City Focus: A web-based interactive 2D and 3D GIS application to find the best place in a city, using open data and open source software

Carolina Arias Muñoz, Simone Corti, Monia Elisa Molinari, Daniele Oxoli, Gabriele Prestifilippo

Department of Civil and Environmental Engineering, Politecnico di Milano Como Campus, Como, Lombardy, Italy

Corresponding Author:
Carolina Arias Muñoz

Email address: carolina.arias@polimi.it

Session: CitySmart, Open Source Apps for Urban Management (chair: Hogan; Brovelli)

ABSTRACT

City Focus is a web-based interactive 2D and 3D GIS application to find the best place in a city to live as well as to pass shorter staying. The user can select among different criteria and decide their importance by assigning weights to each of them. The application provides thematic maps displaying insights on the places which better fit the user's preferences. The resulting map is computed through map algebra by means of Web Coverage Processing Service WCPS provided by RASDAMAN Database Management System. Data visualization is mainly based on NASA Web WorldWind open-source virtual globe. The app exploits exclusively open data as well as Free and Open Source Software (FOSS) for its implementation by enabling continuous improvements while minimizing development costs.

Keywords: Open data Free and Open Source Software NASA Web WorldWind RASDAMAN, MyGEOSS

INTRODUCTION

City Focus is a work-in-progress application, winner of the MYGEOSS third call for innovative Apps, launched by the European Commission (http://digitalearthlab.jrc.ec.europa.eu/mygeoss/results3.cfm). The call focused on developing applications of European relevance able to provide users with quantitative or qualitative information on the changing environment, taking into account broad environmental or social themes across geographic scales, stressing the use of openly available or crowd-generated data in different domains addressing citizens' needs (Raglia & Craglia, 2015).

City Focus is a web-based interactive 2D and 3D GIS application to find the best place in a city to live, or to pass shorter staying. The user can select among different criteria and decide their importance by assigning weights to each of them. The application provides thematic maps displaying insights on the places which better fit the user's preferences. The resulting map is computed through map algebra upon a cascade of pre-built raster maps. Users are guided in the requests through a predefined list of criteria including: air quality, precipitation, terrain morphology (e.g. slope, solar exposure, etc.), temperature, proximity to urban facilities (e.g. schools, transportations, sports fields, cultural attractions, healthcare centers, parking lots, shops, etc.), presence of public spaces, land use characteristics (e.g. commercial areas, residential areas, etc.), population density, traffic or internet connection.

CITY FOCUS APPLICATION DATA

As required by the MyGEOSS call, we focus on fully open data available for the city of Milan, which was selected as initial case study for the first app implementation. Suitable data sources were: Open Data Lombardia (https://www.dati.lombardia.it), Dati Comune Milano (http://dati.comune.milano.it), OpenStreetMap (https://www.openstreetmap.org). These include vector layers regarding urban facilities (e.g. public transports, public green areas, etc.), primary services (e.g. schools, hospitals, commercial centers, etc.) as well as recreational services. Environmental and meteorological data instead were retrieved from different sources. Examples are maps of temperatures from EuroLST dataset (http://www.geodati.fmach.it/eurolst) as well as DTM and land cover maps from SINAnet geoportal (http://www.sinanet.isprambiente.it).

Due to the different formats, spatial extension and resolution of the data, a preliminary processing was required to feed the application database with homogeneous information. The goal was to produce raster maps containing information about *presence*, *spatial concentration* or *proximity* to any of the city features described by the data collected. Regarding point data (e.g. primary services, urban facilities, etc.), data from different sources were merged in a single shapefile and all the possible duplicates were filtered out by means of buffers. Concentration maps were computed using v.kernel module of GRASS GIS (https://grass.osgeo.org). Polygonal features (e.g. parks, water bodies, etc.) were instead first rasterized and then, multiple-distance buffer maps were generated by means of r.buffer GRASS GIS module, to represent both presence and proximity to these features.

Raster maps obtained in the previous steps as well as those data originally sourced in raster format were re-sampled according to an equal grid in order to harmonize both layers extension and cell resolution. Cell values were then normalized to allow the computation aimed to the generation of the output map, performed by the server side as explained in the following paragraphs.

APPLICATION ARCHITECTURE AND PROCESSING

The server side is based on the RASDAMAN Database Management System (http://www.rasdaman.com) for data storage and processing while the client exploits the NASA Web WorldWind open-source virtual globe (https://webworldwind.org) for geospatial data visualization.

Recently, RASDAMAN has arisen as a scalable and cost-effective database solution to store and retrieve multi-dimensional raster data (arrays). RASDAMAN conceptual model consists of multidimensional arrays with n-dimensions extend. Data storage relies on partitioning of the arrays into sub-arrays called tiles, which can be stored either in a relational database, in files in the data systems or archives of files that can be accessed externally (Baumann et al., 1999). Data processing within RASDAMAN is done through the Web Coverage Processing Service WCPS, which defines a protocol-independent language for the extraction, processing, and analysis of arrays (Open Geospatial Consortium, 2009), that can include complex operations such as slicing or trimming, induced operations (arithmetic, exponential, trigonometric among others), condensers (count, avg, some, all), metadata extraction or reprojection (Aiordăchioaie, A. & Baumann, P., 2010).

NASA Web WorldWind is an open-source virtual globe, which allows to display several kinds of geospatial data also in three-dimension. It runs on all the modern browsers thanks to WebGL, JavaScript, and HTML5. Web WorldWind comes along with a rich set of APIs (application programming interfaces) enhancing the possibility for developers of customization as well as implementation of new functionalities. It allows to display different kinds of data in a three-dimensional globe and it’s compliant with different OGC standards such as WMS and KML.

The application’s architecture is summarized in the following figure:

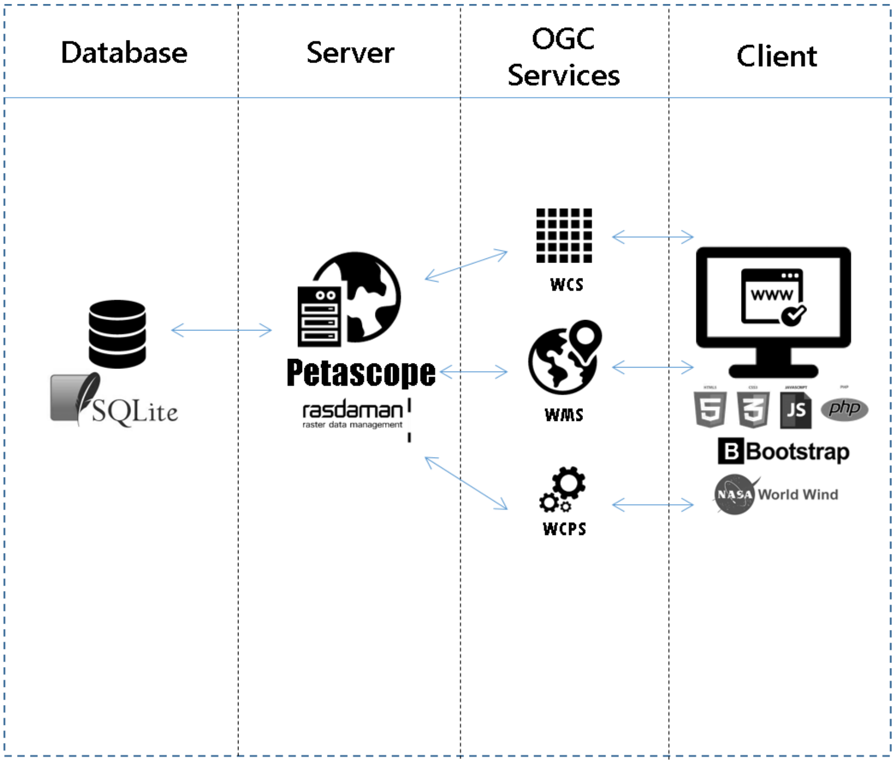
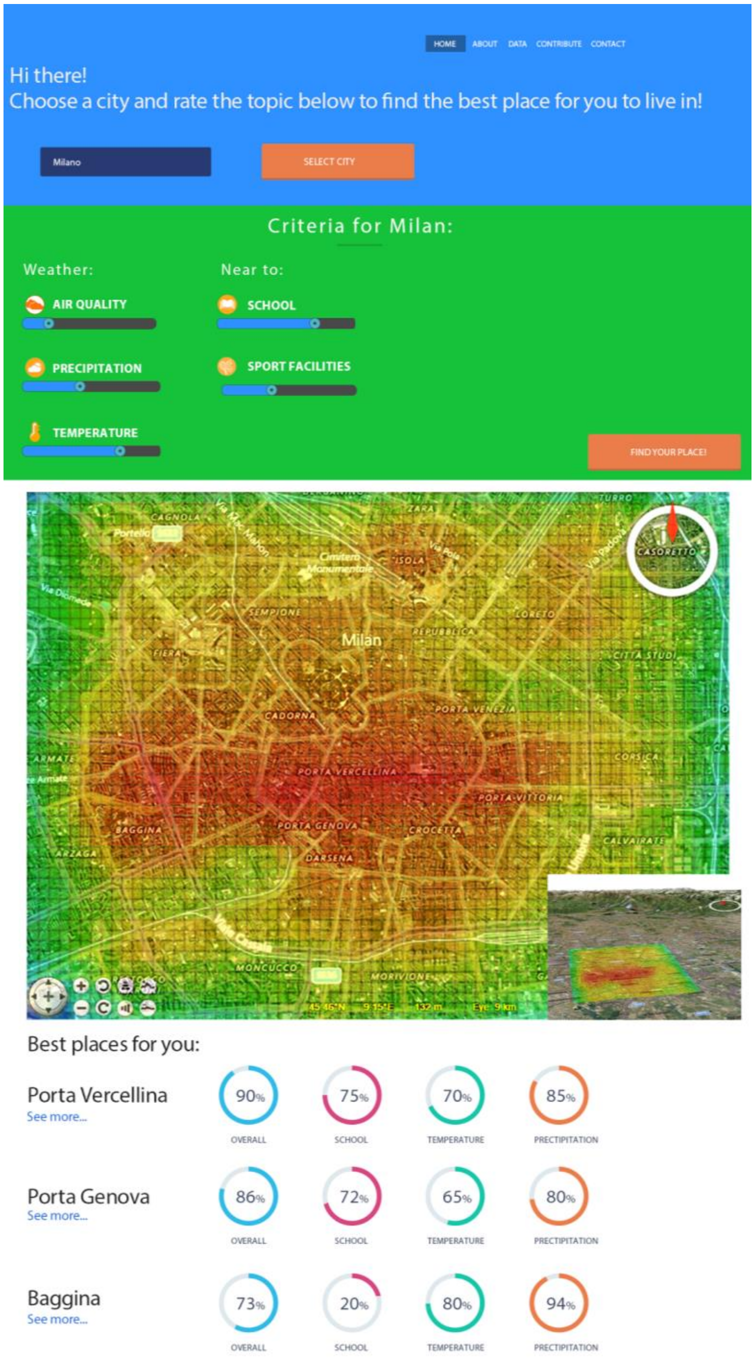


Figure 1. City Focus architecture

On the server side, RASDAMAN provides many functionalities to process raster datasets. The calculations are performed using WCPS requests. Data are accessed over the web by the Petascope component of RASDAMAN, translating incoming WCPS requests into RASDAMAN rasql queries to efficiently process the different raster maps and generate the output map. The City Focus application relies on a standard installation of RASDAMAN server with a SQLite database backend, were both the bulk data and the supplementary spatial semantics (coordinates, resolution, and metadata) reside.

For what it concerns the client side, different tools were considered to build up the interface for maps visualization. These were mainly jQuery (https://jquery.com) a light JavaScript library to interact with the HTML page and NASA Web WorldWind for maps visualization, both in two and three dimensions considering the terrain elevation. Combining jQuery and Web WorldWind, it is possible to retrieve maps from RASDAMAN through the WPCS and show them to the end-user. On the graphical side, jQuery UI (https://jqueryui.com) and Bootstrap (http://getbootstrap.com) allowed creating dynamic components such as sliders, buttons, and forms. These libraries have been adopted to enable users' selection of weights for the available criteria.

As shown in the application mockup included in figure 2, the users will be asked to assign a proper weight to each criterion by simply moving these sliders from the app interface. Users' preferences will be automatically sent to the server employing a WPCS request. The output consists of a raster "score map" computed as a weighted average of the normalized raster maps representing the user's selected criteria. The final map is then displayed with an intuitive color gradient, enabling the user to identify the best places within the city which better fits his/her preferences.



Hi there!
Choose a city and rate the topic below to find the best place for you to live in!

Milano

Criteria for Milan:

Best places for you:

Porta Vercellina See more...	90%	75%	70%	85%
	OVERALL	SCHOOL	TEMPERATURE	PRECTIPITATION
Porta Genova See more...	86%	72%	65%	80%
	OVERALL	SCHOOL	TEMPERATURE	PRECTIPITATION
Baggina See more...	73%	20%	80%	94%
	OVERALL	SCHOOL	TEMPERATURE	PRECTIPITATION

Figure 2. City Focus architecture

DISCUSSION AND FUTURE DEVELOPMENTS

Most of the existing apps (see http://teleport.org, http://www.findyourspot.com, http://where2roost.com) focus on finding a city to live and not on identifying a suitable place within a city, which implies a higher level of information accuracy and a deeper analysis scale.

The existing apps also allow searching for places to live by specifying few parameters such as apartment or house prices. In this cases, once the user has chosen a location, it might need to retrieve additional information about the place in order to make a decision. This application will help to perform this task in an automatic as well as user-friendly way avoiding long and hand-made search on the Web. City Focus takes into account environmental conditions such as air quality levels or microclimate conditions, that existing apps do not consider. The app exploits exclusively open data as well as Free and Open Source Software (FOSS) for its implementation by enabling continuous improvements while minimizing development costs.

Possible improvements regard the possibility for users to get a glimpse of the changing environment within a city through a trend maps or graphs for any specific criterion (e.g. temperature changes in the last five years, etc.) This can also be helpful while selecting weights for each of them. Moreover, name of best scored city blocks from the final map (i.e. the cells with the highest scores) may be displayed too, associating to these cells to neighborhoods by means of geocoding.

As a first case study, we considered the city of Milan. In the future, other Italian as well as European cities are planned to be included. The app's adaptability to a new city should be smooth: Environmental conditions data can be found at a European scale (e.g. Temperatures from EuroLST dataset, air quality from AirBase - the European Air quality database http://www.eea.europa.eu/data-and-maps/data/airbase-the-european-air-quality-database-7) as well as urban facilities, primary services and recreational services that can be obtained from OpenStreetMap; although information can always be enrich with local cities databases.

ACKNOWLEDGEMENTS

The application is being developed within the MyGEOSS project, which has received funding from the European Union's Horizon 2020 research and innovation programme.

REFERENCES

Aiordăchioaie A., Baumann P. 2010. PetaScope: an open-source implementation of the OGC WCS geo service standards suite. In: Scientific and Statistical Database Management. Springer, 160–168.

Baumann P., Dehmel A., Furtado P., Ritsch R., Widmann N. 1999. Spatio-temporal retrieval with RasDaMan. In: VLDB. 746–749.

Raglia E., Craglia M. 2015. MYGEOSS Project. GeoInformatics 18:38.

Open Geospatial Consortium. 2009. Web Coverage Processing Service (WCPS) Language Interface Standard.