ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse116932020.pdf
https://doi.org/10.30534/ijatcse/2020/116932020

Semantic Oriented Data Modeling for Enterprise Application Engineering Using Semantic Web Languages

Khadija Alaoui[1], Mohamed Bahaj[2]
LITEN Lab, Faculty of Sciences and Techniques
Hassan I University, Settat, Morocco
[1]alaoui_khadija@outlook.com
[2]mohamed.bahaj@gmail.com

ABSTRACT

This article proposes the use of the RDF, RDFS and OWL as a basis for data modeling for semantic oriented development of enterprise applications. RDF and OWL ware introduced to guaranty a semantic web to permit a web of interconnected data instead of a static web so that computer can explore web contents. Several efforts have been made to establish rules to convert data of existing enterprise applications into equivalent RDF/OWL data, to make them available on the semantic web. However, the most applications were developed using classical relational database modeling methodologies.

Here beyond publishing data for the semantic web, we give motivations for the necessity of a RDF/OWL semantic data modeling, as well as an approach in form of best practices for this modeling. We also show the effectiveness of our approach by comparing it to the classical relational data modeling.

Key words: Data modeling, RDF, RDFS, OWL, SPARQL, Data integration.

1. INTRODUCTION

Since their creation, the ontology languages RDF (Resource Description Framework), RDFS (RDF Schema) and OWL (Web Ontology Language) have been mainly considered as standards of the semantic web ([3], [5]). Furthermore, numerous researches on RDF have been performed to develop the semantic web and store and query RDF/OWL data.

The aim of this paper is to propose and encourage the utilization of RDF and OWL enterprise applications data modeling as a basis for software engineering, which would help to ensure the semantic oriented software development processes. Our present work constitutes an extended version of our conference paper [13] related to this semantic proposition of data modeling.

Since many years, the development of the enterprise applications was build based on relational databases (RDB) systems and RDB oriented data modeling, hence almost all classical solutions to enterprise data modeling are directed to the use of RDBs. However, many factors such as the use of RDF for data representation in various domains and the maturing of RDF and OWL technologies are launching an alert, indicating that the time has come to think about semantic oriented data modeling based on RDF and OWL. The adoption of RDF for data representation in various domains and the maturing of RDF and OWL technologies are the main examples of such factors.

In this regard, we are initiating and providing a valuable starting modeling proposal reference in the world of semantic oriented software engineering that targets performance in enterprise applications towards efficient semantic information conception, retrieval, and manipulation. More precisely, we propose the adoption of RDF and OWL as means for software engineering data modeling and we provide a modeling approach to this end. For this purpose, we give motivations for a semantic data modeling, taking into consideration some aspects related to data management and ontological applications. Furthermore, in our approach we show how RDF and OWL can be used for architecting data to give data more value. The approach handles relevant tasks related to semantics identification, data interoperability, graph modeling, model reuse, reasoning, semantic documentation, semantic validation, scalability and multi-modeling as well as semantic oriented data integration. Under the proposed approach, we also give some advantages of the proposed semantic oriented approach in comparison with the relational oriented approach.

The sections of this paper are structured as follows. Section 2 reviews the web ontology languages RDF, RDFS and OWL. Section 3 considers research works related to data modeling. In sections 4 we motivate the semantic oriented data modeling approach. In Section 5, we introduce the elements of our suggested approach for an RDF/RDFS and OWL oriented data modeling. Section 6 highlights the issues of the existing related research works and we compare our approach with the relational one. Finally, Section 7 concludes this paper.

2. RDF, RDFS and OWL

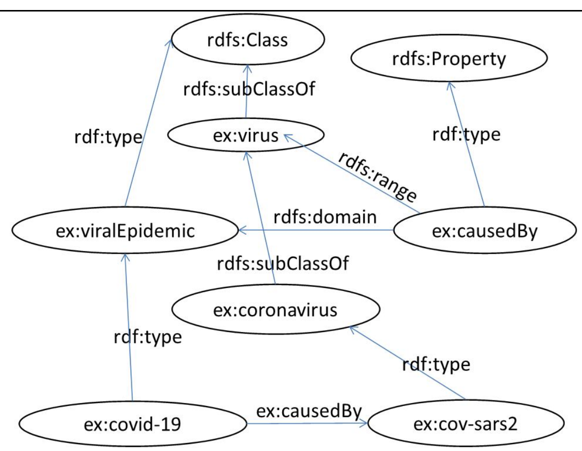
Introduced by the World Wide Web Consortium (W3C), RDF, RDFS and OWL are the main languages proposed for the realization of the semantic web with its primary objective to favor linking data with each other ([3]-[5]).

2.1 RDF and RDFS

In RDF, the information on resources and their relationships is modeled in form of triples (S, P, O). S stands for subject, which is a resource. O stands for object, which can be either a resource or a literal and p for predicate that defines the relation between the subject and the object. RDF has several serialization formats to represent it. The common ones include N-Triples, Turtle, XML, N3 (Notation 3) and JSON formats.

RDF uses various other constructs such as the type "Bag" to represent groups of resources or literal values without order, the type "Seq" to represent ordered groups of resources or literal values, and the type "List" to represent ordered lists of resources or literal values.

Figure 1 gives an RDF/RDFS example illustrating the relationship between the corona-virus "cov-sars2" and the epidemic "covid-19" with their associated classes. In this example we have the triples (ex:virus, rdfs:subClassOf, rdfs:Class), (ex:viralEpidemic, rdf:type, rdf:class), (ex:causedBy, rdf:type, rdfs:Property), (ex:causedBy, rdf:domain, ex:viralEpidemic), (ex:causedBy, rdf:range, ex:virus), (ex:coronavirus, rdfs:subClassOf, ex:virus), (ex:cov-sars2, rdf:type, ex:coronavirus), (ex:covid-19, rdf:type, ex:viralEpidemic) and (ex:covid-19, ex:causedBy, ex:cov-sars2).



(a) Graphical presentation

ex:virus rdfs:subClassOf rdfs:Class .
ex:viralEpidemic rdf:type rdf:class .
ex:causedBy rdf:type rdfs:Property .
ex:causedBy rdf:domain ex:viralEpidemic .
ex:causedBy rdf:range ex:virus .
ex:coronavirus rdfs:subClassOf ex:virus .
ex: cov-sars2 rdf:type ex:coronavirus .
ex:covid-19 rdf:type ex:viralEpidemic .
ex:covid-19 ex:causedBy ex:cov-sars2 .

(b) N3 notation

Figure 1: Example of RDF/RDFS triples

RDFS represents the meta-language for the concrete RDF data [4]. RDFS meta-models are also written in RDF as triples with information on the types and classes of the RDF data model and associated relationships. To this end, RDFS uses various concepts such as "Resource", "Literal", "Class", and "Datatype" and relationships such as "subClassOf", "subPropertyOf", "domain", and "range". RDFS uses those relationships to provide a way to organize the concepts and relationships in an RDF model into hierarchies. These hierarchies can be used to infer new information on the relation between data resources and therefore also new data triples.

2.2 Web Ontology Language OWL

Though RDFS supports ontology specifications, but it does not solve all possible requirements. To remedy to this limitation of RDFS expressiveness, the Web Ontology Language OWL has been developed [5]. OWL is built on top of RDFS with a richer vocabulary. It offers properties and classes for describing entities. Classes represent an abstraction mechanism for grouping instances with similar characteristics (properties). To relate a resource or an instance to another, OWL uses "ObjectProperty" while to relate instances of classes and literal values, OWL uses "DatatypeProperty".

OWL also offers other constructs for ranges and domains such as "ComplementOf", "AllValuesFrom", "SomeValuesFrom", "DisjointWith" and constructs for properties such as "InverseFunctionalProperty", "SymmetricProperty", "TransitiveProperty" and "FunctionalProperty".

OWL also makes it possible to generate new data types from existing ones with the help of constructs like "DataUnionOf", "DataComplementOf" and "DataIntersectionOf".

2.3 SPARQL Query Language

SPARQL is a recursive acronym standing for SPARQL Protocol and RDF Query Language. It represents the querying language of RDF data [6]. SPARQL is an official W3C recommendation that provides a standard format for writing queries that target RDF data.

SPARQL has four query forms: SELECT, CONSTRUCT, ASK and DESCRIBE [7]. The SELECT form returns variables and their bindings in a query pattern match. The CONSTRUCT form returns an RDF graph. The ASK form returns a Boolean value confirming whether a query has a solution or not. The DESCRIBE form that Returns an RDF graph containing data about the resources found. SPARQL can also perform complex joins of various RDF local and remote sources in the same query and put the results together.

3. RELATED WORK

With respect to data modeling using RDF and OWL most research works have mainly dealt with transforming relational models or UML models into RDF/OWL ontology models (e.g., [15], [53], [45], [8], [26]) in order to make data available on the semantic web. In fact, most existing works dealing with ontologies framework, address mainly the topic of how to create ontologies for specific domains in order to publish data in the semantic web, without consideration of the context of enterprise applications data modeling. The authors of [33] provided a review of the basic works in this direction. In [36] a discussion from a business perspective for the transition of data to the semantic web can be found.

To transform ontological data of the semantic web to relational models, other works ([16], [12], [19]) have treated reverse mapping. Reference [47] deals with ontological modeling, however, it only focuses on how the ontology can help in building object models.

With respect to software engineering, ontologies have been mainly used to model the controlling of activities in the development process but not for enterprise applications data modeling. Existing research works in this field (e.g., [31], [21], [54]) addressed indeed the process of the traceability of the software engineering tasks through control ontologies defining the process tasks or of the traceability issues related to components related to the development process such as documents and decisions.

4. MOTIVATIONS FOR SEMANTIC ORIENTED DATA MODELING

Due to the huge use of relational databases (RDB) over the last decades, the modeling of applications data has been influenced by RDB to make it relational oriented. The entities and the relations with attributes are used to describe the various domain elements and the relations between them. This was principally influenced on one hand by the availability of associated relational database management systems (RDBMS) and the vast development of those systems over several years, which allowed them efficient handling of data. On the other hand, the development of various tools and programming APIs for the interaction with such systems has also encouraged the adoption of such a relational data modeling.

In the following, we give some motivations of our idea of proposing a semantic oriented data modeling based on the semantic languages RDF and OWL instead of a relational oriented method.

4.1 RDF Acceptance

During the last years RDF and OWL have found a great acceptance for the use in various domains (e.g., geology [1], [39], smart cities [18], biology, biomedicine and medicine [40], [35], governmental data [2], academic social networks [22], DBpedia [9], [20], tourism [41], [48], collaborative knowledgebase [50], …).

4.2 Diversity of RDF Management Systems

Evidently, such a huge use has been accompanied by the development of RDF/OWL data management systems, namely the triplestores. They have been developed not only for the storage and processing of the data, but also with other tools and APIs for the presentation of RDF/OWL data and the interaction with other triplestore. We refer to [14] for a detailed information on the categories of RDF triplestores.

The list of existing triplestores is long and contains various types of stores ranging from general purpose triplestores [23], NoSQL triplestores [46], P2P triplestores [42] or Cloud triplestores [34]. Also such triplestores range from relational based ones (e.g., [11]) to native ones (e.g., Triplebit [51]) and also from centralized triplestores (e.g., Triplebit [51]) to distributed triplestores (e.g., Virtuoso Cluster Edition [Z15], Trinity.RDF [52], Triad [28]).

Also, various Big Data RDF stores have been developed to propose management solutions of large amounts of RDF data. Such Big Data RDF stores are mainly based on the HDFS file system for data storage and on Map Reduce framework for SPARQL querying of RDF triples. A nice presentation of MapReduce meta-model is given in [27].

Examples of such solutions are those relying on Hadoop such as PRESTO-RDF [37], SHARD [43], HadoopRDF [32] and CliqueSquare [29], and those relying on Spark such as SPARQLGX [30], S2RDF [44], SPARQL-Spark [38], PRoST [24] and TripleRush [49].

Reference [17] gives a comparison of some of Big Data along with some NoSQL RDF management systems.

Another factor that favored the developing of RDF Big Data triplestores is the development during last years of various analytical tools and technologies. We refer to [25] for a review on such tools and technologies.

The aforementioned variety of RDF/OWL data management systems makes a data modeling based on RDF/RDFS and OWL possible in all application domains and for every kind of data.

4.3 RDF Data Tools for Data Modeling

The evolution of semantic web standards has also been accompanied with various associated tools for editing and browsing of RDF schemas and RDF data graphs visualization.

Such tools are among others especially important when dealing with large amounts of data.

The visualization of RDF/RDFS data is of course of great importance. Since it is providing end users with data in a visual context that lets them better understand and deal with the models under hand. Among the tools we cite D3SPARQL,

Protégé, OntoEdit and RDF Instance Creator (RIC).
Also, some triplestores like NoSQL based triplestores have generally their own visualization APIs.

4.4 RDF Programming APIs

Data modeling is generally conducted to provide users with systems and APIs that allow them to organize and handle their processes. In this sense, a great number of triplestores have been developed that provide various programming interfaces for the interaction with RDF data and schemas.
The most known and used one is Jena. Jena is a java-based API for RDF which allows the creation and manipulation of RDF graphs. It supports XML and N-Triples RDF formats and represents RDF graphs as "model".
As another example of a java-based API for handling and processing RDF data and supporting the query language SPARQL is Eclispe-rdf4j. RDF4J supports most RDF file formats, such as XML, Turtle and N-Triples.
There are also other APIs that provide support for other programming technologies. For example, for accessing RDF data from Ruby programs we have is the library ActiveRDF [10].

4.5 Necessity for Semantic Data Modeling

Due to the enormous use of RDF and OWL in several application fields and the progressive development of the storage, presentation and manipulation of RDF and OWL data, we motivate the idea that semantic data modeling should not be limited to publish such data, as it was the aim of the semantic web, but also to develop semantic enterprise applications. A semantic oriented data modeling is of great importance to get knowledge extremely beneficial by making it semantically rich enough to let it express itself through machines, between users and within applications. Within a group, users and machines can share and use the same ontological sets in direct way and without any transformations. Also, various groups can interact with their data models in an easy way and extract semantic information because of a semantic navigation through the data.
For the multiple users groups that have adopted or will adopt RDF and OWL to represent their data, it is of great importance to help them to successfully develop the software they need in an efficient way with a developing process that respects their semantic objectives. This is of crucial role in order to avoid all the problems that the world faceted during the software crisis. To all the challenges of a general software development process other challenges with regards to the semantics have to be considered to add a semantic value with a complete knowledge modeling that is concept and ontological oriented in order to get a software that can also semantically interact with its users.

5. APPROACH FOR A SEMANTIC ORIENTED DATA MODELING

In this section we provide a context for best practices to achieve the semantic oriented data modeling we are proposing.

5.1 Semantic Oriented Analysis and Conception (AOC)

For a semantic oriented data modeling it is necessary to be closer to the targeted domain users and to consider their needs in the obtained models in a way that is understandable by these users and let them interact with such models in an easy and flexible way. Models ought to speak semantics and enable users and machines to be able to follow such semantics.
The main strategies for the proposed semantic oriented analysis and conception are given in the following.

A. Semantics Identification

For the data modeling using RDF or OWL, the focus during the data analysis, should be on the extraction of the semantics within the data. The principal goal is to come up with RDF data models encapsulate semantics. Since RDFS and OWL propose a formal description of data that is independent from implementations, the most relevant task should be to identify the various data resources and the relationships between these resources and also the properties of each resource within the semantics imposed by the use case domain at hand. Meaning of the words will be easily inferred from the context of the ontology or graph within which they are defined.

B. Focus on Semantic Data Interoperability

We also include the notion of semantic data interoperability to describe the interconnection between the data that is semantically oriented. This data interoperability for our semantic oriented approach is to be associated with a graph-oriented conception using RDF and OWL.

C. Graph Modeling

For of the interlinking between the different data pieces RDF oriented graph modeling of (subject, predicate, object) triples with nodes representing subjects or objects and arcs between nodes representing predicates, has the advantage that the information management, retrieval and the addition of new components proceed easily. Additionally, due to graph operations, such as path search, graph isomorphism, graph matching and connectivity check, the graph exploration for a semantic navigation and information extraction is more efficient.

D. Model Reuse

Existing models could be reused. In fact, ontologies include relatively general information that can be reused by different types of applications or tasks.

E. Reasoning over Models

Since RDFS and OWL support reasoning, information that can be inferred through it can be omitted and simply modeled using appropriate modeling elements such as hierarchies between classes. Due to the fact that implicit semantics can be figured out from existing ones, the reasoning capacity over models allows avoiding redundancy. This allows new triples to be simply inferred from existing ones through reasoning rules.

F. Use of Analysis Tools

For the analysis and conception of models various existing user-friendly editors could be used for editing, storing, and manipulating of RDF schemas and RDF data models. Specialized RDF editors, XML or graph tools could be used. XML editors: verify the syntax with respect to XML structure of the elements. They could be extended to make verification according to the RDF rules. Through the graphical presentation of schema and data models, graph visualization tools and editors allow a better control of the dependences between their components, the comparison of models and the identification of errors. Indeed, since the content of RDF schema can be large and needs to be stored in many files, the graphical tools offer better control. Partial or total graph or diagram presentations allow better insight on the structure of the various elements and the interrelationships between them as well as better mastering of their complexity. Schemas can be then generated from the edited graphic models. Such graphical models could also be used as a communication mean between stakeholders.

G. Semantic Oriented Documentation

Semantic oriented documentation, the semantic reasons of considering each graph or subgraph schema are to be documented in detail. This is to be performed with respects to the end users and to the associated domains or subdomains.

H. Semantic Validation of the Models

The compliance of data models is to be validated through interactions with users and through inference tests, in order to grant data consistency and users satisfaction.

5.2 Scalability and multi-modeling

The data modeling should be scalable with respect to big data volumes and data processing.

A. Semantic Distributed Modeling

The scalability is favorized by the graph modeling approach. In fact, the distribution of the data can be easily performed, due to the graphs partitioning ability. Multi-modeling can also be used to arise with linked schema models.

B. Modeling for Cloud, P2P and Big Data

The above data modeling methodology with RDF and OWL can be adopted for the use in peer to peer or cloud based applications without any problems if the generated data are well managed through associated data management systems. As a solution for cloud or P2P data management systems we cite for example those presented in [34] and [42]. The same arguments are also valid for the case of big data for which triple stores based on big data technologies such as Hadoop and Spark could be used.

5.3 Semantic Oriented Data Integration

Nowadays, it is inevitable to be able to handle huge amounts of data resulting from various data sources. It is necessary that different applications including cloud, mobile and big data applications can interact with each other and be able to exchange data for an optimum use.

A complete semantic data modeling art should also provide ways for better semantic data integration. The RDF and OWL models are best suitable for encapsulating the heterogeneity of data from different sources. The integration of data from different sources necessitates the consideration of multiple aspects, i.e., the deal with the used vocabulary and the identification of common vocabulary compared to the existing one. This could conduct to either create new graphs or subgraphs or just to insert the new data into existing graphs.

As stated, nowadays, we have mixed data sources, such as relational, XML, NoSQL, etc., this requires dealing with various data types and formats. So, the heterogeneity of the data is another aspect to consider. Within the semantic contexts, the decisions should be made in a way to integrate the new models in the global ones. For each type and format of data it is then preferable to have a conversion platform to explicitly extract both its RDFS schema and associated data graphs in order to handle the associated data characteristics and structure. Existing mapping and conversion methods and tools could be used for this aim.

6. COMPARISON WITH CLASSICAL RDB ORIENTED DATA MODELING

In this section we compare the semantic-oriented data modeling based on RDF and OWL with the classical oriented data modeling and we give a summarization of their main advantages.

6.1 The Classical RDB Oriented Data Modeling

Classical data modeling is directed towards the use for relational databases. In this model, the requirements are analyzed with the aim to extract a conceptual data model and to identify the entities with their attributes and relationships between them.

A logical data model based on the underlying database system is then constructed from the conceptual data model and is transformed to a physical data model to be implemented.

Those sequences of transformations drive relational database-oriented modeling to be inept for semantic-oriented modeling. Through conversions a loss of semantic information can be caused. Furthermore, there is a risk to get the data semantics lost, because of the restrictive relational rules imposed on relational models. Subsequently, RDB data models are neither flexible nor adequate to web semantic objectives.

6.2 Advantages of a Semantic Data Modeling Approach

For semantic-oriented data modeling, graph-oriented modeling art based on RDF and OWL is of great importance. Compared to relational based models, data modeling based on RDF and OWL is more expressive and grant more flexibility. The users can easily set up or expand their models based on the semantics of the components they want to model. This is done without the need of any restriction or transformation. RDF and OWL based modeling allow the capture of all user requirements accurately with more focus on the nature of the data aspects. Also, this modeling art is implementation independent; hence no technical concerns or transformations are required. Moreover, the expansion and comparison of the models can be done easily.

Other advantages inferred from the previous section are summarized as follows.

A. Powerful data modeling with RDF, RDFS and OWL

The data model forms the basis for discussions with clients and users as well as between developers.

It should therefore be as simple and expressive as possible in order to be easily understandable, manageable and extensible. Such requirements are of course better guaranteed through the proposed RDF modeling approach which focuses on identifying the semantics of the real world and their representation as RDF resources with properties modeling relationships between these entities in form of easily navigable graphs. Such an approach has the following advantages:

- Power of RDF/OWL for explicit semantics formulation where the meaning is simply inferred from this order (S, P, O).
- Fast and easy data modeling through the direct expression of users formulations simply using (Subject, Predicate, Object) triples.
- Efficient model navigation through easy browsing of RDF/RDFS graphs.
- SPARQL search and querying of schema models is possible since these models are also formulated in form of (S,P,O) triples using RDF.
- Redundancy is avoided while adding new schema triples into the schema models.
- Conceptual models are also logical models that most existing RDF triplestores can directly upload. So, neither intermediary models nor database designers are needed.

B. Users oriented data modeling

- RDF and OWL semantic oriented modeling offer the opportunity towards novel enterprise applications and services that better reflect users thinking.
- The semantic modeling based on graphs allows model reuse and distribution. Also, graphs are more expressive and efficient information retrieval is possible by using the multiple existing graph algorithms relative to graph operations such as graph matching or graph isomorphism.
- Visualization with existing RDF tools allows easy browsing of RDF schema constructs.
- Interaction with end users is always possible by simply using the schema RDF graphs.

C. Interoperability and serialization

- RDF models and data can be presented in N3 formats or also XML format which makes them compliant with many existing tools and programming languages, as well as for serialization.
- Various mapping and conversion techniques and tools have been developed during the last decade for the conversion or mapping of other data sources such as XML, UML or relational ones. Integrating other data sources within existing semantically developed models can be performed in an easy and flexible way.
- As already mentioned, interaction with programming languages such as java, Perl and others offers further possibilities for the interaction with RDF data or schemas. These capabilities are of course by far of great importance and unfortunately, they exist in a limited for E/R-modeling.
- The data interoperability through interlinking data within models gives more semantic value to each data property and components.
- Semantic modeling makes information retrieval easy with the use the numerous existing graph algorithms relative to graph operations such as graph matching or graph isomorphism.
- RDF framework allows the processing of the huge amounts of data issued from P2P, cloud or Big Data systems. Such modeling favors a better insight of the systems under hand and leverage the art the research is done and the way to act based on domain understanding and user expectations.

7. CONCLUSION

In this paper we suggested and motivated the use of RDF, RDFS and OWL for a semantically oriented software engineering data modeling, to ensure better enterprise application development. It is for the first time that such a novel idea is proposed. We identified key factors that motivate such use for the modeling of the variety of data produced by multiple services in our daily life instead of the classical entity-relationship modeling approach in order to come up with smart services and intelligent answers to the challenges posed while architecting of such data.

We also gave an approach with best ways to excel in semantically architecting data taking into account several aspects with respect to semantics identification, data interoperability, graph modeling, model reuse, reasoning, semantic documentation, semantic validation, scalability and multi-modeling as well as semantic oriented data integration. We also showed how an RDF/RDFS and OWL based data modeling provides better and sufficient conditions to get rich semantic data models in comparison of the classical relational data modeling. Our approach will help data engineers and enterprise applications developers to treat their data semantically. It will also encourage the struggling implementation of the semantic web through the use of applications directly based on RDF and OWL without any data conversion from other data models into RDF/OWL ones. We are currently working on extending our work to come up with a global approach taking into consideration the use of RDF and OWL in all phases of a semantically oriented software development process starting from the inception phase to the delivery and maintenance one.

REFERENCES

1. http://linkedgeodata.org/
2. http://reference.data.gov.uk/
3. http://www.w3.org/RDF/
4. https://www.w3.org/TR/rdf-schema/
5. http://www.w3.org/TR/owl2-overview/
6. http://www.w3.org/TR/rdf-sparql-query/
7. https://www.w3.org/TR/rdf-sparql-query/#QueryForms
8. https://www.w3.org/2001/sw/rdb2rdf/wiki/Implementations
9. http://wiki.dbpedia.org/services-resources/ontology
10. https://github.com/ActiveRDF/ActiveRDF
11. D. Abadi, A. Marcus, S. Madden, and K. Hollenbach. **SW-Store: a vertically partitioned DBMS for Semantic Web data management**, *VLDB J.* 18(2), 2009. https://doi.org/10.1007/s00778-008-0125-y
12. Z2 H. Afzal, M. Waqas, and T. Naz. **OWLMap: Fully Automatic Mapping of Ontology into Relational Database Schema**, *Int. J. of Advanced Computer Science and Applications*, 7(11), pp. 7-15, 2016. https://doi.org/10.14569/IJACSA.2016.071102
13. K. Alaoui and M. Bahaj. **Semantic Oriented Data Modeling Based on RDF, RDFS and OWL**, in Proc. *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019), Volume 4 - Advanced Intelligent Systems for Applied Computing Sciences*, M. Ezzyani (Ed.), Springer AISC 1105, pp. 411–421, 2020. https://doi.org/10.1007/978-3-030-36674-2_42
14. K. Alaoui. **A Categorization of RDF Triplestores**, in Proc. *Smart City Applications, SCA-2019*, October 2–4, 2019, Casablanca, Morocco, ACM, ISBN 978-1-4503-6289-4/19/10, DOI 10.1145/3368756.3369047, 2019.
15. L. Alaoui, O. El Hajjamy, and M. Bahaj. **RDB2OWL2: Schema and Data Conversion from RDB into OWL2**, *Int. J. of Engineering & Research Technology IJERT*, vol. 3, issue 11, 2014.
16. S. Bagui and J. Bouressa. **Mapping RDF and RDF-Schema to the Entity Relationship Model**, *J. of Emerging Trends in Computing and Information Sciences*, vol. 5, no. 12, 2014.
17. M. Banane and A. Belangour. An Evaluation and Comparative study of massive RDF Data management approaches based on Big Data Technologies, *International Journal of Emerging Trends in Engineering Research*, vol. 7, no. 7, July 2019. https://doi.org/10.30534/ijeter/2019/03772019
18. P. Bellini and P. Nesi. **Performance assessment of RDF graph databases for smart city services**, *J. of Visual Languages and Computing*, 45, pp 24–38, 2018.
19. S. Benslimane, M. Malki, and D. Bouchiha. **Deriving Conceptual Schema from Domain Ontology: A Web Application Reverse Engineering Approach**, *The Int. Arab J. of Information Technology*, 7(2), pp. 167-176, 2010.
20. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. **DBpedia - a crystallization point for the Web of data**, *J. of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009. https://doi.org/10.1016/j.websem.2009.07.002
21. C. Calero, F. Ruiz, and M. Piattini. ***Ontologies for software engineering and software technology***. Springer, 2006.
22. M. Challenger. **The Ontology and Architecture for an Academic Social Network**, *Int. J. of Computer Science*, issue 9, 1, pp 22-27, 2012.
23. T. Chawla, G. Singh, E. S. Pilli, and M. C. Govil. **Research Issues in RDF Management Systems**, in Proc. *ACM India Joint International Conference on Data Science and Management of Data CoDS-COMAD'19*, Kolkata, India, 2019, pp. 188-194.
24. M. Cossu, M. Färber, G. Lausen, **PRoST: Distributed Execution of SPARQL Queries Using Mixed Partitioning Strategies**, in *Proc. 21st International Conference on Extending Database Technology (EDBT)*, March 26-29, 2018.
25. A. Dhankhar and K. Solanki. **A Comprehensive Review of Tools & Techniques for Big Data Analytics**, *International Journal of Emerging Trends in Engineering Research*, vol. 7, no. 11, November 2019. http://www.warse.org/IJETER/static/pdf/file/ijeter257112019.pdf
26. O. El Hajjamy, K. Alaoui, L. Alaoui, and M. Bahaj. **Mapping UML to OWL2 Ontology**, *J. of Theoretical and Applied Information Technology*, vol. 90, no 1, 2016.
27. A. Erraissi, A. Belangour. **Meta-modeling of Big Data management layer**, *International Journal of Emerging Trends in Engineering Research*, Vol.7, No.7, pp.36-43, 2019. https://doi.org/10.30534/ijeter/2019/01772019
28. S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald. **Triad: a distributed shared-nothing rdf engine based on asynchronous message passing**, in Proc. *ACM SIGMOD*, 2014.
29. F. Goasdoué, Z. Kaoudi, I. Manolescu, J. Quiané-Ruiz and S. Zampetakis. **CliqueSquare: Efficient**

Hadoop-based RDF query processing, in Proc. BDA'13 - Journées de Bases de Données Avancées, Oct 2013, Nantes, France. 2013. <hal-00867728>, https://hal.inria.fr/hal-00867728/document, 2013.

30. D. Graux, L. Jachiet, P. Genevès and N. Layaïda. **SPARQLGX: Efficient Distributed Evaluation of SPARQL with Apache Spark**. in Proc. *ISWC 2016*, Groth, P. et al. (Eds.). LNCS, vol. 9982, pp. 80–87. Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-46547-0_9
31. H. J. Happel and S. Seedorf. **Applications of Ontologies in Software Engineering**, in Proc. *Second Int. Workshop Semantic Web Enabled Software Eng.*, 2006.
32. F. Husain, J. McGlothlin,, M.M. Masud, L. Khan and B. Thuraisingham. **Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing**. *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1312–1327, Sep. 2011.
33. R. Iqbal, M. A. A. Murad, A. Mustapha, and N. M. Sharef. **An Analysis of Ontology Engineering Methodologies: A Literature Review**, *Research J. of Applied Sciences, Engineering and Technology*, vol. 6, 2013.
34. Z. Kaoudi and I. Manolescu, **RDF in the clouds: a survey**, *The VLDB J.*, 24(1), pp 67–91, 2015.
35. K. Kochut and M. Janik. **SPARQLeR: Extended Sparql for Semantic Association Discovery**, in Proc. *4th European Semantic Web Conf.*, Innsbruck, Austria, 2007.
36. M. Lytras and R. García. **Semantic Web applications: A framework for industry and business exploitation - What is it needed for a successful semantic web based application**, *Int. J. of Knowledge and Learning*, vol. 4, no. 1, pp. 93-108, 2008. https://doi.org/10.1504/IJKL.2008.019739
37. M. Mammo, M. Hassan and S.K. Bansal. **Distributed SPARQL Querying over Big RDF Data Using PRESTO-RDF**, *Int. J. of Big Data*, vol. 2, no. 3, 2015.
38. H. Naacke, B. Amann and O. Curé. **SPARQL Graph Pattern Processing with Apache Spark**. in Proc. *Fifth Int. Workshop on Graph Data-Management Experiences and Systems, GRADES 2017*, pp. 1:1–1:7. ACM, New York, 2017.
39. G. J. Nalepa and W. T. Furmanska. **Review of semantic web technologies for GIS**, *Automatyka*, 13, 2, pp 485–492, 2009.
40. P. Rajbhandari, R. Gosai, R. C. Shah, and K. C. Pramod. **Semantic Web in Medical Infor-mation Systems**, *Int. J. of Advances in Engineering & Technology*, 5, 1, pp. 536-543, 2012.
41. F. Ricca, G. Grasso, S. Liritano, A. Dimasi, S. M. Lelpa, M. Manna, and N. Leone. **A Logic-Based System for e-Tourism**, *Fundamenta Informaticae*, 105, pp. 35–55, 2010.
42. G. Rizzo, F. Di Gregorio, P. Di Nunzio, A. Servetti, and J. C. De Martin. **A peer-to-peer architecture for distributed and reliable RDF storage**, in Proc. *IEE 1st international conference on networked digital technologies*, Ostrava, 28–31, New York, July 2009, pp.94–99. https://doi.org/10.1109/NDT.2009.5272090
43. K. Rohloff and R. E. Schantz. **High-performance, massively scalable distributed systems using the MapReduce software framework: the SHARD triple-store**, in Proc. *Programming Support Innovations for Emerging Distributed Applications*, pp.1-5, October 17-21, Reno, Nevada, 2010.
44. A. Schätzle, M. Przyjaciel-Zablocki, S. Skilevic and G. Lausen. **S2RDF: RDF Querying with SPARQL on Spark**, *VLDB*, pp. 804–815, 2016. https://doi.org/10.14778/2977797.2977806
45. D. E. Spanos, P. Stavrou, and N. Mitrou. **Bringing relational databases into the semantic web: A survey**, *Semantic Web J.*, 3(2), pp. 169–209, 2012.
46. K. R. Saikaew, C. Asawamenakul, and M. Buranarach. **Design and evaluation of a NoSQL database for storing and querying RDF data**, *KKU Engineering J.*, 41(4), pp. 537-545, 2014.
47. W. V. Siricharoen. **Ontology Modeling and Object Modeling in Software Engineering**, *Int. J. Software Engineering and its Applications*, vol. 3, no. 1, 2009.
48. F. Soualah-Alila, C. Faucher, F. Bertrand, M. Coustaty and A. Doucet. **Applying Semantic Web Technologies for Improving the Visibility of Tourism Data**, in Proc. *Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval ESAIR'15*, Melbourne, Australia, ACM, October 2015, pp. 5-10.
49. P. Stutz, M. Verman, L. Fischer, A. Bernstein. **TripleRush: a Fast and Scalable Triple Store**. in Proc. *9th International Workshop on Scalable Semantic Web Knowledge Base Systems*, Sydney, Australia, 21 October - 22 October, 2013.
50. D. Vrandecic and M. Krötzsch. **Wikidata: a free collaborative knowledgebase**, Communications of the ACM 57(10), pp. 78-85 • September 2014.
51. P. Yuan, P. Liu, B. Wu, H. Jin, W. Zhang, and L. Liu. **Triplebit: a fast and compact system for large scale rdf data**, in Proc. *VLDB Endowment*, 6 (7), May 2013, pp. 517–528. https://doi.org/10.14778/2536349.2536352
52. K. Zeng, J. Yang, H. Wang, B. Shao, and Z. Wang. **A distributed graph engine for web scale rdf data**, in Proc. *VLDB Endowment*, vol 6 Issue 4, 2013, pp. 265-276. https://doi.org/10.14778/2535570.2488333
53. F. Zhang, Z. M. Ma, L. Yan. **Construction of ontologies from object-oriented database models**, *Integrated Computer-Aided Engineering* 18, pp. 327–347, 2011.
54. Y. Zhao and J. Dong. **Ontology Classification for Semantic-Web-Based Software Engineering**, *IEEE Transactions on Services Computing*, vol. 2, no. 4, 2009. https://doi.org/10.1109/TSC.2009.20