urbSAT: from spatial SQL to urban indicators

Erwan Bocher[1], Thomas Leduc[2], Nathalie Long[3], Fernando Gonzales-Cortes[4], Guillaume Moreau[5]

[1]Ecole Centrale de Nantes, IRSTV, Nantes, France, erwan.bocher@ec-nantes.fr
[2]CERMA, IRSTV, Nantes, France, thomas.leduc@cerma.archi.fr
[3]UMR ESO CNRS 6990, IRSTV, Nantes, France, nathalie.long@univ-nantes.fr
[4]IRSTV, Nantes, France, fergonco@gmail.com
[5]Ecole Centrale de Nantes, IRSTV, Nantes, France, guillaume.moreau@ec-nantes.fr

Abstract

This paper proposes both a tool for indicator production and a language with spatial operators in order to narrow the border between severals expert (urban manager and scientist). In the spirit of Spatial Data Infrastructure, the aim is to promote best practices and guidelines around geosciences, not only on sharing data, as focused currently, but also on sharing spatial processing method that are used to build knowledges. Beyond the strict field set of practices, themes and business logics, we aim to set up and deploy a common language and the corresponding framework. Those will provide all researchers the ability to produce complex indicators using knowledge of several different disciplines (geography, sociology, climatology, urban structure, ecology...). We present the urbSAT (urban Spatial Analysis Tool) that includes a workflow and a set of spatial operators that are integrated into a graphical GIS layer named OrbisGIS. To demonstrate the capabilities of urbSAT two indicators are presented.

1. Introduction

GIS have become unavoidable tools for the development and management of territories. Deployed in urban environments, they are used for diagnosis, public debate and as decision-making tools for both experts and non-experts. For urban issues, GIS need several models: 2D and 3D geometric modeling, simulation of natural phenomena (light, heat, water, wind, noise, etc.), multi temporal various scale data representation, visual representation. Due to this complexity and the amount of data, urban GIS are faced with a number of issues concerning data modeling, storage, querying and finally representation, hence the need for indicators to assist urban planning. As far as sustainable development is concerned, the main challenge with indicators is to integrate interdisciplinary knowledge in order to allow a better understanding of urban issues by a larger set of people that thus will be able to share their expertise.

In this paper, we propose both a tool for indicator production and a language with spatial operators in order to narrow the border between each expert (urban manager and scientist). In the first section we will explain the challenges that are specific to the urban context and our vision of knowledge sharing and mutual enrichment of both data and methods. The next section will make a short synthesis of existing works. Then, we will present the urbSAT (urban Spatial Analysis Tool)

Proceedings of the academic track of the 2008 Free and Open Source Software for Geospatial (FOSS4G) Conference, incorporating the GISSA 2008 Conference
29 September - 3 October 2008, Cape Town, South Africa
ISBN 978-0-620-42117-1

that includes a set of spatial operators and urban functions. The last section will present two urban indicators.

2. Motivation

At present, more than a half of the world population is living in urban areas. Urban areas are in fast expansion. Space and energy consumption are increasing. Therefore urbanization control has become a major issue. Moreover, the two most important greenhouse gases producers are transportation and elder buildings. Sustainable development that is seeking the best tradeoff between the fulfillment of human needs maintaining the quality of the natural environment indefinitely by applying different methods and criteria in urban planning and management. Among the social, economical and ecological factors that are taken into account in decision-making, environmental issues now have a greater impact. In this domain, it is necessary to have a wide knowledge of physical, chemical and meteorological mechanisms that operate in urban environments to address such challenges as enhancing the environmental quality of the city, limiting energy consumption and the associated costs, reducing pollutions and preventing risks but also to able to answer public demand in terms of comfort, security and public infrastructures.

To address these issues, researchers are faced with severals problems among which are tool disparity, knowledge sharing and enhancement (both for data and methods). Indeed, different kinds of tools are used ranging from Fluid Mechanics (with Finite Elements Methods computation) to in situ social inquiries, hence the difficulty to integrate the knowledge produced by those tools. What they usually have in common is a geographic reference.

Recently, attention has been focused on the development of Spatial Data Infrastructures (SDI) at different levels (local, national) and for various policy areas and priority of actions such as social, economic, and environmental issues (Clinton, 1994, INSPIRE 2007). All of them share the same goal: maximize user's access to spatial data, minimize the redundancy of efforts and investments (Nebert, 2004). The interoperability of the system is guaranteed by several standards for data exchange and processing (GML, WFS, WMS).

In the spirit of SDI, we will promote best practices and guidelines around geosciences, not only on sharing data, but also on sharing spatial processing method that are used to build urban knowledges. Beyond the strict field set of practices, themes and business logics, we aim to set up and deploy a common language and the corresponding framework. This common will provide all researchers the ability to produce and share complex indicators using a spatial standardized SQL. In terms of uses and learning, the value-added will be a common reusable syntax between many users whatever their speciality (climatology, transport, urban structure...). It provides an opportunity to avoid or eliminate all the barriers through integrated models that transcend disciplines.

3. Related work

3.1 Urban indicators

Within advancements in data collection, computer software including GIS, and image processing, several researchers have developed automated or semi-automated method to extract urban indicators like morphological information: (Ratti and Richens, 1999) use image processing with the MATLAB software package; (Burian et al, 2002) propose an approach using GIS from US database to process 3D building datasets and then, multi-spectral images to compute an expanded set of parameters including surface cover fraction with the ESRI ArcGIS software package (Burian, 2003).

At IRSTV institute a software called DFMap was developed to analyse urban tissue (Long, 2003). It allows computing not only some statistical parameters describing the morphology of the buildings but also aerodynamic parameters like roughness length and displacement height. It was meant as a pre-processor for atmosphere numerical simulation models that are based on a regular vector grid-based map of significant parameters such as rugosities, thermal and radiative flows.

DFMap uses the BDTopo® spatial database that is produced by the French National Geographic Institute (IGN) from aerial photographs. This database is composed of vector land cover data, including vegetation, hydrographic network, roads and buildings. The topography is also included in this database as well as contours and spot heights.

The DFMap computes several functions over a grid :

- Functions describing buildings morphology like height, surface, volume, perimeter,…
- Functions describing the cover mode with density calculation,
- Functions providing aerodynamic parameters like roughness length and displacement height thanks to Bottema's and Raupach's models (Bottema, 1997, Raupach, 1992).

The user can choose cell size, set wind direction to compute frontal and lateral surface densities used in the roughness calculation. The results are presented in cartographic map with associated numerical data. They are used to build a spatial typology about urban tissue.

Nevertheless DFMap has some limitations :

- It can only compute urban indicators from BDTopo® database, stored in DXF format. It is a strong limitation with respect to the availability of various databases like urban GIS, or satellite images.
- It has been developed from scratch as a monolithic application. It does not take advantage of GIS capabilities such as reprojecting and geometric computations or map generation facilities.
- The spatial functions are statics. It's not possible to combine each others to create new indicators.
- It is not pluggable and it is not freely available.

To solve these limitations and to facilitate the knowledge sharing as denote before we decided to extract the main DFMap functions and to integrate then onto a more generic spatial query language.

3.2 SQL spatial query language

The questions of spatial query language have been thoroughly studied over the last years. As reviewed by (Lopes de Oliveira, 1996), the need for spatial query formalism has been clearly identified by and answered in (Egenhofer, 1988), with GEOQL (Ooi, 1990), Spatial SQL (Egenhofer, 1994). The SQL language has already been spatially enabled and extended by the Open Geospatial Consortium in the OpenGIS ”Simple Features Specification for SQL“ (Herring 2006a,b). Indeed, this specification defines a spatial object model and fundamental geometric functions, including spatial predicates and spatial operators (intersection, difference, union, symmetric difference). It proposes a standard SQL schema that supports storage, retrieval, query and update of simple geospatial feature collections but adapted to RDBMS. For example, the following query returns a new geometry defined by buffering a distance "d" around geometry object, where "d" is in the distance units for the Spatial Reference of geometry.

```
SELECT Buffer(geometry, d) FROM roads;
```

where roads is a table stored in a Relational DBMS and geometry is a column that contains the value encoded in Well Known Binary format. The most famous implementation is probably PostGIS, a geographic extension for the RDMS PostgreSQL.

Recently a generic processing library called GDMS (Generic Data Source Management System) has been developed at the IRSTV institute (Bocher et al, 2008). With GDMS, the idea was to postpone all problems of interoperability across data repositories, but also about the SQL semantics by developing a highly flexible, portable and standards compliant tool to build SQL queries. Thus GDMS is an unified abstract layer that allows addressing, managing and modifying any attributes and spatial data's from a single point of view. GDMS is able to make homogeneous SQL queries on data, so as to filter, order, update without worrying about the nature of the data carrier (databases or flat files). However it is possible to extend the semantics from typical SQL with customized functions and queries, developed in Java. At least, it's significant to mention that GDMS is open source and takes advantage of two famous libraries: Java Topology Suite for vector data and ImageJ for raster data. Theses choices allow us to concentrate the effort on the specific spatial processing need to build urban indicators and not on basic spatial operators that already exist.

4. UrbSAT

UrbSAT has been derived of the generic system GDMS. The provided interface is a SQL language that preserves the grammar of the SQL-92 standard and adds to this standard the concept of geometry and spatial functions as defined by the OGC simple features SQL specification (Herring 2006a,b). The most interesting feature of the UrbSAT is the possibility to extend the semantics from typical SQL with customized functions and queries. Those functions and queries are artifacts developed in java that can be easily reused in another SQL statement by referencing its name. For example, the buffer function takes a geometry as input and outputs the buffered geometry. The following statement produces a new data source with as many rows as the original data source (mysource), having only one new field with the buffer of the original geometries:

```
SELECT buffer(the_geom, 20) from mysource;
```

As mentioned in the previous section, the mechanism to build urban indicators is to process vector data and to aggregate information on a vector grid. In order to obtain an abstract reusable workflow that will be used as common basis for all future studies and integrate it in the GDMS SQL semantic we have decomposed the DFMAP methodology. We have identified two types of indicators :

- case 1 : the one that can be compute using a sequence of spatial OGC operators and predicates.
- case 2 : the other that need specific function such as create a regular vector grid.

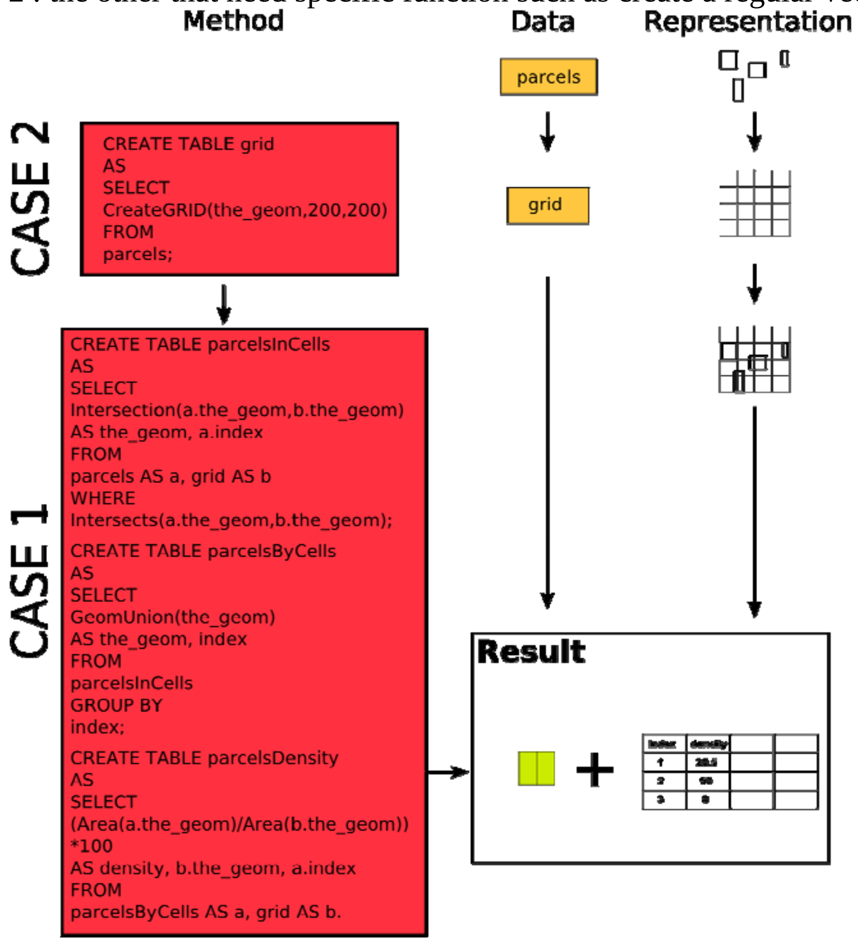

Figure 1: Description of the density computing process

Figure 1 presents an example with the actual process required to obtain a density indicator. It shows step by step the SQL statements that are used to manipulate the different input data to produce expected result. The purpose of this indicator is to build a regular grid of polygons that will contain as an attribute the density of the features in the input data for each one of those polygons. Looking thoroughly to the sequence of instructions that compute the density indicator it is possible to see how UrbSAT takes advantage of some basic SQL functions to calculate the area of the parcels, to intersect different sources and to aggregate the results (case 1). Furthermore, we can show how the SQL semantic is extended by a custom function "CreateGrid" that creates a regular grid of polygons with a 200m cell size (case 2).

UrbSAT is an application of the OrbisGIS open source GIS software developed at IRSTV. The extended query language is available using a SQL console that permits to deal with the layers loading in the OrbisGIS map viewer (Leduc et al, 2007).

5. Example case study

We will present two types of indicators that are computed over a grid with a 200m cell size : average building and clusters classification.

5.1 Average building

We have rebuilt the 3D objects in order to compute the average height for every building. The spatial vector layer, inherited from the new BDTopo® database, contains the altimetry data, relative to the sea level, of each building roof. The aim of the urbSAT GetZDEM custom query is to produce a new vector layer mixing the input DEM and the buildings vector layer (figure 2). This new vector layer enriches the previous one with a new numeric attribute that corresponds to the height of the corresponding building, and it adds to the 2-dimensional geometric field copied from the vector layer a third dimensional value that corresponds to the Z of the underlying DEM cell.

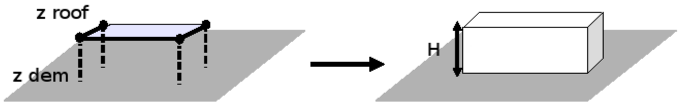

Figure 2: The GetZDEM process

Query : select GetZDEM(r.raster, a.the_geom) from mybuilding a, mydem r;

GDMS underlying layer provides also a full extrude function that does not only create the ground footprint and the roof but also the walls. It gives the possibility to build a real three dimensional representation of parcels and buildings over a DEM. This function, from a single table of 2D geometries (buildings floor) and a numeric field (building height), is able to produce a volume that represents the corresponding building (Figure 3).

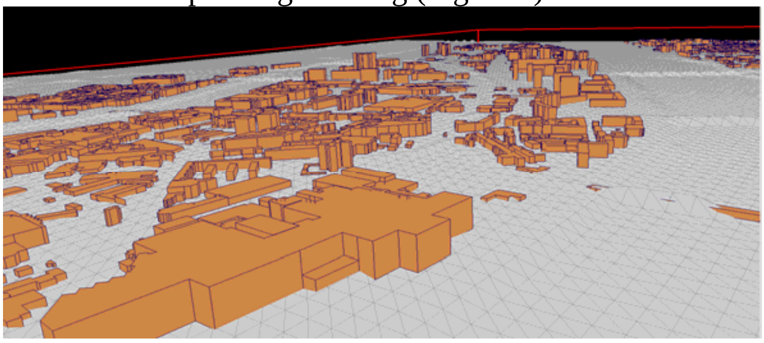

Figure 3: 3D representation of urban buildings over DEM

Query : select Extrude(id, height) from mybuilding;

At the end the new layer that represents the building in 3D is used to compute the average height by cells.

5.2 From densities to clusters classification

This step consists in computing a global envelope based on each input data source boundary. This envelope is the reference used to construct the regular grid (Figure 1). Then, the 5 following densities can be computed:

- Density of built areas
- Density of areas covered with natural vegetation,
- Density of asphalt-covered areas
- Density of water-covered surfaces.
- Average building density.

An example is illustrated in figure 4.

These densities populate the grid and they are classified with the k-means algorithm. The statement is expressed as : SELECT KMeans(9) from mydensities;

where 'mydensities' is a the vector grid that contains for each cells all the densities. '9' represents the number of clusters. Figure 5 shows the result on the French city of Le Havre.

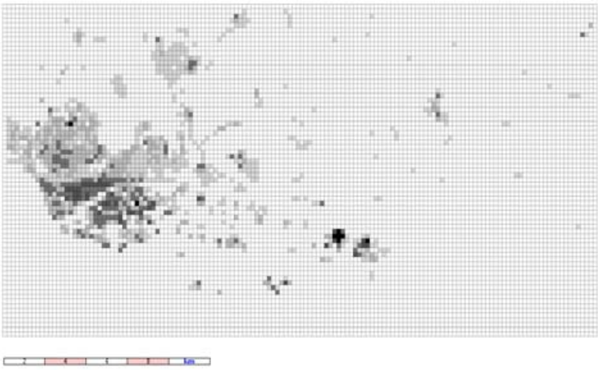

Figure 4: *built area density on Le Havre*

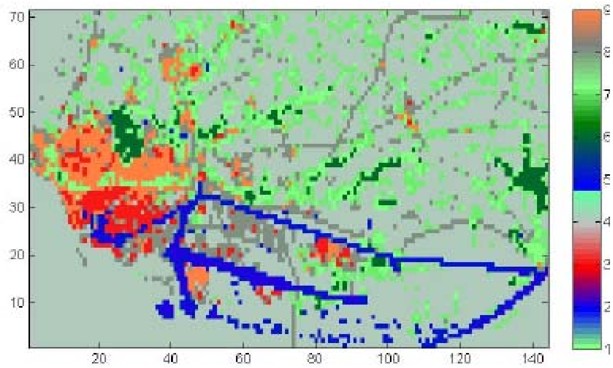

Figure 5: K-means classification on Le Havre.

6. Conclusion

We have presented the UrbSAT tool that is a set of spatial functions dedicated to urban indicators production. UrbSAT is based on GDMS, a generic library, and offers a common way to query spatial data in SQL. It is a large refactoring and generalization of DfMap as it is meant to be used for any simulation and is GIS-based. This latter argument drastically reduces the amount of development needed to produce new indicators and should favor knowledge sharing among researchers.

References

Bocher, E Leduc, T Moreau, G & González Cortés, F 2008. 'GDMS: an abstraction layer to enhance Spatial Data Infrastructures usability', 11th AGILE International Conference on Geographic Information Science - AGILE’2008, Girona, Spain.

Bottema, M 1997, 'Urban roughness modelling in relation to pollutant dispersion', Atmospheric Environment, 31, pp 3059-3075.

Burian, S.J. Brown, M.J. Linger, S.P. 2002, ' Morphological analysis using 3D building databases',Los Angeles California, LA-UR-02-0781, Los Alamos National Laboratory, 66 p.

Burian, S.J Han, W Velegubantla, S Maddula, S.R.K. 2003,'Development of gridded fields of urban canopy parameters for models-3/CMAQ/MM5', U.S. EPA Internal report, RTP, North Carolina.

Clinton, W 1994, Executive Order 12906: Coordinating geographic data acquisition and access: The National Spatial Data Infrastructure.

Egenhofer, M.J. & Frank, A 1988,'Towards a Spatial Query Language: User Interface Consideration', *In Proceedings of the 14th International Conference on VLDB*, L.A., California, pp 124-133.

Egenhofer, M.J 1994,'Spatial SQL: A query and presentation language', *IEEE Transactions on Knowledge and Data Engineering*, 6(1), pp 86–95.

Herring, J.R 2006, 'Opengis® implementation specification for geographic information - simple feature access - part 1: Common architecture'.

Herring, J.R 2006, 'Opengis® implementation specification for geographic information - simple feature access - part 2: Sql option'.

INSPIRE, 2007,'Directive of the European Parliament and of the Council establishing an Infrastructure for SpatialInformation in the European Community (INSPIRE)', http://www.ec-gis.org/inspire/.

Leduc, T Gonzaléz Cortés, F Bocher, E 2007, 'OrbisGIS : A GIS for scientific simulation',*World Free Software Meeting (RMLL)*, Amiens, France.

Long, N Kermadi, S Kergomard, C Mestayer, P.G Trébouet, A 2003, 'Urban cover modes and thermodynamic parameters from urban data base and satellite data: comparison for Marseille during ESCOMPTE', *5th International Conference on Urban Climate*, Lodz, Poland, Proceeding Vol 1, pp 91-94.

Nebert, D 2004, 'Developping Spatial Data Infrastructures: the SDI Cookbook'. Online book: www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf .

Ooi, B.C 1990,' Efficient Query Processing in Geographic Information Systems', Lecture Notes in Computer Science 471, Springer-Verlag.

Raupach, M.R 1992, 'Drag and drag partition on rough surfaces', *Boundary Layer Meteorology*, 60, pp 375-395, 1992.

Ratti, C Richens' P 1999, 'Urban texture analysis with image processing techniques', *Proceedings of the CAADFutures99 Conference*, Atlanta, GA.