Multimodal Planner: From Prototype to Production

Author

Francisco José Peñarrubia, SCOLAB Software Colaborativo, *Spain*

KEYWORDS : planner, multimodal, smart city, bus, metro, bike sharing, open source, routing, public transport

Introduction

In 2010 we started to develop a new Multimodal Web Trip Planner for the city of Valencia. Those days, a new open source project was been developed (OpenTripPlanner [1]), and we decided to use it to achieve our goals: The planner should be able to calculate trips using buses, metro, bikes and/or walking.

In this paper we will present what problems we faced by using a software who was not finished yet, what decisions we took that lead to good (or bad) results, and some lessons learned for the future. We will also explain one innovative algorithm that permits to calculate routes using bike sharing service, mixing bus + metro + bike sharing, and the problems we had (and how we solved them) when the system went to production mode.

Next, we will present the final version of the web [2], remarking some of its more interesting features :

- Multimodal trip planner, offering several alternatives
- Real time information
- Multi language
- Printing capabilites
- Added information (timetables, points of interest, nearest points, etc)
- Integration with OGC Services (the background map is a WMS service, and the bus lines and points of interest and stops come from a WFS service

Implementation

The project has been developed totally with Free Software. The server side is implemented using projects like Mapserver [3], Geoserver [4] and TileCache [5], following the recommendations of OGC (Open Geospatial Consortium [6]). Mapserver is used as a WMS server, and Geoserver implements the WFS services, publishing layers like bus lines, stops and Points of Interest. In order to speed up the rendering of the map, we used Tilecache to publish the tiles generated.

The data is stored in PostgreSQL + PostGIS [7] database, and the graph is generated from GTFS files [8] supplied by EMT (buses) and Metro. The core of the solution is the routing service, based in OpenTripPlanner project.. This project is made up of 3 principal subprojects. The first part is used to generate the graph file from GTFS files + streets. The second part is a routing service (planner) that reads the graph and receives the requests from the client. The third part is in the client side, and offers a user interface to allow specify the origin and destination, hour of the trip, and several parameters more. It uses 2 main JavaScript libraries to achieve a good looking and to support the most important internet browsers in the market (ExtJS [9] + OpenLayers[10]).

Also, to prepare the cartography and the data, we used gvSIG [11], and some new tools have been implemented in order to check topology and incorporate data. About our innovation, the bike sharing problem is a bit different than the rest of the transportation problems. It is similar, although there are some differences :

- Bus and metro gives uses information about timetables in a stop. In bike sharing, you don't wait in the bike station. If there are bikes, you just take the bike and go
- Bus and metro always follow the same path (there are defined lines) but in bike sharing, the user will use the streets network to reach his destination
- Also, there were some minor parameters to take into account (use the bike lines where possible, follow the sense of direction in the streets, etc

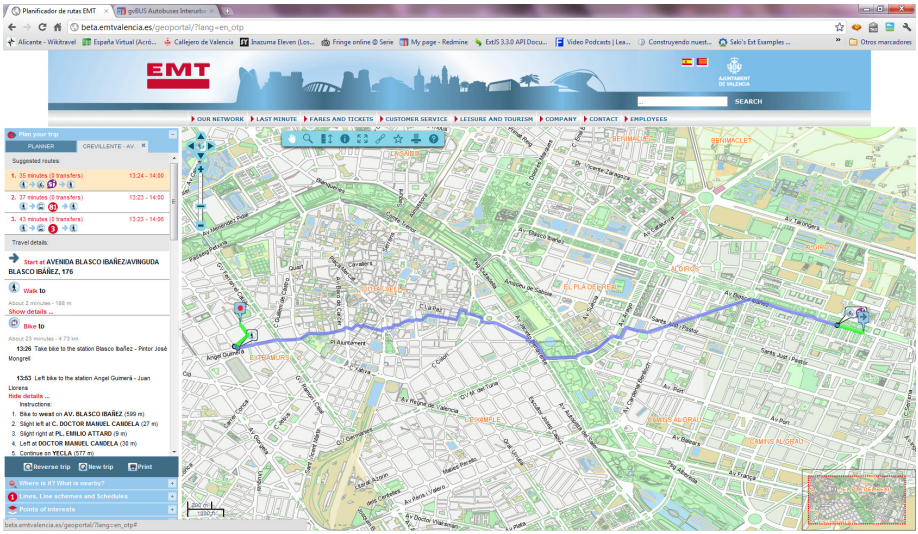

FIGURE 1

Bike sharing calculation

The approach to solve the bike sharing integration with the route planner is based on the idea of creating a virtual network where the bike sharing user (ValenBisi [12]) can go to a bike station, take the bike, and in this moment, the labeling of network nodes will run with the virtual network. This way, the minimal time in a node is different if the user arrives there walking or by bike.

Then, the user rides the bike to a close bike station of the destination and there, he left the bike in the station an start walking, maybe to arrive to the destination, or maybe, to take another transportation mode (bus or metro). This is the improvement, the calculation mixes bus, metro and/or bikes, allowing to mix every kind of transportation.

There was some difficult issues solved that it would be good to take care in the future. One was the requirement to keep compatibility with some old services in use. With this feature in mind, it was a good decision to use a common (and open model). Goggle Transit Feed System (GTFS) file was created to send

data to Google Transit, and this file is the start point to calculate the Graph object suitable to OpenTripPlanner routing service.

We faced also a lot of problems in client side programming because JavaScript behavior is different in Internet Explorer, Chrome, FireFox, etc. The library OpenLayers and ExtJS helped us a lot, but in the end, we expend very much time in debugging JavaScript.
And finally, when we put the production system running, we had problems with memory consumption and the server going down when we received many requests. To avoid this in the future, a good testing plan should be chosen, and is good to be prepared to increase RAM or processors in the server.

Also, is good to separate routing services from mapping services. In our case, the map requests (tile cache) were serviced by MapServer + TileCache, so, each time a new user enter the geoportal, lot of request (each tile for every layer) is requested, and also some WFS requests to get information about the stops in the visible extent. The Apache server was in pre-fork mode, due to some php limitations in other parts of the portal. This means that every single request will be attended by a new process, and this leads to more memory consumption. This was the main reason that put the server down when lot of users accessed the geoportal at the same time. The memory was used by Apache, and Tomcat (where the routing service is) couldn't work well. Luckily, all these problems were solved (and many more that are outside of this paper) and the geoportal now is up and in production mode.

Conclusions

Summing up, the experience has been very positive, and the client is very happy with the result. Each day, around 3000-4000 route calculations are done, and the user experience has been improved very much.

The advantages of this web compared to Google Transit, for example are :

- Better information and easy to maintain. One of the problems that EMT wanted to solve with this system was related to the delay between the GTFS delivery and the use of the new data from Google. Those days,

around 15 days was usual, so, it was impossible to reflect incidences in the web page. (Now, Google is testing in USA a new format that permits real time communication of incidences, but is still not ready in Europe)

- Real time information about arrivals
- Street data and tourist info directly maintain by the cartography service from Valencia's city council
- Better integration with the rest of the web page
- Support for ValenBisi information and really multimodal integration

In near future, a mobile version will be released (for iPhone and Android phones) and the new bike sharing algorithm may be used for other kind of sharing (car sharing, electric bikes, and so on).

[1] OpenTripPlanner wiki. GitHub. Retrieved August 31, 2012 from https://github.com/openplans/OpenTripPlanner/wiki/

[2] EMT. Empresa Municipal de Transportes geoportal page. Retrieved August 15 , 2012 from http://www.emtvalencia.es/geoportal/?lang=en_otp

[3] MapServer project homepage. Retrieved September 3, 2012 from http://mapserver.org/

[4] GARNET J. AND OTHERS. GeoServer web page. Retrieved September 3, 2012 from http://geoserver.org/

[5] TILECACHE CONTRIBUTORS. TileCache wiki. Retrieved September 3, 2012 from http://tilecache.org/

[6] OGC. Open Geospatial Consortium homepage. Retrieved September 3, 2012 from http://www.opengeospatial.org/

[7] Postgis homepage. Retrieved September 3, 2012 from http://postgis.refractions.net/

[8] GOOGLE. General Transit Feed Specification Reference. Retrieved September 3, 2012 from https://developers.google.com/transit/gtfs/reference

[9] SENCHA INC. ExtJS library web page (3.x). Retrieved September 3, 2012 from http://www.sencha.com/products/extjs3/

[10] OpenLayers wiki. Retrieved September 3, 2012 from http://openlayers.org/

[11] GVSIG ASSOCIATION. gvSIG project homepage. gvSIG Portal. Retrieved August 31, 2012 from http://www.gvsig.org

[12] JDCDECAUX ESPAÑA S.L.U. ValenBisi homepage. Retrieved September 3, 2012 from http://www.valenbisi.es/