

Personalized touristic routes implementing Semantic Web on GIS system (Itiner@ project)

Laia Descamps-Vila
Universitat Oberta de Catalunya, h7ux@uoc.edu

Abstract. Nowadays, when a user is planning a touristic route is very difficult to find out which are the best places to visit. The user has to choose considering his/her preferences due to the great quantity of information it is possible to find in the web and taking into account it is necessary to do a selection, within small time because there is a limited time to do a trip.

In Itiner@ project, we aim to implement Semantic Web technology combined with Geographic Information Systems in order to offer personalized touristic routes around a region based on user preferences and time situation. Using ontologies it is possible to link, structure, share data and obtain the result more suitable for user's preferences and actual situation with less time and more precisely than without ontologies. To achieve these objectives we propose a web page combining a GIS server and a touristic ontology.

As a step further, we also study how to extend this technology on mobile devices due to the raising interest and technological progress of these devices and location-based services, which allows the user to have all the route information on the hand when he/she does a touristic trip. We design a little application in order to apply the combination of GIS and Semantic Web in a mobile device.

Introduction

Geographic Information System (GIS) is a combination of different software tools to store, analyze and query spatial data. Some years ago, GIS were only used for professional benefits due to most of software was privative and very expensive, and the unavailability to have the cartography available, since it was also very expensive.

There was a change on this tendency when users had the availability of free maps viewers and editors with the apparition of Web Map Servers like Google Maps [1] (2005), Bing Maps [2] (2005) and later Yahoo Maps [3] (2007). The collaborative project, OpenStreetMap [4] (2004), which is based on user collaboration to construct a worldwide map, it is rising so fast and it is becoming a good alternative from free but privative Web Map Servers. It also appeared different GIS desktop Open Source applications like gvSIG Desktop [5] or OpenJUMP [6] which combined with GIS Servers like MapServer [7] or

GeoServer [8] allow users to construct interesting applications to serve and share geographic data. In this new scenario, different kind of industries started to use GIS for different benefits than professional use, especially based on touristic and commercial sectors.

Touristic sector is becoming so interested in this technology, due to the availability to reach different users from different places in an easy and fast way with all touristic information of a place. The most popular applications are those that calculate routes, show points of interest, show recommendations of other users, etc.

The aim of Itiner@ project is to build a touristic application able to propose (or to build) touristic routes going through different points of interest. In fact, this is not new. The novelty we want to introduce is the adaptation of the route to everyone. If a family is going to Barcelona for a day trip, the application should offer a different route than if the user is going with a group of friends. The route is build based on user preferences and time left.

In order to achieve this, we are going to use Semantic Web technology combined with a GIS. Semantic Web is a relatively new technology, often called web 3.0, as an evolution of web 2.0 based on sharing information over the web, which appeared some years ago with the objective of having a more efficient and easier to use web. Applying this technology, the user can get the exact information that he/she is looking for due to the web is created to be read by a machine, thus not only designed for human understanding. This technology uses ontologies to allow computers read the concepts and get relations in a certain domain.

The general use of ontologies provide us a tool to improve sharing information and knowledge, allowing the integration of data in information systems, but in particular, the advantage of using semantic data and ontologies related with a GIS is what allow us to describe the content and meaning; and to link together different data [9] [10].

With all these characteristics, it is possible to develop a GIS application to find the exact point of interest (POI) that a user is looking for, do a personalized route based on user defined preferences and time restrictions, choose a place based on other user recommendations, view multimedia content related to the place where you are located, etc. Although these applications could be done without Semantic Web integration, we will see that using this technology we have some benefits like: get exact results from a consult for every different user (personalized routes), select from the great amount of information we can find in the web, which provide us time saving and precise results. We avoid the user the task of decide if one point of interest is what he/she is looking for or not.

Another interesting issue is the use of this technology in mobile devices. From some time ago, mobile phones have reached new technologies as Global Positioning System (GPS), fast wireless communications for Internet access and powerful processors to do all kind of operations, like mini-computers. So, GIS applications combined with Location-Based services (LBS) propitiate that in recent time appeared a lot of applications based on tourism, urban route, traffic planning, augmented reality, etc.

The article follows different steps to achieve the previously described facts.

First of all, we study the general situation of the two different areas that compound our project separately, GIS and Semantic Web.

We study the state of the art of actual GIS which include GIS desktop applications and GIS Servers. We do computer tests with studied applications and considering the results, we decide which one is the best option for the project.

About Semantic Web, we study the state of the art of this new technology and which frameworks are nowadays in order to create, edit and instantiate semantic data. We want to do some ontology implementation tests on PC, considering different storage possibilities (in-memory or persistent) and some query tests in order to know if it is efficient to work with semantic data. We also design a touristic ontology to store the touristic data we want to share with the users on the application.

Then, we study one of the critical points of the project, how to integrate semantic web to GIS [12], because it is necessary to relate semantic data with geographic and touristic data, which is not semantic data [13] [14].

As a final research, we extend all this technology to mobile devices [15] in order to have the advantages that provide having the information of a touristic route on the hand. We study which is the state of the art of this mobile applications and show if there is any kind of personalization on them. We also start the implementation of the application on a mobile device.

Objectives

The research of this article is based on achieve different objectives.

The first one is to establish how to combine, efficiently, GIS and Semantic Web. We also will create our own ontology using touristic data and decide which GIS we are going to use.

The other objective is to determine how to extend this combination to mobile devices. We will do a mobile application that creates personalized routes and show points of interest depending on user preferences.

State of the art

First of all, we are going to study which is the state of the art of tools we are going to need for Itiner@ project, in order to know if we can reuse or adapt some of them in our project. We are going to focus on Open Source software due to the possibility to get access to source code and to develop or modify its features.

Geographic Information Systems

A GIS is a system able to store, manage, show and query spatial data and do geographic operations. This involves a wide variety of software and there are different kinds of GIS software: GIS Desktop, GIS mobile, GIS Servers, GIS web clients, spatial databases and spatial libraries. Depending on the needs of the user, there are professional users and non-professional users; he/she chooses the software and applications that offer the capabilities the user want.

For a project based on planning touristic routes and selecting POIs, we need a GIS which combines the following software and capabilities: a desktop or mobile application to visualize the information; a spatial database to add spatial indexes and provide an extension for routing functions; a spatial library in order to have geographic functions and coordinate representation; and finally a source of maps where to get map information.

Considering we want to share the information over the web to provide touristic data to all the users, it is necessary to use a GIS Server like MapServer or GeoServer additionally of previously described tools. We also want to have access to WMS and WFS services.

Once established the requirements, we detail the most used desktop and mobile applications, spatial libraries and spatial databases used in GIS and GIS Servers.

Spatial libraries

A Spatial library is a collection of resources to support spatial applications like GIS Desktop, adding new functionalities to those applications. The most complete spatial libraries are described here [16] and we can summarize them classifying by the functionality they provide to the applications:

- JTS Topology Suite [17] and Geotools [18] (Java) provide implementations of all the simple features objects found in the OpenGIS specification and the methods defined for those objects. JTS Topology Suite is specially used for geometry representation and it is the most used in Java (due to the extensive use of the library, there is a porting to C++ called GEOS). Geotools is usually used as a complement of JTS topology suite.
- PROJ.4 [19] (C) is a Cartographic Projection library which proper tools for cartographic visualization of geographic data.
- Focusing on spatial data formats, in GIS there are two types of representation, raster data (describe discrete objects like a digital image with a tree) and vectorial data (using vectors to describe the geometric characteristics of a shape). Libraries to add support for representing data formats are: GDAL [20] (C), used to read and write raster geospatial formats, like GeoTIFF, JPEG2000, ECW, etc. and OGR [21](C++) which allows to read some vector geospatial formats like ESRI shapefile, PostGIS, etc.

GIS desktop applications

Some of the most popular Open Source GIS desktop applications are gvSIG Desktop, which uses JTS and Geotools libraries, OpenJump (JTS) and uDig [22] (JTS and Geotools). All of them are Open Source and have very similar functions. They all read and write raster and vectorial maps, they have WFS and WMS web-services, they have database connection and they have tools to edit geographic data.

uDig can be used as a standalone desktop application but it is also a GIS framework for Eclipse, which allow to build complex analytical capabilities. OpenJUMP is the evolution of JUMP GIS and it has a lot of plug-ins to adapt to user utilities and gvSIG Desktop is a desktop application also ported to mobile device with less tools, but very interesting.

Spatial Databases

In computing, when a great amount of information, it is usual to store them in a relational database in order to have the data organized and related. If we use geographic data we need to extend the relational database to a spatial database, which are already optimized to work with spatial data. Traditional databases have not the possibility to use the location of a place as an index to a spatial structure and neither has spatial functions like near that point, intersection with a line, inside this region, etc. Using spatial databases we can do all this.

Spatial databases have two important characteristics; they use spatial indexes (R-tree, QuadTree) to optimize query operations and they provide spatial operations like spatial measurements or spatial functions. Most used Open Source spatial databases nowadays are:

- PostGIS [23] extends PostgreSQL database. It is a very complete system, probably the most used actually in computer environment to use with GIS applications. There is the possibility to add pgRouting [24] extension to provide geospatial routing functionality with routing algorithms like Dijkstra, A-Star, Shooting-Star.
- SpatiaLite [25] extends the lightweight database SQLite, especially interesting for mobile devices because SQLite acts as a library that becomes an integral part of the application program, instead of being a classical client-server database management system. SpatiaLite also has a routing tool, the VirtualNetwork module, which implement the Dijkstra's Shortest Path algorithm.

GIS applications in mobiles

Now, we focus on GIS mobiles applications because they are an emerging business which is growing so fast. A great amount of actual mobile applications are dedicated to tourism, urban route, traffic planning, augmented reality, etc, so we have analyzed them.

In the table below (table 1) there are the most popular GIS mobile apps nowadays, with the characteristics that are important to locate POIs and plan routes. Another important

thing is the license of the project, because if it has useful tools, we can reuse the code for our project.

	User location	POI	Maps	Route finder	Multimedia content	Personalized routes	License	Platform
<u>gvSIG Mini v.1.1.0</u>	Yes	Yes	Offline – Online	Yes – Online	No	No	GNU/GPL	Android - Java -Blackberry
<u>gvSIG Mobile</u>	Yes	No	Online	No	No	No	GNU/GPL	Windows Mobile
<u>ArcPad</u>	Yes	No	Online	No	No	No	Proprietary	Windows Mobile
<u>Bluemapia Mobile</u>	Yes and other users	Yes	Online	No	Yes. On the Map	No	GNU/GPL	Windows Mobile - Iphone
<u>OSMTracker v.0.5.2</u>	Yes	Yes	Online	No	No	No	GPL3	Windows Mobile - Android
<u>OsmAnd v.0.5.1</u>	Yes	Yes – OSM data	Offline – Online	Yes – Online	No	A little in type of route (pedestrian/bike/car)	LGPL	Android
<u>Wikitude v.9.08</u>	Yes	Yes	Online	Wikitude drive	Yes. Augmented Reality	No	Proprietary (Free)	Android – Iphone – Symbian
<u>Mobile GMaps</u>	Yes	No	Online	No	No	No	Creative Commons	Windows Mobile
<u>MapDroyd v.0.9.9a</u>	Yes	No	Offline - Online	No	No	No	Proprietary (Free)	Android
<u>Google Maps 5 v.5.0</u>	Yes	Yes	Online – Offline	Yes – Online & Offline re-routing	No	A little in type of route (pedestrian/bus/car)		Android
<u>TravelDroyd v.1.2.6</u>	Yes	Yes	Offline	Yes – Offline	Wikipedia	No	Pay for each guide	Andorid

Table 1 - Mobile GIS applications with its characteristics (versions actualized at January 2011)

All of them provide the user location, using the GPS of the device or through internet connection and some of them (gvSIGmini, Bluemapia, OSMtracker, OSMAnd, Wikitude, Google Maps 5 and TravelDroyd) show the points of interest (POIs) of a region. Besides that, a few applications show pictures and videos (Bluemapia) or Wikipedia content using Augmented Reality (Wikitude).

These lasts months appeared some applications that allow the user to download the map tiles and browse them without Internet connection (gvSIGmini, OsmAnd and MapDroyd), but already it is not possible to do a route without server connection, it is only possible if the user is connected to servers like Yours [26] (gvSIG mini and OsmAnd), Cloudmade [27] or Google maps (Google Maps 5). Google Maps have a re-routing function, if the user starts the route online and you drop the server connection the application will be able to guide you to the right path, but you still need a server for the initial draw.

Personalization of the routes is the less achieved functionality, only OsmAnd and Google Maps have a very little personalization when you select a route; you can choose if you are using a car, a bus, a bicycle or go by walk.

GIS Server

In the world of Open Source GIS Servers [29] there are two clear dominators, MapServer and GeoServer. Both of them allow users to share and edit spatial data, they read a great variety of formats and they support standard protocols as WMS, WFS, WCS.

WMS (Web Map Service), WFS (Web Feature Service) and WCS (Web Coverage Service) are international OGC (Open Geospatial Consortium) standards to transfer map over Internet. WMS serves georeferenced map images, WFS provides an interface allowing requests for geographical features using platform-independent calls and WCS provides an interface allowing requests for geographical coverages. These standards are widely used in the cartography world to share the information.

GeoServer was specially designed to act as a server for geospatial data and MapServer is a development environment for GIS applications which has the possibility to be used as a Map Server.

MapServer is written in C and uses GDAL, GEOS & Proj4 libraries, but GeoServer is written in Java and uses Geotools library.

Semantic Web

Semantic Web is a relatively new project appeared at the beginning of 2000s, which include different technologies with the objective of having a more efficient and easier to use web. The principal objective is to have a web able to be read by a computer because nowadays the web is only designed to be read by humans.

The concept "Semantic Web" is based on interlink metadata using the technologies proposed by the W3C (World Wide Web Consortium), where RDF [30] (Resource Description Framework) is a standard model for data interchange on the Web which is based on Triples (subject-predicate-object expressions). The idea is to represent everything we can as Triples, and save them to a Triple Store.

Semantic Web uses ontologies to add this metadata to the Web and provide formal description of concepts, terms, and relationships within data.

Ontologies use different technologies like; Ontology Web Language [31] (OWL) and RDF Schema languages build on RDF for authoring ontologies and different data interchange formats like XML/RDF, N3 or N-Triples notations to construct the model (with the classes and relationships).

Once we have the model and structure of the ontology, we can add the resources we decided as instances of the ontology and relate those using properties. There is specific software, like Protégé, to do these operations with a friendly graphical interface. One of the most valuable characteristic of ontologies is the possibility to link between them. We can use different ontologies and link with our own ontology, like people do in Linked Data project [32], to have all the metadata connected together.

One important part of the ontology management are the reasoners or inference engines, programs that generate deductions following logical rules, what allows us to deduce information from the ontology model. For example, if a trucker is a person who drives a truck and a truck is a vehicle, then a trucker, for driving a vehicle, must be a driver.

The abstraction of RDF information is done using RDF query language, called SPARQL [33], with which we can obtain the exact information we want, like the previous example of the trucker.

Focusing on touristic area [34] [35], the most popular projects based on semantic web and used ontologies are: *GeoNames Ontology* [36], which makes possible to add geospatial semantic information (6.2 million toponyms) to the web, *DBpedia Ontology* [37] based on the most commonly used infoboxes within Wikipedia which provide descriptions of points of interest (DBpedia Ontology currently contains about 1,667,000 instances), LinkedGeoData (LGD) project [38] uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base according to the Linked Data principles and *LGD ontology* is a class hierarchy for LGD elements as well as description of popular properties. LinkedGeoData describes 320 million nodes and 25 million ways at June 2009. On the other hand, there are a lot of little ontologies that describe small regions of the territory like *GeoLinkedData Ontology* [39], which describe geographic resources of Spain country, *Cruzar Ontology* [40] which describe touristic resources to realize routes and visit POIs of Zaragoza city.

Semantic Web in GIS

After explaining what is the Semantic Web and which benefits can provide to the user, we study which is the actual situation of the combination of GIS and Semantic Web technology looking for any existing project based on that.

Introduce the Semantic Web technology in GIS is also interesting because it is possible to use geographic data linked with the available touristic information of a region from different sources.

There is one project that has already done this combination with good results. This kind of project is developed by Zaragoza council and based on Semantic Web and touristic information of Zaragoza, called "*Planifica Tu Visita*" [41].

It is a web application that offers personalized routes in Zaragoza taking into account preferences and circumstances of each visitor. They designed the Ontology "*Ontologia Cruzar*" which allows the integration of the touristic and spatial information of Zaragoza databases. Data is transformed to RDF and combined with semantic model. There is also a query processor engine for ontology data. It uses semantic matching to select POIs, and they use route planning to calculate the best itinerate considering user profile or monuments timetable.

It is an interesting project but simple, because it is very local and it has not so much preferences to choose or a good interface for the users. Neither has a user configuration system and cannot get automatic preferences from a user.

Semantic Web and GIS in mobile

When a user is doing a touristic visit it is important to have real time information in case you want to check or view some touristic data, so it would be very interesting to have, for example, the project of Zaragoza council available in a mobile device. Now we are going to focus on projects with this objective that are on development phase:

- Mediatour [42] project is developed by a Spanish consortium of Universities and companies that develop a technological platform to manage personalize and reuse multimedia content for touristic use and their publication in a multiformat environment. It is oriented to travel agencies or tourism managers to organize, catalog and reuse touristic content to generate personalized touristic packs in real time.

With mobile device you can do different operations like download multimedia content associated to a touristic resource, geographic visualization and upload personalized routes. It is still in development, the forecast is to be finished in 2011.

- Infutur [43] is a project based on e-Tourism (travel technology based on online information) which has the objective to transform mobile phone to a personalized tourist guide in real time based on user preferences, localization, visualization and personalization of services. They used new technologies associated to Web 3.0 in generation and interchange of information between users, combining tools of semantic web and social web.

Infutur is an application accessible from a web page or from mobile device that offers different characteristics depending on the environment used. It is still in development.

Case of Study

After studying the state of the art of all the tools we need for the project, we do some tests with different tools to decide which the best option to choose for each case is.

To develop our application we need to integrate two different parts, a GIS and Semantic Web technology, so first of all we analyze the GIS part.

GIS tests and selection

We do different tests on computer which involve previously studied software in the state of the art section. As we advance with the tests and get results, we decide if a software is a good option or not for our project.

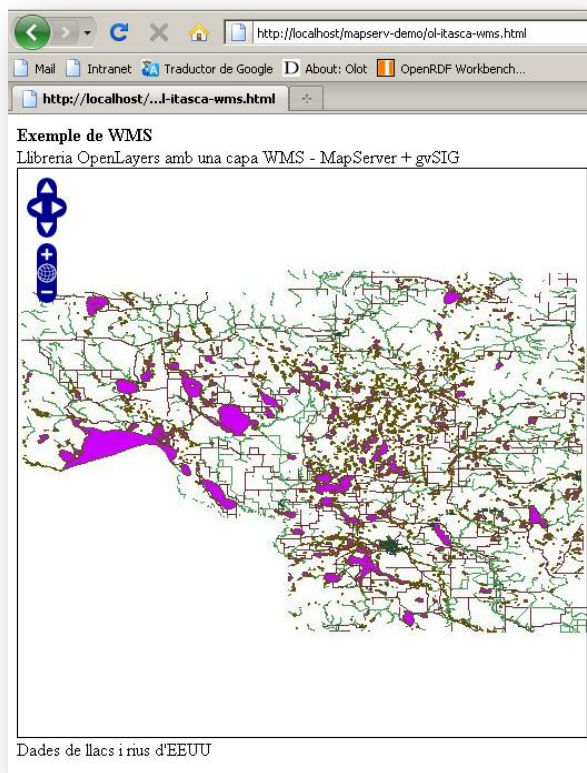
Based on state of the art classification, we consider different sections for software we need to choose.

GIS Desktop and GIS Server

We do some functionality tests with GIS desktop applications importing shapefiles and connecting with PostGIS database using gvSIG Desktop, uDIG and OpenJUMP. We confirm they provide very similar functions which only differ on the graphical interface and some different extensions.

For the project, we want to share all the information with the users. Then, a part of the GIS Desktop, we need to additionally implement a GIS server. A GIS server is simply a GIS that allows sharing the resources and the functionalities with web clients. Now, besides choosing the appropriate software, the question is; which architecture must have the GIS Server to feed our application functionalities, i.e., which different software we are going to select to construct the server?

We do a test to visualize geographic data in the web using the WMS protocol. With this test we know how to edit and publish geographic data on the web. The reason for the software selected for the test is the availability of good manuals and community support to install and configure them.



We install and configure MapServer with Apache HTTP Server. We install gvSIG Desktop plus the plug-in for web publishing and we create a mapfile using different shapefile layers choosing the Server (MapServer) and protocol (WMS) we want to use. We can personalize the mapfile with gvSIG software before publishing on the web; we change the color of the map water from blue to purple. Using OpenLayers we create an html page which shows the mapfile. In Fig 1 there is the result.

Fig. 1 - Geographic data publication on the web using gvSIG Desktop software with a MapServer and OpenLayers

After doing this test, we realize that it is easy to edit and publish geographic data using a GIS Server plus a GIS Desktop. After doing the test, we realize that for us it is not necessary to use a GIS Desktop. If you want to edit geographic data is useful to deal with a GIS Desktop, but as we do not need to edit cartographic data, we will not use a GIS Desktop. We only want to show a map with another layer on top with touristic information. So, the ideal way to do that is using a mashup like [48] inside a GIS Server combining a Map Server plus a layer with the touristic information. This is better because we can have the cartography actualized from the Map Server and then add our own layer of personalization.

As a GIS Server, we decide to select GeoServer because it is simple to manage and has a strong community support, which is very important in Open Source software.

To construct the mashup, we also need a library to put the map on the web. At the moment we have the Server and the map source but we need to display the map. The Open Source, most complete and most used library is OpenLayers [44] which has an API that supports Web Map Services (WMS), commercial Map Servers (Google Maps, Bing, Yahoo), Web Features Services (WFS), OpenStreetMap data and vectorial format.

The Map Server allows us to get Open Source cartography data. Nowadays, there are a lot of map sources, as we described in the introduction, and one advantage of using a mashup for the application is that we can feed the GIS Server with more than one map source, but at the moment we only use one source, to simplify the application.

Proprietary Web Map Servers as Google Maps, Bing Maps, Yahoo Maps or WMS Servers like ICC and IDEE can lead us to a license problem because they are free but property.

We decide to use an open source data font like OpenStreetMap (OSM). Although OSM has a low level of definition and detail of small towns and less populated areas, it has a strong community support and it is rising so fast. If there is a concrete lack of streets, roads or places we can edit ourselves.

If we take into account all the project objectives, there is another reason to choose OSM instead of ICC or IDEE sources. The final objective is to port all this technology on a mobile device. WMS transmission is slower than tiles using Google Maps or OSM because of map tile format. The time necessary to show a map on a mobile device is important. Map tiles get a quick map display, if the user pan and zoom the map the interaction is faster and responsive.

Below there is a test using OpenLayers library to show ICC cartographic data and another test which is a mashup between Google Maps and Panoramio photos, also using OpenLayers:

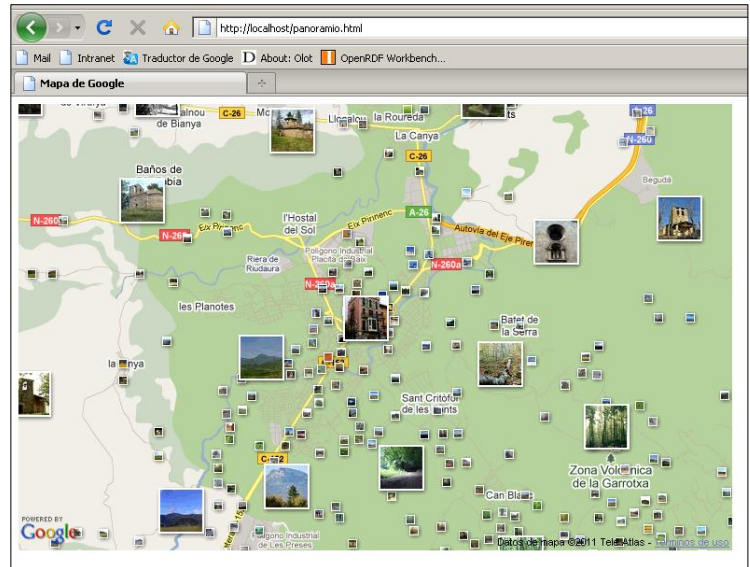
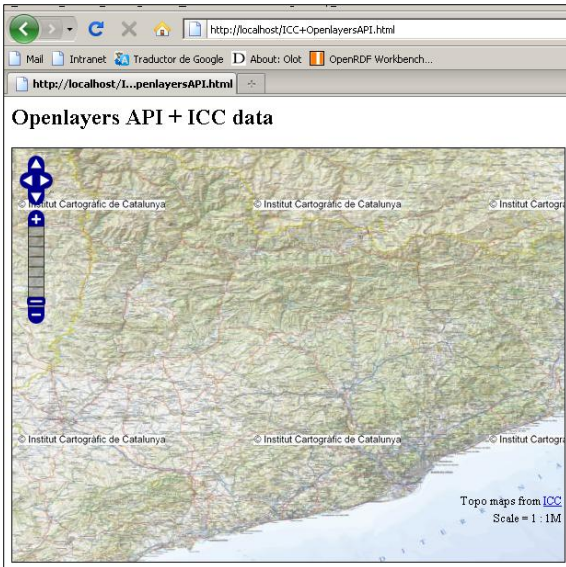


Fig. 2 - At the left. Cartographic data from ICC published on the web using OpenLayers.
 Fig. 3 - At the right. Mashup using OpenLayers API, between map source from Google Maps and Panorámico photos.

Spatial databases

Examining spatial databases, we decide to use PostGIS because it is more complete than SpatiaLite and has more functionalities. For example, SpatiaLite is an extension of SQLite which is a not server-side database (serverless), which disable the option of being used in the GIS Server.

We need PostGIS database to store vectorial map data from OSM (ways, nodes and relations) in order to calculate the routes with pgRouting. This vectorial data is stored in a local OSM format, .osm, which is an XML-based data format that GeoServer cannot read natively. But there is a program, osm2pgsql, which allows the user to load the OSM data into PostGIS. This will process the XML information and load the data into a PostGIS database called "osm". There is also a library, called Osmosis which can do the same in a Java application.

The mashup is build using the map tiles layers plus the on top touristic information layer. This is the information stored in the ontology as Semantic Data and it will be showed on the map. This part is critical, because we have to relate RDF data with non-RDF data.

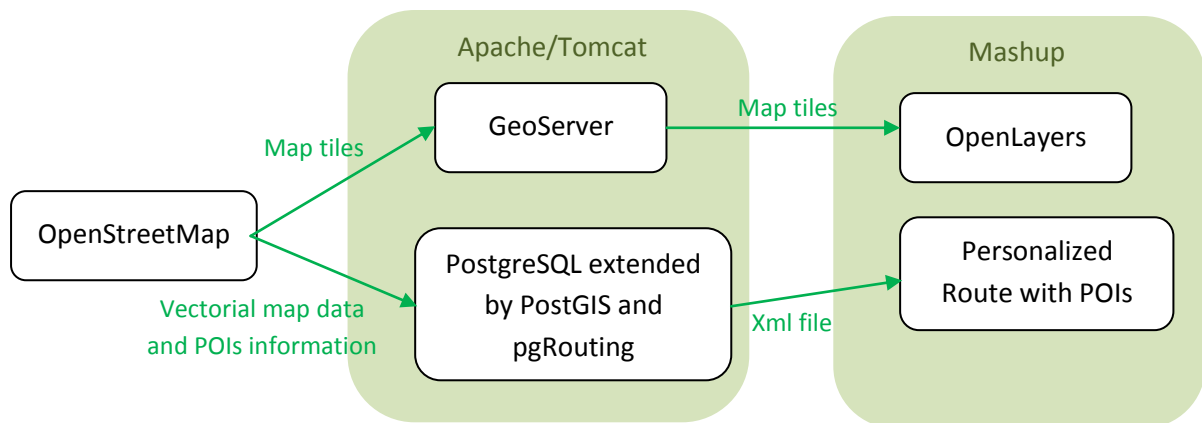


Figure 4 – Architecture of GIS

Now, we have accomplished one objective, to decide which GIS architecture (Figure 4) to use, we study the Semantic Web part.

Implementation of the ontology

In our project, the Semantic Web implementation involves two different systems, the computer and the mobile device. We study and do the tests apart, because the implementation is quite different. In a computer, the process of store and query semantic data is more studied and tested, but in mobile device we can encounter more implementation problems due to the lack of software and the limited mobile memory capacity.

Implementation on a computer

First of all, we design an ontology based on touristic resources, detailed as follows.

We design the *POI* entity reusing some entities from LinkedGeoData ontology, discarding the ones that are not necessary to describe touristic resources. LGD ontology is based on the OSM features, so it is good to reuse them because we use OSM touristic data for POIs and the subtypes and classes will coincide. Apart of POI entity, we add the *Resources* entity with the subtype's text, PDF, video, picture and HTML page to describe the resources of each POI. With this, we can relate each POI with the comments, photos and video that other users do.

Once we have the ontology, we must decide which the best way to store it is, considering that later we have to query with SPARQL language. When you work with ontologies, the critical point is the efficiency of the operations; we do some tests to choose the best option.

There are two different ways to store an ontology, memory storage, where data is stored as a variable on Java program, or persistent storage, using a database, a dataset or a file. The advantage of memory storage is that the load time is smaller than persistent storage, but you need to load the ontology every time. So, if you need to do some consults, it is better to use a persistent storage because you only need to load the ontology once.

To do this, we study the two more common RDF frameworks nowadays, they have a great support from the Open Source community, Jena [45] and Sesame [46], both build in Java. Both offer an RDF API to store, inference and query RDF information using a SPARQL engine and reasoners able to infer logical consequences from a set of asserted facts or axioms.

Sesame is more adaptable for Server functionality and Jena has a robust API. They offer different systems for RDF and OWL data persistent storage; Jena offers TDB and SDB (described below) and Sesame have native and RDBMS store. All storage systems are designed to be queried using SPARQL language.

TDB is a pure-Java, non-transactional database for both embedded use and very large datasets. SDB is a RDF storage using SQL databases, including transactional operations

and the most important databases supported are PostgreSQL, MySQL, Oracle, SQL server, H2.

Sesame Native store stores and retrieves its data directly to/from disk. The advantage of this over the memory store is that it scales much better as it isn't limited to the size of available memory. Of course, since it has to access the disk, it is also slower than the in-memory store, but it is a good solution for larger data sets.

An RDBMS store is an RDF repository that stores its data in a relational database. PostgreSQL and MySQL are supported.

As a test, we load four different ontologies with different quantity of triplets to different store types (in-memory, in PostgreSQL and MySQL databases) using Jena framework. In Figure 5 there is a graphic that shows the ontology load types for different storage systems. We can see that in-memory storage is the fast way to load the ontology, and that MySQL is the worst case, as it was expected.

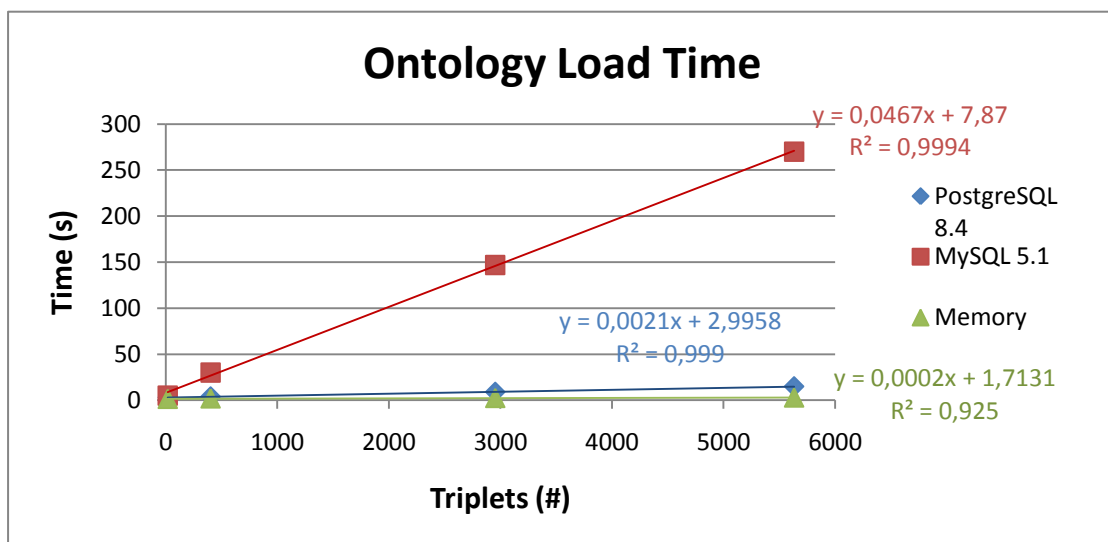


Fig.5 - X Axis shows the number of triplets of the ontology and the Y Axis shows the time in seconds, necessary to load the ontology.

Next, we do tests to know the query time. The tests were done in a PC with a 1.5GHz CPU and 2GB of RAM memory. Doing a SPARQL consult to MySQL and PostgreSQL databases, the query time was lower than 1 second for each load ontology (three different ontologies). When we do the query using memory storage, the time increases as the number of triplets increases, reaching a maximum time of 6 seconds for the ontology with 5633 triplets. It is confirmed that in-memory storage has a good load time but a bad query time and the persistent storage is very good in PostgreSQL and bad using MySQL.

There is a study which compare [47] load and query time of ontologies using RDFLib 2.5 (another RDF framework) which supports MySQL and PostgreSQL. The tendency is the same that we find with our tests.

So, taking into account the efficiency, the best option using Jena framework in a computer is to use persistent storage and PostgreSQL database. While the ontology remain stored in

the database, load time is a little worst than memory storage, but it is only the first time you load the ontology and then every query you need to do is very fast.

We can use Sesame as a server and/or in our application as a java library. As a server, when we have internet connection, we can do the request to the Server, and as a java library when we need to have the libraries in our application. As we did some tests with Jena as a library in a Java program, now we do some tests implementing Sesame Server (with Tomcat 6.0) and using the OpenRDF workbench.

We store different ontologies (the same as Jena tests) in different types of repositories, in-memory, native store and PostgreSQL database. Load time is quite fast, only two or three seconds and query time is practically instantaneous. There is not so much different query time in-memory or persistent storage. So, work with Sesame Server is a good option too.

Implementation on a Mobile

Considering that the final objective of the project is to load and query the ontology or a part of it in a mobile device, it is necessary to find a solution that is also able to work in a mobile device. Then, before deciding which framework and storage system we are going to use, we do some tests on the mobile. We decide to work with Android platform, due to it is Open Source under Apache 2.0 and GNU GPL licenses and there is an API to build applications, easy and simple to use. Tests are done in a mobile with a processor of 600MHz and 256MB of RAM memory.

Although Android applications are developed in Java, in Android it is no possible to run Java applications because “normal” Java Applications execute in a different virtual machine than Dalvik, which is the virtual machine of Android. Android is basically an alternative to J2ME, a rewrite, so it is necessary to port applications written in Java to Android.

Only some utilities of Jena framework are ported to Android, the application is called Androjena [49]. Sesame does not have Android version, so we do the same load and query tests that we have done on the computer in the mobile device using Androjena libraries. If we use Androjena, it is only possible to work with in-memory storage or TDB (porting of SDB is not able yet). We do the tests using in-memory storage because for us it is not useful to work with TDB due it doesn't work with a relational database.

The results in the mobile device are very inefficient, because it takes more than 4 seconds to load and query the ontology. More than 4 seconds waiting for a mobile device answer is too long. For an ontology of 18 triplets it takes around 7 seconds to load and query with SPARQL (in PC takes 2 seconds) but for and ontology of 402 triplets takes around 50 seconds (in PC takes 4 seconds) and for and ontology of 5633 triplets takes about 245 seconds (in PC takes 6 seconds). Our ontology will be over 402 triplets, so it is not possible to wait more than 50 seconds for a simple SPARQL query.

Ontology implementation alternative

As it is not possible to work with RDF data and SPARQL query language in mobile devices, we decide to transform the ontology model to a database schema (described on Figure 5) with the instances from the ontology to database columns maintaining the ontology entities relationships. We do this to improve the efficiency of querying the ontology information. SQL consults are faster than SPARQL, so the consults will be instantaneous. Next to, there is the database schema of the ontology.

As we decided before, for the GIS Server the database is a PostgreSQL database with PostGIS extension. For the mobile device we use the same database schema but in a SQLite database because it is embedded and designed for Android system.

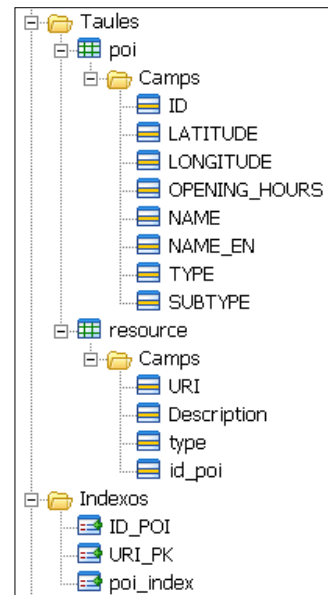


Fig. 6 - Database Schema

As we can see in Figure 6, the database schema has the same tables as entities described for the ontology. The database has the table *POI* with the name of each POI (name) and the English name (name_en), the geographic position (latitude, longitude), the timetable (opening_hours), the kind of POI (type) depending if it is an historic monument, a food amenity, etc, and subtype if it is a museum, a restaurant, etc. The table *Resource* gives as an URI to find where is the resource and the kind (type) of the resource; it is an image, text, video, HTML page.

Next, we need to add a table with user preferences in order to allow the user to choose the preferences of a trip and give them a personalized route.

After studying with is the best GIS architecture to use and decide how to implement an ontology to a computer and mobile device, we explain how to combine together, one of the important objectives of the project.

Integrate Semantic Web in GIS

The use of mobile devices for the project, has led a problem when the user do SPARQL queries to semantic data (RDF data) and we had to apply a countermeasure to not use SPARQL queries. To avoid this kind of queries and to improve the efficiency, we decide to use a database schema (Figure 6) to represent the ontology model. This, in fact, is an advantage to do the integration between geographic and semantic data.

This is so, because now we store semantic data in a relational database. Geographic data is always stored in a spatial database because this kind of databases has geographic functions and spatial indexes necessary to search POIs and to construct a routing system. Then, all the data can be stored in the same database engine but in different tables, which give us an instantaneously connected data. There is a table with the geographic data,

necessary to assign a latitude and longitude to the map, and a table with the POIs and their properties. Finally, there is the table of user preferences.

With this, we also can integrate information from other databases to our own POI information table to have more complete information, for example, we can take the OSM database information for POIs that the volunteers of OSM update continuously.

Another advantage is that we can do traditional SQL queries, which are faster than SPARQL.

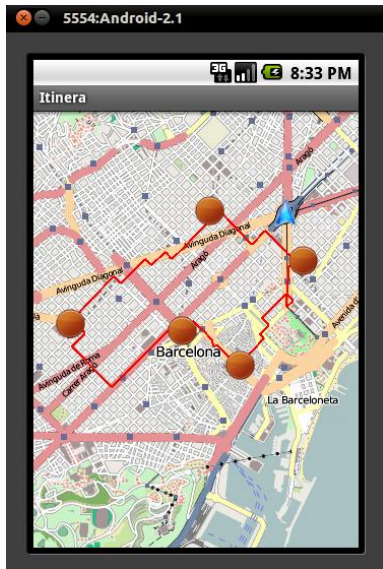
Mobile application

The final objective of this project is to port all this technology to the mobile device, so in this section we explain the tests we do, the problems we encounter and how we solve them.

The biggest problem we encounter with the mobile device part is that there are not spatial databases for Android. If we use SQLite as a database, the spatial extension is SpatiaLite, which is not ported to Android. Although the objective of this study is not porting applications to Android, we study if a spatial database is essential for our mobile application.

What give us the spatial database are the geographic functions and the R-tree indexes.

First of all, we can create our own geographic functions because we don't need a professional GIS. For the routing functions that pgRouting provide with PostGIS, there is a java library, called Traveling-Salesman that can be adapted easily to Android and can do routes from start to end node of OSM vector map data.



We also do some tests in order to know if R-tree indexes are really necessary to improve query efficiency. We are not going to query a large number of data and the geographic functions we need for the application are simple. Results demonstrate that we can leave aside R-tree indexes. The test was based on looking for nearest POIs (geographic function created by us) within a distance of 100 meters; it takes 2 seconds to find 596 nearest POIs and 1 second to find 73 POIs.

Fig.7 – Route with POIs in Barcelona

Considering the fact that we can create the geographic functions in Java and we do not need to use R-tree, brings to the conclusion that we can avoid to use a spatial database in the mobile device.

At the previous page there is an example of the mobile application, which provides a route in Barcelona passing through 5 points of interest. We previously get the map tiles,

map vector data and xml from the GIS Server and know we can show the POIs information or recalculate the route depending on our interests.

Conclusions

After all the tests we did using different semantic web frameworks on computer and on an Android mobile device, we realized that the integration of Semantic Web in GIS could not be done using ontologies and RDF metadata due to the inefficiency of SPARQL queries. So, we decide to import the touristic ontology we created to a database schema (Figure 6) and the instances of the ontology as data to the tables of the database maintaining the relationships of the ontology. With this, is easier to integrate geographic data and other touristic information with our ontology data, because all the data is stored on a relational database and we can do SQL queries.

We achieved to design the architecture of the application for the Itiner@ project (Figure 4). The final architecture of the GIS Server is a GeoServer plus a PostgreSQL database extended with PostGIS spatial database and pgRouting extension to design routes using one of the algorithms it provides. We also use the OpenLayers library to publish the maps on the web, and we get all the data from OpenStreetMap; the map tiles in .png format, the vector map data (ways, nodes and relations) to calculate the routes and the touristic information uploaded from volunteers which include the POIs of different regions. Vector map data and touristic information are stored on PostgreSQL databases.

The mobile application is build for Android and gets the route information (map tiles, xml with POIs and vector map data) from the GIS Server, but instead of a PostgreSQL database uses a SQLite database to store the data and uses a java library instead of pgRouting to calculate the route. We could avoid using a spatial database in the device and we achieved to construct an application that calculates a route for a user, show map tiles and touristic information with POIs. The application can work without Internet connection which is a differentiation from the actual mobile GIS applications.

As future work, we can add more map sources than OSM tiles to show map.

References

- [1] Google Maps [web page]. [last visit 19th January 2011]. <<http://maps.google.es/>>
- [2] Bing Maps [web page]. [last visit 19th January 2011]. <<http://www.bing.com/maps/>>
- [3] Yahoo Maps [web page]. [last visit 19th January 2011]. <<http://maps.yahoo.com/>>
- [4] OpenStreetMap [web page]. [last visit 19th January 2011]. <<http://www.openstreetmap.org/>>
- [5] GIS Desktop Maps [web page]. [last visit 19th January 2011]. <<http://www.gvsig.org/web/home/projects/gvsig-desktop>>

- [6] OpenJUMP [web page]. [last visit 19th January 2011]. <<http://www.openjump.org/>>
- [7] MapServer [web page]. [last visit 19th January 2011]. <<http://mapserver.org/>>
- [8] GeoServer [web page]. [last visit 19th January 2011]. <<http://geoserver.org/>>
- [9] Oliva-Santos, R.; Mendoza-Robaina, L.; Costales-Llerandi, C.; Garea-Llano, E.; Maciá-Pérez, F., Modelo de Anotación semántica para Sistemas de Información Geográfica, VI *Congreso Internacional de Geomática 2009*. XIII Convención y Feria Internacional INFORMATICA 2009, La Habana, Cuba. ISBN: 978-959-286-010-0 (13-02-09), 2009, http://www.cenatav.co.cu/doc/PapersGarea/2009/04_2009.pdf
- [10] Zagal, R.; Torres, M.; Ramirez, T.; Moreno, M., Diseño de una aplicación Web híbrida aplicada a servicios turísticos descritos semánticamente, *5^o Congreso internacional de ingeniería electromecánica y de sistemas*, México, D.F. del 13 de noviembre de 2008, <http://www.tonahtiu.com/SIS032.pdf>
- [12] Hiramatsu, K.; Reitsma, F., GeoReferencing the Semantic Web: ontology based markup of geographically referenced information. Joint EuroSDR/EuroGeographics *workshop on Ontologies and Schema Translation Services*, Paris, France, April 2004, http://www.mindswap.org/2004/geo/geoStuff_files/HiramatsuReitsma04GeoRef.PDF
- [13] Hage, W.; Wielemaker, J.; Schreiber, G. (2010), The Space Package Tight Integration, *TransactionsInGIS*, 2010, 14(2): p131-146
- [14] Batcheller, J.; Reitsma, F., Implementing feature level semantics for spatial data discovery: Supporting the reuse of legacy data using open source components, *Computers, Environment and Urban Systems* 34 (2010) 333–344, 2010.
- [15] Della Valle, E.; Dell'Aglio, D.; Celino, I. (2010), The Experience of Realizing a Semantic Web Urgan Computing Application, *TransactionsInGIS*, 2010, 14(2): p163-181.
- [16] Ramsey, P., The State of the Art of Open Source GIS, *FOSS4G 2007*, 2007, http://2007.foss4g.org/presentations/view.php?abstract_id=136
- [17] JTS Topology Suite [web page]. [last visit 19th January 2011]. <<http://www.vividsolutions.com/jts/jtshome.htm>>
- [18] GeoTools [web page]. [last visit 19th January 2011]. <<http://www.geotools.org/>>
- [19] Proj4 [web page]. [last visit 19th January 2011]. <<http://trac.osgeo.org/proj/>>
- [20] GDAL [web page]. [last visit 19th January 2011]. <<http://www.gdal.org/>>
- [21] OGR [web page]. [last visit 19th January 2011]. <<http://www.gdal.org/ogr/>>
- [22] uDig [web page]. [last visit 19th January 2011]. <<http://udig.refractions.net/>>
- [23] PostGIS [web page]. [last visit 19th January 2011]. <<http://postgis.refractions.net/>>
- [24] pgRouting [web page]. [last visit 19th January 2011]. <<http://www.pgrouting.org/>>

- [25] SpatialLite [web page]. [last visit 19th January 2011]. <<http://www.gaia-gis.it/spatialite/>>
- [26] YOURS [web page]. [last visit 19th January 2011]. <http://wiki.openstreetmap.org/wiki/YOURS_>
- [27] Cloudmade [web page]. [last visit 19th January 2011]. <http://wiki.openstreetmap.org/wiki/Cloudmade_>
- [28] SpatialLite [web page]. [last visit 19th January 2011]. <<http://www.gaia-gis.it/spatialite/>>
- [29] Web Services Geográficos, INCO - Facultad de Ingeniería Montevideo Uruguay, 2010, www.fing.edu.uy/inco/cursos/tsi/TSIG/clases2010/GeoWS.pdf
- [30] RDF [web page]. [last visit 19th January 2011]. <<http://www.w3.org/RDF/>>
- [31] OWL [web page]. [last visit 19th January 2011]. <<http://www.w3.org/2007/09/OWL-Overview-es.html>>
- [32] Linked Data project [web page]. [last visit 19th January 2011]. <<http://linkeddata.org/>>
- [33] SPARQL [web page]. [last visit 19th January 2011]. <<http://www.w3.org/TR/rdf-sparql-query/>>
- [34] Gutierrez, I.; Conesa, J.; Geva, F., Ontologías Turísticas Geográficas: Creación de una ontología sobre rutas turísticas (a Pie o en Bicicleta) por espacios naturales, 2010, http://openaccess.uoc.edu/webapps/o2/bitstream/10609/2284/1/igutierrezl_articulo.pdf
- [35] Prantner, P.; Ding, Y. & others. (2007), Tourism Ontology and Semantic Management System: State-of-the-art analysis, International Conf Internet 2007.
- [36] Geonames Ontology [web page]. [last visit 19th January 2011]. <<http://www.geonames.org/ontology/documentation.html>>
- [37] Dbpedia project [web page]. [last visit 19th January 2011]. <<http://dbpedia.org/About>>
- [38] Auer, S.; Lehmann, J.; Hellmann, S., LinkedGeoData - Adding a Spatial Dimension to the Web of Data, *8th International Semantic Web Conference (ISWC2009)*, 2009, <http://data.semanticweb.org/pdfs/iswc/2009/in-use/paper207.pdf>
- [39] GeoLinkedData project [web page]. [last visit 19th January 2011]. <<http://geo.linkeddata.es/>>
- [40] Cruzar Ontology [web page]. [last visit 19th January 2011]. <http://www.zaragoza.es/ciudad/turismo/es/visitar/ontologia.htm_>
- [41] Turismo de Zaragoza – Ontología Cruzar [web page]. [last visit 14th January 2011]. <http://www.zaragoza.es/turruta/Turruta/index_Ruta>

- [42] Mediatour project [web page]. [last visit 19th January 2011]. <<http://www.mediatur.net/>>
- [43] Infutur project [web page]. [last visit 19th January 2011]. <<http://www.infutur.es/>>
- [44] OpenLayers [web page]. [last visit 19th January 2011]. <<http://openlayers.org/>>
- [45] Jena [web page]. [last visit 19th January 2011]. <<http://jena.sourceforge.net/>>
- [46] Sesame [web page]. [last visit 19th January 2011]. <<http://www.openrdf.org/>>
- [47] Study to compare relational databases backend [web page]. [last visit 19th January 2011]. <http://code.google.com/p/rdfliib/wiki/SQL_Backend>
- [48] Li, S.; Gong, J., Mashup: a New Way of Providing Web Mapping and GIS Services. In *ISPRS Congress Beijing 2008*, Proceedings of Commission IV, 2008, pp. 639-649, http://www.isprs.org/congresses/beijing2008/proceedings/4_pdf/114.pdf
- [49] Androjena [web page]. [last visit 19th January 2011]. <<http://code.google.com/p/androjena/>>

Journal to send:

TranscationsInGIS

(<http://www.wiley.com/bw/journal.asp?ref=1361-1682&site=1>)