PostgreSQL con PostGIS 2.0 aplicados al mundo de la ingeniería hidráulica y fluvial
PostgreSQL + PostGIS 2.0 aplicados al mundo de la ingeniería hidráulica y fluvial
1. Open source software

The myths of open source

1) Open software is poor quality.

2) Open software is not professional.

3) With open software you can not do everything.
2. Water management. World situation

Aqueduct combines maps with 12 different water risk indicators, including water quality
Source: http://www.wri.org/our-work/project/aqueduct/aqueduct-atlas
2. Water management. World situation

GIS SOFTWARE
DATABASE SOFTWARE
HIDRAULIC SOFTWARE

€?
3. What is Giswater?
3. What is Giswater?
3. What is Giswater?

Smart cities, smart countries, smart world
4. Technical approach

Giswater is a wide software ecosystem with different technologies. The released versions are developed with:

- 'WATER' MODELS: EPANET, EPA SWMM, HEC-RAS
- GIS PROJECT: QGIS
- DATABASE STORAGE: PostgreSQL with spatial extension
4. Technical approach

EPANET CAPABILITIES

- Places no limit on the size of the network that can be analyzed
- Models constant or variable speed pumps, with pumping energy and cost
- Allows storage tanks to have any shape (i.e., diameter can vary with height)
- Considers multiple demand categories at nodes
- Models pressure-dependent flow issuing from emitters (sprinkler heads)

- Models the age of water throughout a network
- Models reactions both in the bulk flow and at the pipe wall
4. Technical approach

EPA SWMM CAPABILITIES

- Networks of unlimited size
- Variety of standard conduit shapes
- External flows and water quality input
- Kinematic wave or full dynamic wave flow
- User-defined dynamic control rules

- Dry Weather flow (DWF)
- Pollutants, (buildup, washoff, routing, reduction)
4. Technical approach

HEC-RAS CAPABILITIES

Computer program that models the hydraulics of water flow through natural rivers and other channels.

Is one-dimensional (by the moment), meaning that there is no direct modeling of the hydraulic effect of cross section shape changes, bends, and other two- and three-dimensional aspects of flow.

Developed by the US Department of Defense, Army Corps of Engineers in order to manage the rivers, harbors, and other public works under their jurisdiction.
4. Technical approach

GIS INTERFACE

GIS O/S project
World wide comunity of users
Better than more proprietary GIS software

DATABASE

PostgreSQL: The World's Most Advanced
Open source database
PostGIS OS4G project
4. Technical approach
4. Technical approach
4. Technical approach

```sql
SELECT
    node1.arc_id,
    node1.node_1,
    node2.node_2
FROM
    SELECT arc.arc_id, node.node_id AS node_1
    FROM epaswmm.arc, epaswmm.node
    WHERE (node.the_geom = st_startpoint (arc.the_geom)) node1,

    SELECT arc.arc_id, node.node_id AS node_2
    FROM epaswmm.arc, epaswmm.node
    WHERE (node.the_geom = st_endpoint (arc.the_geom)) node2

WHERE
    ((node1.arc_id) :: TEXT = (node2.arc_id) :: TEXT);
```
4. Technical approach
4. Technical approach

```sql
SELECT
nextval('epaswmm.inp_vertex_id_seq' :: regclass) AS id,
arc.arc_id,
(st_x(arc.point)) :: NUMERIC (16, 3) AS xcoord,
(st_y(arc.point)) :: NUMERIC (16, 3) AS ycoord
FROM
  SELECT
    st_dumppoints (arc.the_geom)).geom AS point,
    st_startpoint (arc.the_geom) AS startpoint,
    st_endpoint (arc.the_geom) AS endpoint,
    arc.sector_id,
    arc.arc_id
  FROM epaswmm.arc ) arc
JOIN epaswmm.sector_selection ON arc.sector_id = sector_selection.sector_id
WHERE
  arc.point < arc.startpoint OR arc.point > arc.startpoint AND
  arc.point < arc.endpoint OR arc.point > arc.endpoint
ORDER BY id;
```
4. Technical approach

Node Results at 0:00:00 hrs:

<table>
<thead>
<tr>
<th>Node</th>
<th>Elevation (m)</th>
<th>Demand (L/s)</th>
<th>Head (m)</th>
<th>Pressure (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>28.46</td>
<td>0.32</td>
<td>74.07</td>
<td>45.87</td>
</tr>
<tr>
<td>1002</td>
<td>28.17</td>
<td>0.32</td>
<td>74.03</td>
<td>45.86</td>
</tr>
</tbody>
</table>

Link Results at 0:00:00 hrs:

<table>
<thead>
<tr>
<th>Link</th>
<th>Length (m)</th>
<th>Diameter (mm)</th>
<th>Flow (L/s)</th>
<th>Velocity (m/s)</th>
<th>Headloss (m/1000m)</th>
<th>Setting</th>
<th>Reaction</th>
<th>F-Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>2041</td>
<td>63.05</td>
<td>110.00</td>
<td>1.12</td>
<td>0.12</td>
<td>0.16</td>
<td>150.00</td>
<td>0.00</td>
<td>0.025</td>
</tr>
<tr>
<td>2037</td>
<td>63.76</td>
<td>110.00</td>
<td>2.40</td>
<td>0.25</td>
<td>0.66</td>
<td>150.00</td>
<td>0.00</td>
<td>0.022</td>
</tr>
</tbody>
</table>
4. Technical approach

```
SELECT
    node.node_id,
    result_selection.result_id,
    max(rpt_node.demand) AS max_demand,
    min(rpt_node.demand) AS min_demand,
    max(rpt_node.press) AS max_pressure,
    min(rpt_node.press) AS min_pressure,
    node.the_geom
FROM
    epanet.node
JOIN epanet.rpt_node ON rpt_node.node_id = node.node_id
JOIN epanet.result_selection ON rpt_node.result_id = result_selection.result_id
GROUP BY
    node.node_id,
    result_selection.result_id,
    node.the_geom
ORDER BY
    node.node_id;
```
PostgreSQL + PostGIS 2.0 aplicados al mundo de la ingeniería hidráulica y fluvial
Thank you

Source code: https://github/giswater/giswater
website: www.giswater.org