GDMS-R: a mixed SQL to manage raster and vector data

T. Leduc, E. Bocher, F. González Cortés, and G. Moreau
IRSTV – CERMA Laboratory – Ecole Centrale de Nantes
GIS 2009, Ostrava (Czech Republic), January 26th 2009
Outline

- Context and motivations,
- Internal architecture and data model,
- Some use cases,
- Need of some enhancement...
The context

Urban remote sensing

Urban microclimatology

Urban data
  - Spatial Data Infrastructure
    → OrbisGIS

Urban sound environment

Sustainable Urban Project

Pluri-disciplinary workshop sector
Motivations

- Deals with several formats (some of them are mandatory due to existing simulation software) but also with complex and heterogeneous simulation softwares,
- Respect users practices constraints but also international standards,
- Provides pre-processed data for the SDI,

⇒ GDMS: Generic Data source Management System = API + extensible OGC compliant SQL.
Internal architecture

- A layered architecture that stacks:
  - a driver layer,
  - an adapter layer,
  - an application layer...
- That can be extended
  - I/O,
  - functionalities.
Extended data model

- `<DataSource, row, field>` close the GeoAPI model
- `<FeatureCollection, Feature, Feature attribute>`. A typical tabular model close to the JDBC standard.
GDMS in action

• simple semantic use case (1):

```sql
select Buffer(the_geom, 6) from roads;
```

before...

... and after
GDMS in action

• simple semantic use case (2):

```sql
select b.* from buildings b, isle_of_nantes i where Contains(i.the_geom, b.the_geom);
```

before... ... and after
GDMS in action

- urban indicators, UrbSAT plugin (1):

```sql
select register(grid);
create table grid as select creategrid(500, 500) from buildings;
select register(gridIntersectWithBuildings);
create table "gridIntersectWithBuildings" as select intersection(g.the_geom,b.the_geom) as the_geom, g.index from grid g, buildings b where isValid(b.the_geom) and intersects(g.the_geom,b.the_geom);
select register(exploded);
create table exploded as select explode() from "gridIntersectWithBuildings";
select register(filterPointsAndLines);
create table "filterPointsAndLines" as select * from exploded where dimension(the_geom) = 2;
select register(unionGridBati);
create table "unionGridBati" as select geomunion(the_geom) as the_geom, index from "filterPointsAndLines" group by index;
select register(batiDensityPerCell);
create table "batiDensityPerCell" as select area(the_geom) as Area, index from "unionGridBati";
select register(density);
create table density as select a.the_geom, a.index, b.Area/area(a.the_geom) as area from grid a, "batiDensityPerCell" b where a.index=b.index;
```
GDMS in action

• urban indicators, UrbSAT plugin (2):

Automatic tool to extract urban tissue's main direction
From vector to raster

```sql
select register('multipoints');
create table multipoints as select ToMultiPoint(the_geom) as the_geom, altitude from contour;
select register('explode');
create table explode as select explode(the_geom) as the_geom from multipoints;
select register('points');
create table points as select adds(the_geom, altitude) as the_geom from explode;
select register('dem');
```
SELECT D8Watersheds(D8Direction(raster)) from dem;

SELECT RasterToPolygons(raster) from watersheds;
Need of a mixed SQL

SELECT CropRaster(raster, the_geom) FROM dem, fence;
Road impact on runoff path

SELECT
D8Accumulation(D8Direction(raster) from dem;

SELECT RasterizeLine(the_geom, raster, 1) from roads, dem;

SELECT
D8ConstrainedAccumulation(d.raster, r.raster) from direction d, rasterized_roads r;
Mixing both raster & vector

```sql
select register('buildings_explode');
create table buildings_explode as select Explode() from buildings;
select register('ground');
create table ground as select GetZDEM('Nantes_dem', the_geom) from buildings_explode;
select register('id:tmp/IsleOfNantesIn3D.cir', 'r3d');
create table r3d as select Extrude(id, height, the_geom) from ground;
```
50,000 triangles (DEM) and 30,000 polygons (buildings)
GeoCognition concept (1)
GeoCognition concept (2)

The tabular data source

Click & drag

Compute a buffer around a geometry. Usage: `Buffer(the_geom, bufferSize, 'butt'='round')`
The SIF framework (API)

- a framework to build UI panels (forms) with an advanced input field validation mechanism,
- all interface's input fields are processed as if they were a single row in a SQL table,
- a SQL expression is built that returns a boolean result – it has to respect the where's clause!

```java
DynamicUIPanel myDynamicUIPanel = UIFactory.getDynamicUIPanel("1st panel", null, new String[] { "field1", "field2" },
                                                                   new int[] { SQLUIPanel.STRING, SQLUIPanel.DOUBLE },
                                                                   new String[] { "field1 LIKE 'abc%'", "field2 > 12" },
                                                                   new String[] { "field1 must start with 'abc'", "field2 must be greater than 12 !" });
UIFactory.showDialog(myDynamicUIPanel);
```
- GDMS SQL syntax extension to provide SQL script re-usability without any modification,
- Provide a mechanism to obtain informations about internal operations so that 2 users interfaces can be used:
  - a wizard that ask for each of the required items the script uses,
  - a graphic modeler (graphic query builder) that lets the graphic connection between script inputs and outputs,
- Define and process schemas to filter tables based on their metadata.
-- Creates a buffer on the specified table
-- with the specified size and shape

DOC('Table containing the geometric field to create the buffer');

DECLARE source:TABLE(the_geom:GEOMETRY);

DOC('Buffer size. It must be positive');

DECLARE size:INT(value > 0);

DOC('Buffer type');

DECLARE buffer_type:STRING VALUES ('CAP', 'BUTT', 'ROUND') DEFAULT 'CAP';

BEGIN;

CREATE TABLE result AS SELECT buffer(the_geom, size, buffer_type) from source;

COMMIT;
Future works

- query and display 3D spatial data sets,
- integrate the temporal dimension,
- deals with multiplexing as outputs,
- from tabular to hierarchical approach…
Why not try it by yourself…

● a FOSS scientific GIS platform developed by a French research institute:
  
  http://orbisgis.cerma.archi.fr/

● … and hosted by a French academic forge:
  
  http://sourcesup.cru.fr/projects/orbisgis