

An Efficient Web-GIS Solution based on Open Source Technologies: A Case-Study of Urban Planning and Management of the City of Zagreb, Croatia

Mario MILER, Drazen ODOBASIC and Damir MEDAK, Croatia

Key words: geoinformation/GI; Open Source; spatial information management; web-mapping

SUMMARY

This paper describes the progress made by the Open Source community in displaying maps on the web by FLOSS AJAX based framework (GeoExt, ExtJS and OpenLayers) and WMS-C (TileCache, Mapnik) and WFS (FeatureServer) server architecture. ExtJS is a popular client-side JavaScript library for building user interfaces. GeoExt adds extensions to ExtJS that bind basic ExtJS components to the spatial features of OpenLayers. TileCache intelligently store and serve map tiles using standard web protocols for requests and responses. Mapnik is an open source mapping toolkit for tile rendering. FeatureServer is an implementation of a RESTful Geographic Feature Service used for serving vector data using standard HTTP methods. Putting maps on the web alone can be very difficult, especially if there is a requirement that those maps are displayed at fast rate. In order to create such a service, specialized software and, what is more important, specialized knowledge is needed. People today got used to web mapping services like Google Maps or Bing Maps which display maps at very high rate and everything that works even little slower than those services is not satisfactory. Advanced users want to display and frequently edit the vector data, which makes the building of the system even more challenging task. We developed an original and efficient solution for visualization of a very large amount of spatial data on the web using Open Source technologies which combine tiled raster maps and vector WFS data. We present a case study of a fast web mapping solution for Urban Planning and Management of the City of Zagreb, Croatia. This approach represents a major step forward in spatial information management and it increased the capacity of small and medium enterprises to develop similar spatially-aware applications without high starting investment.

An Efficient Web-GIS Solution based on Open Source Technologies: A Case-Study of Urban Planning and Management of the City of Zagreb, Croatia

Mario MILER, Drazen ODOBASIC and Damir MEDAK, Croatia

1. INTRODUCTION

This paper describes the progress made by Open Source community in displaying maps on the web by FLOSS AJAX based framework ((Sayar et al. 2006), (Murray 2006). The team at the Faculty of Geodesy has already been active in this area (Medak et. al 2003), but recent improvements of available FLOSS tools (Hall and Leahy 2008) led to much more effective solutions. Several papers on FIG events stressed importance of the methodology (Holmes et al 2005), (Steudler 2009). In following sections we describe the background of Free Open Source Software in general and the software which is related to web-GIS. The tools which are used in the case-study are described in detail.

1.1 FLOSS – General Aspects

Free/Libre/Open Source Software or FLOSS is software with liberal license which grants users the right to use, study, change and redistribute software in modified or unmodified forms. Commercial software licenses usually restrict users of these rights allowing only restricted use in specific environment.

OSI approved licenses are not restricting any party from selling or giving the software away, but the software must include source code or a way to access the source code, like a publicly accessible web-site. Redistribution of modified software is only possible if modified software grants users same rights as original version of the software.

FLOSS development model differs from standard commercial closed source development model allowing users to directly participate and influence software development. Common examples for user's participation are writing code, documenting or bug reporting. Openness of development process allows anyone to comment and share ideas, which allows experts with different area of interest to collaborate leading to a better software product.

FLOSS enables reuse of source code, so rather than building from the scratch, projects can use existing source code reducing the development. Reusing and modifying FLOSS software also enables creation of much better end-user solutions, which provides better value for money than proprietary software.

FLOSS projects use existing standards and protocols for communication, storage, data transfer, etc. Open standards and protocols grant users more mobility and choices when choosing a software platform. Commercial software providers tend to use closed proprietary storage formats and protocols which lock-in users to their specific platform.

TS 5B - Information Management Software

Mario Miler, Dražen Odobasic and Damir Medak

2/16

Efficient Web-GIS Solution based on Open Source Technologies: Case-Study of Urban Planning and Management of the City of Zagreb, Croatia

FIG Congress 2010

Facing the Challenges – Building the Capacity

Sydney, Australia, 11-16 April 2010

European commission funded a project “Free/Libre/Open Source Software: Worldwide impact study” (FP6-IST-015722) which has included Croatian FLOSS users too. Research concluded that more skills are learned in the FLOSS community, and these skills are not only technical. For example, users in a FLOSS community have better understanding of legal rights and restrictions. The study concluded that even those with prior skills learn new skills and formal education rates poorly against wide variety of learning strategies available by FLOSS. Because of this FLOSS can be considered as a training environment which increases learning capacity of community participants without any explicit investment in training, which is a novel form of technology transfer.

1.2. Standardization efforts in GIS

The OGC was founded in 1994 as the OpenGIS Consortium. It is a non-profit, international, voluntary consensus standards organization specializing in geospatial data and Web services. The Open Geospatial Consortium (OGC) in 1997 became involved in developing standards for web mapping. They established a task force to come up with a strategy, and organized the "Web Mapping Testbed" initiative. The initiative resulted in pilot projects that were demonstrated in September 1999 which ended in April 2000. As the time passed by, OGC specifications found their way to most of the major GIS vendors. In 2007, 381 vendor products either implement or directly conform to OGC specifications (Open Geospatial Consortium 2007).

The Open Geospatial Consortium released WMS version 1.0.0 in April 2000, followed by version 1.1.0 in June 2001, and version 1.1.1 in January 2002. The last WMS version 1.3.0 was released in January 2004. Today, almost all of the open source and commercial web mapping solutions provide WMS capability. WMS is a widely supported format for maps and GIS data accessed via the Internet and loaded into GIS software, on the client side. In 2006, OGC released WMS-C (Tiled Maps) version which replaces arbitrary resolution Web mapping approaches with “tiled” maps, which can be managed in an underlying cache mechanism by software. This allowed the development of rich web mapping applications that were faster than previous WMS approach of rendering maps.

1.3 History of Web-GIS Development

In the 1994, the World Wide Web Consortium (W3C) developed interoperable technologies (specifications, guidelines, software, and tools) to facilitate minimal levels of conformity in Web standards. This marked the beginning of web mapping as standardization on the web was established. Core specifications include Hypertext Transfer Protocol (HTTP), the Uniform Resource Locator (URL), and Hypertext Markup Language (HTML), which have become building blocks for other W3C specifications such as the Document Object Model (DOM), Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), Scalable Vector Graphics (SVG), and Cascading Stylesheets (CSS).

Today, many popular internet sites have some kind of geospatial data pinned to them, but in

the past it was not so common. MapGuide and MapQuest are the pioneers of web mapping. In the late 1990s, MapQuest virtually owned the online web mapping market. In the following years many other products emerged, like Geomedia WebMap, Sharp Map, ka-Map, MapServer, Geoserver, MultiMap, ArcIMS, OpenStreetMap, Google Maps, Bing Maps, Yahoo Maps etc. Today we have large variety of web mapping tools available. Some of them are commercial but most of them are free and open to general public. The most popular products are Google Maps maintained by Google Inc. and Bing Maps maintained by Microsoft Inc. There is a very limited capability of displaying your own spatial data by their application programming interfaces.

2. METHODOLOGY

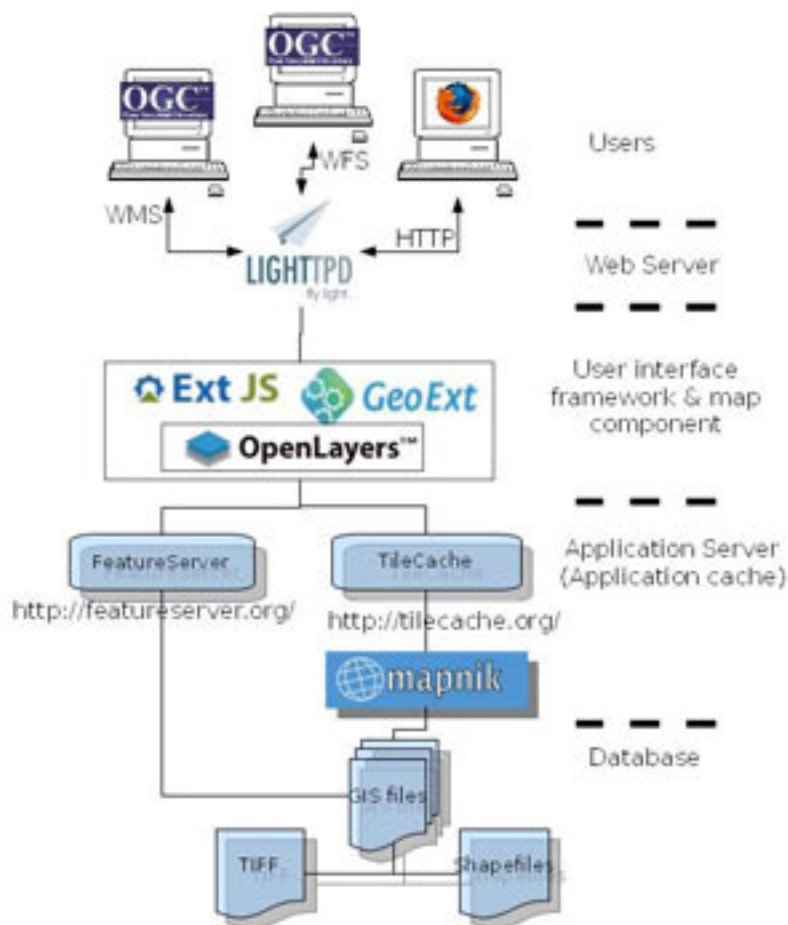


Fig. 1: Flowchart of Open Source technologies used in the Case-Study.

2.1 Mapnik

Mapnik is an open-source rendering library written in C++ allowing users to render spatial data using styles and rules to raster (PNG) or vector (SVG) output formats. The Mapnik

project was started by Artem Pavlenko who released it as open source under LGPL license. Mapnik has Python bindings that support fast-paced agile development and enable easier integration with other Python-based FLOSS GIS tools.

Input format plug-in architecture enables usage of various spatial data formats like PostGIS, ESRI Shape, TIFF, Oracle Spatial, SpatialLite. Mapnik can render all of the data formats supported by GDAL/OGR which allows rendering of almost all known data formats even some closed proprietary data formats.

For raster rendering Mapnik uses AGG (Anti Grain Geometry) library or Cairo library for rendering in SVG, PDF and other formats supported by Cairo. Final output is defined in an XML file which defines styles and rules for layers of spatial data. A layer defines a data source that will be rendered and which style will be used. Each style is using rules to define how Mapnik is going to render the particular data source. For instance, polygons will be rendered in yellow with 50% opacity if attribute value 'area' is larger then 1000 squared km. Mapnik is using a painter algorithm which paints current layer over the previous one. Thus, the order of layers is important.

The largest project that uses Mapnik for rendering of vector data set is OpenStreetMap. OpenStreetMap uses Mapnik as a default renderer producing beautiful maps for everyone.

2.2 FeatureServer

FeatureServer is an implementation of RESTful Geographic Feature Service released under BSD OSI approved license. It enables users to fetch a representation of a geospatial feature or a collection of features, add new data or delete data using standard HTTP methods. Simple REST interface enables easier interaction with FeatureServer.

FeatureServer is implemented in Python programming language and enables storage and retrieval of geospatial features from various data sources. Some of the data sources are DBM, BerkeleyDB, PostGIS, WFS, etc. Geospatial features can be managed using one of the following formats GeoRSS, KML, GML, etc. All of these formats are approved and accepted by OGC enabling true information interchange.

Although it can function as a standalone WSGI server, it works best behind an HTTP server as a CGI or FastCGI service. FeatureServer's light footprint enables its usage in resource-restricted environments and it integrates easily with OpenLayers/GeoExt.

2.3 TileCache

TileCache implements WMS-C protocol that enables access to spatial data using fixed image width and height referenced to some fixed geographic grid at fixed scales, also called tiles. TileCache is written in Python under BSD license and provides WMS-C/TMS server with pluggable caching mechanisms and rendering backends.

TS 5B - Information Management Software

Mario Miler, Dražen Odobasic and Damir Medak

5/16

Efficient Web-GIS Solution based on Open Source Technologies: Case-Study of Urban Planning and Management of the City of Zagreb, Croatia

FIG Congress 2010

Facing the Challenges – Building the Capacity

Sydney, Australia, 11-16 April 2010

TileCache caches WMS-C/TMS requests and stores tiles to one of selected caching backends: a disk which stores tiles on a hard disk or memory which stores tiles in memcached service. Mapnik and Mapserver are two rendering backends which can be used by TileCache using Python bindings. TileCache can also connect to remote WMS services and locally cache fetched tiles.

Although TileCache can fetch and cache tiles per request, it can also be used to pre-fill cache repository with tiles speeding up server responses to client requests. Similar to FeatureServer, TileCache can function as a standalone WSGI server but it works best behind an HTTP server as CGI or FastCGI service.

2.4 Lighttpd

Lighttpd is a secure, fast, compliant, and very flexible web-server that has been optimized for high-performance environments. It has a very low memory footprint compared to other webservers and takes care of CPU-load. Lighttpd was first released in March 2003 and today it is running big traffic sites such as Youtube and Wikimedia. Webserver receives client request and responds, returning the result, usually a HTML page that can be displayed in an internet browser. Lighttpd is specially optimized for a large number of parallel connections, making it perfect for utilization in a Web-GIS environment. Lighttpd also supports features like FastCGI and load-balancing which are used to serve FeatureServer and TileCache requests.

2.5 OpenLayers

OpenLayers is an Open Source map viewing framework, written in pure JavaScript. The OpenLayers framework provides a JavaScript application programming interface (API) which makes it easy to incorporate maps from a variety of sources into your webpage or application, similar to Google Maps and Bing Maps. OpenLayers currently has support for OGC WMS, WFS, GML and many other layers, navigation, icons, markers, and layer selection. As open source code with a practical respect for standards, OpenLayers offers an open market for all geographic data. OpenLayers is agnostic of server-side technologies. It allows anyone to rapidly construct applications using layers from different providers of georeferenced data.

2.6 ExtJS

ExtJS is a client-side JavaScript framework for building Rich Internet Applications. The ExtJS JavaScript framework allows developers to easily implement desktop functionality on the web. It provides a base framework for manipulating Document Object Model (DOM) elements, making AJAX requests, managing events and listeners, and best of all, creating object that are only familiar in desktop user interfaces like tree interfaces, drag and drop functionality, interactive data grid and more. ExtJS was originally built on Yahoo's YUI! Framework, but has since abstracted the implementation of its base library.

TS 5B - Information Management Software

Mario Miler, Dražen Odobasic and Damir Medak

6/16

Efficient Web-GIS Solution based on Open Source Technologies: Case-Study of Urban Planning and Management of the City of Zagreb, Croatia

FIG Congress 2010

Facing the Challenges – Building the Capacity

Sydney, Australia, 11-16 April 2010

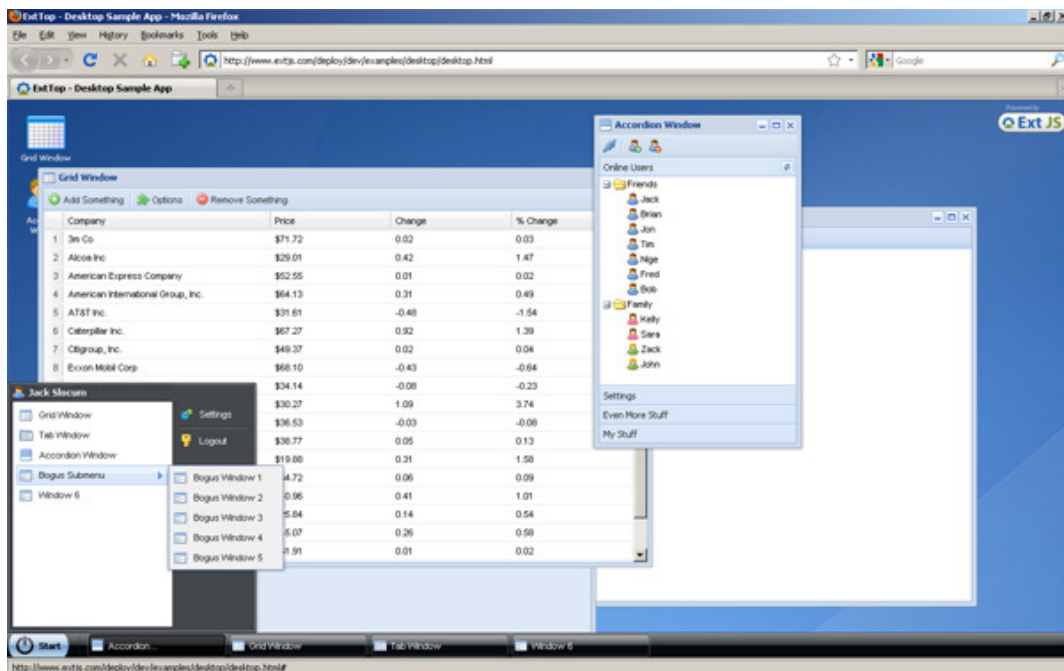


Fig. 2. An example of a desktop-like application in the web.

2.7 GeoExt

GeoExt is extensions to ExtJS that bind basic ExtJS components to the spatial features of OpenLayers that provides a ground-work for creating web-mapping applications. GeoExt brings together the geospatial know how of OpenLayers with the user interface of Ext JS to help build powerful desktop style GIS applications on the web with JavaScript. It is a set of components that allow ExtJS and OpenLayers to work together as a single spatial framework. For example, it bind together user interface components (button, grid, tables, tree lists etc.) of ExtJS with OpenLayers map component so that they interact with each other.

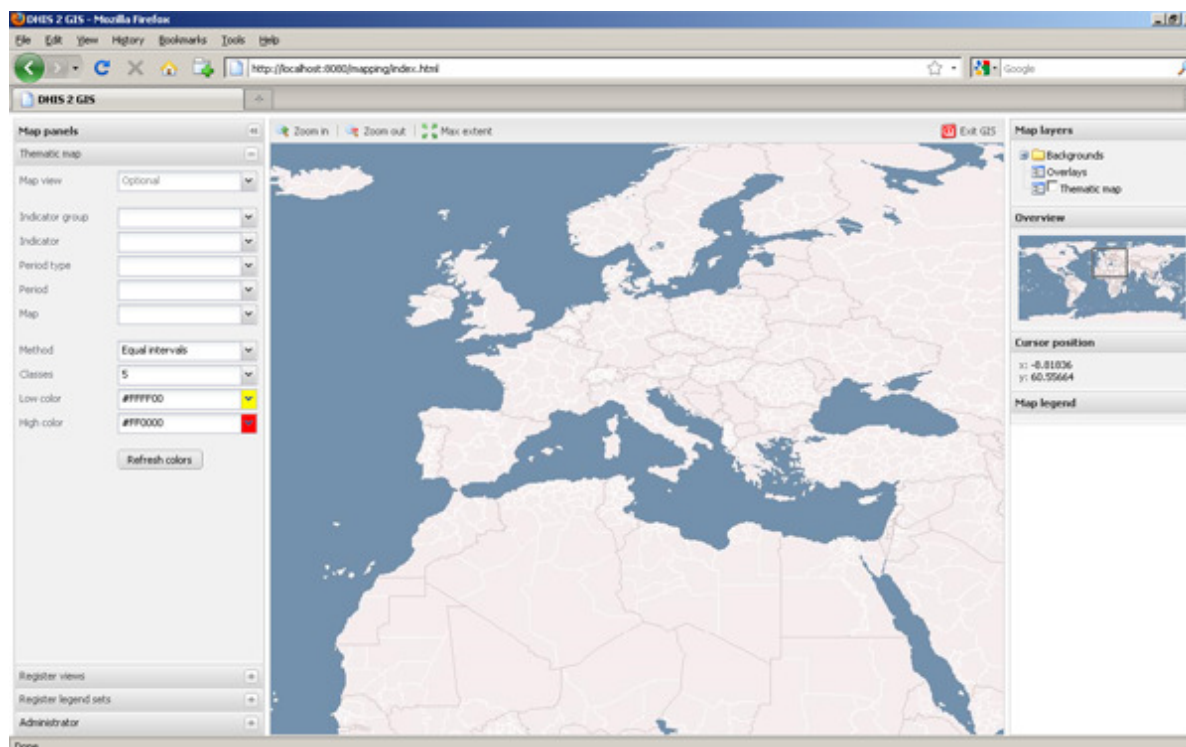


Fig. 3. An example of District Health Information Software 2 created with GeoExt.

3. CASE-STUDY: WEB-GIS FOR THE CITY OF ZAGREB

This section describes data-sources, modeling issues and visualization of the results in a web-GIS that is related to a urban planning and management spatial application prepared for the City of Zagreb.

3.1 Data-sources

Spatial data-sources can be categorized into two groups: raster data and vector data. Raster data sources are digital orthophoto images, HOK (Croatian base map), TK25 (topographic map). Raster data will be used for base layers which can be selected by users. Vector data sources are cadastral data, geological data (soil types and faults), geodynamic data, general urban plan (types of urban areas) and digital elevation model which will be used to create a color relief map.

Digital orthophoto images of Zagreb area are used as the default base layer, enabling users to have a good overview with enough details at maximal zoom level. Digital orthophoto images are stored in TIFF graphic file format. HOK is a topographic map also known as Croatian base map in scale 1:5000. TK25 is also a topographic map but in scale 1:25000. Both topographic maps are also stored in TIFF file format and they are created and maintained by State Geodetic Administration.

Cadastral data consists of cadastral boundaries and parcels with associated information about parcel numbers. Cadastral data is stored in ESRI shape file format and it will be used as overlay. Geological data includes information about soil types and location of faults which was available as a paper map. Therefore, it was necessary to vectorize it before it can be used. Geodynamic data is stored in comma separated text files, with information about movement of fixed points in the City of Zagreb metropolitan area. General urban plan is available in ESRI shape file format. It contains information for planned types of urban areas in the City of Zagreb like industrial, civil, cemetery or business areas also used as overlay. Digital elevation model will be used to create a color relief map that will be used as another base layer.

3.2 Data modeling

All of the data sources must be prepared before they can be used in Mapnik XML file as layers. During the modeling phase FLOSS tools like GDAL and ImageMagick were used. The Python programming language was used to automate several tasks.

Table 1: Spatial datasets used in the case-study

<i>Layer</i>	<i>Type</i>	<i>File size</i>	<i>Number of tiles</i>	<i>Total volume</i>
Digital orthophoto	Raster	600Mb	~1 400 000	~8Gb
HOK	Raster	400Mb	~1 400 000	~7Gb
TK25	Raster	300Mb	~1 400 000	~7Gb
Cadastral data	Vector	80Mb	~1 400 000	~5Gb
Geological	Vector	10Mb	~1 400 000	~5Gb
Geodynamic	Vector	30kb	GML	~40kb
GUP	Vector	5Mb	~1 400 000	~5Gb
DEM	Raster	300Mb	~1 400 000	~6Gb

First of all, data sources were converted to a single spatial reference system. This operation was not necessary because Mapnik and other tools allow on-the-fly re-projection, but conversion enabled easier integration of data for the final product.

Image file formats are processed with GDAL and post processed using ImageMagick tools: first, each of all images per dataset are merged into one big image that simplifies handling; secondly, this base image is used to create overview images, each with doubled resolution which are then used for faster tile rendering.

Spatial indexes are created for shape files, which are readable by Mapnik and FeatureServer. Datasets stored in text files are converted to GML which will be displayed as overlay layers. Digital elevation model was processed using GDAL Digital Elevation Model tools which

enable creation of color relief images. These images are then processed in the same manner as other image files.

After preparation of datasets, the next step is to include them as layers in Mapnik XML file. A style for each layer is defined, and each style contains one or more rules for feature rendering. Style contains rules which define how the feature is rendered. There are PolygonSymbolizer, LineSymbolizer and PointSymbolizer for rendering of polygons, lines and points, and the TextSymbolizer for text rendering. Filters are used to select a data subset that needs to be rendered in a specific way, for instance usage of a different foreground color for specific area type in the general urban plan.

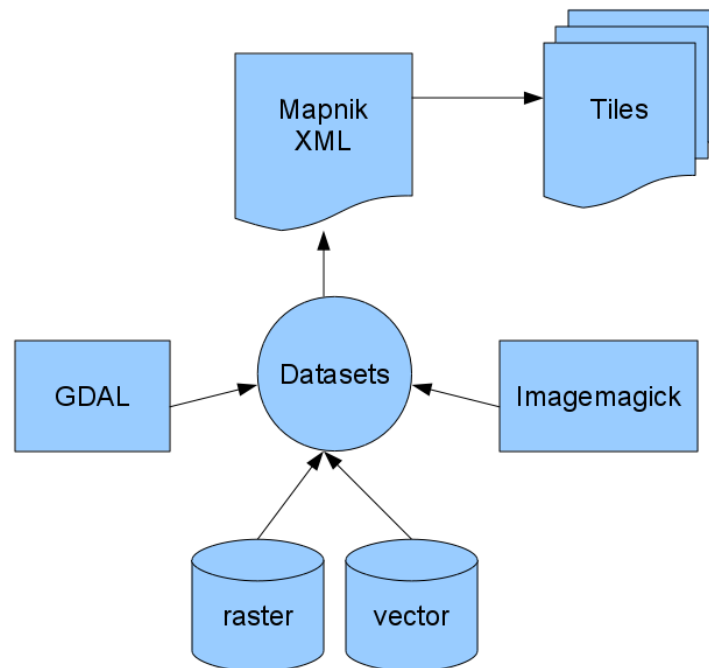


Fig. 4: Data modeling.

In total there is around 3GB of images, shape files and text files that are rendered by Mapnik into more than 10 millions tiles using up more than 50GB of disk space. As tiles are pre-rendered, Lighttpd responds to client requests in less than 5 ms per tile. With average of 75 tiles per client request it displays map in around 400ms. Without pre-rendering, a tile request can last more than 10sec depending on selected layers. Tile pre-rendering (also known as *seeding*) takes a relatively long time: seeding of all layers in our case took 36 hours. The seeding time depends only on computer processing resources. The procedure can be scripted and distributed at several computers, reducing the time to just a couple of hours. Pre-rendered datasets are relatively static with changes on a year or half year basis. Data with more dynamic nature can be served directly from dataset as GML layer.

3.3 Visualization

With web mapping developing rapidly, new web technologies are emerging. GeoExt (ExtJS) is one of them. Utilizing power of OpenLayers, GeoExt uses AJAX (Asynchronous JavaScript and XML) to display dynamic map content on the web. With the power of AJAX technology, web applications today can look and feel like any other desktop application without relying on plug-ins and browser specific features. Earlier web application developed without AJAX, had to be refreshed with every change of content which made browsing very slow, when compared to other products. Technologies such as JavaScript and CSS (Cascading Style Sheets) have evolved to the point where you can create any dynamic web content that will display on every major browser on the market. For example, Google Maps is using AJAX technology to display maps on the web, too. Google Maps also provide powerful application programming interface (API) to display its maps on users' web pages, but if a user wants to add her own data, or customize some user interface components it might be easier to use some other API. The Open Source community has developed many such tools, GeoExt being one of the recent.

What makes GeoExt unique is that it has a power to integrate OpenLayers map components with nice and appealing user interface of ExtJS. This is why we decided use this technology for the project. In comparison to some other mapping systems where tiles are cached/created the first time user displays the map, we rendered all tiles before, creating a huge file database of pre-rendered images. As all of our data was tiled and used with WMS-C specification, this was not suitable for any attribute or coordinate data extraction. This drawback was solved by using FeatureServer for attribute and coordinate data. FeatureServer was implemented to work as a WFS service and serve XML/GML data on request. WFS instances serve and store spatial and theirs properties. With request from any client, WFS instance, in our case FeatureServer, creates and sends XML formatted data in GML. Some of WFS instances provide transactional and lock feature services, but in our case that was not necessary because we only needed to read the data, not to modify it.

```

- <wfs:FeatureCollection xsi:schemaLocation="http://www.opengis.net/wfs
- <gml:featureMember>
- <fs:aaa fid="0">
- <fs:geometry>
- <gml:Point>
  <gml:coordinates>5595973.72,5067373.73</gml:coordinates>
</gml:Point>
</fs:geometry>
<fs:ADMAPKEY>1.0</fs:ADMAPKEY>
<fs:PO__ETAK_R>01.01.1878</fs:PO__ETAK_R>
<fs:KOTA_NULE_>95.61</fs:KOTA_NULE_>
<fs:KRAJ_RADA>-</fs:KRAJ_RADA>
<fs:__IFRA_POS>3096</fs:__IFRA_POS>
<fs:MJERI>DHMZ</fs:MJERI>
<fs:X__M_>5595973.72</fs:X__M_>
<fs:Y__M_>5067373.73</fs:Y__M_>
<fs:POSTAJA>Rugvica</fs:POSTAJA>
<fs:VODOTOK>SAVA</fs:VODOTOK>
<fs:EXCELLINK>3096.xls</fs:EXCELLINK>
</fs:aaa>
</gml:featureMember>
</wfs:FeatureCollection>

```

Fig. 5. An example of a GML-generated request for a single feature.

Using FeatureServer, we solved two problems. One was how to display selected feature on the map. This is not possible with WMS-C tiled map. The second problem was how to get attribute data of some feature on the map. This could also be done with database backend, like PostgreSQL/PostGIS but FeatureServer allows much more flexibility and reduces the time of system development without any speed degradation.



Fig. 6. Example from the case-study showing the planned land-use

In our client application we implemented some of the familiar features from similar desktop applications, like mouse wheel zoom, distance and area measurement, pan, scale-bar, loading bar etc. We also implemented a search feature that searches WFS instance and displays found content on the map, for example, search for particular cadastral parcel by entering number of parcel and zoom to it or zoom to specific street number. All of these search operations would not be possible with only WMS service. This way we got the speed of WMS-C service maps, and functionality of WFS service maps.

4. CONCLUSION

Free/Libre/Open Source Software can be efficiently used for spatial applications. Such tools are especially useful for those small and medium companies and their clients in public sector who cannot afford expensive proprietary licenses of commercial GIS packages. Average user of web-mapping services like Google Maps is used to a fast response of the service. Achieving such efficiency in practical applications which frequently mix raster and vector spatial data is a challenging task. The solution requires not only the proper FLOSS packages but a lot of research work in order to put all the pieces together. We demonstrated an original and efficient solution for visualization of a very large amount of spatial data on the web using Open Source technologies which combine tiled raster maps and vector WFS data. This is the major scientific contribution of this paper. The Case-Study of a fast web mapping solution for Urban Planning and Management of the City of Zagreb, Croatia, confirmed that FLOSS can result in the same usability and efficiency as a commercial package. This approach represents a major step forward in spatial information management and it is increasing the capacity of small and medium enterprises to develop similar spatially-aware applications without high starting investment.

REFERENCES:

- Hall, B., Leahy, M.G. 2008. Open Source Approaches in Spatial Data Handling, Series. : Advances in Geographic Information Science. Springer.
- Holmes, C., Doyle A., Wilson, M. 2005. Towards a Free and Open Source (FOSS) Spatial Data Infrastructure. From Pharaohs to Geoinformatics, FIG Working Week 2005 and GSDI-8, Cairo, Egypt April 16-21, 2005.
- Medak, D., Pribicevic, B., Djapo, A., Medved, I. 2003. Open Source based Spatial Data Infrastructure - Why and How? /Proceedings of the ISPRS WG VI/3 Workshop: Geoinformation for Practice, Vol. XXXIV, Part 6/W11, 193-196, Zagreb.
- Murray, G. 2006. Asynchronous JavaScript Technology and XML (AJAX) With Java 2 Platform, Enterprise Edition, available online:
<http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>
- Sayar, A., Pierce, M., Fox, G. 2006. Integrating AJAX Approach into GIS Visualization Web Services, Proceedings of IEEE International Conference on Internet and Web Applications and Services ICIW'06 February 23-25, 2006 Guadeloupe, French Caribbean.

Stuedler, D. 2009. Open Source Software for Cadastre and Land Registration – A Viable alternative? FIG Working Week 2009, Surveyors Key Role in Accelerated Development, Eilat, Israel, 3-8 May 2009.

URL addresses:

URL-1: White Paper : The OpenGeo Architecture, <http://opengeo.org/publications/opengeo-architecture/>, (20.09.2009)

URL-2: OpenLayers webpage, <http://openlayers.org/>, (20.09.2009)

URL-3: OSGeo wiki page, http://wiki.osgeo.org/wiki/WMS_Tile_Caching (20.09.2009)

URL-4: ExtJS web page, <http://www.extjs.com/>, (20.09.2009)

URL-5: GeoExt web page, <http://www.geoext.org/>, (20.09.2009)

URL-6: Lighttpd web page, <http://www.lighttpd.net/>, (20.09.2009)

URL-7: Mapnik web page, <http://mapnik.org/>, (20.09.2009)

URL-8: TileCache web page, <http://tilecache.org/>, (20.09.2009)

URL-9: FeatureServer web page, <http://featureserver.org/>, (20.09.2009)

URL-10: History of web mapping, http://en.wikipedia.org/wiki/Web_mapping (20.09.2009)

BIOGRAPHICAL NOTES

Mario Miler is a research assistant the Department of Geoinformation Science, Faculty of Geodesy, University of Zagreb, Croatia. After the graduation, he got interested in CAD and GIS systems, and current research activities are on design and implementation of GIS systems and geographic visualization in Internet.

Dražen Odobasic is a research assistant the Department of Geoinformation Science, Faculty of Geodesy, University of Zagreb, Croatia. After the graduation, he continued working with FLOSS, especially FLOSS used for solving geospatial problems. His current research activities are on design and implementation of GIS systems and creation of web-based geospatial services.

Damir Medak is a full professor at the Department of Geoinformation Science, Faculty of Geodesy, University of Zagreb, Croatia. After he graduated surveying in 1993, he made his PhD in 1999 at the Department of Geoinformation, Vienna University of Technology. Since then, he accelerated the new curricula development at the Faculty of Geodesy, University of Zagreb, which encompasses both geodesy and geoinformatics at BSc, MSc and PhD level. He is teaching Modeling of Geoinformation, Spatial Databases, Mobile GIS, Spatial Data Analysis and Programming in GIS. He is leading several national and international scientific and educational projects related to geoinformation science and technology, most notably TEMPUS CARDS Joint European Project “Geographical Information Science and Technology in Croatian Higher Education”. His research team promotes the usage of open standards and Open Source software both for educational and research purposes. He is the chief editor of Geodetski list, a scientific journal with a long tradition, an associate member of Croatian Academy of Technical Sciences.

CONTACTS

Mario Miler, MSc.
Department of Geoinformation Science
Faculty of Geodesy, University of Zagreb
HR-1000 Zagreb, Kaciceva 26
CROATIA
Tel. +385 1 4639 227
Fax. +385 1 4828 081
e-mail: mmiler@geoinfo.geof.hr
website. <http://www.geoinfo.geof.hr>

Drazen Odobasic, MSc.
Department of Geoinformation Science
Faculty of Geodesy, University of Zagreb
HR-1000 Zagreb, Kaciceva 26
CROATIA

TS 5B - Information Management Software
Mario Miler, Dražen Odobasic and Damir Medak
Efficient Web-GIS Solution based on Open Source Technologies: Case-Study of Urban Planning and
Management of the City of Zagreb, Croatia

15/16

Tel. +385 1 4639 227
Fax. +385 1 4828 081
e-mail: dodobas@geoinfo.geof.hr

Prof. Damir Medak, PhD
Department of Geoinformation Science
Faculty of Geodesy, University of Zagreb
HR-1000 Zagreb, Kaciceva 26
CROATIA
Tel. +385 1 4639 227
Fax. +385 1 4828 081
e-mail: dmedak@geoinfo.geof.hr