

# 1 Semantic GIS, GeoSPARQL

## How to link data without links

- Which bus stops are within 5 minutes walk from home?
- In which municipal district is the highest area of park areas?
- What historical monuments are visible from my hotel room?

## Spatial operations

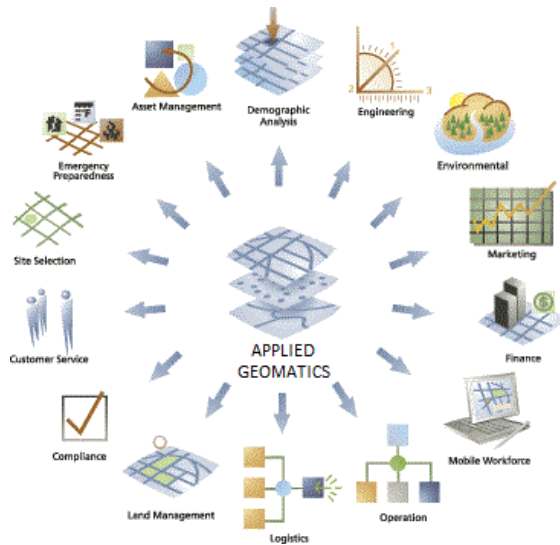


## 1.1 WTF is GIS??

### Geomatics

#### Geomatics – ISO/TC 211

Discipline concerned with collection, distribution, storage, analysis, processing, presentation of geographic data or geographic information.



## GIS

Term Geomatics is often used in the meaning of Geographic Information System (GIS) and vice versa.

## Geographic Information System

System designed to capture, store, manipulate, analyze, manage and present spatial or geographic data. GIS applications are tools that allow users to create interactive queries (user-created searches), analyze spatial information, edit data in maps, and present the results of all these operations.

## Geographic Data

Also called:

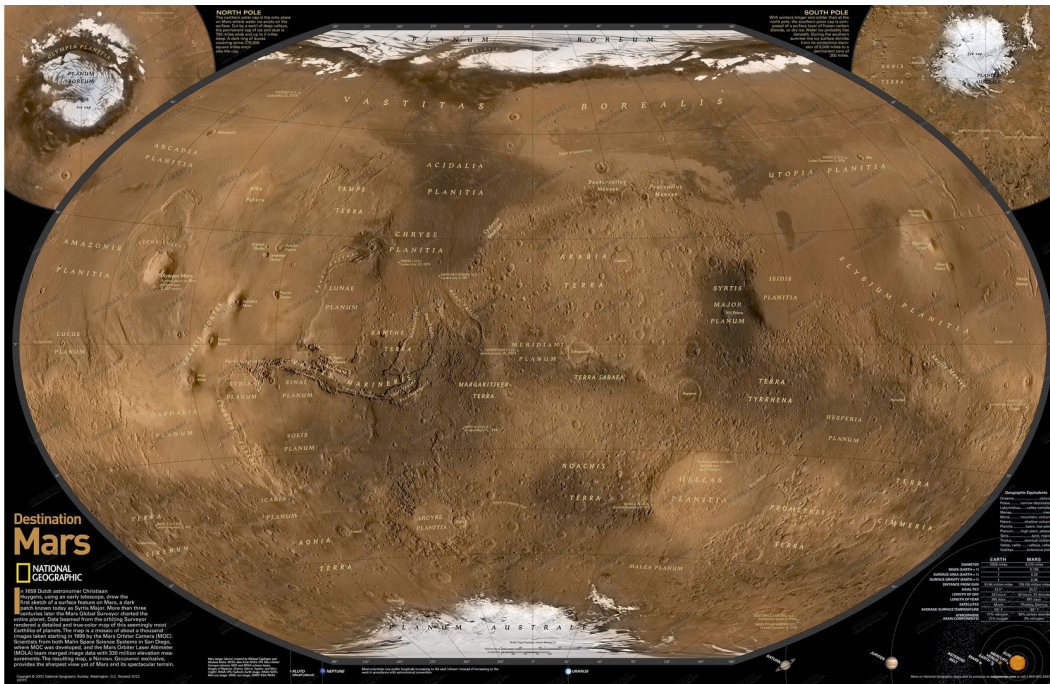
- Geospatial data and information,
- Georeferenced data and information,
- Geodata,
- Geoinformation,
- Spatial data.

## Geographic Data – ISO/TC 211

Data and information having an implicit or explicit association with a location relative to the Earth.

## Geographic Data

... or in wider context any other space object. (*Martian spatial data*)



## Geographic Data

### Geographic Data – ISO/TC 211

Data and information having an **implicit** or **explicit** association with a location relative to the Earth.

**Implicit** – coordinates, direction and distance,

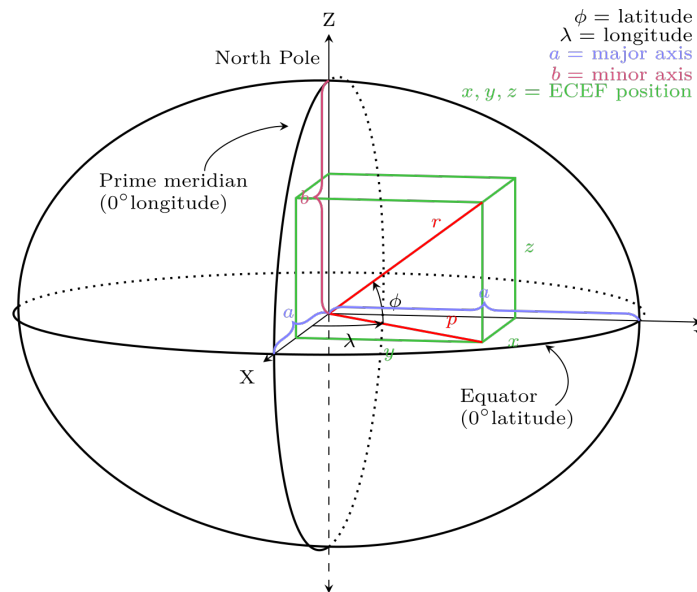
**explicit** – geographic name, address.

Depends on semantics.

## Coordinates

### Geographic Coordinate System

System that enables every location on Earth by set of numbers, letters and symbols.



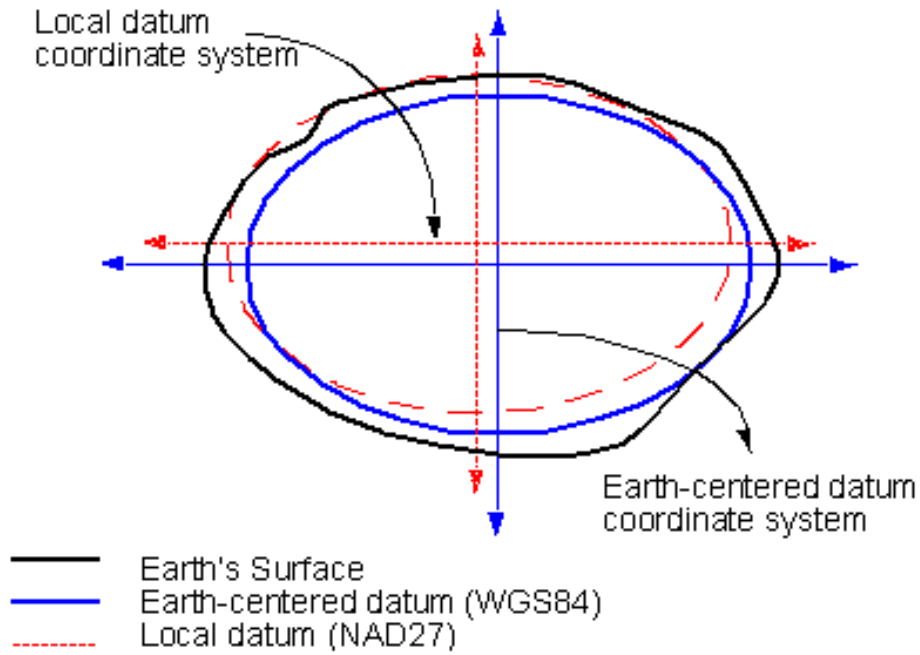
## Coordinate Reference Systems

**Reference ellipsoid** – approximation of geoid,

**geodetic datum** – mapping of spherical coordinates onto ellipsoid,

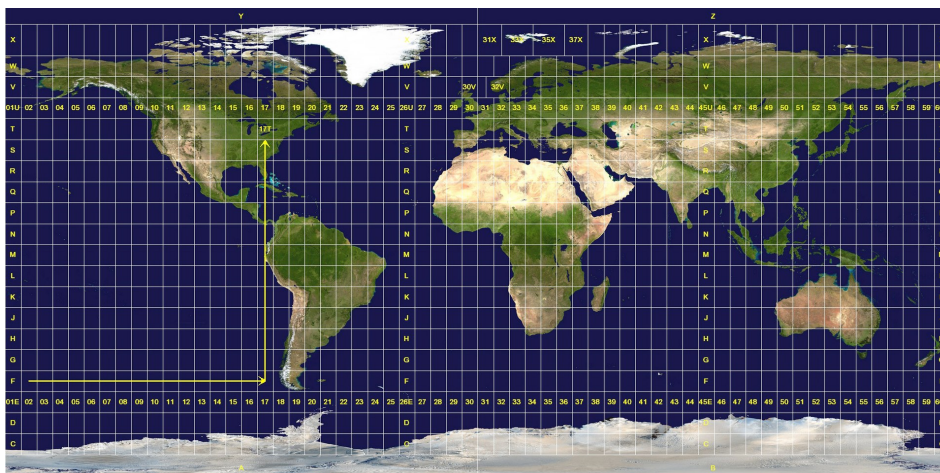
**map projection** – conversion of geodetic coordinates to the plain map.

## Reference ellipsoid and datum



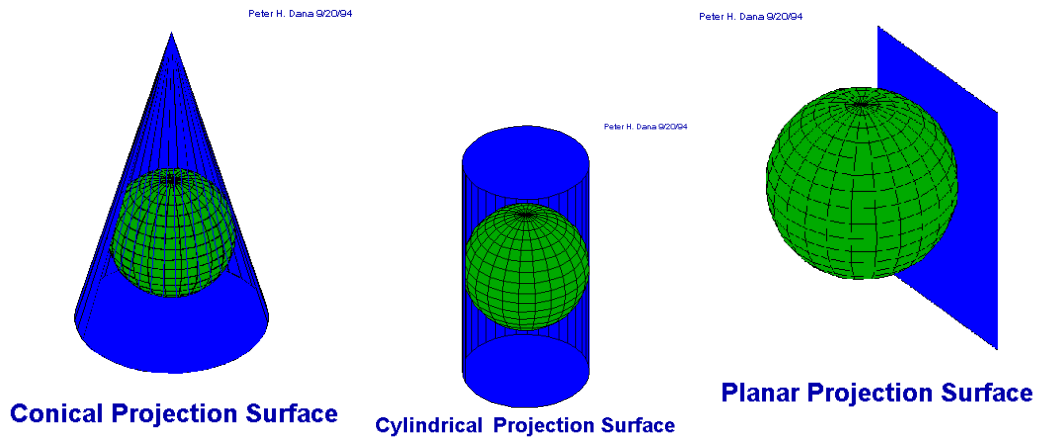
### Projection

Best known coordinate system is WGS-84, also known as **GPS coordinates** in a form of latitude and longitude.



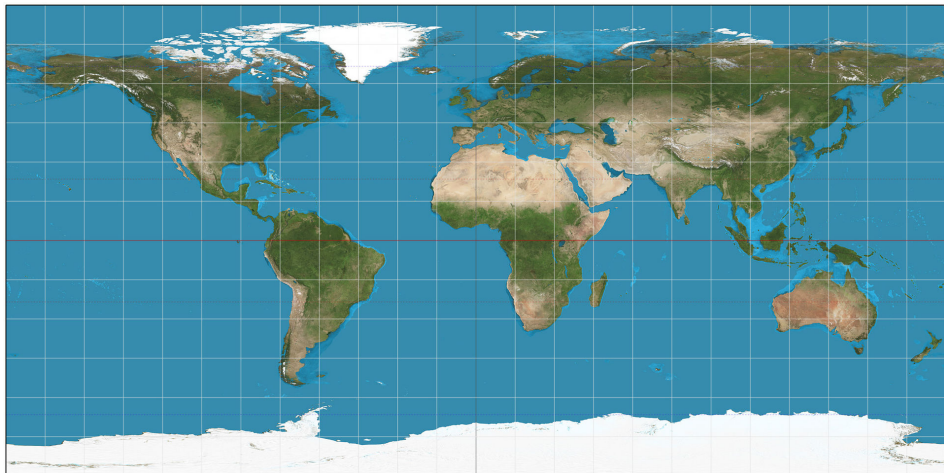
### Projection

## 1 Semantic GIS, GeoSPARQL



### Why is CRS important?

Same coordinate system may be projected differently.



### EPSG codes

## EPSG:4326

Geodetic coordinate system



WGS 84 -- WGS84 - World Geodetic System 1984, used in GPS

Transform coordinates

Get position on a map

### Attributes

**Unit:** degree (supplier to define representation)

**Geodetic CRS:** WGS 84

**Datum:** World Geodetic System 1984

**Ellipsoid:** WGS 84

**Prime meridian:** Greenwich

**Data source:** OGP

**Information source:** EPSG. See 3D CRS for original information source.

**Revision date:** 2007-08-27

**Scope:** Horizontal component of 3D system. Used by the GPS satellite navigation system and for NATO military geodetic surveying.

**Area of use:** World.

**Coordinate system:** Ellipsoidal 2D CS. Axes: latitude, longitude. Orientations: north, east. UoM: degree

### Covered area



### Center coordinates

0.00000000 0.00000000

### WGS84 bounds:

-180.0 -90.0

180.0 90.0

World.

<http://epsg.io>

## Coordinate Reference Systems

### WGS-84

World Geodetic System 1984, EPSG code 4326, used in GPS and Google maps.

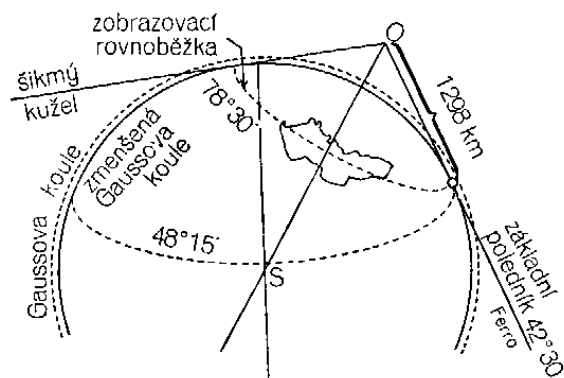
### ETRS-89

European Terrestrial Reference System 1989, EPSG code 4258, based on GRS 1980 ellipsoid, made to be precise in Europe.

### S-JTSK

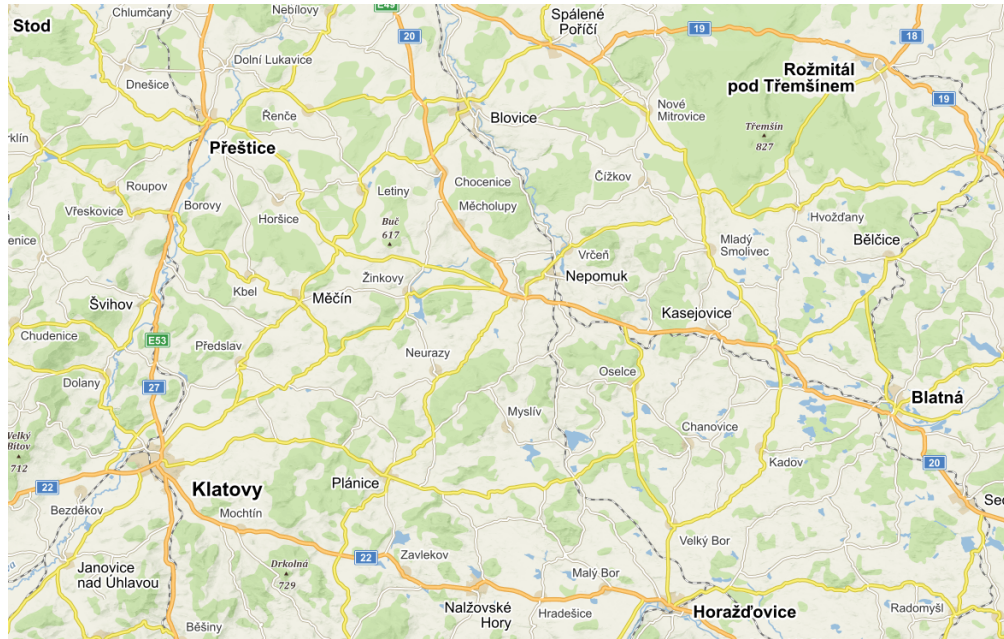
Systém jednotné trigonometrické sítě katastru, EPSG code 5514, cone based system with double transformation made for the Czechoslovakian area, using Křovák projection.

### S-JTSK

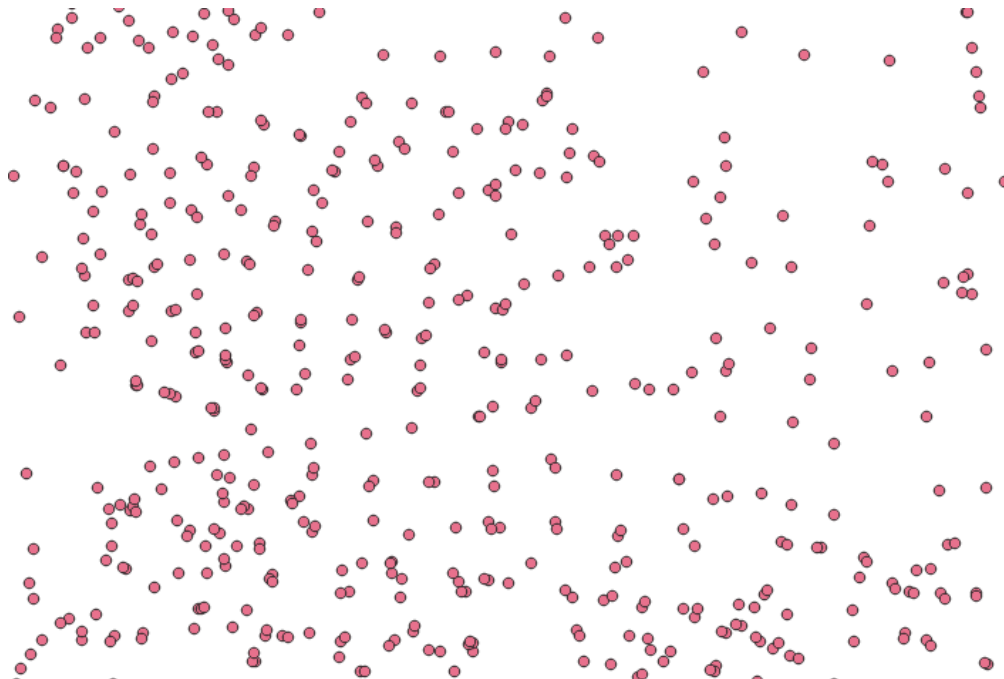


## 1.2 Usage of GIS

### Geodata representation – raster



### Geodata representation – vector



**Geodata representation**

id	x	y
102151	14.251	49.321
102152	14.632	48.956

**Geodata representation****Geodata representation****Why?**

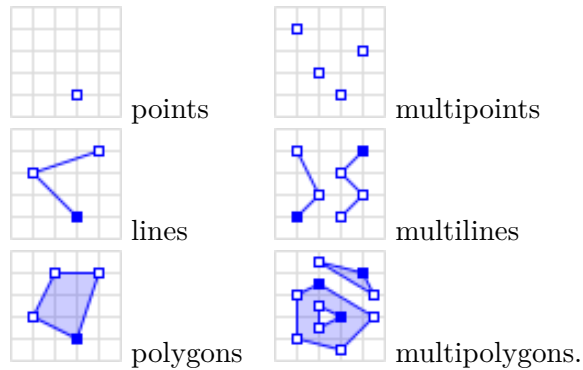
Missing coordinate reference system,  
 expresses only points,  
 every single point has single id,  
 does not support any spatial operations.

**Certification and standards**

<http://www.opengeospatial.org/>

## Spatial Objects

Geographic features are handled as objects:



### Spatial objects representation Well-Known Text (WKT)

```
POINT(50.056 14.434)
LINESTRING(50.056 14.434, 50.064 14.442, 50.042 14.445)
```

### Geography Mark-up Language (GML)

```
<gml:Point srsName="http://opengis.net/def/crs/EPSSG/0/4326" srsDimension="2"/>
  <gml:pos>50.056 14.434</gml:pos>
</gml:Point>

<gml:Curve srsName="http://opengis.net/def/crs/EPSSG/0/5514" srsDimension="2">
  <gml:segments>
    <gml:LineStringSegment>
      <gml:posList>-641126.76 -1093821.18 -641119.35 -1093831.05
        -641109.75 -1093844.44</gml:posList>
    </gml:LineStringSegment>
  </gml:segments>
</gml:Curve>
```

## Spatial data formats

### Geography Mark-up Language

- OGC standard and XML based format,
- geometry described as GML objects,
- complex format allowing any type of geometry object,
- allows any type of attributes,
- described by xml schema (XSD).

### Spatial data formats – GML

```
<ad:Address gml:id="AD.22547665">
  <ad:inspireId>
    <base:Identifier>
      <base:localId>AD.22547665</base:localId>
      <base:namespace>CZ-00025712-CUZK_AD</base:namespace>
    </base:Identifier>
  </ad:inspireId>
</ad:Address>
```

```

</ad:inspireId>
<ad:alternativeIdentifier>K Pitkovicům 1, Benice, 10300 Praha 10</ad:alternativeIdentifier>
<ad:position>
  <ad:GeographicPosition>
    <ad:geometry>
      <gml:Point gml:id="P.AD.22547665" srsName="urn:ogc:def:crs:EPSG::5514"
        srsDimension="2">
        <gml:pos>-731037.56 -1053052.98</gml:pos>
      </gml:Point>
    </ad:geometry>
    <ad:specification
      xlink:href="http://inspire.ec.europa.eu/codelist/
        GeometrySpecificationValue/entrance"
      xlink:title="entrance"/>
    <ad:default>true</ad:default>
  </ad:GeographicPosition>
</ad:position>
<ad:component xlink:href="#AA.MOP.108" xlink:title="Praha 10"/>
<ad:component xlink:href="#AA.MOMC.538078" xlink:title="Praha-Benice"/>
<ad:component xlink:href="#AA.2585" xlink:title="Benice"/>
<ad:component xlink:href="#TF.498211" xlink:title="K Pitkovicům"/>
<ad:component xlink:href="#PD.10300" xlink:title="10300"/>
</ad:Address>

```

## Spatial data formats

### GeoJSON

OGC standard and JSON based format,

records as single objects,

not so robust, but simpler,

CRS as object,

exchange format on the internet,

extension – GeoJSON-LD.

## Spatial data formats – GeoJSON

```

{
  "geometry": {
    "coordinates": [
      14.419134,
      50.090122
    ],
    "type": "Point"
  },
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:EPSG::4326"
    }
  },
  "properties": {
    "cislo_orientacni": "22",
    "cislo_popisne": "128",
    "druh_mista": "RESTAURAČNÍZÁHRÁDKY",
    "druh_zbozi": "",
    "mome": "Praha 1",
    "ulice": "Pařížská"
  },
  "type": "Feature"
}

```

## Spatial data formats

### GeoPackage

OGC standard and SQLite based format,  
allows both vector and raster data (embedded image file),  
complex spatial structure may be inserted as attribute,  
basically database file.

## Spatial data formats

### Comma Separated Values

table way of description,  
rows of text file separated by UTF-8 character U+200C (,),  
geometry object contains commas, therefore it has to be closed in quotation marks,  
geometry recommended to be described as WKT,  
not really recommended.

## Spatial data formats – CSV

```
{
  Kód,Název ulice,Kód Obce,Název Obce,Kód Okresu,Název Okresu,WKT_Geometry,CRS
442666,Adamovská,554782,Praha,3100,Hlavní město Praha,"LINESTRING(14.45032 50.05789,
14.45094 50.05791, 14.45121 50.058034, ...)",http://www.opengis.net/def/crs/EPSSG/0/4258
442674,Africká,554782,Praha,3100,Hlavní město Praha,"LINESTRING(14.34946 50.09616,
14.34978 50.09607, 14.34996 50.09575, ...)",http://www.opengis.net/def/crs/EPSSG/0/4258
442682,Akátová,554782,Praha,3100,Hlavní město Praha,"LINESTRING(14.41953 50.07761,
14.41979 50.07764, 14.42133 50.07749, ...)",http://www.opengis.net/def/crs/EPSSG/0/4258
}
```

## 1.3 Linked Geo Data

### How is it usually done?

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
  <geo:Point>
    <geo:lat>49.701</geo:lat>
    <geo:long>14.552</geo:long>
  </geo:Point>
</rdf:RDF>
```

**How is it usually done?**



**What do we need to do it right?**

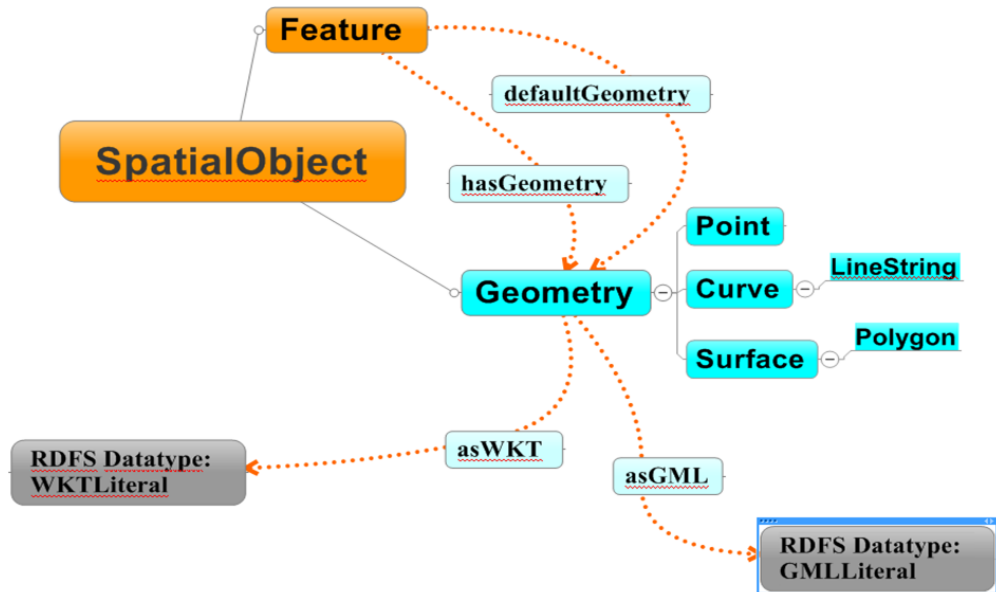
- coordinate system support,
- spatial objects support,
- spatial operations support,
- ontology describing relations between objects.

**GeoSPARQL**

**Why is it so great?**

- it is an ontology,
- it is a query language supporting spatial operations,
- it supports spatial objects in WKT and GML.

**GeoSPARQL ontology**



## GeoSPARQL representation

```

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geosparql: <http://www.opengis.net/ont/geosparql#> .
@prefix ds-par: <http://onto.fel.cvut.cz/ontologies/town-plan/parcely/> .
@prefix databaseTableParcely: <http://onto.fel.cvut.cz/ontologies/town-plan/databaseTableParcely/> .
@prefix par-geometry: <http://onto.fel.cvut.cz/ontologies/town-plan/parcelakn_dokm_p/geometry/>
@prefix townplan: <http://onto.fel.cvut.cz/ontologies/town-plan/>

townplan:parcelakn_dokm_p/1/2018-01-29T14:36:24.178617 a ds-par:Parcely,
    geosparql:Feature ;
    rdfs:label "parcelakn_dokm_p/1/2018-01-29T14:36:24.178617" ;
    databaseTableParcely:dat_vznik "2008-09-25"^^xsd:date ;
    databaseTableParcely:existujedi "A" ;
    databaseTableParcely:id 2087553101.0 ;
    databaseTableParcely:id_poskyt 397 ;
    databaseTableParcely:katuze_kod 727164 ;
    databaseTableParcely:nazev_ku "Vinohrady" ;
    databaseTableParcely:ogc_fid 1 ;
    databaseTableParcely:par_id 2087553101.0 ;
    databaseTableParcely:parcela "1057" ;
    databaseTableParcely:shape_area 260.475900002 ;
    databaseTableParcely:shape_length 65.6304823872 ;
    databaseTableParcely:tid_parcelakn_dokm_p 61534.0 ;
    databaseTableParcely:vymera 260 ;
    geosparql:hasGeometry par-geometry:1/2018-01-29T14:36:24.178617 .

par-geometry:1/2018-01-29T14:36:24.178617 a geosparql:Geometry ;
    rdfs:label "parcelakn_dokm_p/geometry/1/2018-01-29T14:36:24.178617" ;
    geosparql:asWKT "MULTIPOLYGON((( -742241.02 -1045480.81, -742242.84 -1045482.35,
        -742257.059 -1045469.76, -742246.0798 -1045456.9, -742237.98
        -1045465.82, -742241.02 -1045480.81)))" .
    
```

## GeoSPARQL querying

```

geo: <http://www.opengis.net/ont/geosparql#>
geof: <http://www.opengis.net/def/function/geosparql/>
    
```

## GeoSPARQL querying

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX vocab-vyuziti: <http://onto.fel.cvut.cz/ontologies/town-plan/
resource/vocab/urk_ss_vyuzitizakl_p/>
PREFIX vocab-fvu: <http://onto.fel.cvut.cz/ontologies/town-plan/
resource/vocab/pvp_fvu_p/>

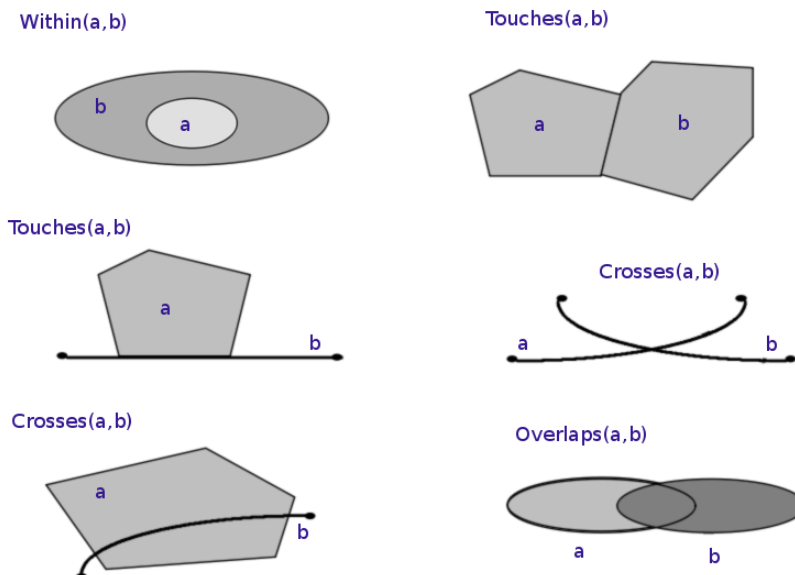
SELECT ?var1 ?var2
WHERE {
  ?var1 vocab-fvu:wkb_geometry ?geometry1.
  ?var2 vocab-vyuziti:wkb_geometry ?geometry2.
  ?geometry1 geof:intersects ?geometry2
}

```

## GeoSPARQL spatial relations

### ogc:equals

Two objects have the same geometry.



## GeoSPARQL filter functions

### ogc:relate

Returns true if two objects are spatial related.

### ogcf:relate

```

(geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
relation: xsd:anyURI): xsd:boolean

```

### ogc:distance

Returns distance in given units between two objects.

**ogcf:distance**

(geom1: **ogc:GeomLiteral**, geom2: **ogc:GeomLiteral**,  
units: **xsd:anyURI**): **xsd:double**

**GeoSPARQL filter functions**

**ogcf:buffer**

Returns geometric object representing all points whose distance from geom1 is within radius in units.

**ogcf:buffer**

(geom: **ogc:GeomLiteral**, radius: **xsd:real**,  
units: **xsd:anyURI**): **ogc:GeomLiteral**

**ogcf:convexHull**

Returns geometric object representing all points in convex hull of geom1.

**ogcf:convexHull**

(geom1: **ogc:GeomLiteral**): **ogc:GeomLiteral**

**GeoSPARQL filter functions**

**ogcf:intersection**

Returns a geometric object that represents all Points in the intersection of geom1 with geom2.

**ogcf:intersection**

(geom1: **ogc:GeomLiteral**, geom2: **ogc:GeomLiteral**,  
) : **ogc:GeomLiteral**

**ogcf:union**

Returns a geometric object that represents all Points in the union of geom1 with geom2.

**ogcf:union**

(geom1: **ogc:GeomLiteral**, geom2: **ogc:GeomLiteral**,  
) : **ogc:GeomLiteral**

**GeoSPARQL filter functions**

ogcf:difference,

ogcf:symDifference,

ogcf:envelope,

ogcf:boundary.

GeoSPARQL documentation

**GeoSPARQL filter example**

```

PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof:
  <http://www.opengis.net/def/function/geosparql/>

SELECT ?f
WHERE {
  ?f my:hasPointGeometry ?fGeom .
  ?fGeom geo:asWKT ?fWKT .
  FILTER (geof:sfWithin(?fWKT, '''
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Polygon ((-83.4 34.0, -83.1 34.0,
              -83.1 34.2, -83.4 34.2,
              -83.4 34.0))
    '''^geo:wktLiteral))
}

```

**GeoSPARQL query example**

```

PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX ogcf:
  <http://www.opengis.net/def/function/geosparql/>

SELECT ?f ?fWKT
WHERE {
  my:A my:hasExactGeometry ?aGeom .
  ?aGeom ogc:asWKT ?aWKT .
  ?f my:hasExactGeometry ?fGeom .
  ?fGeom ogc:asWKT ?fWKT .
  ?aGeom ogc:sfContains ?fGeom.
}
}

```