Beyond points: How to turn SMW into a complete Geographic Information System

SMWCon 2013, New-York

Pierre Racine

Contributor to PostGIS

Research Professional

Center for Forest Research

Québec, Canada

cef Centre d'étude de la forêt

UNIVERSITÉ LAVAL

Objectives

- Demonstrate the need to support complex geometries in SMW.
- Emphasis the need to store geometries natively in the DB.
- Propose some geospatial functionalities.
- Discuss how to transpose common GIS concepts into SMW.

Semantic Mediawiki is a Database

- **Meaningful data and types**
- **Queries**
- **More than a database, it is a web data management system**
  - Forms for input and edition.
  - Display and presentation of data (wikitext and template).
  - Possible to develop data driven web applications without having to write low level code (PHP, Ruby, Java, Python, etc…)

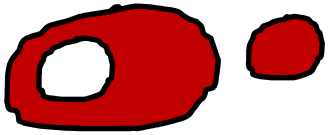
Semantic Maps is a Spatial Database

- **A spatial database**
  - Has special types to store complex geographical entities.
  - Provide functions to manipulate and query spatial entities.
- **Semantic Maps can store points (geographic coordinates)**
  - Human constructions (towns, villages, building, etc…).
  - Geographic features (peaks, falls, islands, etc…).
  - Events (shows, conferences, scientific observations, etc…).
  - Moving objects (trucks, plane, boats, cars).
- **Queries**
  - By any attributes returning a geographic coordinate.
  - By distance to a specific coordinate.
- **Display as maps**
  - Basemaps: Google Map, OpenLayers and OpenStreetMap.
- **Forms**
  - Map allowing to point to generate a geographic coordinate.
- **Perfect to create what we call a Web GIS application!**
- **But the world is not made just of points…**

What’s missing?

1) Types for representing complex geographic entities

- **Professional spatial databases allows storing more complex geometries**
- **LineStrings & MultiLineString**
  - Roads, rivers, transmission lines, plane, boat or human trajectories or directions, contour lines, etc…
- **Polygons & MultiPolygons**



  - Administrative limits (countries, states, agglomerations, postal codes).
  - Ecological delimitations (lakes, land covers, ecologic stratifications, mountain ranges, animal or plant distributions, geological deposit).
  - Footprints (satellite or aerial images, paper maps).
  - Polygons and MultiPolygons may contain holes…
- **A common type: GEOMETRY**
  - Most spatial databases store all those vector shapes as a unique type named GEOMETRY.
  - A typical geographic coverage store all its geometries in one table column.

Advantages of the native GEOMETRY type

- **MySQL, PostgreSQL (with PostGIS) and SQLite (with Spatialite) all have a GEOMETRY type.**
- **There is a spatial extension to SPARQL called GeoSPARQL.**
- **All have a function to convert a text representation (WKT) into GEOMETRY (GeomFromText) and back.**
- **All have special spatial indexes for fast retrieval of geometries based on spatial operators.**
  - e.g. All points inside a boundary or the bbox of the map
- **All have a set of functions to manipulate and query geometries.**
  - Buffer(geom, dist)
  - Within(geom, geom)
  - Distance(geom, geom) (great circle distance on PostGIS!)
  - AsKML(geom) (not in MySQL : (
  - And many, many more… ()

What’s missing?

2) Display of complex geometries and symbology

- **Think big!**
  - In a typical geographic coverage, many geometries may have thousands of vertexes.
  - People CAN’T manually edit textual representation of such geometries.
  - Preferable to NOT display WKT in wikitext, only display an identifier to the DB.
  - What if there are 1 000 000 features to display in a query map?
- **For maps, KML code can easily be generated directly from the DB (AsKML)**
- **Symbology (styling of features)**
  - KML and SLD are standards ways to describe features symbologies
  - Complex styling can be quite verbose. They could be defined as pages in a MapStyles namespace.
  - A form would help defining the most important SLD or KML properties for point, line, polygon and toponym styles.
  - Semantic Maps could provide a libraries of predefined MapStyles pages for points, lines and polygons.

What’s missing?
4) Edition of complex geometries

- **4 ways to create/edit a complex geometry (in order of usability)**
  - Upload a row from a shapefile or a KML (or a whole)
  - Semantic Maps MapEditor (no snapping, no holes)
  - A textarea form input to drop a WKT or a KML string created with another sophisticated (more) map editor
  - Wikitext: NO! Only for points.

What’s missing?

5) Import/export of geospatial data

- Import and export shapefiles, CSV with a WKT column and KML with the Data Transfert extension
- Shapefile ARE simple tables (with a geometry column) so the logic is similar to importing a CSV file.
- One row = one page
- A PHP shapefile reader exist. KML is XML.
- KML also contains symbology info that can be imported to MapStyle pages.

What’s missing?
6) Parser functions & Queries

- **Some useful parser functions**
  - {{#**area**: GEOMETRY}}, {{#**lenght**: GEOMETRY}}
  - {{#**geometrytype**: GEOMETRY}}
  - {{#**envelop**: GEOMETRY}}, {{#**buffer**: GEOMETRY|dist}}

- **Spatial Comparators (Operators)**
  - Show all cities (points) intersecting Germany’s border (multipolygon):

    [[Has location:: **intersects** {{#show: Germany | ?Has border}}]]

What’s missing?

7) Optional support for different coordinate systems

- Google Map shows everything in WGS 84 but OpenLayer can project in other coordinate systems.
- A database of standard coordinate system exist: Each CS has a unique ID called SRID.
- It is important to be able to reproject a geometry in another SRID. Most spatial databases support ST_Transform(geom, srid).
- However there is no support for reprojection in MySQL…
- Support for different coordinate systems can be avoided if we require that all imported data to be mapped together most be in the same coordinate system (SRID) BEFORE they are imported in SMW.

What’s missing?
8) Support for raster

- **Raster data is generally used to represent continuous values like temperature or elevation.**
- **They are however of a quite different nature than points, lines and polygons:**
  - A polygon representing a geographic feature can have a name, an area, a description, and many other properties.
  - A raster however do not have other descriptive properties than the values of the pixels themselves.
- **In a wiki where different Categories of features are represented by many instances (or Pages), there would always be only one instance of a raster coverage type.**
- **The only application that would imply many properties per raster would be a catalog of georeferenced satellite or aerial images.**

Exemples of applications

- **Any application having to describe an object represented by a geographic entity.**
- **A gazetteer**
  - A catalog of geographical feature with their names in different languages, their synonyms, their parent and children features and the geometry representing them.
- **Cadastral survey**
  - A catalog of every property of a city with their owners names, history of values, etc…
- **Catalog of geospatial data**
  - Catalog of geospatial datasets with their name, type of data they store, info about the producer and the distributors, geographic coverage, etc…
- **OpenStreetMap**
  - Every feature stored in OpenStreetMap with revision history, tags, source and the feature itself.
- **An Open Atlas**
- **Every articles having a georeference in Wikipedia.**

A roadmap

1. Make WKT the text representation of the "Geographic coordinate" type.
2. Store the "Geographic coordinate" values as native "geometry" type in the DB.
3. Rename "Geographic coordinate" "Geometry".
4. Add a "Special:ImportSHP" to the Data Transfer extension to import shapefiles as SMW pages.
5. Reuse the Special:ImportSHP code to create a "Load Shape" form input in Semantic Form.
6. Do a "Special:ImportKML" and a form input.
7. Replace complex shape WKT in pages with an ID to the DB shape.
8. Add an "intersects" operator.
9. Define and create predefined MapStyle pages.

Thanks

Geospatial Elucubrations

@geoelucubration

cef
Centre d'étude de la forêt

UNIVERSITÉ LAVAL

Wikidata?

What’s missing?

2) Storage of complex geometries

- **As text in the database (not really a good option)**
  - KML
    - Includes symbology and supported by Google Map, Earth
    - Supports only one coordinate system (WGS 84).
  - WKT (Well Known Text)
    - Does not include symbology, data only.
- **As native spatial database types (better option)**
  - MySQL, Spatialite or PostGIS geometry
  - Advantages:
    - Access to a multitude of manipulation functions
    - Indexation for fast search and spatial operators
    - Examples:
- **As an annotation in a SMW page (minimalist)**
  - Textual representations for complex shapes are way too verbose.
  - Impossible to create/edit by hand.
  - Better simply put an identifier of the real stored native geometry.
  - A new geometry type replacing “Geographic coordinates”.