

Increasing Participation for Mobile Map Applications

Anis Yazidi , Morten Goodwin Olsen

Faculty of Engineering and science, Agder University College

Serviceboks 509, N-4898 Grimstad, Norway

anisy06@student.hia.no, morten.g.olsen@hia.no

Abstract

In this paper we present a Free and Open Source Software (FOSS) map server centered on the vector format SVG. The main motivation behind this work is to build a simple map server that can be accessed easily using a standard web browser that does not support SVG.

Due to the fact that most of the currently available cellular devices support raster formats such as GIF (Graphic Interchange Format) and PNG (Portable Network Graphics), a solution that increases the accessibility of map services should be based on a raster format and subsequently avoid the use of vectorial formats. Using raster formats to display maps on mobile devices does not only increase accessibility, but also increases the potential market of map services. Commercial solutions exist, however they are to the best of our knowledge not FOSS solutions.

In this paper, we present a prototype of a web mapping solution designed for mobile devices. The solution is a client server solution. The server is equipped of a module that converts on the fly the rendered SVG graphic corresponding to the mobile query into standardized bitmap images that are sent back to the client. Experimental tests on mobile browser simulators and J2ME clients demonstrate the feasibility of the solution.

1 Introduction

SVG (Scalable Vector Graphics) is a W3C (World Wide Web Consortium) recommendation that has gained a lot of attention in GIS (Graphical Information Systems) applications. In contrast to the raster formats, the SVG format offers flexibility in Geo data treatment such as infinite zoom without loss of quality and searchable text.

Nevertheless, the use of SVG creates some accessibility problems. Even though SVG is an open standard, is not supported by a wide range of web browsers. Besides, SVG is not yet fully deployed on mobile devices due to limited computational capacity on the devices. Because of this, displaying SVG maps in a client browser necessitates either an SVG viewer plugin or a browser with native SVG support. In contrast, the raster format is widely deployed in the web browser alleviating the accessibility barriers. However, this format does not provide facilities to treat the Geo data efficiently.

In this perspective, our web mapping application combines the advantages of both the vector and the raster format resulting in an interactive application which can easily be accessible by standard web browsers and have functionality such as the possibility to zoom on the map and to navigate in all the directions. This notion is enhanced by filtering unnecessary image details when a user zooms out.

The spatial data is provided as an ensemble of shapefiles that are loaded in a Postgis database. Our application is a client server architecture based on J2EE (Java 2 Platform, Enterprise Edition). As a web server we utilize the Tomcat application server. In addition, we make use of a servlet container that queries the Postgis spatial database and formats the query result in SVG. In our solution, an SVG raster conversion module is embedded on the sever side. This module converts on the fly the rendered SVG graphic corresponding to the query into a standardized bitmap image that is sent back to the client.

2 Survey of cartographic solutions

Following, we will survey the existing cartographic solutions, and then we will compare them. However, for lack of documentation about the cartographic services dedicated to function on mobile terminals and considering the recentness of this type of services, we proceeded by analogy compared to the existing web mapping solutions. In this direction, we started by reviewing the existing cartographic solutions on the market. Following we utilized the characteristic mobile terminals and adopted solutions only to those terminals compatible. Taking the above into consideration, we rejected the unrealizable solutions in a mobile context, for instance cartographic services based on the applet.

First of all, we can differentiate two large groups' solutions: "static" solutions and "dynamic" solutions. Technically the interactive cartography is returned in two different ways:

- In the form of raster images that are generated in a timely manner and on demand by a server specialized in the geographical treatment. The navigator displays images GIF or png equipped with interactive zones. Furthermore, we felt it was judicious to avoid the the JPEG format which is only partially deployed on the mobiles.

- The form of images displayed by the navigator via inherited programs or via installed plug-ins.

Diffusion of static maps in the raster format

There is no doubt that it exists methods to implement that guarantees restitution in conformity with expectations. For this purpose, it is sufficient to produce a map using usual GIS software and to record the map produced in a bitmap format. Additionally, pre-computed Zooms on portions of images makes possible to increase the definition on significant zones. In order to visualize only the useful portion of a large map, it is divided in small entities makes possible

Diffusion of static maps in vectorial format

An alternative to the preceding solution consists in exporting a map in a vectorial format. A vectorial representation is richer in information than a bitmap presentation, which permits local treatments of GIS data. If the data preserve their initial structuring, several possibilities can be offered to the mobile. The most radical solution result in exporting the totality of the data on the mobile device. The situation then becomes similar to that of a "mono user" application and all the operations of visualization or analysis envisaged can be carried out locally. It should be emphasized that an operation of transcribing of data is not always simple and that the vectorial data contained in the databases contains usually too much details and are not adapted to the mobile devices which are not able yet to interpret these vectorial formats with the same performance as an PC. A work of simplification of the vectorial representation is to be considered. The SVG and the Flash remain the two principal formats of the vectorial solution implemented partially on the mobiles.

The format SVG is carried by W3C for vectorial posting on Internet. We note that a light version (SVG Tiny or SVG Mobile) of SVG is available on certain

mobiles. The SVG is an application side customer. It requires the installation of a plug-in and the transfer of data for visualization. The motivation behind this is the need of light-weight, dynamic and interactive graphs. Its functioning principle is as follows: In the server side, a preliminary transformation of GIS data towards SVG format is necessary. In the client side the data are stored in a cache memory. The principal characteristics of the SVG, as for all the vectorial solutions are: aestheticism and good ergonomics of navigation.

Cartographic servers:

A cartographic server produces maps of spatially referenced data dynamically from geographic information. Most of the servers available in the market are proprietary servers. Generally, map servers are focused on rendering spatial data for the web, however these servers are not optimized to support mobile devices. Another difficulty is interfacing these cartographic servers with a J2ME application.

3 Comparison between the solutions of interactive cartography

There is no ideal and universal solution for cartographic mapping. However, these various solutions respond to various needs. One of the major advantages of the use of vectorial formats is to be able to pass all the control of zoom and panoramic to the client side. Thus, all navigation necessary to visualize a map on a screen, once all elements charged, are independent from the application server. This characteristic confers to the vectorial format an ergonomics of displacement of very high level. In addition, this technology allows a good view thanks to the infinite zoom capacity of the vectorial format, without pixelization phenomena.

On the other hand, the obligation to download the totality of the data on the mobile customer can become problematic for big size files which introduces undesirable latency. The vectorial format is very compact for a small number of geometric objects. In addition there are possibilities of compression limiting the map size. In contrast, when the number of objects becomes high, this type of file becomes very cumbersome and heavier than a Bitmap representation. Thus, the unnecessary and invisible details with low resolution are simplified to limit the number of objects. With some subtlety it is possible to mitigate this disadvantage by managing the level of detail depending on the threshold of zoom.

The first technique is most robust: it guarantees that the cartographic application will be visible immediately on all the navigators of the mobiles. It is also the lowest in visual quality, in displaying speed and in interactivity. The second technique, which transfers a significant part of the treatments towards the client side, is thus used in a very diverse way. It initially opposes specialized technologies of large editors GIS (Esri, Mapinfo, Autodesk...) to the vectorial formats general editors (Flash, Svg). However it is necessary to underline that the use of a vectorial format can block accessibility because this type of format is not yet largely deployed on the mobiles of new generation. Moreover, even if the navigator of the mobile supports the vectorial format, Javascript format should be supported to manage the interactivity. These two conditions are seldom joined together in mobile devices. If we seek to make a data dissemination towards the greatest number it seems that the solution of a cartographic server is most

adequate. Indeed this system is most tolerant with respect to the user, and corresponds perfectly to a desire of decentralization of geographical computations. However it will not allow a high interactivity. On the other hand, the basic functionalities that it proposes (navigation, selective display) are in the majority of the cases sufficient.. The commercial cartographic servers offer variable possibilities, but their very high price poses a use barrier.

4 The solution design

Conceptual choice

Considering the above description, our solution will be based on three points:

A client server application:

This choice is legitimate for at least one reason. Such a need raises with regards to the limited capacity of the mobile. Indeed, the computational power of mobiles devices is not yet sufficient to embark the totality of the service. Thus, the choice of such an architecture allows to carry heavy calculations towards the server part resulting in a better performance

Use of the vector format SVG

Having as source of information an image bitmap stored in the server part results in a deterioration of the quality of the image when a zoom is performed.

One disadvantage is the high weight of the file image generated starting from the data source ESRI. In contrast, the vector mode indefinitely has the quality of zooming on the image without quality loss. Having the data in the form of vectorial layers meets a need for filtering the geographical according to the level of zoom (Concept of intelligent zoom). One of the major advantages of the use of vectorial formats is to be able to pass all the control of zoom and panoramic to the client part. However this requires the addition of a module to the mobile navigator managing the transcribing of the vectorial mode in raster mode (image). This assumption is seldom realized with regards the weak deployment of the vectorial formats even on the most sophisticated mobiles. Since accessibility is a concern, we gave up the solution which consists in sending to the mobile the data under a vectorial format.

An alternative to this solution is to provide to the mobile the map a raster format. This solution seems to be most adequate because it makes it possible to reconcile the contributions of the vectorial mode with the portability of the raster format. Thus it is judicious to pass the control of zoom to the server which will manage the vectorial format. From this point of view the applicative server is supposed being in measurement, each time that a customer asks for a map, to extract the adequate vectorial data, to treat it then to convert it to the raster format, format interpreted by the near total of the mobiles.

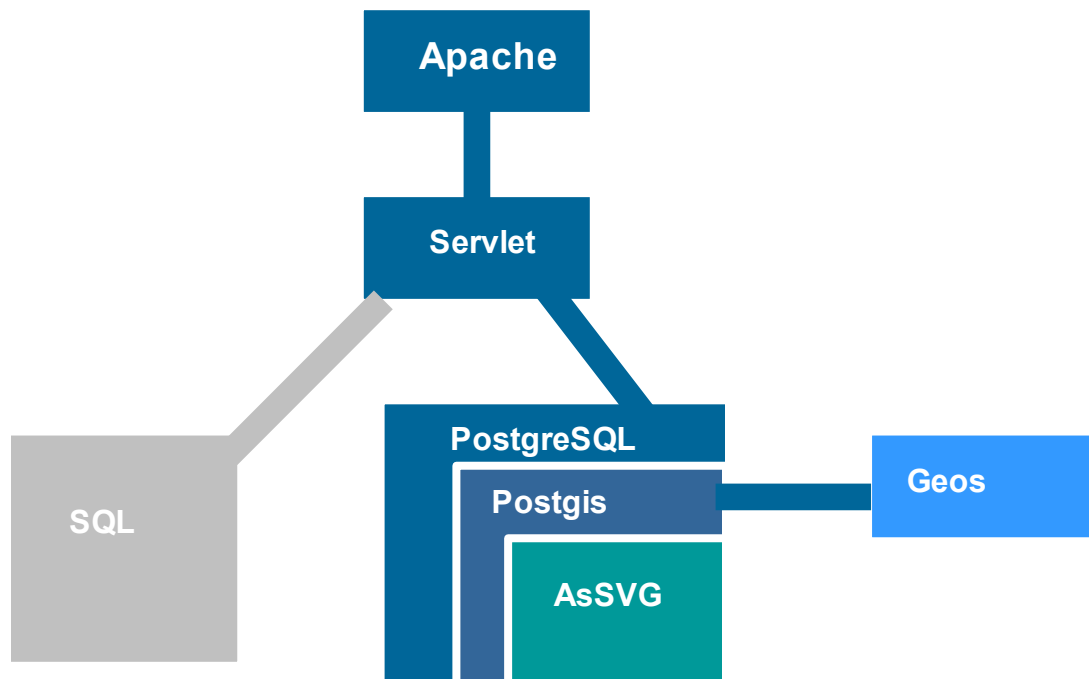
Use of a an Open-Source Spatial Database: Postgis

Nowadays, the Geo data is usually presented in ESRI shape-files. Disposing of ESRI shape-files, it is judicious to store them into a data base such as Postgis or spatial Oracle. In fact, a data base has the advantage of a fast access to the

data compared with the access times to files SIG. This rapidity in access is enhanced by the use of spatial indexes, which was carried out in our application.

Software architecture

The initial objective that we had fixed is to explore the tracks leading to an interactive cartographic service, implemented on the mobile, starting from the offers of Open Source for geographical data. The software architecture of our application is given by the figure below:



PostGis :

PostGIS is an extension to the PostgreSQL relational data base. It is used to store spatial data (so-called simple features: points, lines, polygons, and combinations of them) and provides spatial geometry tests (e.g. Touches, Intersects, Contains) and operations (Buffer, GeomUnion, Distance) and coordinate transformation (Transform).

GEOS (Geometry Open Engine Source)

GEOS is a complementary library to PostGIS that provides an implementation of the space operators. It was developed separately, although there is (before) a version Java named JTS (Java Topology Suite) of Vivid Solutions, proposing the same functionalities. GEOS is in that only one bearing in C++ of this solution. This tool provided us an invaluable help at the time of the formulation of our space requests since it presents the asset to support a multitude of space operators. Indeed, any calculation of intersection of geometrical forms carried out within the framework of our application relies on GEOS.

Batik is developed in Java for the development of applications calling upon images to format SVG. Batik is able to convert SVG files to a raster format. The Batik library is one of the main parts of this project. Indeed, our script returns a geometrical data in SVG which that is not supported by Internet explorer of the mobile. Therefore, the transcribing of data SVG to the raster format by the means of Batik is essential.

Application server (Tomcat)

A request from a mobile client to visualize a portion of a map is carried via the use of servlet. The parameters of the query involves the coordinates of the zone to visualize. These parameters are passed to a servlet. The servlet connects to the Postgis database and performs a “bounding clipping” on the different layers of the Geo data. In this sense, we make use of the functionalities offered by Geos for bounding box clipping. The result of this query is an SVG representation of the different layers of the clipped zone. After that, the server converts on the fly this SVG data via the Batik module to an image and sends it back to the client.

The images are generated dynamically in the cache memory without stocking them on the server. In fact storing the images on the hard disk is susceptible of creating two problems:

- Identification of the various images stored on the hard disk relative to the various mobiles

- Increasing the latency of the application because the access to the hard disk is generally slower than the access to the cache memory.

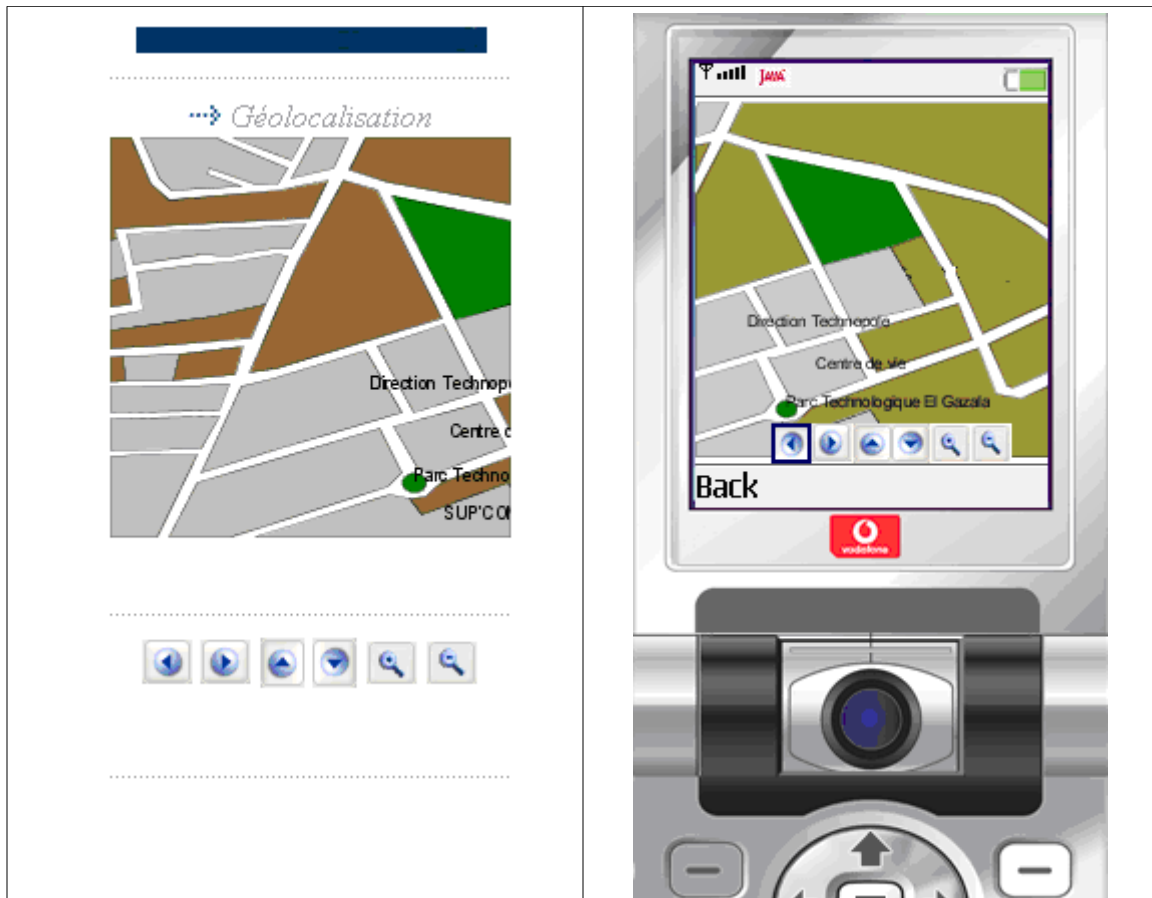
Another key point to underline is filtering unnecessary details when a zoom out is performed. In this sense, we filter the layer representing the roads names whenever the client demands from the server to zoom out the selected zone. This notion is illustrated below:



Figure 1: Reducing encumbering details when a zoom out is applied by filtering the layer representing the streets.

Client side

An interesting characteristic of our application is the duality of interfaces Web and J2ME. A duality which derives its originality of the uniqueness of the server part. Indeed, far from duplicating the server part to interface the two types of customers: Web and J2ME, we conceived an server part. These two interfaces are characterized by a design concern. Indeed, we paid a great attention to the aestheticism of our application.



Service on a Mobile browser	Service on J2ME midlet
------------------------------------	-------------------------------

5 Conclusion

In this paper, we presented a prototype of an interactive cartographic application for mobile devices. The application is based on the offer of open source. The accessibility in this application was enhanced by exporting all the computations to the server side.

The development of the mobile terminals inaugurates an innovation. In fact, the increasing computational power of mobile devices is susceptible of creating an evolution in the cartographic services running on mobiles. Such an evolution is subject to the adoption from the vectorial formats by the manufacturers of mobile terminals.