Knowledge Technologies

An Introduction to Linked Geospatial Data

Manolis Koubarakis
http://www.di.uoa.gr/~koubarak

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

Knowledge Technologies

The Theory

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

Representation and Querying of Linked Geospatial Data

- Early papers:
  - Kollas (2007)
  - Perry’s Ph.D. dissertation (2008)
  - Koubarakis and Kyzirakos (2010)

- We concentrate on:
  - **GeoSPARQL (2012)**

Approach

- The proposal offers constructs for:
  - Developing **ontologies** for spatial data.
  - Encoding **spatial data** that use these ontologies **in RDF.**
  - **Extending SPARQL** to query spatial and temporal data.

GeoSPARQL

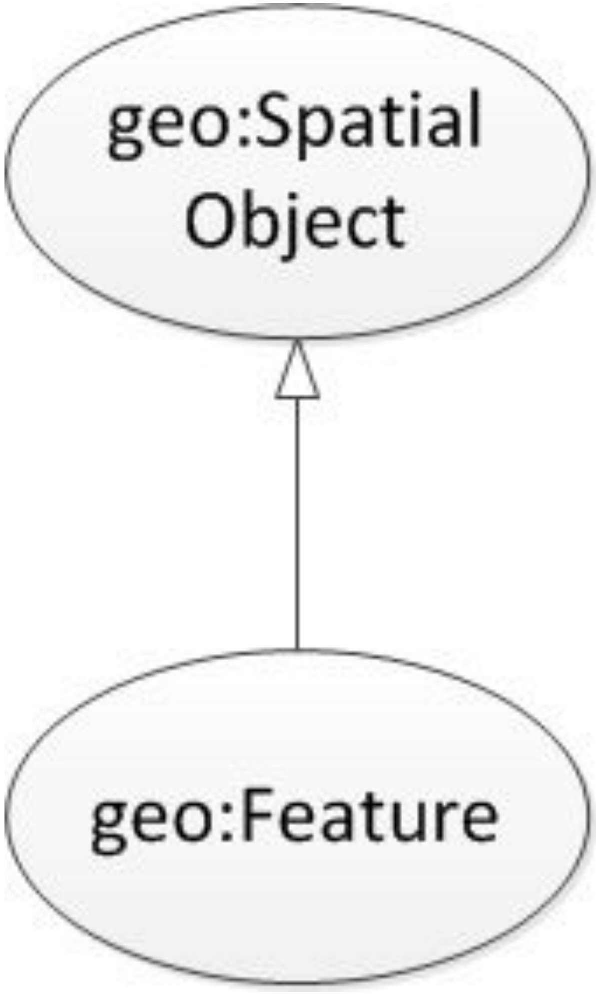
GeoSPARQL is an **OGC standard**. [Perry and Herring, 2012]

Main functionalities:

- Representing geospatial information is done using **high level ontologies** inspired from GIS terminology.
- Geometries are represented using **literals** of **spatial datatypes**.
- Literals are serialized using OGC standards **WKT** and **GML**.
- Families of **functions** are offered for querying geometries.

GeoSPARQL Core

Defines two **top level classes** that can be used to organize geospatial data.



Geographic Feature

- **Geographic feature or geographic object:** a domain entity that can have various **attributes** that describe **spatial and non-spatial** characteristics.

- **Example:** the country Greece with attributes
  - Population
  - Flag
  - Capital
  - Geographical area
  - Coastline
  - Bordering countries

Geographic Feature (cont’d)

- Geographic features can be **atomic** or **complex.**
- **Example:** According to the Kallikratis administrative reform of 2010, Greece consists of:
  - 13 **regions** (e.g., Crete)
  - Each region consists of **regional units** (e.g., Heraklion)
  - Each regional unit consists of **municipalities** (e.g., Dimos Chersonisou)
  - …

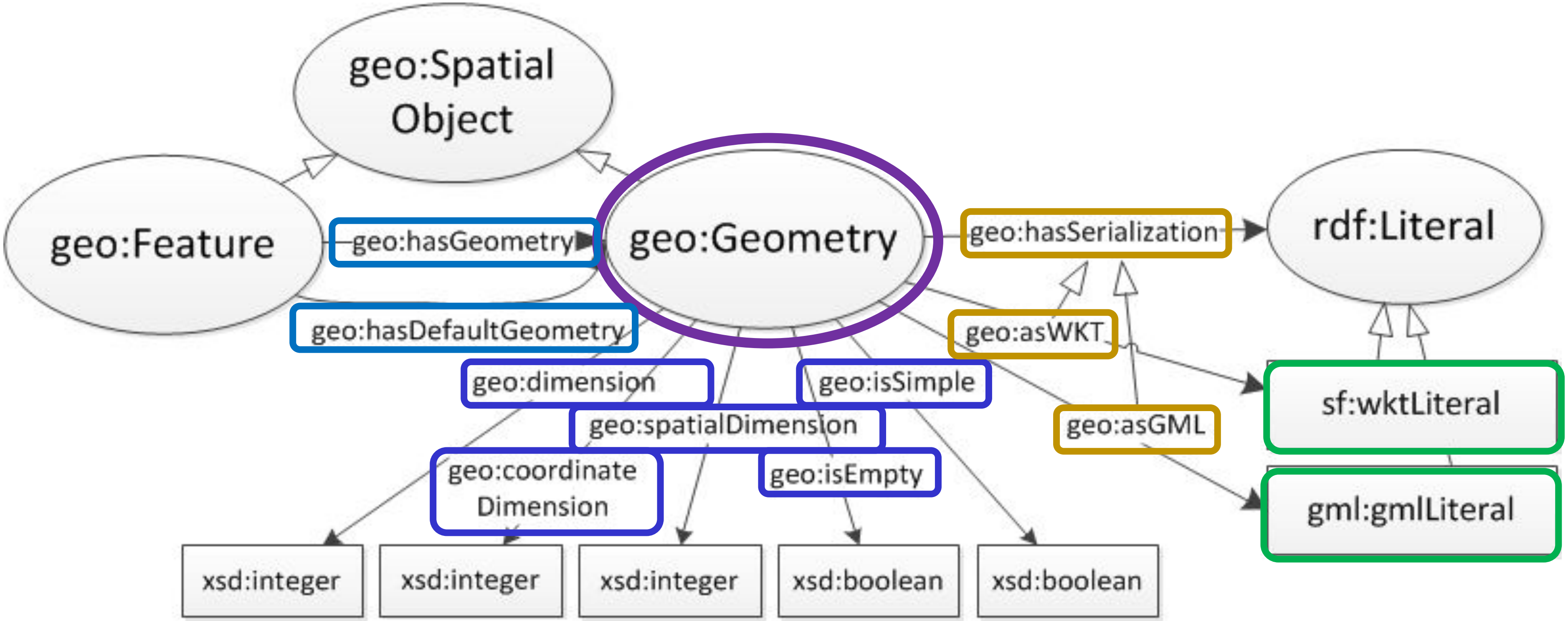
Geographic Feature (cont'd)

- The spatial characteristics of a feature can involve:
  - **Geometric information** (location in the underlying geographic space, shape etc.)
  - **Topological information** (containment, adjacency etc.).

Municipalities of the regional unit of Heraklion:
1. Dimos Irakliou
2. Dimos Archanon-Asterousion
3. Dimos Viannou
4. Dimos Gortynas
5. Dimos Maleviziou
6. Dimos Minoa Pediadas
7. Dimos Festou
8. Dimos Chersonisou

GeoSPARQL Geometry Extension

Provides vocabulary for asserting and querying data about the **geometric attributes of a feature.**



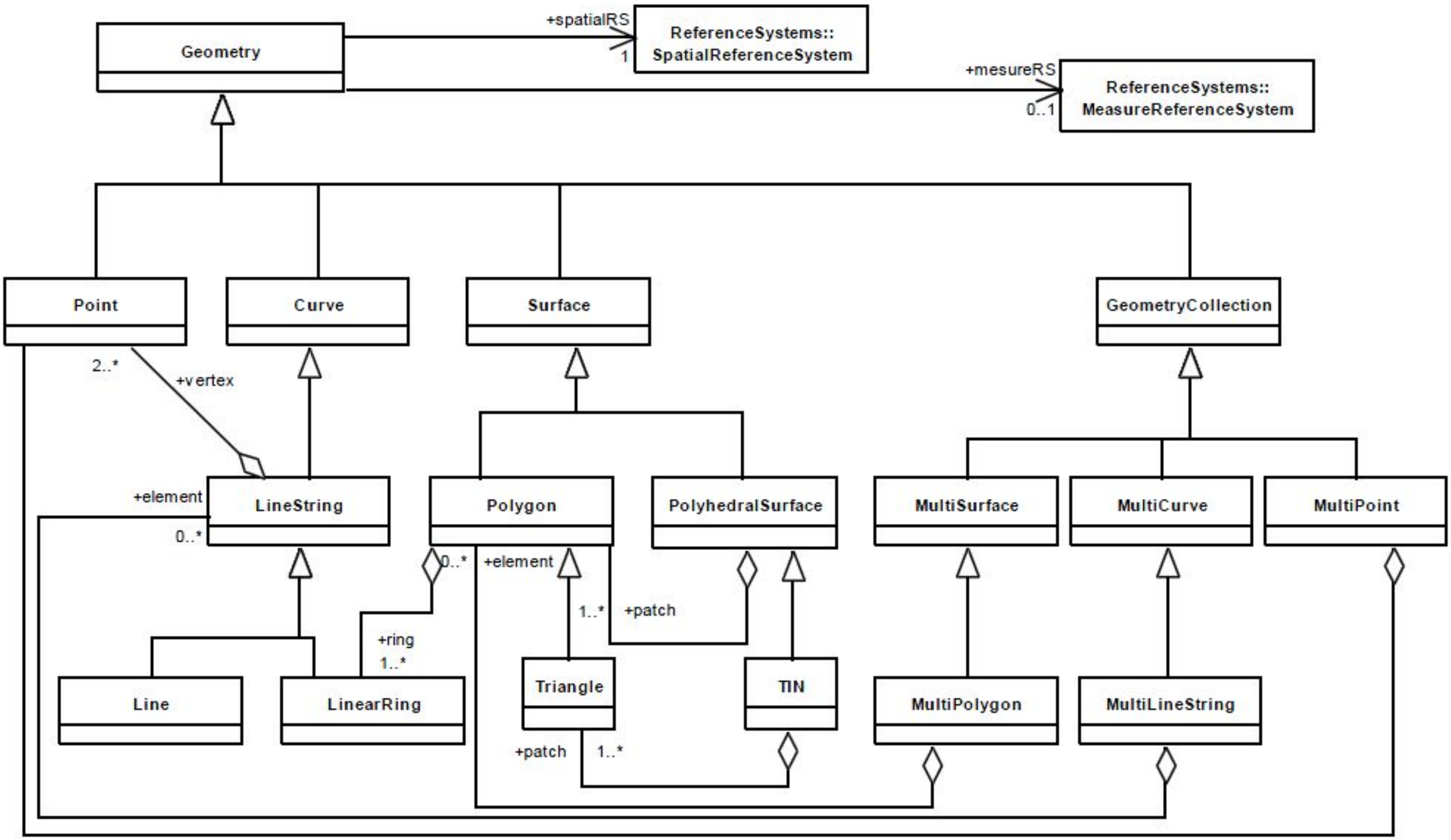
Well-Known Text (WKT)

- WKT is an OGC and ISO standard for representing **geometries**, **coordinate reference systems**, and **transformations** between coordinate reference systems.

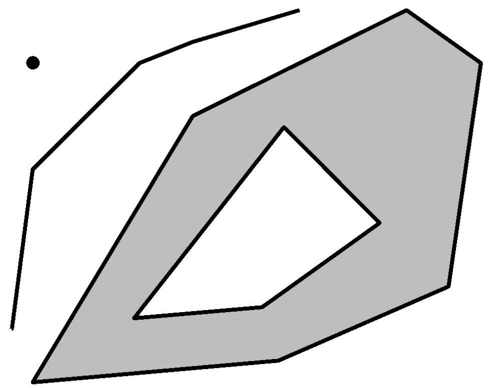
- WKT is specified in **OpenGIS Simple Feature Access - Part 1: Common Architecture** standard which is the same as the **ISO 19125-1** standard. Download from http://portal.opengeospatial.org/files/?artifact_id=25355 .

- This standard concentrates on **simple features:** features with all spatial attributes described piecewise by a straight line or a planar interpolation between sets of points.

WKT Class Hierarchy



Example



WKT representation:

```
GeometryCollection(
  Point(5 35),
  LineString(3 10,5 25,15 35,20 37,30 40),
  Polygon((5 5,28 7,44 14,47 35,40 40,20 30,5 5),
          (28 29,14.5 11,26.5 12,37.5 20,28 29))
  )
```

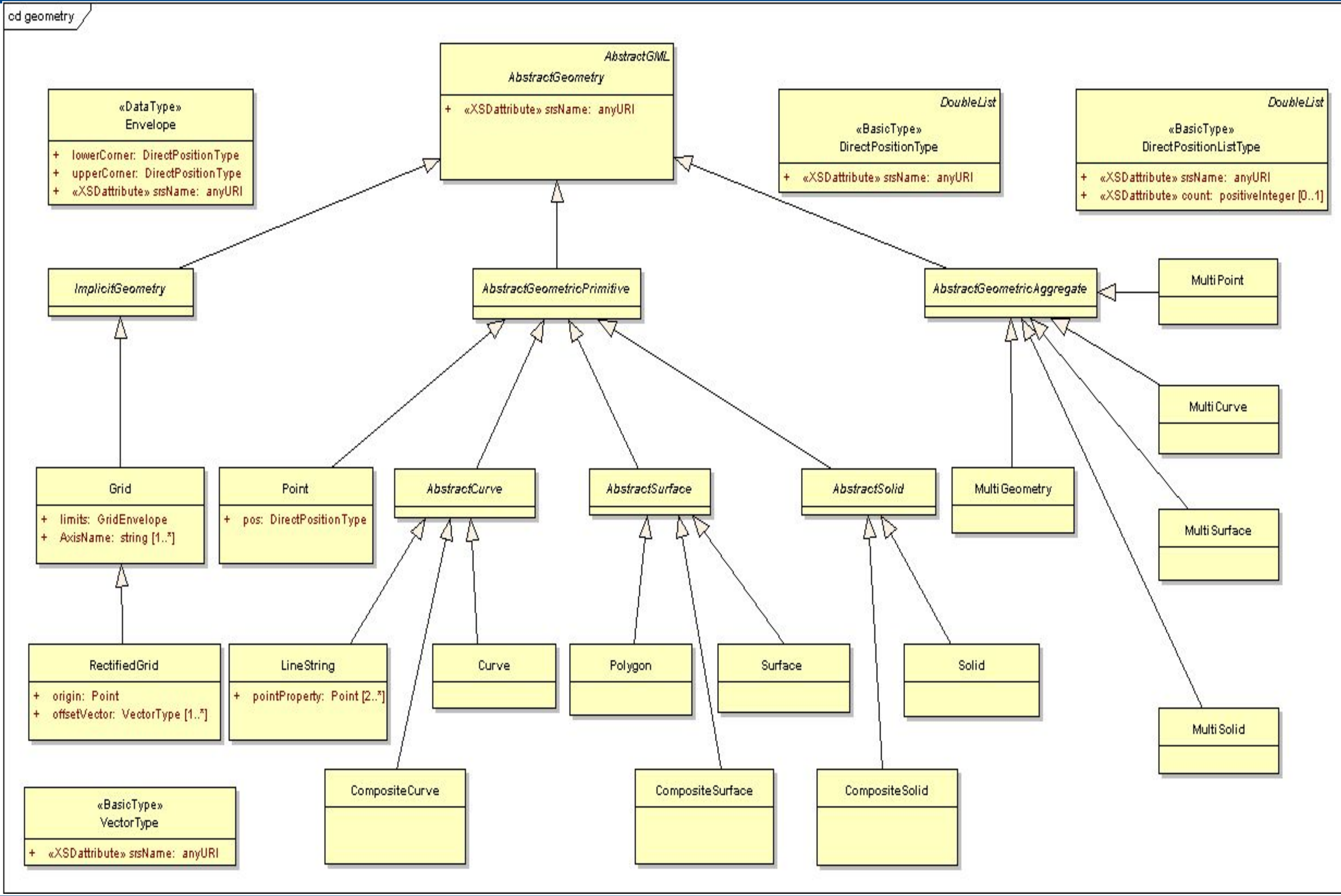
Geography Markup Language (GML)

- **GML** is an **XML-based encoding standard** for the representation of geospatial data.

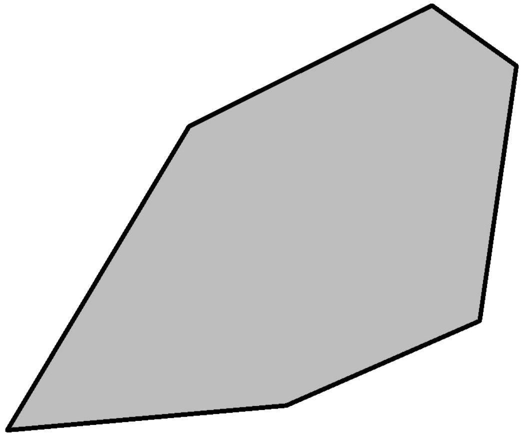
- GML provides XML schemas for defining a variety of concepts: **geographic features, geometry, coordinate reference systems, topology, time** and **units of measurement**.

- **GML profiles** are subsets of GML that target particular applications.
  - **Examples**: Point Profile, GML Simple Features Profile etc.

GML Simple Features: Class Hierarchy



Example

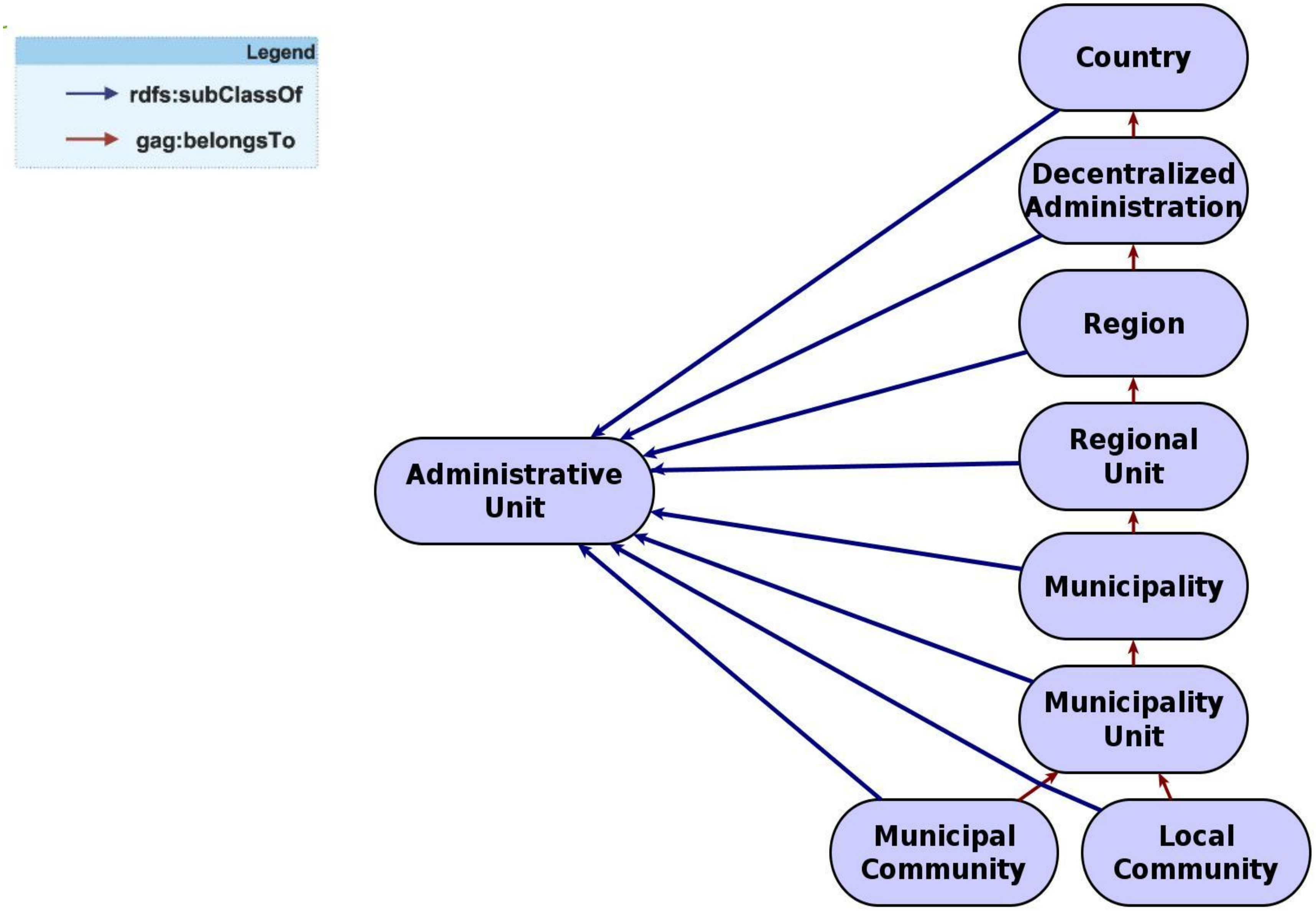


GML representation:

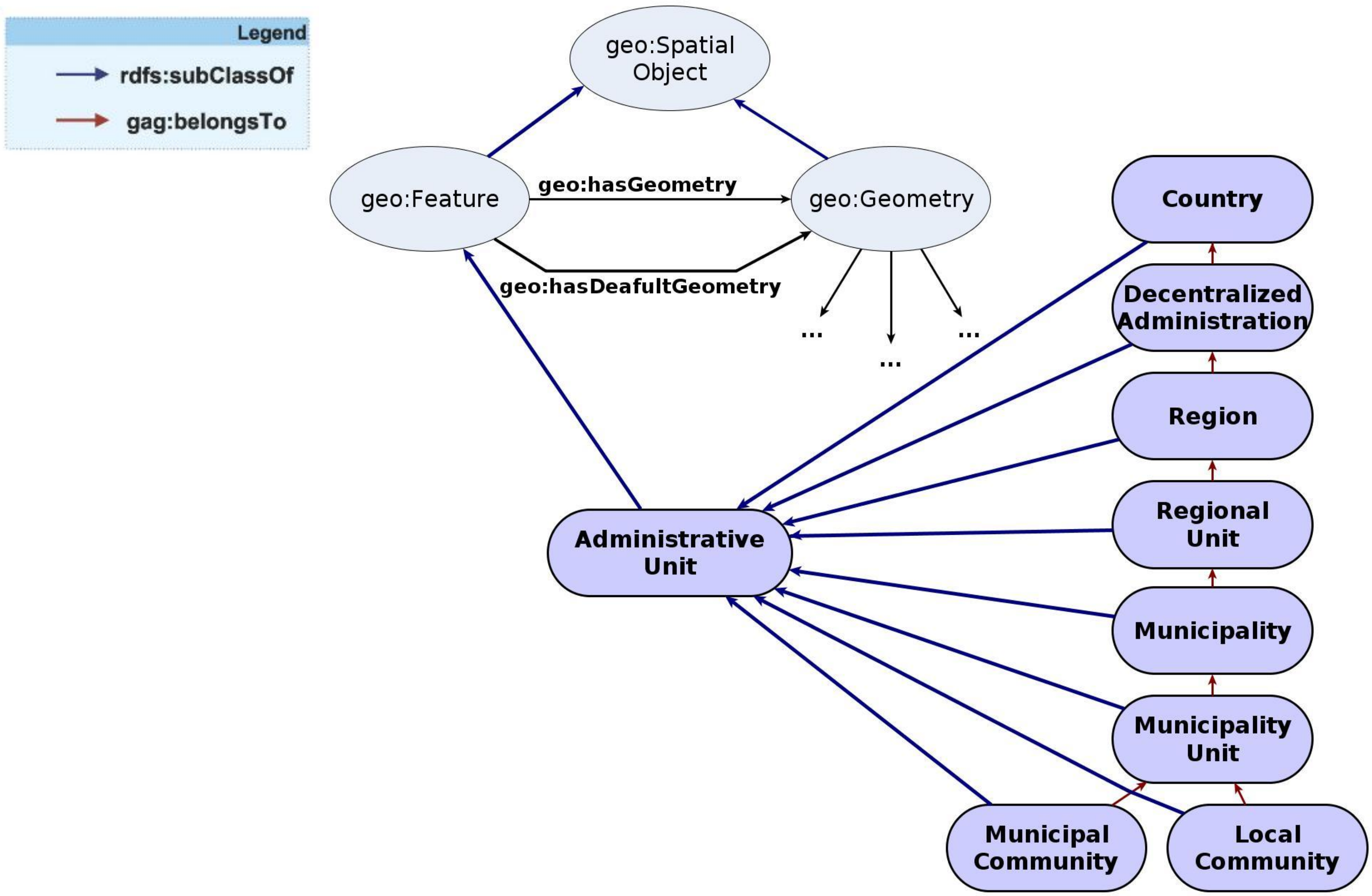
```
<gml:Polygon gml:id="p3" srsName="urn:ogc:def:crs:EPSG:6.6:4326”>
    <gml:exterior>
        <gml:LinearRing>
            <gml:coordinates>
                5,5 28,7 44,14 47,35 40,40 20,30 5,5
            </gml:coordinates>
        </gml:LinearRing>
    </gml:exterior>
</gml:Polygon>
```

Example: Greek Administrative Geography

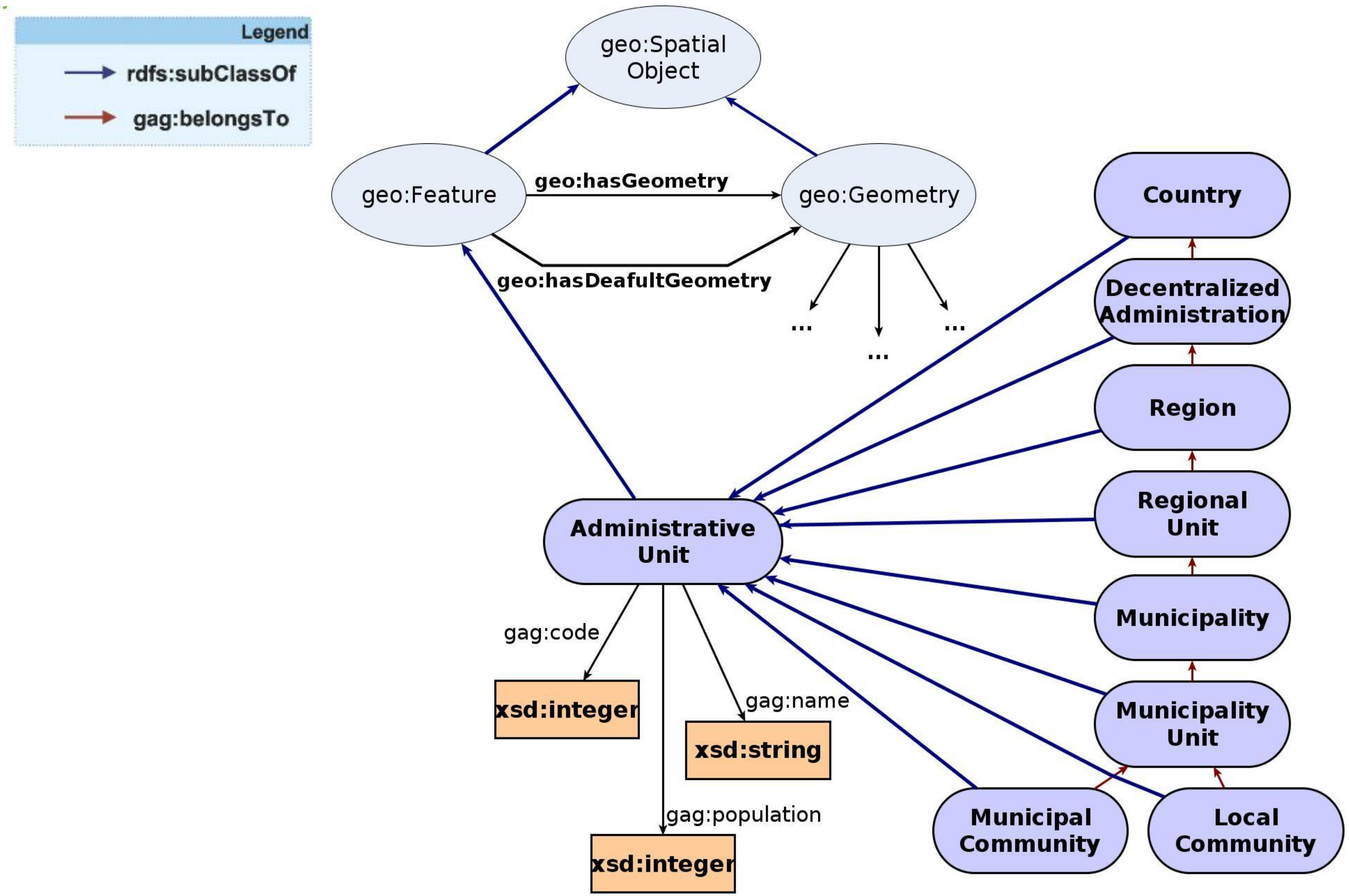
Domain Ontology



Greek Administrative Geography



Greek Administrative Geography

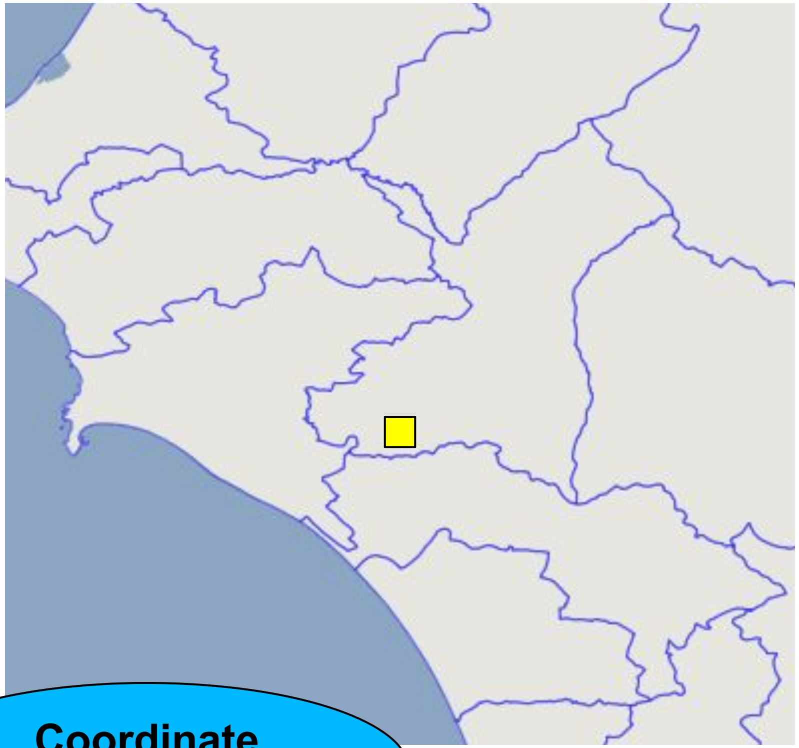


Example Data

```
gag:Olympia
  rdf:type gag:MunicipalCommunity;
  gag:name "Ancient Olympia";
  gag:population "184"^^xsd:int;
  geo:hasGeometry ex:polygon1.
```

Property from Geometry extension

Class from Geometry extension



```
ex:polygon1
  rdf:type geo:Geometry;
  geo:asWKT "http://www.opengis.net/def/crs/OGC/1.3/CRS84
             POLYGON((21.5 18.5,23.5 18.5,
                       23.5 21,21.5 21,21.5 18.5))"
      ^^sf:wktLiteral.
```

Coordinate Reference System

Geometry literal

Property from Geometry extension

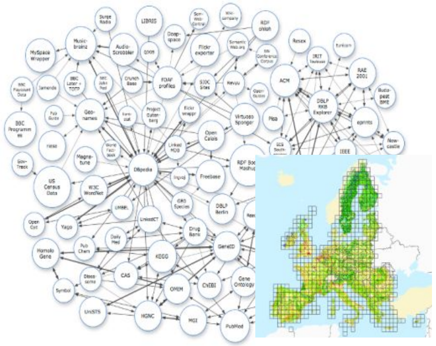
Datatype from Geometry extension

Non-Topological Query Functions of the Geometry Extension

- The following non-topological query functions are also offered:
  - `geof:distance`
  - `geof:buffer`
  - `geof:convexHull`
  - `geof:intersection`
  - `geof:union`
  - `geof:difference`
  - `geof:symDifference`
  - `geof:envelope`
  - `geof:boundary`

GeoSPARQL: An example

Find forests near municipal communities



```
SELECT ?r ?c
WHERE {
   ?r rdf:type clc:Region;
      geo:hasGeometry ?rGeom;
      clc:hasCorineLandCover ?f.
   ?rGeom geo:asWKT ?rGeomWKT .
   ?f rdfs:subClassOf clc:Forest.
   ?c rdf:type gag:MunicipalCommunity;
      geo:hasGeometry ?cGeom.
   ?cGeom geo:asWKT ?rGeomWKT .

FILTER(geof:distance(?rGeomWKT,?cGeomWKT,uom:metre) < 1000)}
```

GeoSPARQL Topology Vocabulary Extension

- The extension is used for representing **topological information** about features.
- Topological information is **inherently qualitative** and it is expressed in terms of **topological relations** (e.g., containment, adjacency, overlap etc.).
- Topological information can be **derived from geometric information** or it might be captured by **asserting explicitly the topological relations** between features.

Topological Relations

- The study of topological relations has produced a lot of interesting results by researchers in:
  - GIS
  - Spatial databases
  - Artificial Intelligence (qualitative reasoning and knowledge representation)

DE-9IM

- The **dimensionally extended 9-intersection model (DE-9IM)** of Clementini and Felice.

- It is based on the **point-set topology** of R^2.
- It deals with **simple, closed and connected geometries (areas, lines, points).**

- It is an extension of earlier approaches: the **4-intersection (4IM)** and **9-intersection (9IM)** models by Egenhofer and colleagues.

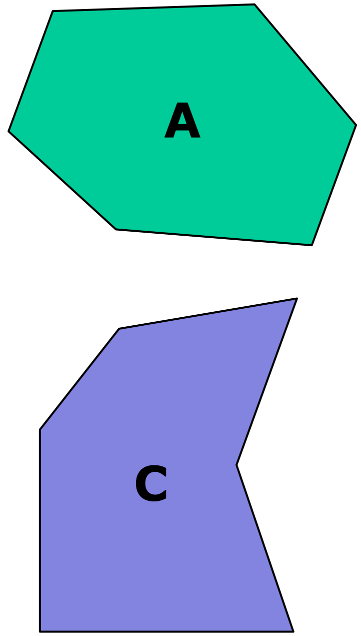
Topological Relations in DE-9IM

- It captures **topological relationships** between two geometries *a* and *b* in R^2 by considering the **dimensions of the intersections of the boundaries, interiors and exteriors** of the two geometries:

$$\text{DE-9IM}(a,b) = \begin{bmatrix} \dim(I(a) \cap I(b)) & \dim(I(a) \cap B(b)) & \dim(I(a) \cap E(b)) \\ \dim(B(a) \cap I(b)) & \dim(B(a) \cap B(b)) & \dim(B(a) \cap E(b)) \\ \dim(E(a) \cap I(b)) & \dim(E(a) \cap B(b)) & \dim(E(a) \cap E(b)) \end{bmatrix}.$$

- The dimension can be **2, 1, 0** and **-1** (dimension of the empty set).

Example



	I(C)	**B**(C)	**E**(C)
I(A)	-1	-1	2
B(A)	-1	-1	1
E(A)	2	1	2

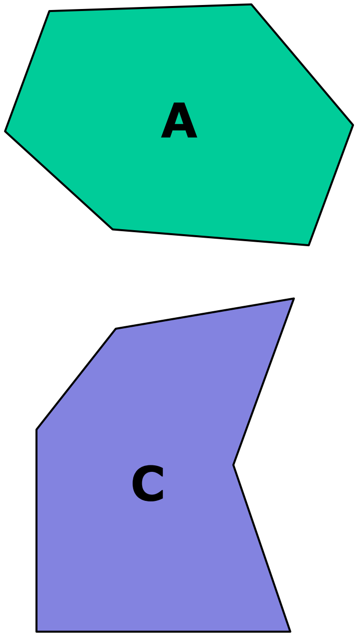
Topological Relations in DE-9IM

- The following five **named relationships** between two different geometries can be distinguished: **disjoint, touches, crosses, within** and **overlaps**.

- The named relationships have a **reasonably intuitive meaning** for users. They are **jointly exclusive and pairwise disjoint (JEPD).**

- The model can also be defined using an appropriate **calculus of geometries** that uses these 5 binary relations.

Example: A disjoint C

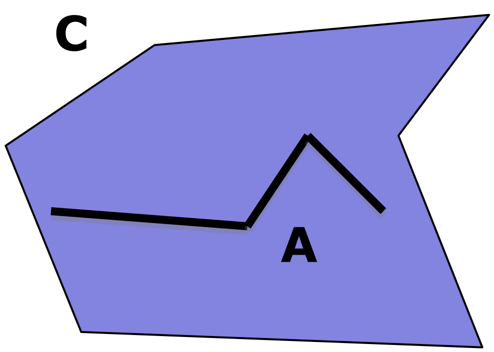


	I(C)	**B**(C)	**E**(C)
I(A)	F	F	*
B(A)	F	F	*
E(A)	*	*	*

Notation:

- T = { 0, 1, 2 }
- F = -1
- * = don't care = { -1, 0, 1, 2 }

Example: A within C

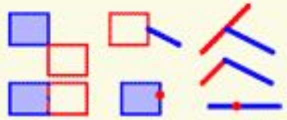
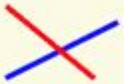
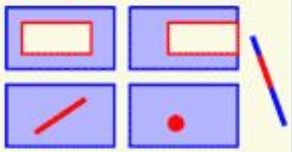
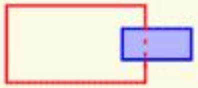


	I(C)	**B**(C)	**E**(C)
I(A)	T	*	F
B(A)	*	*	F
E(A)	*	*	*

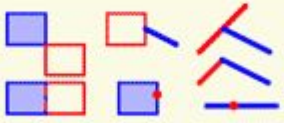
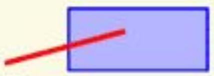
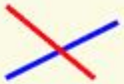
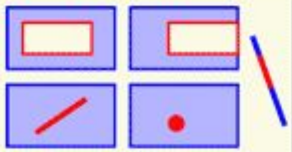
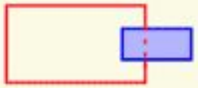
Notation equivalent to 3x3 matrix:

- String of 9 characters representing the above matrix in row major order.
- In this case: T*F**F***

DE-9IM Relation Definitions

Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

Simple Features Relation Definitions

Beziehung	Definition	Beispiele
A disjoint B	$\begin{bmatrix} F & F & * \\ F & F & * \\ * & * & * \end{bmatrix}$	
A touches B (d(A) > 0 ∨ d(B) > 0)	$\begin{bmatrix} F & T & * \\ * & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ T & * & * \\ * & * & * \end{bmatrix} \vee \begin{bmatrix} F & * & * \\ * & T & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) < d(B))	$\begin{bmatrix} T & * & T \\ * & * & * \\ * & * & * \end{bmatrix}$	
A crosses B (d(A) = d(B) = 1)	$\begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$	
A within B	$\begin{bmatrix} T & * & F \\ * & * & F \\ * & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B), d(A) ≠ 1, d(B) ≠ 1)	$\begin{bmatrix} T & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	
A overlaps B (d(A) = d(B) = 1)	$\begin{bmatrix} 1 & * & T \\ * & * & * \\ T & * & * \end{bmatrix}$	

- **A equals B** can be defined by the pattern TFFFTFFFT.
- **A intersects B** is the negation of **A disjoint B**
- **A contains B** is equivalent to **B within A**

The Region Connection Calculus (RCC)

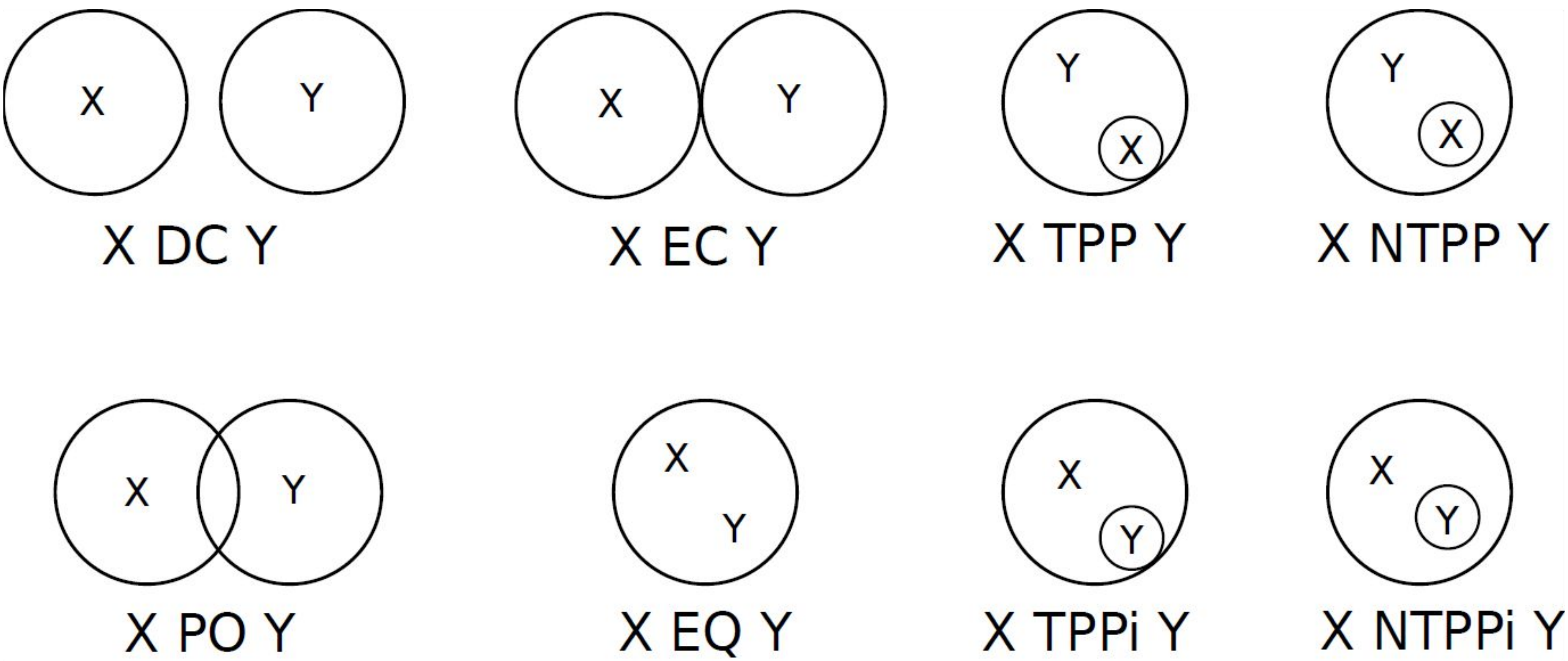
- The primitives of the calculus are **spatial regions**. These are **non-empty, regular closed** subsets of a topological space.

- The calculus is based on a single binary predicate *C* that formalizes the “**connectedness**” relation.
  - ***C(a,b)*** **is true** when the closure of *a* is connected to the closure of *b* i.e., they have at least one point in common.

- It is axiomatized using first order logic.

RCC-8

- This is a set of **eight JEPD binary relations** that can be defined in terms of predicate *C*.



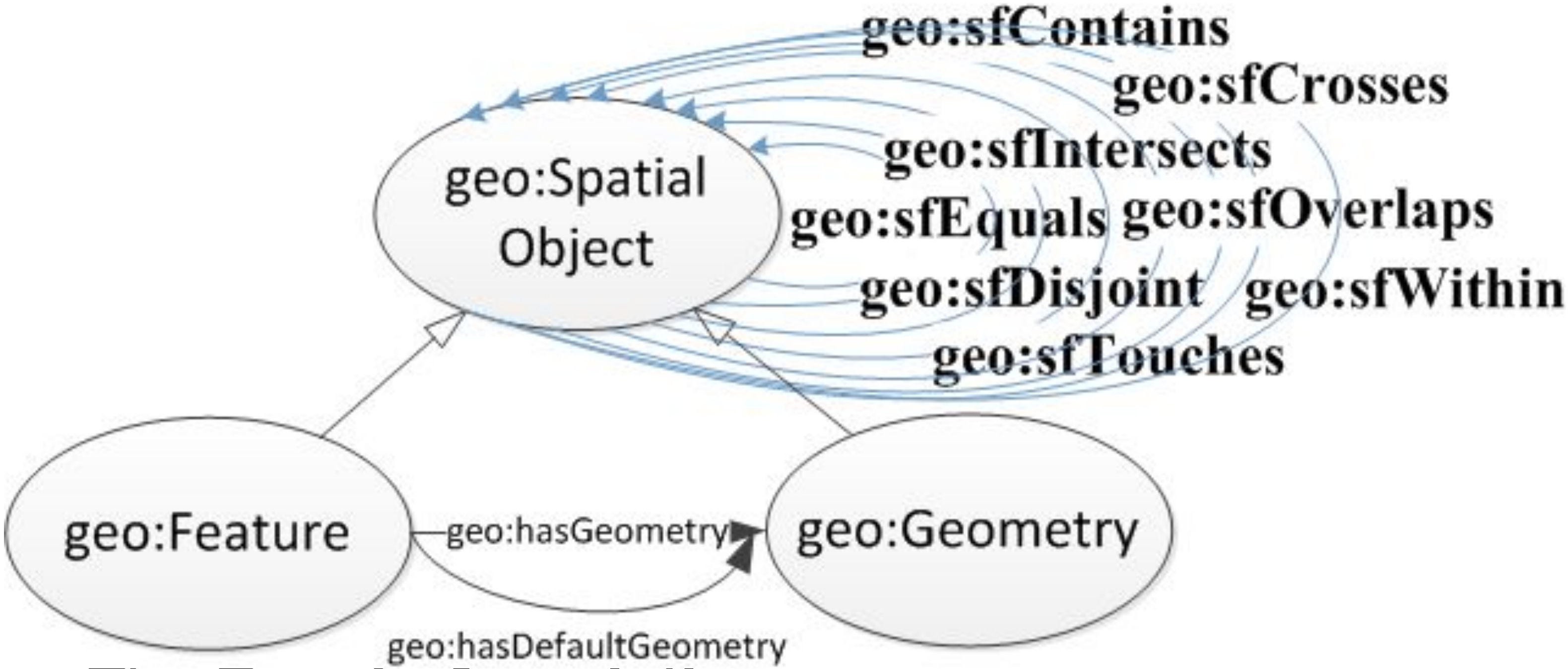
RCC-5

- The **RCC-5 subset** has also been studied. The granularity here is coarser. The boundary of a region is not taken into consideration:
  - No distinction among DC and EC, called just DR.
  - No distinction among TPP and NTPP, called just PP.

- RCC-8 and RCC-5 relations **can also be defined using point-set topology**, and there are very close connections to the models of Egenhofer and others.

GeoSPARQL Topology Vocabulary Extension

- The extension is parameterized by the **family of topological relations** supported:
  - Topological relations for **simple features**



  - The **Egenhofer relations** e.g., `geo:ehMeet`
  - The **RCC-8 relations** e.g., `geo:rcc8ec`

Greek Administrative Geography

```
gag:Olympia
  rdf:type gag:MunicipalCommunity;
  gag:name "Ancient Olympia".

gag:OlympiaMUnit
  rdf:type gag:MunicipalityUnit;
  gag:name "Municipality Unit of
              Ancient Olympia".

gag:OlympiaMunicipality
  rdf:type gag:Municipality;
  gag:name "Municipal
              Anc
```

Simple Features topological relation

```
gag:Olympia geo:sfWithin gag:OlympiaMUnit .

gag:OlympiaMUnit geo:sfWithin gag:OlympiaMunicipality.
```

GeoSPARQL: An example

Find the **municipality unit** that contains the community of Ancient Olympia

```
SELECT   ?m

WHERE {

  ?m rdf:type gag:MunicipalityUnit.

  ?m geo:sfContains gag:Olympia.

}
```

Simple Features topological relation

Answer: **?m = gag:OlympiaMUnit**

GeoSPARQL: An example

Find the **municipality** that contains the community of Ancient Olympia

```
SELECT   ?m

WHERE {

 ?m rdf:type gag:Municipality.

 ?m geo:sfContains gag:Olympia.

}
```

Answer: **?m = gag:OlympiaMunicipality**

Example (cont'd)

The answer to the previous query can be computed by **reasoning about the transitivity** of relation **`geo:sfContains`**.

Implementation options:

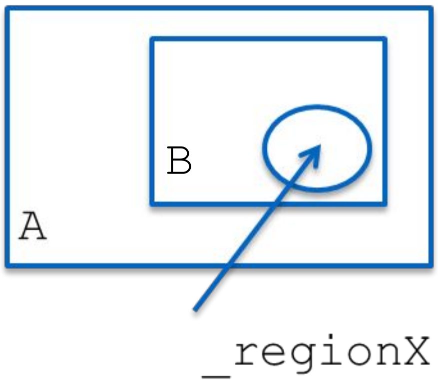
- Use rules
- Use description logic reasoners for topological information like PelletSpatial and RacerPro
- Use constraint solvers for topological constraints (our favourite approach)

Beyond the Topology Vocabulary Extension

Triples:

```
ex:regionA strdf:hasGeometry "POLYGON(A)"^^strdf:WKT .
ex:regionB strdf:hasGeometry "POLYGON(B)"^^strdf:WKT .

_:regionX geo:sfWithin ex:regionB
```



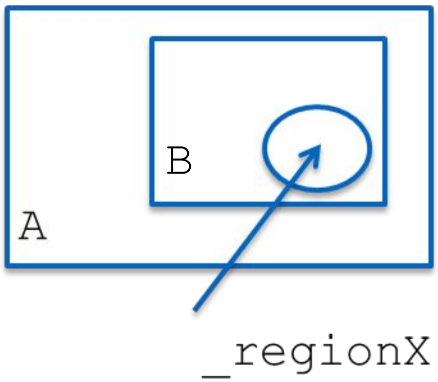
Query: Is `regionX` contained in `regionA`?

Beyond the Topology Vocabulary Extension (cont'd)

Triples:

```
ex:regionA strdf:hasGeometry "POLYGON(A)"^^strdf:WKT .
ex:regionB strdf:hasGeometry "POLYGON(B)"^^strdf:WKT .

_:regionX geo:sfWithin ex:regionB
```



Query: Is `regionX` contained in `regionA`?
Answer: Yes

Now the transitivity reasoning has to take into account the geometries of regions `A` and `B`.

GeoSPARQL Geometry Topology Extension

- Offers vocabulary for **querying topological properties** of geometry literals.

- **Simple Features**
  - `geof:relate`
  - `geof:sfEquals`
  - `geof:sfDisjoint`
  - `geof:sfIntersects`
  - `geof:sfTouches`
  - `geof:sfCrosses`
  - `geof:sfWithin`
  - `geof:sfContains`
  - `geof:sfOverlaps`

- **Egenhofer** (e.g., `geof:ehDisjoint`)
- **RCC-8** (e.g., `geof:rcc8dc`)

Example Query

Return the names of local communities that have been affected by fires

```
SELECT  ?name
WHERE {
   ?comm   rdf:type gag:LocalCommunity;
        gag:name ?name;
        geo:hasGeometry ?commGeo .
  ?commGeo geo:asWKT ?commGeoWKT .
  ?ba rdf:type noa:BurntArea;
        geo:hasGeometry ?baGeo .
  ?baGeo geo:asWKT ?baGeoWKT .

FILTER(geof:sfOverlaps(?commGeoWKT,?baGeoWKT))
}
```

The Practical Stuff

Dept. of Informatics and Telecommunications
National and Kapodistrian University of Athens

GeoSPARQL

- Offers vocabulary for **querying topological properties** of geometry literals, as well as useful functions for making calculations.

- **Simple Features**
  - `geof:relate`
  - `geof:sfEquals`
  - `geof:sfDisjoint`
  - `geof:sfIntersects`
  - `geof:sfTouches`
  - `geof:sfCrosses`
  - `geof:sfWithin`
  - `geof:sfContains`
  - `geof:sfOverlaps`
  - `geof:sfDistance`
  - `geof:area`
  - `geof:envelope`

GeoSPARQL Examples

- PREFIXES used in the examples

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
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 yago: <http://yago-knowledge.org/resource/>
PREFIX y2geor: <http://kr.di.uoa.gr/yago2geo/resource/>
PREFIX y2geoo: <http://kr.di.uoa.gr/yago2geo/ontology/>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX km: <http://rdf.useekm.com/ext#>
```

Example Query: hasGeometry & asWKT

What is the geometry of **Londonderry**?

```
SELECT ?lWKT WHERE {
   y2geor:osnientity_Londonderry_1414
      geo:hasGeometry ?lGeo .
   ?lGeo geo:asWKT ?lWKT .
}
```

Example Query: sfContains

Does **Wales** contain **Londonderry**? 🔄

```
ASK WHERE {
  y2geor:osnientity_Londonderry_1414
    geo:hasGeometry ?lGeo .
    ?lGeo geo:asWKT ?lWKT .

  yago:Wales geo:hasGeometry ?wGeo .
    ?wGeo geo:asWKT ?wWKT

  FILTER geof:sfContains(?wWKT, ?lWKT) 🔄
}
```

Example Query: sfTouches

Is **County Mayo** bordering **County Sligo**?

```
ASK WHERE {
  yago:County_Mayo geo:hasGeometry ?o .
     ?o geo:asWKT ?WKT .

  yago:County_Sligo geo:hasGeometry ?o1 .
     ?o1 geo:asWKT ?WKT1

  FILTER geof:sfTouches(?WKT, ?WKT1)
}
```

Example Query: sfDistance

What is the distance between **Oxfordshire** and **Kent**?

```
SELECT ?distance
WHERE {
  yago:Oxfordshire geo:hasGeometry ?geo1 .
  ?geo1 geo:asWKT ?oxfWKT .

  yago:Kent geo:hasGeometry ?geo2 .
  ?geo2 geo:asWKT ?kentWKT .

  BIND(geof:distance(?oxfWKT, ?kentWKT, uom:metre)
     AS ?distance)
}
```

Example Query: area

Is **York** larger than **New York**?

```
ASK WHERE {
   yago:York geo:hasGeometry ?yGeo .
      ?yGeo geo:asWKT ?yWKT .

   yago:New_York_City geo:hasGeometry ?nyGeo .
      ?nyGeo geo:asWKT ?nyWKT .

   FILTER( km:area(?yWKT) > km:area(?nyWKT) )
}
```

Example Query: area and envelope

Is the **York** larger than the bounding box of **York**?

```
ASK WHERE {
   yago:York geo:hasGeometry ?yGeo .
      ?yGeo geo:asWKT ?yWKT .

   FILTER
   (
     km:area(?yWKT) > km:area(geof:envelope(?yWKT))
   )
}
```