GeoSPARQL: an introduction

C. Métral

Semantic Web Technologies

Handling Spatial Data

More and more **spatial** data

=> Need **spatial reasoning**

Examples

- What are the monuments in parks of Geneva?
- What are the universities within 20 km?
- What are the commercial land parcels that touch some arterial streets?

=> Need **geospatial concepts and properties** if the relationships (between monuments and parks or between parcels and streets) are not explicit

Handling Spatial Data with semantics

Examples

- What are the monuments in parks of Geneva that have been made by Paul Landowski?
- What are the universities within 20 km that propose courses on RDF?

=> Need RDF(S) + geospatial concepts and properties for representing such resources

Simple Features Model

Definition

- It is a specification of the Open Geospatial Consortium, which defines a general architecture for geographic data and for their geometries
- More precisely
  - It describes a way to represent geospatial data using a hierarchy of classes
  - It defines functions for accessing, operating and constructing these data

Features and Geometries

Feature
- Any entity in the real world with some spatial location
- Examples: a university, a parcel, a street…

Geometry
- Any geometric shape, such as a point, a line, a polygon, that is used as a representation of a feature's spatial location
- Examples: a point for a university, a polygon for a parcel…

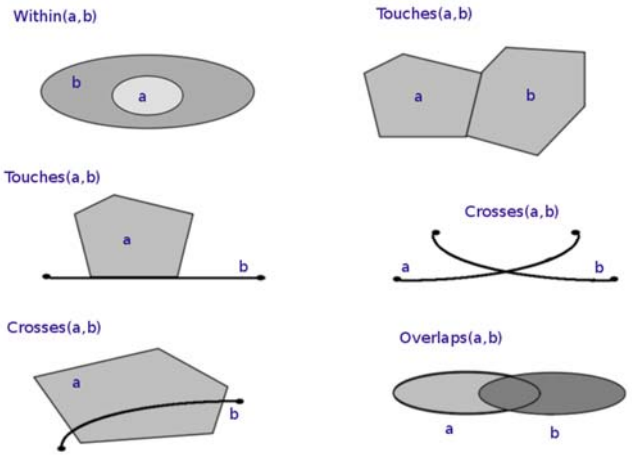
Coordinate Reference System/Spatial Reference System
- Part of the metadata associated with a geometry
- Examples: (X,Y,Z) coordinates, longitude+latitude…

Topological relationships

Each spatial entity is inherently related to some other spatial entities -> 8 geospatial topological relations to describe relationships between entities in space

equals
disjoint
intersects
touches
within
contains
overlaps
crosses

Examples of topological spatial relations



From http://en.wikipedia.org/wiki/Spatial_relation

Handling and Querying Spatial Data

Examples
- What are the universities within 20 km that propose courses on RDF?
- What are the monuments in parks of Geneva that have been made by Paul Landowski?

=> Need of **spatial reasoning with semantic reasoning**

=> Need geospatial concepts and properties for representing such resources + **geospatial queries**

GeoSPARQL

Definition

A **spatial extension to** the **SPARQL** query language for geospatial information

Built on existing standards

- World Wide Web Consortium W3C
  Semantic Web: RDF, RDFS, SPARQL…
- Open Geospatial Consortium OGC
  Simple Features model, geometry models…
- ISO/TC 211
- …

GeoSPARQL

Provides the following features

- An RDF/OWL vocabulary for representing spatial information consistent with the Simple Features Model
- A set of SPARQL extension functions for spatial computations
- A set of RIF (Rule Interchange Format) rules for query transformation

From [1]

SPARQL query

What are the universities that propose courses on RDF?

```
…
@prefix cui: <http://cui.unige.ch/> .

SELECT ?i
WHERE {
    ?i a cui:University .
    ?i cui:course ?c .
    ?c cui:keyword ?k .
    FILTER ( ?k = "RDF" )
}
```

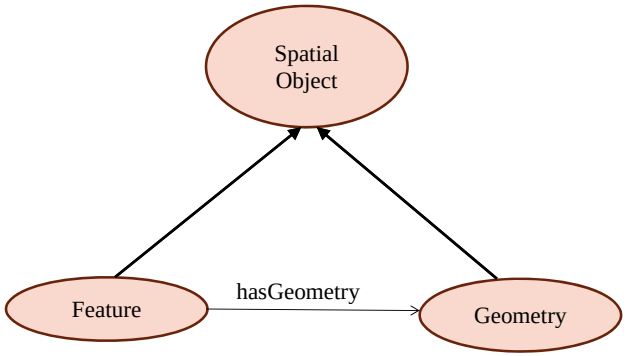

GeoSPARQL query

What are the universities **within 20 km** (from a reference point with lat=46.202 and lon=6.146) that propose courses on RDF?

```
…
SELECT ?i
WHERE {
    ?i a cui:University .
    ?i spatial:nearby(46.202 6.146 20 'km') .
    ?i cui:course ?c .
    ?c cui:keyword ?k .
    FILTER ( ?k = "RDF" )
}
```

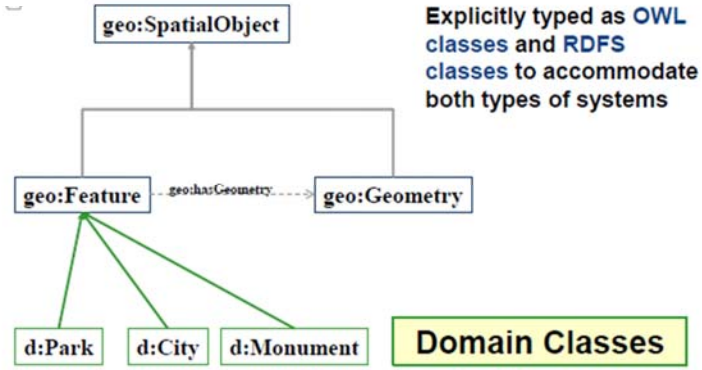
Spatial RDFS

Main geospatial classes and property



From [2]

Geospatial and domain classes



From [3]

Main geospatial classes and property

geo:Feature class
A thing with a spatial location (a city, a monument…)

geo:Geometry class
A representation of a spatial location (a set of coordinates)

geo:SpatialObject class
A superclass of both Feature and Geometry classes

geo:hasGeometry property
To link a feature to its geometry that represents its spatial extent
A given feature may have many associated geometries for varying purposes
Example: a city can be represented either by a point or a polygon according to the scale or to the size of the city

See http://www.opengis.net/ont/geosparql

Definition of *geo:SpatialObject* class

geo:SpatialObject a rdfs:Class,
owl:Class;

rdfs:label "Spatial Object"@en;

rdfs:comment "The class Spatial Object represents everything that can have a spatial representation. It is superclass of feature and geometry"@en .

From [1]

Definition of *geo:Feature* class

geo:Feature a rdfs:Class,
owl:Class;
rdfs:label "Feature"@en;
rdfs:subClassOf geo:SpatialObject;
owl:disjointWith geo:Geometry;
rdfs:comment "This class represents the top-level feature type. This class is equivalent to GFI_Feature defined in ISO 19156, and it is superclass of all feature types"@en .

From [1]

Definition of *geo:Geometry* class

geo:Geometry a rdfs:Class,
owl:Class;
rdfs:label "Geometry"@en;
rdfs:subClassOf geo:SpatialObject;
owl:disjointWith geo:Feature;
rdfs:comment "The class represents the top-level geometry type. This class is equivalent to the UML class GM_Object defined in ISO 19107, and it is superclass of all geometry types"@en .

From [1]

Definition of *geo:hasGeometry* property

geo:hasGeometry a rdf:Property,
owl:ObjectProperty;
rdfs:label "has Geometry"@en;
rdfs:comment "A spatial representation for a given feature."@en;
rdfs:domain geo:Feature;
rdfs:range geo:Geometry .

From [1]

Main geometry representations

WKT (Well-known text)
Text markup language for representing vector geometry objects
Defined by the Open Geospatial Consortium (OGC)

Example: the point with 2D coordinates (x=10, y=23) defined by the string "*POINT (10,23)*"

GML (Geography Markup Language)
XML-based language
Defined by the OGC

Example: definition of the point (10,23)

```
<gml:Point srsDimension="2" >
    <gml:pos>10 23</gml:pos>
</gml:Point>
```

Main geometry types (WKT)

TYPE	SHAPE	Geometry Class	SYNTAX
POINT	•	`sf:Point`	`POINT(longitude latitude)`
LINESTRING		`sf:LineString`	`LINESTRING(long1 lat1, long2 lat2, …)`
POLYGON		`sf:Polygon`	`POLYGON((long1 lat1, long2 lat2, … , long1 lat1))`
POLYGON (WITH HOLE)		`sf:Polygon`	`POLYGON((long1 lat1, long2 lat2, … , long1 lat1), (longA latA, longB latB, …, longA latA))`

To create a WKT geometry, a resource should be declared to be the appropriate type from the table above, and given an **asWKT** property with a literal of the appropriate form.

From [2]
See also http://www.opengis.net/ont/sf

Datatype properties for *geo:Geometry*

- Explicitly typed as owl:DatatypeProperty and rdf:Property
  - geo:dimension
  - geo:coordinateDimension
  - geo:spatialDimension
  - geo:isEmpty
  - geo:isSimple
  - geo:is3D
  - geo:asWKT
  - geo:asGML

  Only one of these -- based on the conformance class
- Implementations may do both.

From [3]

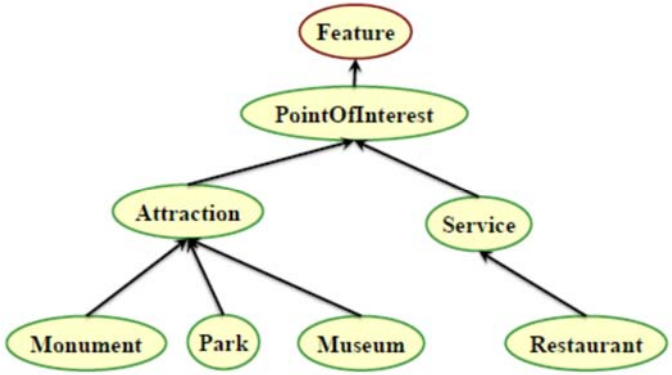
Example: representation of a spatial resource

```
ex:WashingtonMonument a geo:Feature;
                      rdfs:label "Washington Monument";
                      geo:hasGeometry ex:WMPoint .
ex:WMPoint a sf:Point;
           geo:asWKT "POINT(-77.03524 38.889468)"^^geo:wktLiteral.
```

1. The resource is a feature
2. This feature has a label
3. This feature has a geometry (geo-location)
4. This geometry is a point
5. This point is defined according the WKT representation by 2 coordinates (longitude latitude)

From [2]

Another example: semantic part



From [3]

Example: definition of the (semantic) structure

@prefix…

```
ex:Monument a owl:Class;
                    rdfs:subClassOf ex:Attraction .
ex:Park a owl:Class;
            rdfs:subClassOf ex:Attraction .
ex:Museum a owl:Class;
                  rdfs:subClassOf ex:Attraction .
ex:Restaurant a owl:Class;
                   rdfs:subClassOf ex:Service .
ex:Attraction a owl:Class;
                   rdfs:subClassOf ex:PointOfInterest .
ex:Service a owl:Class;
                rdfs:subClassOf ex:PointOfInterest .
ex:PointOfInterest a owl:Class;
                          rdfs:subClassOf geo:Feature .
```

From [2]

Example: definition of the geospatial data

@prefix…

```
ex:Monument1 a ex:Monument;
                   rdfs:label "Washington Monument";
                   geo:hasGeometry ex:Point1 .

ex:Point1 a sf:Point;
             geo:asWKT "POINT(-77.03524 38.889468)"^^geo:wktLiteral .

ex:Park1 a ex:Park;
             rdfs:label "Example Park";
             geo:hasGeometry ex:Polygon1 .

ex:Polygon1 a sf:Polygon;
              geo:asWKT "POLYGON((-77.05 38.87, -77.02 38.87, -77.02 38.9,
                           -77.05 38.9, 77.05 38.87))"^^geo:wktLiteral .
```

From [2]

Example of a query

Retrieve the geometry information of *ex:Monument1*

```
PREFIX…

SELECT ?wkt
WHERE {
            ex:Monument1 geo:hasGeometry ?g .
            ?g geo:asWKT ?wkt .
}
```

-> "POINT(-77.03524 38.889468)"

From [2]

GeoSPARQL relationships

Topological relationships :

- equals
- disjoint
- intersects
- touches
- within
- contains
- overlaps
- crosses

Different syntaxes according to the relations family:
for example, *geo:sfEquals* (Simple Features), *geo:ehEquals* (Egenhofer), *geo:rcc8eq* (RCC8) for the *equals* relation

Example of a query

Find the monuments that are within *ex:Park1*

PREFIX…

```
SELECT ?f
WHERE {
        ex:Park1 geo:hasGeometry ?g1 .
        ?f a ex:Monument;
         geo:hasGeometry ?g2 .
        ?g2 geo:sfWithin ?g1 .
}
```

-> ex:Monument1

From [2]

GeoSPARQL query functions

Topological functions:

- equals
- disjoint
- intersects
- touches
- within
- contains
- overlaps
- crosses

As topological relationships but use the prefix *geof* instead of *geo*: for example, *geof:sfEquals* (and not anymore *geo:sfEquals*)

Example of a query

Find whether there are monuments that are within *ex:Park1*

PREFIX…

```
SELECT ?f
WHERE {
        ex:Park1 geo:hasGeometry ?g1 .
        ?f a ex:Monument;
         geo:hasGeometry ?g2 .
        ?g2 geof:sfWithin ?g1 .
}
```

-> true

From [2]

Other query functions

Properties	Parameters	Returns
geof:distance	Geom1, Geom2, unitsURI	xsd:double
geof:buffer	Geom1, radius, unitsURI	Geometry literal
geof:convexHull	Geom1	Geometry literal
geof:intersection	Geom1, Geom2	Geometry literal
geof:union	Geom1, Geom2	Geometry literal
geof:difference	Geom1, Geom2	Geometry literal
…		
geof:envelope	Geom1	Geometry literal
geof:boundary	Geom1	Geometry literal
geof:getsrid	Geom1	SRID of literal

From [2]

Query Transformation Rules

- Allow for an additional layer of abstraction in SPARQL queries
- If a feature is used as the subject or object of a topological relation, the query is automatically rewritten to compare the geometry linked, thus removing the abstraction for processing

From [4]

Before:

```
…
SELECT ?f
WHERE {
      ?f a ex:Monument;
      ?f geo:sfWithin ex:Park1.
}
```


Query Transformation Rules

After:

```
…
SELECT ?f
WHERE {
            ex:Park1 geo:hasGeometry ?g1 .
            ?f a ex:Monument;
             geo:hasGeometry ?g2 .
            ?g2 geo:sfWithin ?g1 .
}
```

From [2]

Implementation: some sites

Parliament Triple Store
http://parliament.semwebcentral.org/

AllegroGraph
https://franz.com/agraph/allegrograph/

Strabon (spatiotemporal RDF store with some GeoSPARQL components)
http://www.strabon.di.uoa.gr/

GeoSPARQL sites (Online queries)
http://www.geosparql.org/
http://linkedgeodata.org/sparql

Some references

[1] **OGC GeoSPARQL – A Geographic Query Language for RDF Data**, Open Geospatial Consortium, OGC 11-052r4
http://www.opengis.net/doc/IS/geosparql/1.0

[2] **GeoSPARQL user guide**, Dave Kolas & Robert Battle, 1/19/2012

[3] **Getting started with GeoSPARQL**, Dave Kolas, Matt Perry & John Herring, OGC, Oct 29 2013
http://www.ssec.wisc.edu/meetings/geosp_sem/presentations/GeoSPARQL_Getting_Started%20-%20KolasWorkshop%20Version.pdf

[4] **GeoSPARQL: Enabling a Geospatial Semantic Web**, Robert Battle, Dave Kolas
http://www.semantic-web-journal.net/sites/default/files/swj176_0.pdf

Some references

WKT and WKB https://www.gaia-gis.it/spatialite-3.0.0-BETA/spatialite-cookbook-fr/html/wkt-wkb.html

W3C Geospatial Ontologies
http://www.w3.org/2005/Incubator/geo/XGR-geo-ont-20071023/
Geonames interface http://www.geonames.org/
Vocabularies for geospatial modelling http://geovocab.org/

OGC's geoSPARQL vocabulary
http://www.opengeospatial.org/standards/geosparql

Basic Geo (WGS84 lat/long) Vocabulary
https://www.w3.org/2003/01/geo/

Annex: acronyms

GFM	General Feature Model (as defined in ISO 19109)
GML	Geography Markup Language
KML	Keyhole Markup Language
OWL	OWL 2 Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
RIF	Rule Interchange Format
SPARQL	SPARQL Protocol and RDF Query Language
WKT	Well Known Text (as defined by Simple Features or ISO 19125)
W3C	World Wide Web Consortium (http://www.w3.org/)
XML	Extensible Markup Language

From [1]

Annex: XML namespaces

geo:	http://schemas.opengis.net/geosparql/1.0/geosparql_vocab_all.rdf
geof:	http://www.opengis.net/def/function/geosparql/
geor:	http://www.opengis.net/def/rule/geosparql/
gml:	http://www.opengis.net/ont/gml#
owl:	http://www.w3.org/2002/07/owl#
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
sf:	http://www.opengis.net/ont/sf#
xsd:	http://www.w3.org/2001/XMLSchema#

From [1]

Annex: SPARQL prefixes

```
PREFIX spatial: <http://jena.apache.org/spatial#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX gn: <http://www.geonames.org/ontology#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX loticoowl: <http://www.lotico.com/ontology/>
PREFIX spatial: <http://jena.apache.org/spatial#>
```

From [1]

Annex: Turtle prefixes

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix geo: <http://www.opengis.net/def/geosparql/> .
@prefix geof: <http://www.opengis.net/def/geosparql/function/> .
@prefix sf: <http://www.opengis.net/def/sf/> .
@prefix gml: <http://www.opengis.net/def/gml/> .
```

From [1]

Annex: SPARQL prefixes

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX geo: <http://www.opengis.net/def/geosparql/>
PREFIX geof: <http://www.opengis.net/def/geosparql/function/>
PREFIX sf: <http://www.opengis.net/def/sf/>
PREFIX gml: <http://www.opengis.net/def/gml/>
```

From [1]