From HTML to PostGIS
GIS infrastructure legal regulations OGC spatial web services

Michał Okulewicz, Aneta Rosłan

Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Public GIS infrastructure

Legal regulations

- 2007/2/EC INSPIRE Directive
- Technical guidance
- 4th March 2010 statute on **Spatial information infrastructure**,
- 17th May 1989 statute on **Geodesic and cartography law** et al.,
- Minister of Internal Affairs and Administration 20th November 2010 regulation on **Cataloging data sets and services of public spatial information infrastructure** [theoretically repealed]
- Minister of Digitization 8th February 2017 regulation on Council of Spatial information infrastructure

Spatial information infrastructure

- Spatial data sets / Zbiory danych (Art. 7. + Annexes)
- Metadata / Zbiory metadanych (Art. 5.)
- Discovery services / Usługa wyszukiwania (Art 9.) - free of charge (Art 12.)
- View services / Usługa przeglądania (Art 9.) - free of charge (Art 12.)
- Download services / Usługa pobierania (Art 9.) - charges may apply (Art 12.)
- Transformation services / Usługa przekształcania (Art 9.) - charges may apply (Art 12.)
- Services allowing spatial data services to be invoked / Usługa uruchamiania usług (Art 9.) - charges may apply (Art 12.)

Spatial data themes I

Theme	Entity responsible
1. Coordinate reference systems	GUGiK
2. Geographical grid systems	GUGiK
3. Geographical names	GUGiK
4. Administrative units	GUGiK
5. Addresses	GUGiK
6. Cadastral parcels	GUGiK
7. Transport networks	GUGiK
8. Hydrography	Prezes KZGW
9. Protected sites	MŚ i MKiDN

Comment on South Korean Google Maps data

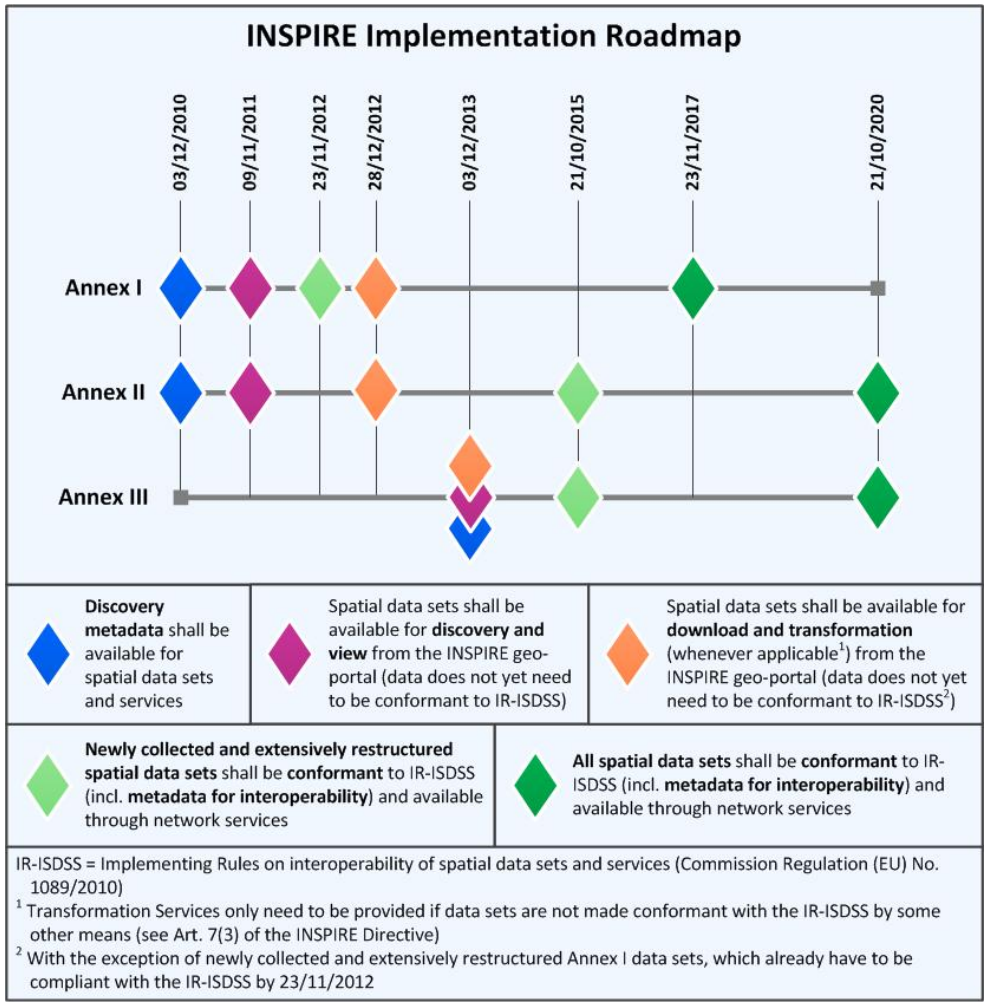
Spatial data themes II

Theme	Entity responsible
1. Elevation	GUGiK
2. Land cover	GUGiK
3. Orthoimagery	GUGiK
4. Geology	Gł. Geolog Kraju

Spatial data themes III

Theme	Entity responsible
1. Statistical units	Prezes GUS
2. Buildings	GUGiK
3. Soil	GUGiK
4. Land use	MI
5. Human health and safety	MZ
6. Utility and governmental services	GUGiK
7. Environmental monitoring facilities	GIOŚ
8. Production and industrial facilities	GUGiK
9. Production and industrial facilities	MR
10. Population distribution — demography	Prezes GUS
11. Area management/restriction/regulation zones and reporting units	GUGiK
12. Natural risk zones	MŚ
13. Atmospheric conditions	MŚ
14 .Meteorological geographical features	MŚ
15. Oceanographic geographical features	MI
16. Sea regions	MI
17. Bio-geographical regions	Gł. Konserwator Przyrod y
18. Habitats and biotopes	Główny Konserwator Przyrody
19. Species distribution	MŚ
20. Energy resources	Gł. Geolog Kraju
21. Mineral resources	Gł. Geolog Kraju

INSPIRE Roadmap



Spatial data sources

- OpenStreetMap - map wikipedia
- GoogleMaps - more than just maps
- INSPIRE Geoportal
- Polish Geoportal
- Polish spatial data web servers
- GraphHopper.com
- GraphHopper API - remember about starting GH

  (HOME/XTrack_SAPD_2017/gh/graphhopper-web-0.10.0-RC1-bin)

Web Map Service

Specification

Web Map Service is responsible for providing raster spatial data (with some exceptions) and text information about the objects in a specified point.

WMS does not provide the ability to search through database using the objects features values.

WMS - Request

WMS utilizes HTTP protocol. Following parameters are sent with each request:

- **service** - service name (WMS)
- version - currently 1.3.0, 1.1.1
- **request** - type of the request

WMS implements the following requests types:

- **GetCapabilities** - metadata of the service and its data
- **GetMap** - maps in a raster form
- GetFeatureInfo - features of objects in a gicen location
- *GetLegendGraphic*, *GetStyles*, *GetPrint*, ...

WMS - GetMap

GetMap takes a following additional parameters:

- **layers** - comma separated list of layers names (eg. layer1,layer2)
- **styles** - comma separated named styles for layers (styles have defined colors, transparencies, line widths etc.) (can be empty)
- **crs/srs** - projection (EPSG codes - eg. EPSG:4326)
- **bbox** - bounding box in a corresponding coordinates system - its lower left corner and upper right corner (eg. for EPSG:4326 - 19.6875,53.7487,20.390625,54.1624)
- **width** - image width (eg. 256)
- **height** - image height (eg. 256)
- **format** - MIME type (eg. image/png)
- transparent - should the image have a transparent background
- ...

Example 1 Example 2

WMS - GetFeatureInfo

GetFeatureInfo takes a following additional parameters:

- **copy of the GetMap request**
- **query_layers** - layers which objects features should be displayed
- **i/y** - vertical image coordinate
- **j/x** - horizontal image coordinate
- **feature_count** - max. number of returned features
- **info_format** - expected MIME type (e.g. application/xml, text/html)
- ...

Example

Web Map Tiled Service

Specification

WMTS is similar to a WMS service. Data is provided in a raster form, but only in predefined tiles instead of custom bounding boxes and sizes.

Sometimes tiled data are also avaialble in a following pattern: http://hosta.adres.domain/zoom/x/y.png

Example

Web Feature Service

Dokumentacja

Web Feature Service provides vector raster data (usually in a form of GML). It allows for both spatial and feature filtering. It also supports a CRUD model of service operation (although usually only Read is available).

WFS - Request

WMS utilizes HTTP protocol. Following parameters are sent with each request:

- **service** - service name (WFS)
- version - currently 2.0.0, 1.1.0, 1.0.0
- **request** - request type

The following WFS request types are available:

- **GetCapabilities** - service and database metadata (Example)
- **DescribeFeatureType** - description of a given data structure (Example)
- **GetFeature** - return all data of a given type (Example)
- *GetGmlObject*, *Transaction*, *LockFeature*, ...

WFS - GetFeature + filter

```
<wfs:GetFeature
 xmlns:wfs="http://www.opengis.net/wfs"
 service="WFS" version="1.0.0"
 xsi:schemaLocation="http://www.opengis.net/wfs
                     http://schemas.opengis.net/wfs/1.0.0/WFS-transaction.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <wfs:Query typeName="feature:test" xmlns:feature="http://www.qgis.org/gml/">
    <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:BBOX><ogc:PropertyName>geometry</ogc:PropertyName>
        <gml:Box xmlns:gml="http://www.opengis.net/gml" srsName="EPSG:4326">
        <gml:coordinates decimal="." cs="," ts=" ">
        18.010204007256877 65,52.23902294775036381
        20.99333728592148 19,53.13053438171262144
        </gml:coordinates>
        </gml:Box>
      </ogc:BBOX>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

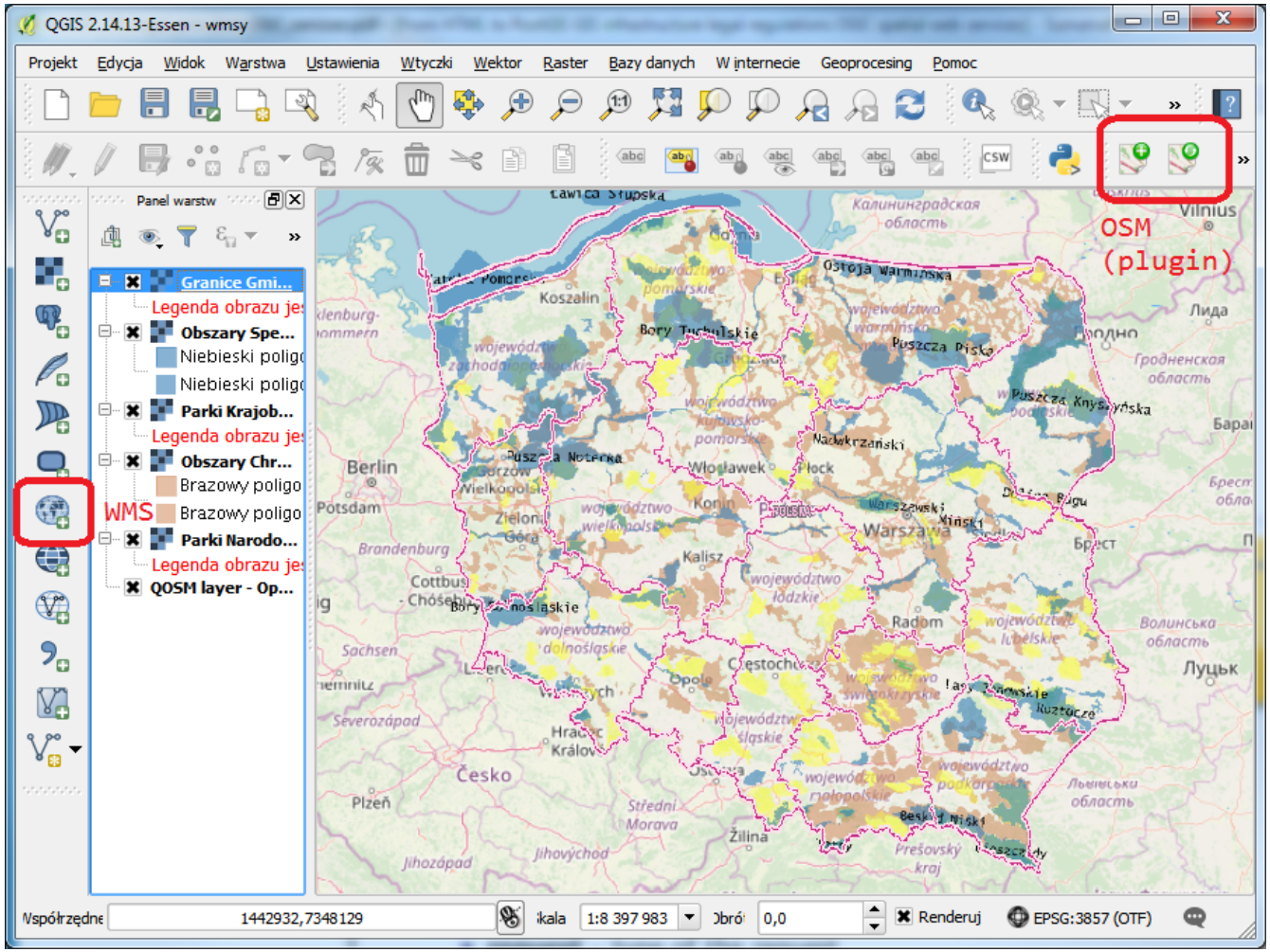
Other services

- Catalogue Service
- GeoNames - free of charge geocoding
- Nominatim - free of charge geocoding based on OpenStreetMap data
- GraphHopper - free of charge routing / charges apply for advanced features / OpenStreetMap
- Google Gecoding - 1000 - 2 500 / day (user)
- Yahoo Geocoding - 50 000 / day (IP)
- ...

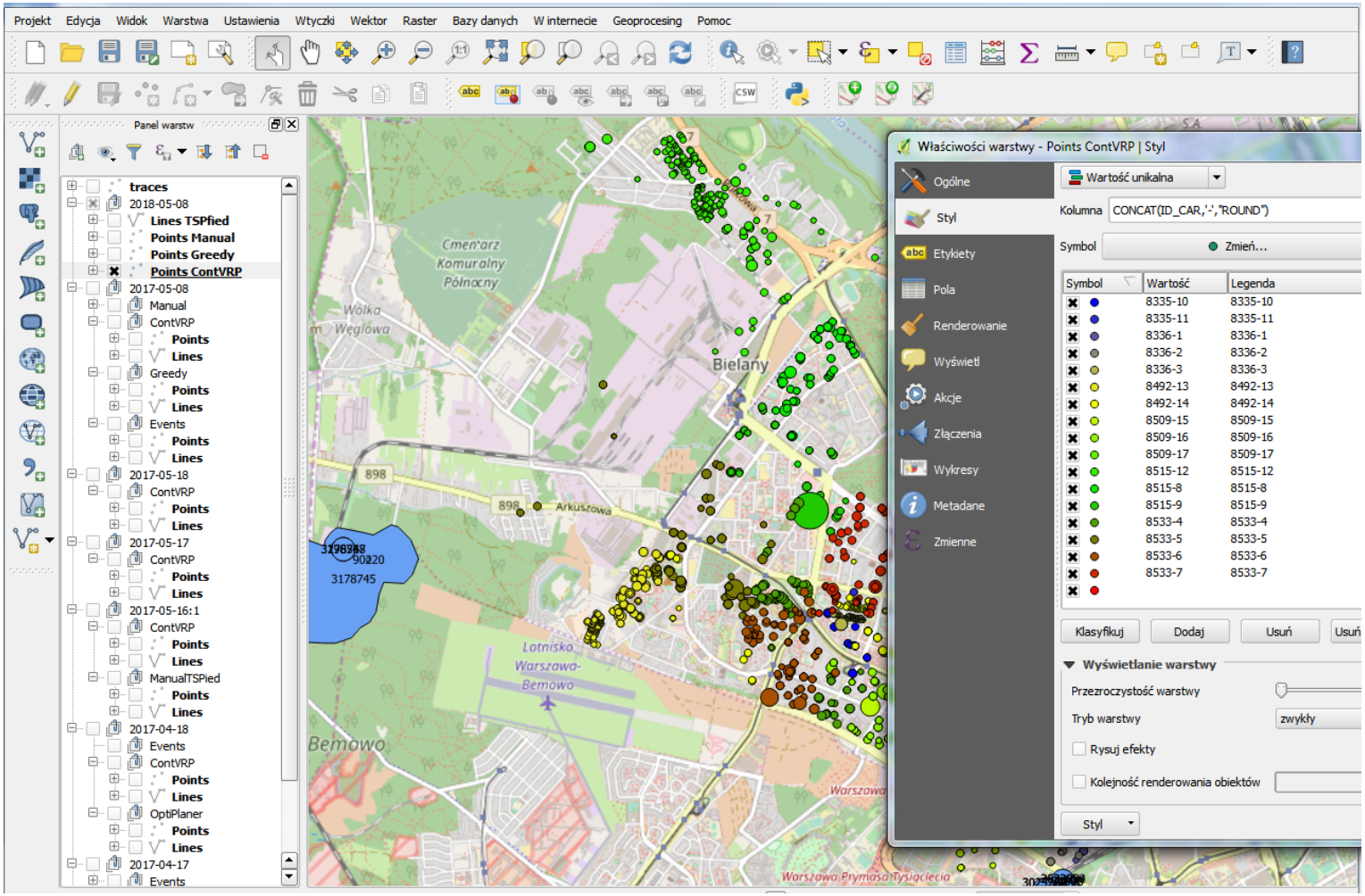
Spatial data and services handling tutorial

- QGIS with PostGIS data, WMS services and OSM plugin
- Query example to get Nominatim data
- Open Street Map object details
- Download and run GraphHopper
- Query example to get GH route
- Parsing GH response in order to get distance matrix

QGIS: WMS and OSM



QGIS: PostGIS data stylization



OSM: checking object properties

Lets consider results for *Emilii Plater St.*:

- https://www.openstreetmap.org/way/11917798
  - two way part
- https://www.openstreetmap.org/api/0.6/way/11917798
  - two way part XML details
- https://www.openstreetmap.org/way/184674429
  - near central railway station
- https://www.openstreetmap.org/way/466325491
  - near our University

Nominatim: searching for objects

https://nominatim.openstreetmap.org/search?:

- format
- limit
- polygon_geojson
- q OR street, city, county, state, country, postalcode

Examples:

- https://nominatim.openstreetmap.org/search?format=json&street=Emilii%20Plater&City=Warsaw&limit=100
- https://nominatim.openstreetmap.org/search?format=json&street=Emilii%20Plater&City=Warsaw&limit=1&polygon_geojson=1

source: https://wiki.openstreetmap.org/wiki/Nominatim

GraphHopper: running your own service instance

- https://github.com/graphhopper
- https://oss.sonatype.org/content/groups/public/com/graphhopper/graphhopper-web/0.10.0-RC1/
- Change configuration filename to `config.properties`
- http://download.geofabrik.de/europe/poland/mazowieckie.html
- `java -jar *.jar jetty.resourcebase=webapp config=config.properties datareader.file=europe_poland_mazowieckie.osm.pbf`
- http://127.0.0.1:8989/route?point=52.2688%2C21.04005&point=52.30123%2C21.03277&locale=pl-PL&instructions=false&vehicle=car&weighting=fastest&elevation=false&points_encoded=false&use_miles=false&layer=Omniscale

Querying GraphHopper

```
public async Task<Tuple<double, int>> GetDistance(Location from, Location to, bool withCoerce = true)
{
    var builder = new UriBuilder(_webServiceAddress);
    var parameters = new List<(string, string)>();
    parameters.Add(("point", from.ToString()));
    parameters.Add(("point", to.ToString()));
    parameters.Add(("locale", "pl-PL"));
    parameters.Add(("instructions", "true"));
    parameters.Add(("vehicle", "car"));
    parameters.Add(("weighting", "fastest"));
    parameters.Add(("elevation", "false"));
    parameters.Add(("points_encoded", "false"));
    parameters.Add(("use_miles", "false"));
    parameters.Add(("layer", "Omniscale"));
    builder.Query = GetQueryString(parameters);
    var response = await _client.GetAsync(builder.Uri);
    if (!response.IsSuccessStatusCode)
        throw new HttpException((int)response.StatusCode, response.ReasonPhrase);
    var resp = JsonConvert.DeserializeObject<ResponseModel>(await response.Content.ReadAsStringAsync());
    var dist = withCoerce ? CoerceDistance(from, resp, to) : resp.Paths[0].Distance;
    return new Tuple<double, int>(dist, resp.Paths[0].Instructions.Length-2);
}
```

source: https://github.com/bakala12/VrpTestCasesGenerator