspects are the main subject of mapping.

2.1.3 Web GIS

A Web GIS application includes some more GIS functions than a Web Map. An important difference is the access to attribute data. The user should be able to perform functions like attribute queries, search functions, area and distance measuring or the construction of buffer zones.

2.1.4 GIS functions

All GIS applications contain a group of functions which allow to qualify them as real GIS. This group contains about twenty functions classified in six categories: *Search, Location Analysis, Terrain Analysis, Distribution / Neighborhood, Spatial Analysis and Measurements*. A list of GIS analysis functions consisting of: *Measure, Query, Reclassification, Overlay, Buffering, Neighborhood, Spatial Interpolation, Terrain Analysis, Networks*, can also be considered as the basic functions requested in every GIS application. However, not all GIS functions are suited for all kinds of data and projects.

2.1.5 Cartographic functions

The most important cartographic functions are *symbolology changes and management* and *layer control*. Besides, *exploratory data analysis* can also be considered to be a cartographic function.

2.1.6 MapServer

MapServer is an Open Source development environment for building spatially-enabled internet applications. *MapServer* is neither a full-featured GIS system, nor able to aspire to be one. Instead, *MapServer* excels at rendering spatial data (maps, images, and vector data) for the web. Beyond browsing GIS data, *MapServer* allows to create "geographic image maps", that is, maps that can direct users to content. The same application serves as a "map engine" for other portions of the site, providing spatial context where needed.

MapServer was originally developed by the University of Minnesota (UMN) ForNet project in cooperation with NASA and the Minnesota Department of Natural Resources (MNDNR). Currently, the *MapServer* project is hosted by the TerraSIP project, a NASA sponsored project between the UMN and consortium of land management interests.

The software is maintained by a growing number of developers (nearing 20) from around the world and is supported by a diverse group of organizations that fund enhancements and maintenance. (Mapserver, 2008).

Mapserver's main features are:

- Advanced cartographic output
- Support for popular scripting and development environments
- PHP, Python, Perl, Ruby, Java, and C#
- Cross-platform support
- A multitude of raster and vector data formats
- Map projection support

2.1.7 Interoperability

The interoperability is a standardization of the transmission of data, that is the ability of several systems to communicate even if their natures are different. Besides, the interoperability allows to favor the choice, competition and its innovation, and an opened access to information.

It also favored the development of free and Open Source GIS software opening the way to the expansion of geographical information.

2.1.8 Interactivity

From a systematic point of view, interactivity is a relation between two systems in which changes in the first one change the behavior of other one. From this basic definition, interactivity implicates a duality. In an interactive system there is the necessity of two parts, defining so a receiver and a transmitter.

An interactive application can be defined as a more or less opened system, composed of elements in dynamic correlation and organized in perspective to allow the user to attain a more or less predetermined purpose in a relation of reciprocal

exchange fulfilled at the level of interface by a defined potential of action.

3. STUDY AND AUTOMATIZATION OF MAPFILE

To be able to take into account the cartographic legends, all queries and to show all different layers with a specific symbology in the created geographic map, it is essential to store these component elements in a specific file called the Mapfile. Thus, the Mapfile becomes the key file of applications using the cartographic *MapServer*. Study of Mapfile seemed essential for understanding the functioning of the system.

3.1 Mapfile

ASCII file containing the general shape of the geographic map, the Mapfile can be acquired by two manners: from a manually created text file or by using free software allowing to export the data and structure of the geographic map in a Mapfile.

(Amein, 2007) for *ArcGIS 9.x*, (Wortomap, 2005) for *MapInfo 7.8* are examples of such tools. Besides, the edition of the Mapfile can be made from the module *MapEdit* of (*Maplab*, 2008).

Mapfile has a hierarchic organization based on blocks or objects inside a main block, the block MAP, and every block begins with its name and ends by keyword END. Figure 1 shows the structure of Mapfile.

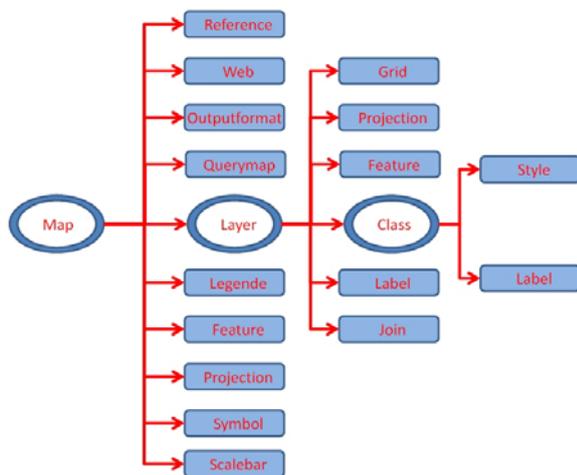


Figure 1. Hierarchic structure of Mapfile

3.2 PHPMapscript and Mapfile

The *PHPMapScript* module is a PHP dynamically loadable module that makes *MapServer's MapScript* functions and classes available in a PHP environment. (*MapScript*, 2008)

With functions of *PHPMapScript*, Mapfile becomes less heavy because the functions allow to create some elements or to call them even if not available in the Mapfile.

3.3 Dynamic Mapfile creation

Mapfile is the necessary element in the production of the geographic map with the proposed software architecture. The

objective is to construct manually, but in an assisted way the initial Mapfile. A form has been used to request the different parameters which allow the operator to define the manner its geographic map will be displayed. When all requested parameters are entered, the user has only to apply the form and PHP program will create the relevant Mapfile. When Mapfile is created, it is possible to generate a geographic map with different layers, to display or not each of them.

A method *post* is used to recover parameters from the form (see Figure 2).

Then, in a second step, the use of the dynamically produced geographic map can need some changes in the map representation. A symbology change of a layer representation is a simple example of what is often needed. To do that, the advanced user can edit the existing Mapfile and modify the corresponding parameters. The not expert user will use a form to indicate the new values for the symbology parameters. By applying the form, the Mapfile is updated. Any possible kind of modifications of the geographic map can in this way be stored in the new generated Mapfile. But more complex changes are expected if not all features of a same layer have to be displayed, or if different symbologies have to be used. To realize such more complex dynamic modifications, the Mapfile has to integrate queries in the definition of its layers. This is possible with *Mapserver*, spatial queries are possible with the combination of *PostgreSQL / PostGIS* DBMS.

```

//debut du bloc barre d'echelle
if($_POST['sta_ech']=="ON")
{
  $scalebar["n scalebar"]="";
  if($_POST['bar_ech_1']=="||$_POST['bar_ech_2']=="")
  {
    $scalebar["n size "]=$SESSION['bar_ech']=$_POST['bar_ech_1'];
  }
  $_POST['bar_ech_2'];
}
if($_POST['nb_sub_ech']=="")
{
  $scalebar["n intervals "]=$SESSION['nb_sub_ech']=$_POST['nb_sub_ech'];
}
$scalebar["n style "]=$SESSION['sty_ech']=$_POST['sty_ech'];
$scalebar["n status "]=$SESSION['sta_ech']=$_POST['sta_ech'];
$scalebar["n units "]=$SESSION['uni_ech']=$_POST['uni_ech'];
}
//fin du bloc barre d'echelle
foreach($Scalebar as $Cle => $valeur)
{
  fwrite($handle, strtoupper($Cle).$valeur);
}
    
```

Figure 2. Method *post* for scalebar parameter recovering

4. POSTGRESQL / POSTGIS DBMS

Today, there exist a lot of DBMS. As Open Source DBMS *MySQL* and *PostgreSQL* are the most famous. In this project, it is mainly focused on the *PostgreSQL* and its geo-spatial extension *PostGIS*.

4.1 PostgreSQL

(*PostgreSQL*, 2008) is a powerful, Open Source RDBMS. It has more than 15 years of active development and a proven architecture that has earned a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX and Windows. It is fully ACID

compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL92 and SQL99 data types. It also supports storage of binary large objects, including pictures, sounds, or video.

PostgreSQL / PostGIS is nowadays part of object-oriented DBMS and is considered as one of the most competitive Open Source spatial DBMS.

4.2 Spatial component PostGIS

(*PostGIS, 2008*) 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, much like *ESRI's SDE* or *Oracle's Spatial extension*. *PostGIS* follows the OpenGIS "Simple Features Specification for SQL" and has been certified as compliant with the "Types and Functions" profile.

PostGIS has been developed as a project in Open Source spatial database technology. *PostGIS* is released under the GNU General Public License. *PostGIS* has now user interface tools, basic topology support, data validation, coordinate transformation, programming APIs and much more. Coupled with the analysis functionalities of *PostGIS*, the *PostgreSQL* DBMS allows to perform geometric queries in 2D, 3D or 4D environments. The DBMS gives great possibilities for the integration of standard SQL queries. Through its spatial cartridge, links can easily be established between the DB and the Mapfile, interpreted by *PHPMapScript*.

4.3 Shapefile and PostgreSQL/PostGIS database

In addition, *PostGIS* includes two converters *shp2pgsql* and *pgsql2shp* which allow respectively to import and to export data into the Shapefile format (*ESRI* format for vector data, very widely used in GIS data exchanges). The importation and conversion of shape data into database allow to create a geometric column used by *PHPMapScript* for the display of the feature geometry.

4.4 PHPMapscript and PostgreSQL / PostGIS DBMS

To complete successfully this project, the *PHPMapScript* library method was selected because it enhances the performance of *Mapserver* on its dynamic level. Indeed, this module embedded a functions library which uses *Mapserver's* C functions to create dynamically the required well documented Mapfile.

PHPMapScript allows to read Shapefiles mentioned in the Mapfile and to show as layers into the resulting geographic map. However, the access to data stored in DBMS is not possible in the same way to data structured as Shapefiles. While the access to the Shapefile is a direct access, the access to the geometry stored in the DBMS is possible through the key DATA (Figure 3). In the Mapfile, DATA key allows to insert a SQL query. The result of the SQL query can be the whole layer or only some features from the layer responding to the WHERE criteria of the SQL statement. All spatial and semantic combinations can be used to perform complex queries and in this way to restrict the feature set to be displayed in the resulting geographic map.

```
CONNECTIONTYPE POSTGIS
CONNECTION "user=postgres dbname=france password=postgre host=localhost"
DATA "the_geom from (select * from departements where nom_dept like 'B%') as foo
USING UNIQUE gid USING SRID =27582 "
```

Figure 3. Extract of Mapfile structure using DATA key

4.5 Interest in PostgreSQL / PostGIS for mapping

Once data have been imported in the *PostgreSQL* DBMS, a "the_geom" field is created. It is this column which contains the geometry of the features. All queries linked in the Mapfile will concern this column. Thus, *PostgreSQL* gives the possibility to manage geographic features with geometry and semantic description. It also allows to make complex queries combining geometry, semantic description and topology.

However, it is necessary to notice that queries on *PostgreSQL* through *PHPMapScript* require a particular syntax.

From now, the project has to take into account the *PostgreSQL / PostgreSQL* DBMS as well as Shapefile structures. The two types of data sources will also allow different possibilities to query the data.

To conclude, it is useful to notice the following elements:

- *PostgreSQL / PostGIS* is *MapServer* compliant.
- The object DATA in the LAYER block is the key object for connection to *PostgreSQL / PostGIS* DBMS.
- The *psql* monitor is very helpful. It is a good way to connect and to manage *PostgreSQL* DBMS in a dynamic way.
- *PostGIS* allows to integrate 2D, 3D and 4D spatial data. These spatial data can be exported in Shapefile format.

5. APPLICATION DESIGN AND IMPLEMENTATION

Once the *Mapserver* and *PostgreSQL / PostGIS* DBMS being installed, it is possible to design and to develop a specific application to help the final user in his try to compose a complex map.

5.1 Data structure

To organize the project, different folders were created. Datafiles were stored in corresponding folder to make the data manipulation easier (Figure 4).

```
\bak: for backups
\bal: for batch files
\css: for css cascading style sheets
\html: for HTML files
\js: for javascripts
\map: for Mapfiles
\php: for PHP files
\phtml: for PHTML files
\shp: for ESRI Shapefiles
\tmp: for temporary files
\img: for image files like JPEG, GIF, TIFF
\ogr: for Bentley design files .dgn
\wms: for Web Map Services connection parameters
```

Figure 4. Project file structure

5.2 Application

The application has been designed at server-side. Following factors were decisive in the different choices made:

- *PHP* includes a library of very rich functions,
- The choice of *MapServer* as most used cartographic server,
- The library of functions *PHPMapScript*.

Since the interactivity is not guaranteed by *PHP*, some functions have been developed in *JavaScript* and installed on client-side allowing so to improve the application of the server-side. There exist different solutions to design an interface on Internet: div, frames, tables. The use of tables has been chosen because the developed application doesn't need a great amount of memory.

The application passes through different stages from the homepage to the visualization of geographic map.

The homepage proposes to the user four functionalities:

- Edit the Mapfile: create one manually (without assistance), editing and modifying parameters on existing one,
- Transfer Shapefiles to *PostgreSQL* DBMS,
- Open display of geographic map if Mapfile available,
- Create Mapfile (with assistance).

5.2.1 Functioning principles

Final objective has been to create a Mapfile allowing to display data at the end of process. Figure 5 illustrates the functioning principle of the developed application.

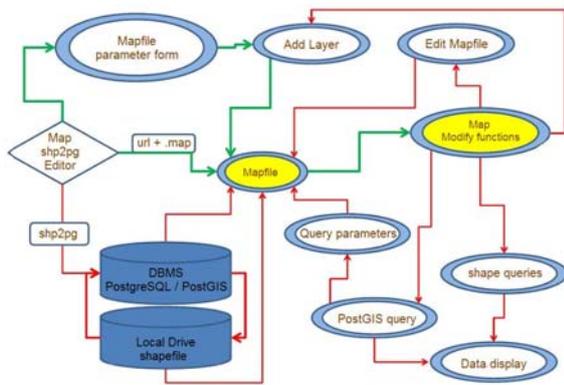


Figure 5. Application workflow

5.2.2 Structure of implemented application

Mapfile editor

The Mapfile editor consists in a script allowing to edit/modify the Mapfile parameters (text file). The objective of this module is to allow the user to create his own Mapfile connected to available data on local disc or in a DBMS.

Mapfile creation

The process of Mapfile creation includes two stages. The first stage consists in valuing the fields of a first form (Figure 6) allowing to create the header, to assign spatial references, to define scalebar and the legend of the Mapfile.

The second stage consists in valuing fields of a second form to be able to add new layers. Other data can be added afterwards by editing the Mapfile or by adding directly a new layer.

Figure 6. Mapfile creation form

5.2.3 Application functions

According to (Hächler, 2003) it is not sufficient to have an online map with functions of zoom and displacement to define Web GIS. Thus, a group of functions allowing to manage the map and to access to more information has been added.

Functions

The functions for panning, zoom in/out and zoom to extend view are classical one. All these functions are calculated according to the geographical extend of the map defined in the Mapfile. Besides, the application of these functions on the map has the same effect on reference map, that is the corresponding zone in the map will be visible in reference map.

Additional functions like reference map, map plotting, cursor position coordinates are also implemented.

Legend creation process

The essential element for the understanding of a geographic map is the legend. The application includes a module for the design of the map's legend.

The ergonomics of the legend displaying is assured by *PHPMapScript* and simplicity to be able to convert *PHP* code into *HTML*.

In effect, for the legend creation, available *PHPMapScript* scripts read the Mapfile and extract from it the number of layers as well as the style parameters for layer symbology. Thus, the corresponding icon to each layer is created and *PHP* code is converted into *HTML* to show legends. (Figure 7).

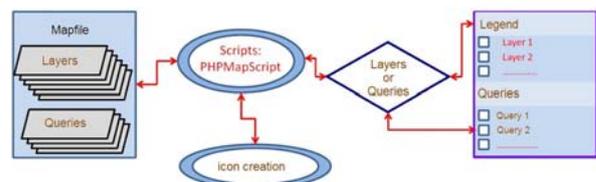


Figure 7. Legend creation process

6. QUERIES

With a drawn of a general geographic map the user gets an overview. However, the user's interest is to be able to acquire information on objects displayed in the map. The legend gives general information. But to get the particular attributes of a specific feature the legend is not sufficient. With the perspective to allow the user to have information on each feature of the map, two types of queries are available: queries on Shapefiles and those on *PostgreSQL* Data.

6.1 Shapefile queries

Queries on Shapefiles allow to get the attribute values of a specific feature which geometric position correspond to where a click was performed.

Moreover, other solutions were explored in order to change the displaying color of objects queried in the map...

6.2 PostgreSQL / PostGIS queries

PostGIS queries are the most efficient. In fact, *PostgreSQL* DBMS supports any kind of SQL queries once the syntax respects the rules. Thus, almost all possible *PostGIS* queries can be inserted into the Mapfile to construct the resulting layer.

To allow the display of the queried objects, a new form has been designed to define SQL queries (Figure 8) directly in the map. These queries has been performed in *PostgreSQL* DBMS. To take action, it is necessary that there exist a layer in the Mapfile allowing to show the result of this query (Figure 9).

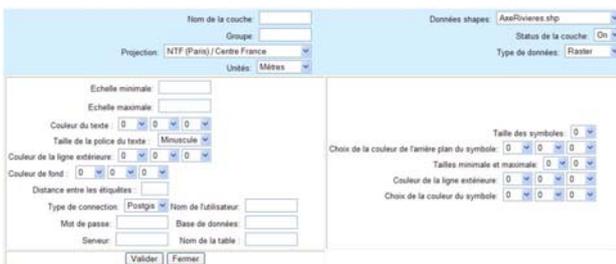


Figure 8. Query layer form

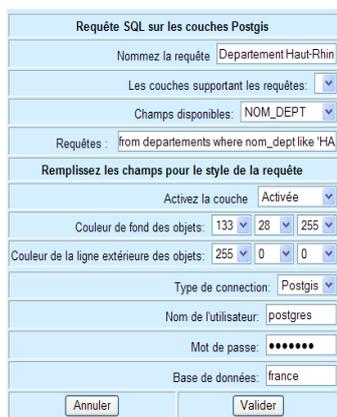


Figure 9. SQL query builder

6.3 Queries on query resulting layers

Application was conceived so that all performed queries are recorded in a new table in DBMS. This new table will be created with the name given for the layer.

The new table created with the query parameters can afterwards be the basic layer for a new query such as illustrated in the Figure 10. Now SQL queries can be implemented in cascading mode. At first, a new layer is constituted by the geometric and semantic queried data. Then it can be used as the next reference layer for a new sub-query that handles on it.

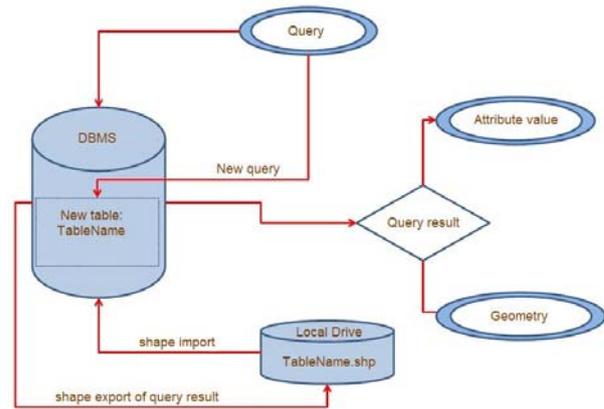


Figure 10. Query building workflow

7. CONCLUSION

In the perspective of the extension of the mapping on the web, the dissemination of geographical information appears to be a necessity. Groups, research departments, research centers concentrate more and more interest for the web mapping since they have taken in account the possibilities that this last offers. Acquired results allow to point out not only what was made but also what it remains to be made.

Moreover, these jobs give to the users a synthetic vision of the available tools and their potentialities and a possibility of being able to perform complex requests.

Delivered application remains in the state of prototype, but allows to test the great potential of spatial queries as part of Internet GIS. In the same time, it allows to work interactively with the GIS in the Internet environment.

REFERENCES

- AMEIN, 2007, <http://www.mapbender.org/index.php/AMeIN>
- Bernier, E., Bédard, Y., Hubert, F., 2005. UMapIT: An on-demand Web Mapping tool based on a multiple representation database. In: *8th ICA Workshop on Generalisation and Multiple Representation*, A Coruña, July 7-8th 2005, 9 pages.
- Desjacques, J., 2005. Visualisation et édition d'une carte dynamique sur le Web au laboratoire COGIT de l'IGN. In: *Diploma-thesis*. Department of Surveying, INSA de Strasbourg. 60 pages.

Formats-ouverts, <http://www.formats-ouverts.org>

Geomatica, 2007. Presentazione di MapServer, Marco Negretti, http://geomatica.como.polimi.it/corsi/sw_gis/mapserver.pdf

Hächler, T., 2003. Online visualization of spatial data. A prototype of an open source internet map server with backend spatial database for the swiss national park. In: *Diploma Thesis*. Department of Geography, University of Zürich. 119 pages.
internetgis, 2004. *PostGIS / MapServer*, http://www.internetgis.net/documents/mapserver/MapServer_Users_Meeting-2/PostGIS-Mapserver.doc

MapFile Reference – MapServer 3.7. http://fa.vdb.free.fr/MapServer/doc/mapfile-reference_fr.html
MapLab, <http://www.maptools.org/maplab/>

Mappemonde, 2005. La cartographie en ligne ou Web mapping : les outils “libres“, état des lieux des solutions

“applicatives“ autour de *MapServer*. <http://mappemonde.mgm.fr/num8/internet/int05401.html#A2>

Mapscript, http://www.maptools.org/php_mapscript/

Mapserver, 2008, <http://mapserver.gis.umn.edu/>

Mapstorer, 2007, <http://www.mapstorer.org/>

OpenWeb, <http://www.openweb.eu.org/>

PostGIS, <http://www.postgis.org/>

PostgreSQL, <http://www.postgresql.org/>

Tyler Mitchell, 2005. *Web Mapping Illustrated. Using Open Source GIS Toolkits*. First Edition June 2005, 367 pages.

