

Design of an Oracle Spatial and Graph server with GeoSPARQL implementation, for real case application

Evangelos Theocharous

09 November 2017

Contents

1	Introduction	1
2	Related work	3
2.1	RDF	3
2.2	Linked Data	4
2.3	SPARQL	4
2.4	GeoSPARQL	4
2.5	Oracle Spatial and Graph	6
2.6	Spatial Functions	8
3	Research questions	8
4	Methodology	8
5	Time planning	9

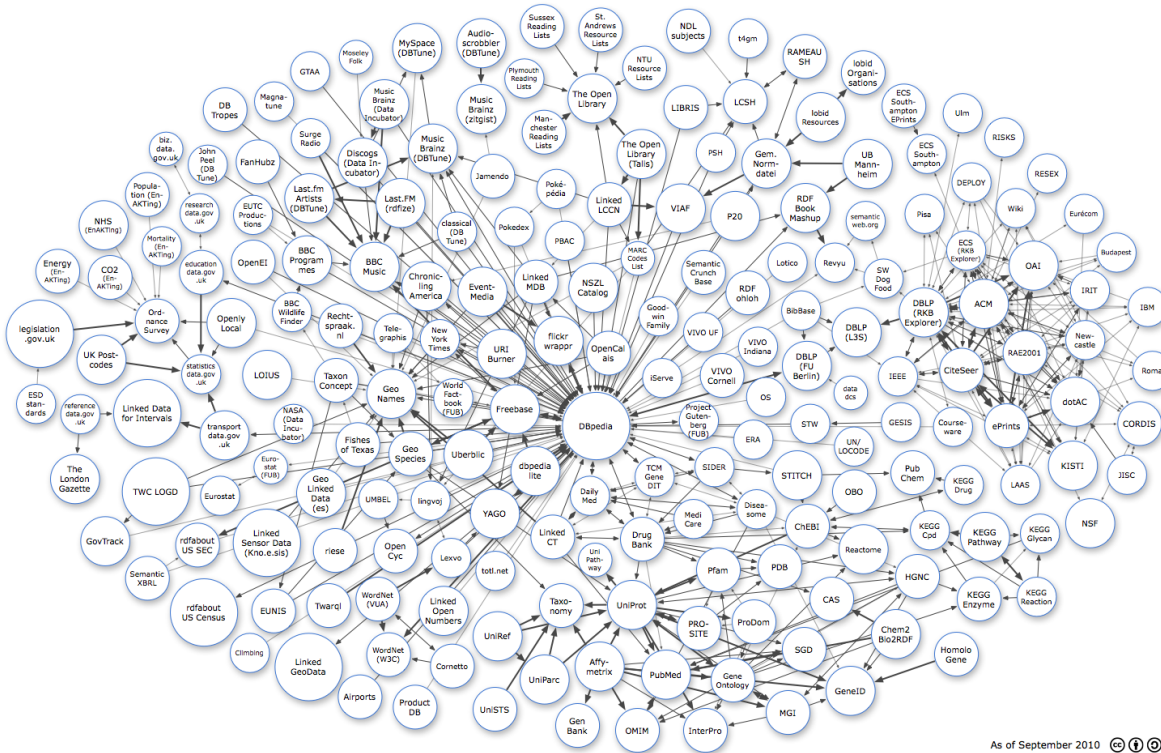
1 Introduction

European Union (EU) promotes the re-use of public government data, in order to increase transparency in government and the growth of information industry, through the implementation of “ Directive on reuse of public sector information ”, known as the PSI Directive (Kulk and van Loenen, 2012). Moreover, many projects, activities and policies regarding environmental protection and restoration have been developed between different member states of European Union. To support these actions, EU developed the INSPIRE Directive in May 2007, that establishes an infrastructure for spatial information in Europe (European Parliament, 2007). Such an infrastructure enables exchange, sharing, access and use of inter-operable spatial data and spatial data services across the various levels of public authority and across different sectors and it is known as Spatial Data Infrastructure (SDI) (European Parliament, 2007).

Along with EU, another attempt for publishing and reusing data began, with a completely different background and aim. Tim Berners-Lee, published his idea about the Semantic Web (Berners-Lee et al., 2006). According to World Wide Web Consortium (W3C),

"The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". The Semantic Web, however, is more about creating machine and human readable links between published data that help explore the web of data. Those data along with the links are called Linked Data.

Figure 1: The Cloud of Linked Data



Nevertheless, Linked Data cannot be just published with the traditional way one can post a web page. Linked Data should be published using the Resource Description Framework (RDF), which was originally developed to be a metadata data model which is a directed, labeled graph data format for representing information on the web. Linked Data can be accessed using the SPARQL Protocol and RDF Query Language (a recursive acronym for SPARQL) and an SPARQL endpoint. The RDF statements contain a subject, a predicate and an object; those three are commonly referred as a triple (Lassila and Swick, 1999). The triples are a way to define (predicate) something(object) about someone(subject). From the triple, the subject must be either a blank node or a Unified Resource Identifier (URI), the Predicate must be a URI and the Subject can be blank node, URI or a literal. From all three, only the predicate should definitely be a URI that semantically connects Object and Subject. For the purpose of describing the relationships between entities, vocabularies have been developed, using the RDF Vocabulary Definition Language(RDFS) and the Web Ontology Language (OWL)(Bizer et al., 2009).

Although SPARQL can handle all kinds of data, expressing spatial data as linked data does not offer as much. Hence, Open Geospatial Consortium (OGC) developed GeoSPARQL, a standard for representation and querying geospatial linked data, which contains an ontology based on OGC standards that supports qualitative and quantitative spatial reasoning and querying using SPARQL (Perry and Herring, 2012). Within the GeoSPARQL standard there are 10 spatial functions defined while another five spatial function are defined as possible for

development (?). More specifically, the GeoSPARQL already includes:

1. Relate
2. Distance
3. Buffer
4. Convex hull
5. Intersection
6. Union
7. Difference
8. Symmetric difference
9. Envelope
10. Boundary

While the possibilities for development include functions like:

1. Area. A function that returns the area of an object.
2. Nearest neighbor. A function that returns the nearest neighboring object.
3. Generalization. A function that returns the generalized type of a geometry eg. A polygon is the generalization of a triangle.
4. Centroid. A function that returns the center point of a geometry or a collection of geometries.
5. Bounding box. A function that returns a geometry that contains all the selected geometries.

Those functionalities are some of possible improvements of GeoSPARQL specified by the OGC, that aim to make it more simple, include some more expressive properties and geometry types and make it more specific.

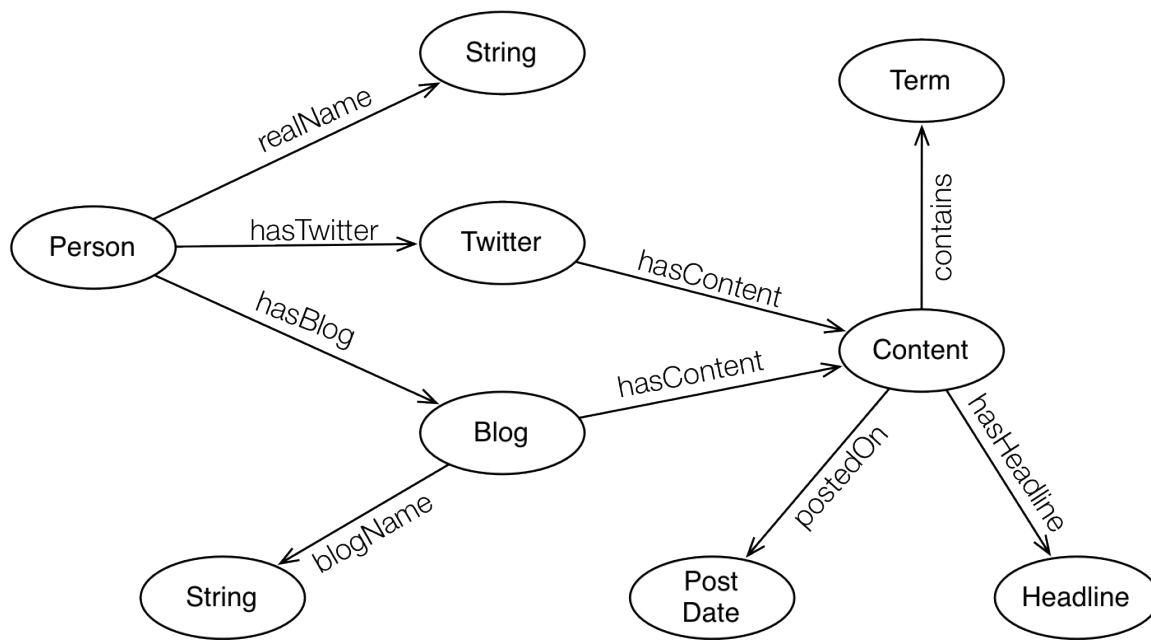
2 Related work

The following section presents the relevant literature and links it to the project.

2.1 RDF

The Resource Description Framework is a family of World Wide Web Consortium (W3C) specifications. It has been introduced on 1999. The RDF 1.0 specification was published in 2004 and the RDF 1.1 specification in 2014. RDF which was originally developed to be a metadata data model, and it is a directed, labeled graph data format for representing information on the web. RDF has been adopted as the medium to post and connect linked data. RDF statements, also called triples, can be encoded in a number of different formats that might be XML based (e.g., RDF/XML) or not (e.g. Turtle).

Figure 2: Linked Data as Directed Graph



2.2 Linked Data

As stated by Berners-Lee (2001), the Semantic Web is an extension to the existing web, which attempts to give computer and human meaning to existing dataset. Due to the global character of World Wide Web (www) and the huge volume of information that is published there, one of the initial steps is to connect this information to the according dataset, thus creating "a cloud of linked data", in sort Linked Data. Berners-Lee (2009) also stated four rules that optimize the exploration of data published on the web.

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. so that they can discover more things.

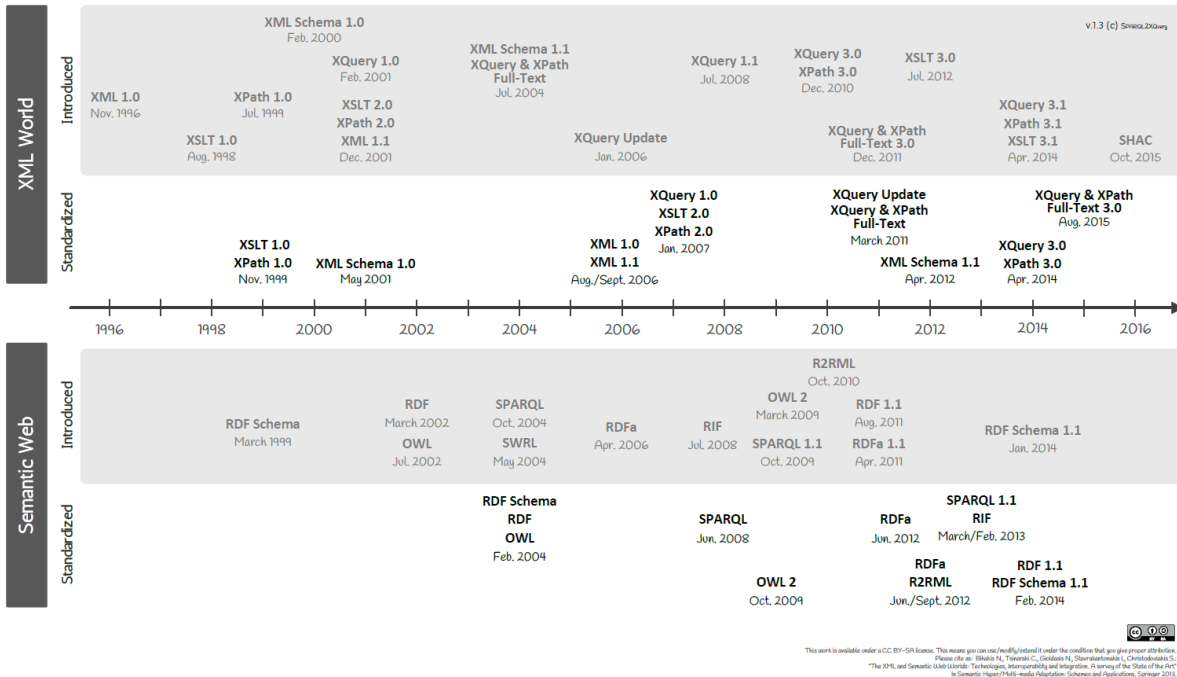
2.3 SPARQL

On 2008, SPARQL 1.0 became an official W3C Recommendation and SPARQL 1.1 in 2013. SPARQL is an RDF query language, able to retrieve and manipulate data stored in RDF format. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium, and is recognized as one of the key technologies of the semantic web (Harris et al., 2013).

2.4 GeoSPARQL

In the cloud of Linked Data there are many types of data, such as dates, values, names and places. Some of those data however, might refer to features that also have a spatial di-

Figure 3: Timeline of W3C Standards



<http://www.dblab.ntua.gr/bikakis/XMLSemanticWebW3CTimeline.pdf>

mension. An example would be TU Delft, which is located in Delft, is called "Delft University of Technology" and has a geometry e.g. a position, as shown in the figure 4.

Figure 4: Query for TU Delft

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?ConceptGeom where {?Concept dbo:city dbr:Delft;
rdfs:label "Delft University of Technology"@en;
geo:geometry ?ConceptGeom.
} LIMIT 100
```

ConceptGeom
"POINT(4.3724999427795 52.001667022705)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>

Features, as "TU Delft", with spatial dimension started populating the cloud of Linked Data, hence geospatial extensions of SPARQL have been developed to help manage them. Such extensions are stSPARQL and GeoSPARQL. GeoSPARQL has been published as an official OGC standard in 2012 (Perry and Herring, 2012). It defines classes, properties and extension functions that can be used in SPARQL queries with features that have spatial dimension (Koubarakis, 2013). On 2016 (W3C, 2016) future developments for GeoSPARQL were published that were aiming to give GeoSPARQL more expressive properties and allow handling of more geometry types, while making it more simple and intuitive. The GeoSPARQL comprises several different components.

- Core component defines top-level RDFS/OWL classes for spatial objects.
- Topology vocabulary component defines RDF properties for asserting and querying topological relations between spatial objects.
- Geometry component defines RDFS data types for serializing geometry data, geometry-related RDF properties, and non-topological spatial query functions for geometry objects and defines topological query functions.
- RDFS entailment component defines a mechanism for matching implicit RDF triples that are derived based on RDF and RDFS semantics.
- Query rewrite component defines rules for transforming a simple triple pattern that tests a topological relation between two features into an equivalent query involving concrete geometries and topological query functions.

According to GeoSPARQL's wiki page, there is (almost) no complete implementation of GeoSPARQL, but some partial or vendor implementations. Namely:

1. Apache Marmotta
2. Parliament
3. Strabon
4. OpenSahara uSeekM IndexingSail Sesame Sail plugin
5. GraphDB
6. Stardog

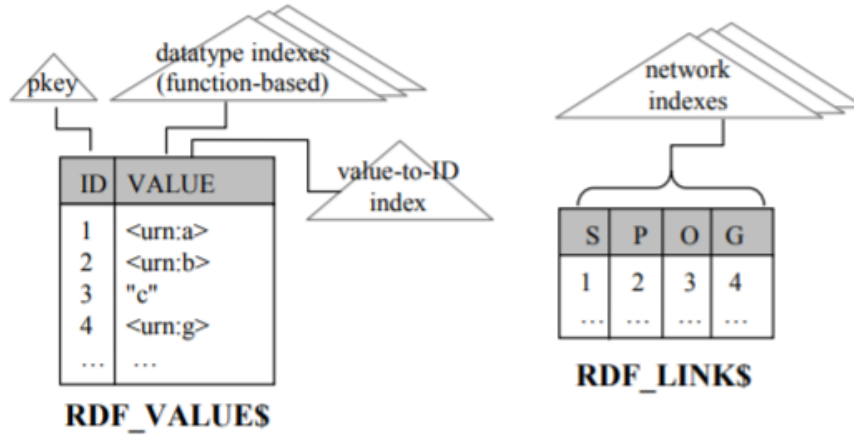
Those implementations, contain some or all GeoSPARQL classes, properties and extensions. Oracle Spatial and Graph, since version 12c, supports the full implementation of GeoSPARQL (Abhilakh Missier, 2015).

2.5 Oracle Spatial and Graph

Oracle Spatial and Graph is a database that supports large-scale Geographic Information Systems, and location-based services applications in the cloud. Oracle Spatial and Graph is a RDF triplestore that supports significant components of the OGC GeoSPARQL standard. Oracle Spatial and Graph uses well-known text serialization and Simple Features relation family to support the OGC GeoSPARQL standard conformance classes.

1. Core
2. Topology Vocabulary Extension (Simple Features)
3. Geometry Extension (WKT, 1.2.0)
4. Geometry Topology Extension (Simple Features, WKT, 1.2.0)
5. RDFS Entailment Extension (Simple Features, WKT, 1.2.0)

Oracle Spatial and Graph uses a relational schema to store RDF data and processes SPARQL queries by translating them to equivalent SQL queries that are executed by the parallel SQL engine in ORACLE Database. A simplified Version of ORACLE's relational schema for RDF data consists of an *RDF_LINK* table and an *RDF_VALUE* table. The *RDF_VALUE* table stores and ID to lexical value mapping for RDF terms and the *RDF_LINK* table stores 4-tuples of IDs representing subject, predicate, object and graph Perry et al. (2015). This basic schema is as shown in the Figure 5a, while in the Figures 5b and 5c, an example of it, is shown.



(a) Relational Schema RDF

1	<http://www.example.org/family/Cathy>	<http://www.example.org/family/sisterOf>	<http://www.example.org/family/Jack>
2	<http://www.example.org/family/Cindy>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.example.org/family/Female>
3	<http://www.example.org/family/Jack>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.example.org/family/Male>
4	<http://www.example.org/family/Tom>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.example.org/family/Male>
5	<http://www.example.org/family/siblingOf>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#Property>
6	<http://www.example.org/family/parentOf>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#Property>
7	<http://www.example.org/family/Person>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2000/01/rdf-schema#Class>
8	<http://www.example.org/family/sisterOf>	<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>	<http://www.example.org/family/siblingOf>
9	<http://www.example.org/family/brotherOf>	<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>	<http://www.example.org/family/siblingOf>
10	<http://www.example.org/family/motherOf>	<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>	<http://www.example.org/family/parentOf>
11	<http://www.example.org/family/fatherOf>	<http://www.w3.org/2000/01/rdf-schema#subPropertyOf>	<http://www.example.org/family/parentOf>

(b) The *RDF_VALUES* Table

1	1 [MDSYS.SDO_RDF_TRIPLE_S]
2	2 [MDSYS.SDO_RDF_TRIPLE_S]
3	3 [MDSYS.SDO_RDF_TRIPLE_S]
4	4 [MDSYS.SDO_RDF_TRIPLE_S]
5	5 [MDSYS.SDO_RDF_TRIPLE_S]
6	6 [MDSYS.SDO_RDF_TRIPLE_S]
7	7 [MDSYS.SDO_RDF_TRIPLE_S]
8	8 [MDSYS.SDO_RDF_TRIPLE_S]
9	9 [MDSYS.SDO_RDF_TRIPLE_S]
10	10 [MDSYS.SDO_RDF_TRIPLE_S]
11	11 [MDSYS.SDO_RDF_TRIPLE_S]

(c) The *RDF_LINK* Table

Figure 5: An example of Linked Data stored in Oracle Spatial and Graph Database

It is important to note that in Oracle Spatial and Graph, from the spatial functions that are to be developed in GeoSPARQL, only Generalization has not been implemented yet.

2.6 Spatial Functions

Regarding the Spatial functions, with the exception of Generalization, all the other functions are available with general solutions on Wikipedia. Moreover, the formulas for handling the Area and Centroid of polygons has been described by Bourke (1988), however it doesn't work for complex polygons. Regarding the Bounding box, Freeman and Shapira (1975) described methodologies that provide minimum bounding rectangles with different characteristics (area, perimeter etc.) Finally, Roussopoulos et al. (1995) described the process to find the K nearest features to a given feature.

Furthermore, the specifications of those functions can be found on ISO (a) and ISO (b)

3 Research questions

The main research question that will be investigated in the context of this graduation project is: **How to design an efficient Oracle GeoSPARQL implementation?**

To answer this question, a series of sub-questions should be answered:

1. Why the added GeoSPARQL functionalities have not been implemented yet?
2. What semantic information can be derived by the spatial functions in linked spatial data?
3. What benefits linked spatial data offer over traditional spatial data?
4. Can the semantics be utilized to improve the performance of the implementation?

As already noted, the speed of the functions and the queries will not be taken into consideration in this project.

4 Methodology

In order to answer the Research Question, we have to start with the sub-research questions. More specifically, the first sub-research question, *Why the added functionalities have not been implemented yet?*, will possibly indicate problems that this research will meet on its course. Answering it first can probably change the course of the research and it is one of the most important ones as it can help against pitfalls. Following, the second sub-research question, *What semantic information can be derived by the spatial functions in linked spatial data?*, will possibly lead to new semantic information that can add to a use case. Furthermore, by answering *What benefits linked spatial data offer over traditional spatial data?*, beyond the justification of using linked spatial data, that sub-research question aims to determine whether linked data and the corresponding information that can be deduced by the semantics can improve the efficiency of the implementation. Consequently, the last sub-research question, *Can the semantics be utilized to improve the performance of the implementation?*, will be more interesting in the case of larger data volumes, where functions would require a considerable amount of time. Hence, in addition to any other technique that might be implemented, filtering with semantics will possibly reduce the workload.

Beside answering those sub-research questions, in order to answer the Research question, ORACLE software will be used because with its Oracle Spatial and Graph, it supports Linked data with GeoSPARQL. Moreover, almost all spatial functions have been implemented in it. It allows user-defined functions with SQL or Java, thus it will make the development easier.

The developed implementation will be tested on data both created for that purpose and with data from a real use case and the results are going to be presented after the proper fixes

have been implemented. The real data, will come from LinkedGeoData or from any interested party. LinkedGeoData uses data that are collected by the OpenStreetMap project and publishes them as Linked Data.

An example use case would be a server that is used for determining where in Delft is allowed to build a certain building eg. a 10-floor mall. In that case, BGT data (contains detail maps of The Netherlands), BAG data (contains information about all addresses and buildings in The Netherlands) and urban laws should be joined together. Initially, selecting only data in Delft, then selecting lots that allow such a building to be build (have the proper area and side size) and that have a side in a large road (plot touches a main road).

5 Time planning

1. P2 Retake 09/11
2. Collection and linking of the data
3. Design of implementation.
4. Testing with mock data.
5. Fix of problems encountered.
6. P3-Midterm
7. Use functions on real data.
8. Documentation of findings and publishing of code.
9. P4 04/Dec-12/Dec
10. Documentation of findings and publishing of code. CONT.
11. Write of thesis.
12. P5 24/Jan-02/Feb

References

- Stefan Kulk and Bastiaan van Loenen. Brave new open data world? *IJSDIR*, 7:196–206, 2012.
- Council of the European Union European Parliament. Directive 2007/2/ec establishing an infrastructure for spatial information in the european community (inspire), 2007. URL <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:en:PDF>.
- Tim Berners-Lee, J Handler, and Ora Lassila. The semantic web. *Database and Network journal*, 36(3):7, 2006.
- Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1999.
- Christian Bizer, Freie Universität Berlin, Germany, Tom Heath, Talis Information Ltd, United Kingdom, Tim Berners-Lee, Massachusetts Institute of Technology, and USA. Linked data - the story so far. Technical Report context, 2009.

- Matthew Perry and John Herring. Ogc geosparql-a geographic query language for rdf data. *OGC Implementation Standard*. Sept, 2012.
- James Hendler; Ora Lassila; Tim Berners-Lee. The semantic web. 2001.
- Tim Berners-Lee. Tim berners-lee on the next web. *TED video*, 16:17, 2009.
- Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. Sparql 1.1 query language. *W3C recommendation*, 21(10), 2013.
- George Garbis; Kostis Kyzirakos; Manolis Koubarakis. Geographica: A benchmark for geospatial rdf stores. In H. Alani et al., editors, *Geographica: A Benchmark for Geospatial RDF Stores*, volume II of *Part*, page 8219. Springer, 2013.
- W3C. Further development of geosparql, 2016. URL https://www.w3.org/2015/spatial/wiki/Further_development_of_GeoSPARQL.
- GM Abhilakh Missier. Towards a web application for viewing spatial linked open data of rotterdam. 2015.
- Matthew Perry, Ana Estrada, Souripriya Das, and Jayanta Banerjee. Developing geosparql applications with oracle spatial and graph. In *SSN-TC/OrdRing@ ISWC*, pages 57–61, 2015.
- Paul Bourke. Calculating the area and centroid of a polygon. *Available from WWW: <http://paulbourke.net/geometry/polygonmesh>*, 1988.
- Herbert Freeman and Ruth Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.
- Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest neighbor queries. In *ACM sigmod record*, volume 24, pages 71–79. ACM, 1995.
- PNEN ISO. 19125-1. *Geographic Information—Simple Feature Access—Part, 1, a*.
- PNEN ISO. 19125-2. *Geographic Information—Simple Feature Access—Part, 2, b*.