Storing and Querying Ontologies in Relational Databases: An Empirical Evaluation of Performance of Database-Based Ontology Stores

Raoul Kwuimi
Department of Information and Communication Technology
Vaal University of Technology
Private Bag X021, Andries Potgieter Blvd,
Vanderbijlpark, 1900
Vanderbijlpark, South Africa
e-mail: kwuimi@gmail.com

Jean Vincent Fonou-Dombeu
Department of Software Studies
Vaal University of Technology
Private Bag X021, Andries Potgieter Blvd,
Vanderbijlpark, 1900
Vanderbijlpark, South Africa
e-mail: fonoudombeu@gmail.com

***Abstract*—Mapping ontologies to relational databases is an active research topic in Semantic Web. Therefore, several platforms have been developed to enable the storage and query of ontologies in relational databases. However, only a few studies have empirically measured and compared their performances in terms of speed and scalability. In this paper, two popular database-based ontologies stores, namely, Jena API and Sesame are used to load and query five selected ontologies of different sizes into MySQL relational database. Various metrics including (1) the loading times of ontologies into the relational databases, (2) the response times of SPARQL queries executed on the stored ontologies databases and (3) the sizes of the ontologies databases are used to measure and compare the performance of the two Semantic Web platforms. Experiments show that (1) both platforms are scalable and could successfully parse, load and query ontologies of different formats (OWL/RDF) and sizes into relational databases, (2) Jena API performs faster with small size ontologies, whereas, Sesame is more efficient with bigger size ontologies with regards to loading of ontologies into relational databases, (3) Sesame provides quicker responses to SPARQL queries compared to Jena API and (4) the disk space required to store the resulting ontologies databases in both platforms are proportional to the initial sizes of the ontologies and is higher in Jena API than in Sesame.**

Keywords-Jena API; Sesame; SPARQL; Ontology Storage; Relational Databases .

I. INTRODUCTION

The Semantic Web is an improvement of the current World Wide Web (WWW) in which web contents are represented on the basis of their meaning rather than web links as in the current internet. The meaning of web content are represented with ontology. Ontology as explained in [2] is a knowledge base system that contains a vocabulary of basic terms concerning a particular domain and semantic interconnections between those terms. It is the formal representation of data used on the semantic web. Several languages are used to represent ontologies in Semantic Web; they include Extensible Markup Language (XML), DARPA Agent Markup Language (DAML), Resource Description Framework (RDF), RDF Schema (RDFS) and Web Ontology Language (OWL) [3]. Two of those languages are widely used and recommended by the World Wide Web Consortium (W3C) including RDF/RDFS and OWL languages [1] [2] [4]. Ontology generated in these languages need to be persistently stored and used within Semantic Web applications.

In the Semantic Web domain, 3 techniques are used for ontologies storage, namely, (1) In-memory storage, (2) File or native storage, and (3) database storage. The in-memory storage is efficient only for small size ontologies, i.e., when the ontology has less instances or statements. It provides quick query response times because the ontology is residing in the main memory of the computer. When the ontology is large in size, persistent storage is appropriate as the ontology can no longer be stored in the main memory of the computer. Native storage makes use of files to store ontologies. The database technology has been used for more than 30 years [3]. In Semantic Web database storage is useful in many cases where storage is required on the web [5]. In fact, ontologies used in online systems today are of hundreds of Megabytes to thousands of Gigabytes in size; they need to be stored in relational databases for their efficient and optimal utilization [6] [7] [8].

Several platforms have been developed to enable the persistent storage and query of ontologies in relational databases. Relational databases are mostly used over object and object relational databases because, it provides performance, maturity, availability and reliability [43]; the most commonly used platforms are: AllegroGraph, Jena API, Open Anzo, Oracle Semantic [8], Minerva [18] [42] and Sesame [12]. Oracle semantic and AllegroGraph are currently available only in the form of trial versions [8]. Further, Open Anzo, AllegroGraph and Minerva do not process ontologies written in RDF syntax. Jena API and Sesame support both OWL and RDF ontologies as well as MySQL which is a widely used Relational Database Management System (RDMS) on the web. Further, Sesame and Jena API are both open source platforms and are accessible free of charge with full functions and supports. To date, only a few studies have empirically measured and compared their performances in terms of speed and scalability.

In this study, Jena API and Sesame are used to load and query five selected ontologies of different sizes into MySQL relational databases. Various metrics including (1) the loading times of ontologies into the relational databases, (2) the response times of SPARQL queries executed on the stored ontologies databases and (3) the sizes of the ontologies databases are used to measure and compare the performance of the two Semantic Web platforms. Experiments show that (1) both platforms are scalable and could successfully parse, load and query ontologies of different formats (OWL/RDF) and sizes into relational databases, (2)

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-420-6

Jena API performs faster with small size ontologies, whereas, Sesame is more efficient with bigger size ontologies with regards to loading ontologies into relational databases, (3) Sesame provides quicker responses to SPARQL queries when compared to Jena API and (4) the disk space required to store the resulting ontologies databases in both platforms are proportional to the initial sizes of the ontologies and higher in Jena API than in Sesame.

The rest of the paper is organized as follow. Section 2 discusses existing approaches for storing ontology on the Semantic Web. Characteristics of existing platforms for ontologies storage and query are presented in Section 3. Section 4 describes the experimental design of the study in terms of the dataset, performance metrics and tools employed. The last part of Section 4 presents and discusses the experiments and results of the study. Related studies are discussed in Section 5 and a conclusion ends the paper in Section 6.

II. ONTOLOGY STORAGE TECHNIQUES

Ontology storage is based on 3 main models (Figure 1). These include: (1) In-memory storage, (2) Native or File-based storage and (3) Databases-based Storage [9] [11]. In-memory or Memory-based storage uses the central memory of the computer to store ontologies. It is very efficient and fast with small scale ontologies. The drawback of this technique is that as the ontology get larger, it becomes more difficult to manipulate. In fact, ontologies stored using the in-memory storage technique need to be loaded in the memory every time a user wants to run an application that is using it. The native storage technique uses files to store ontologies. Ontologies statements are stored in triple store in the form of (S, P, O) where S is the Subject, P the Predicate and O the Object. The advantages of native storage are that data loading and data query are fast [37]. In order to retrieve data easily and quickly with fewer errors, index algorithms such as the B-tree or B+ [10] [37] are used. Structuring and editing of ontologies are very efficient as well [39]. The main drawback of this technique is that large scale ontologies are difficult to process. Furthermore Native storage needs to implement functionality such as data recovering, query optimization, controlled access and transaction processing in order to improve its data processing and management [37]. In the database-based storage, the ontology is stored in a Relational Database (RDB). Ontology storage in RDB needs to provide at least three of the following technologies: store and scalability, support for reasoning, and SPARQL query facilities [38] [39] [41]. Database-based storage is usually grouped into 2 main types [39], namely, generic and ontology specific (Figure 1). The generic schema [11] uses one table to store all triples or statements in the ontology. The table contains 3 columns, each representing an element of the ontology statement including Subject, Predicate and Object. Every row in the table is an ABox fact [11]. ABox are statements that describe the relationship between instances of the ontology [18]. TBox facts are ontology statements that describe relationship between classes and properties [18]. Many tables are required to store axioms or TBox facts of the ontology. The ontology specific format (Figure 1) creates tables according to the contents of the ontology. It has 3 modes of representation: horizontal, vertical and hybrid [11] [39]. In the horizontal mode also called one-table-per-class mode, every class is represented by a table with 2 columns. The first column represents the instance ID and the second column represents the predicate in which the instance ID belongs to. Properties are stored as values in the second column in the class table. In the vertical representation, also called one-table-per-property mode or decomposition storage model [11], tables are created for all properties of the ontology. Every table contains two columns as in the horizontal model including the Subject and Object columns to record the subjects and objects of ABox and TBox facts of the ontology. The hybrid model combines both vertical and horizontal representations in which tables are created for classes and properties.

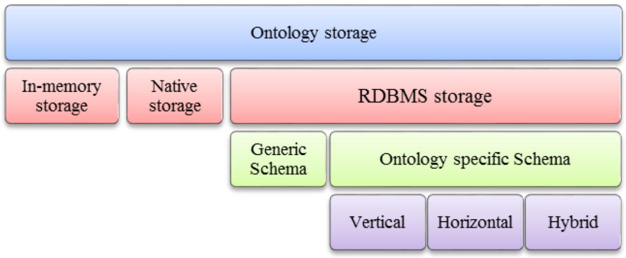

Figure 1. Ontology storage models

As shown in Figure 1 above, unlike in-memory and native storages, only RDBMS storage gives the possibility to elaborate further on the storage technics employed.

III. ONTOLOGY STORAGE AND QUERY PLATFORMS

As mentioned earlier, several platforms have been developed to enable the store and query of ontologies in relational databases. The commonly used platforms are: AllegroGraph, Jena API, Open Anzo, Oracle Semantic [8], Minerva [42] and Sesame [12]. AllegroGaph store ontologies as graphs [8]. It is installed as a server application and requires client applications such as Java, C#, Python, Ruby, Perl or Lips to access it. It supports SPARQL as query language but provides API for direct access to Subject, Predicates and Objects of ontology triples or statements without any use of SPARQL queries. Minerva [18] [42] is a component of the Integrated Ontology Development Toolkit (IODT). It is used as a plugin in Eclipse IDE. It stores OWL ontologies and supports the SPARQL query language. It also supports IBM DB2 and Derby as backend databases. Open Anzo is a Semantic Web platform developed by IBM. It can be used in three different modes: (1) embedded in an application, (2) installed as a server application and accessed remotely by clients or (3) run locally [8]. It supports the SPARQL query language. Further, it supports persistent storage through its Storage Service Layer which interacts with Relational Databases. In order to interact with Open Anzo, the client stack layer uses three different languages, namely, Java, Java Script or dot Net [8]. Open Anzo supports DB2 and Oracle as backend databases. Jena API is integrated into Eclipse IDE as a library and uses a

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-420-6

variety of DBMS such as Oracle, PostgreSQL and MySQL [8] [16]. It enables ontologies to be stored in three storage models: in-memory, native or RDB. The query languages supported by Jena API are SPARQL and RDQL. The Oracle Semantic [8] is a Jena Adapter that works with Oracle databases [8]. It is a plugins that implements Jena Graph and Jena Model interfaces. It also supports the SPARQL query. Sesame is a Software Development Kit (SDK) that was developed in the European IST project On-to-Knowledge [12]. It enables ontologies to be queried or exported. Two languages are used for ontology query in Sesame, namely, SPARQL and SeRQL. The Sesame architecture [12] has one component called the SAIL API which translates an ontology file into its RDB representation as well as enables Sesame to interface 2 DBMS, namely, MySQL and PostgreSQL. A comparative of the characteristics of the abovementioned platforms is provided in Table 1. The columns OWL and RDF show the platforms that support ontologies in these formats. The third column indicates those that are open source or not. Jena API and Sesame are used in this study as they both support RDF and OWL ontologies as well as MySQL RDBMS. Furthermore, Sesame and Jena API are both open source platforms and are accessible free of charge with full functions and support from the Internet.

TABLE I. CHARACTERISTICS OF ONTOLOGY STORAGE PLATFORMS

Ontology	OWL	RDF	Open Source	Availability
Allegrograph	no	yes	no	Commercial/free
Jena API	yes	yes	yes	free
Sesame	yes	yes	yes	free
Open Anzo	no	yes	yes	free
Oracle Semantic	yes	yes	no	Commercial/free
Minerva	no	yes	no	free

TABLE 1 shows a résumé on different platforms' attributes that guided us to select the two platforms used on the experiments.

IV. EXPERIMENTS

A. Dataset

The dataset is constituted of five ontologies, namely, Gene Ontology (GO) [20] [21] [24] [26] [27], WordNet [29] [30] [31], OntoDPM [32], Biological Top Level (BioTop) [33] [34] and Central Government ontology (CGOV) [28]; they have all been used intensively in related studies.

The GO ontology describes the biology domain in terms of molecular function, cellular components and biological process. It contains the vocabulary used in the biology field and the relationship between terms [21] [22] [23]. The WordNet ontology is an electronic lexical database for the English language [36]. It contains verbs, nouns, adverbs and adjectives. Written in a machine readable format, online dictionaries access it for public usage [29]. The OntoDPM ontology is a knowledge-based model for e-government monitoring of development projects in developing countries [32]. The BioTop ontology is an ontology of the life sciences domain which focuses on molecular biology [33]. It is used as a top level ontology to link the Open Biomedical Ontologies (OBO). The CGOV is an ontology of the UK central government; it models the structure of the UK central government [28].

TABLE II. CHARACTERISTICS OF ONTOLOGIES IN THE DATASET

Ontologies	Format	Size (Bytes)	No. Classes	No. Properties	No. Individuals
OntoDPM	OWL	38,578	30	19	18
CGOV	RDF	68,551	46	46	-
BioTop	OWL	429,989	389	92	-
WordNet	RDF	100,428,111	-	-	-
GO	OWL	106,912,638	-	-	-

Table 2 provides some metadata on the abovementioned ontologies constituting the dataset in this study in terms of their formats (RDF/OWL), sizes, and number of classes, properties and individuals. Some cells of Table 2 were not filled in due to the fact that the expected values were unavailable. In fact, in order to get the metadata in Table 2, an online ontology documentation tool called parrot is used [25]. Ontologies to be analysed are loaded within Parrot in three different ways including (1) uploading the ontology file, (2) pasting the code of the ontology or (3) providing the http address of the ontology. After loading the ontology and executing Parrot, ontologies characteristics such as the number of classes, properties and individuals are displayed. The loading of large ontologies such as GO and WordNet resulted in errors and therefore no characteristics were retrieved.

B. Performance Metrics

Three standard database performance metrics were used to measure and compare the performance of Sesame and Jena API in storing and querying ontologies in relational databases including,

(1) The loading time which is a common performance metric used in RDBMS studies [17] [18] [19]; it represents the time taken by a platform to process, parse and load an ontology into a relational database, (2) The query response time (QRT) [40] which represents the time taken by a platform to display the result of a query and (3) The repository size [19] which is the space disk needed for the storage of the resulting ontologies databases. In this study, the query response time is the average response times of several consecutive executions of the same query.

C. Computer and Software Environments

The experiments were carried out on a computer with the following characteristics: 64-bit Genuine Intel 2160 processor, Windows 8 release preview, 4 GB RAM and 160 GB hard drive. Protégé version 4.3 was installed in the computer and used to create the OWL code of OntoDPM ontology. The Apache tomcat server version

6.0 was installed in order to deploy the Sesame server. The Wamp server was installed as well to enable access to MySQL backend DBMS via Sesame and Jena API. Finally, Jena API was configured in the Eclipse IDE version 4.2. The metadata on the ontologies in the dataset such as the numbers of classes, properties, instances, etc. were determined with the online Semantic Web ontology documentation software, named, Parrot [25].

D. *Experimental Results*

1) *Data Loading into RDB*

The ontologies were loaded into MySQL relational databases via Sesame and Jena API, respectively. In Sesame, ontologies were loaded in command line mode [35]. A sample code used to load the ontologies in Jena API is provided below. The code shows part of the Jena application that reads and loads ontologies into MySQL databases.

```
1. ModelMaker maker = ModelFactory.createModelRDBMaker(conn);
2. Model loader = maker.createDefaultModel();
3. FileInputStream inputStreamfile = null;
4. File file = new File ("c:\\Devel\\gene.owl");
5. inputStreamfile = new FileInputStream(file);
6. InputStreamReader reader = null;
7. reader =new InputStreamReader(inputStreamfile, "UTF-8");
8. loader.read(readed, null);
9. reader.close();
10. loader.commit();
```

In the above code, line 1 creates a model, namely, *maker* which will be used to create the link between a model and the relational database. Line 2 creates a new model which will be used to store the ontology. Lines 3, 4 and 5 create a *FileInputStream* and file objects and then loads the file into the newly created file object. Line 6 creates the reader and line 7 loads the ontology into the reader. Line 8 reads the file from the reader and loads it into the ontology model. Finally line 9 closes the model and line 10 commits the model into the database.

Table 3 shows the loading times of the 5 ontologies presented in Sub-Section IV.A into MySQL databases with both Sesame and Jena API. It shows that Jena loads smaller ontologies (in the range of Kilobytes) (Table 2) faster than Sesame. But, for bigger ontologies (in the range of Megabytes) Sesame performs better with regards to loading ontologies into MySQL RDBMS.

The reason is the fact that Sesame opens an ontology file (OWL in this case), reads and loads it straight into MySQL database, whereas, Jena needs to first load the ontology into a RDF graph in the main memory before transferring it into MySQL database.

TABLE III. LOADING TIMES OF ONTOLOGIES INTO MYSQL DATABASES

Ontologies	Sesame Time (hh:mm:ss.000)	Jena API Time (hh:mm:ss:000)
OntoDPM	00:02:27.0	00:00:32.325
CGOV	00:05:15.776	00:00:45.318
BioTop	00:11:35.95	00:04:23.858
WordNet	14:27:34.387	17:44:10.365
GO	15:50:51.910	16:20:48.830

The data in Table 3 is represented graphically in Figure 2 in which the blue and red bars represent the loading times of ontologies into MySQL databases with Sesame and Jena API, respectively. The blue bars show that in Sesame the loading time is proportional to the size of the ontology, whereas, the red bars suggest that the loading time in Jena is disproportional to ontology sizes. In fact, the Gene Ontology which is bigger than WordNet (Table 2) took less time to be loaded into MySQL database. Figure 2 also shows that Jena loads small ontologies faster (less than a minute) and is slower in loading big ontologies compared to Sesame.

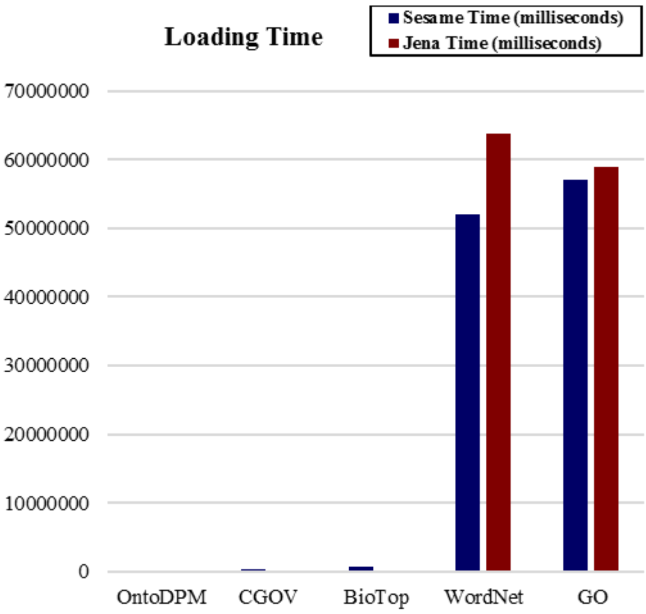

Figure 2. Chart of Loading Times of Ontologies into MySQL Databases

2) *Queries Response Times*

The query response time (QRT) [40] is the average time taken by a query to return a result. A sample SPARQL [13] query that searches for classes and their subclasses in the MySQL ontologies databases is given in the code below.

```
PREFIX rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject ?object
WHERE {?subject rdfs:subClassOf ?object} LIMIT 10.
```

The sample SPARQL query above was executed five consecutive times on each MySQL ontology database and the average response/execution times were recorded in Table 4.

TABLE IV. AVERAGE QUERIES RESPONSE TIMES ON ONTOLOGY DATABASES

Ontologies	Average Time in Sesame (ms)	Average Time in Jena API (ms)
OntoDPM	616.2	2393
CGOV	732.4	2321.6
BioTop	824.4	2369.4
WordNet	91.2	2428.2
GO	4135	2424

Table 4 represents the chart in Figure 3. It shows that the average queries response times are generally lower in Sesame (blue bars) than in Jena API (red bars). Further, the average queries response times in Jean API are almost constant on all ontology databases (red bars).

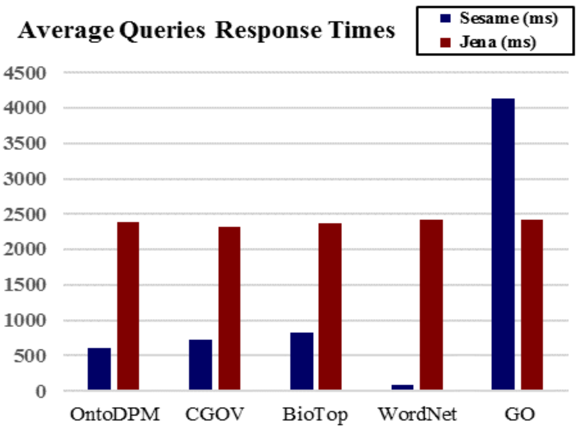

Figure 3. Chart of Average Queries Response Times on Ontologies Databases in Sesame and Jena API

3) Disk Space for Storing Ontologies Databases

Figure 4 is a comparison of the disk space used to store the resulting 5 ontologies databases into MySQL RDBMS via Sesame and Jena. The orange bars represent the initial sizes of the ontologies; it can be observed that OntoDPM and CGOV ontologies are very small as described in Table 2. The blue bars show the space required to store the ontologies in the Sesame repository; WordNet and GO required more space due their initial sizes. The red bars show the space required to store the ontologies into MySQL databases via Jena; the spaces used to store the ontology databases for WordNet and GO, are almost double of the space used in Sesame. The ontology databases for OntoDPM, CGOV and BioTop occupied less disk space due to their initial small sizes (Table 2).

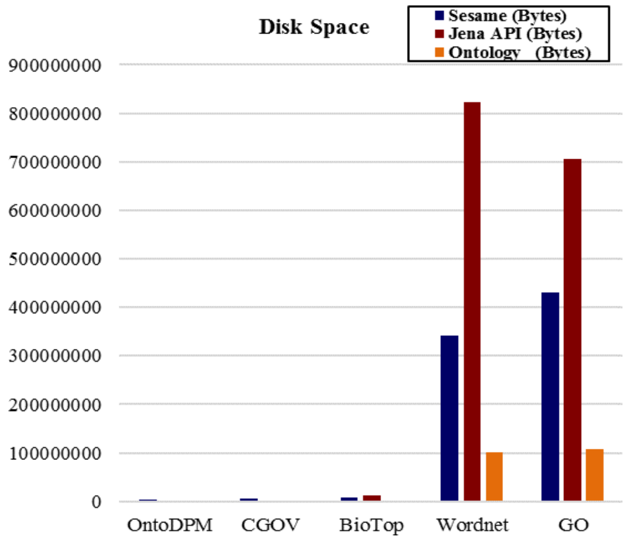

Figure 4. Disk Space Occupied by Ontologies Databases in Sesame and Jena API

As represented above, the graph in Figure 4 clearly shows that the required space is proportional to the ontologies independently of their format.

V. RELATED WORK

Mapping ontology to relational database is an active research topic in Semantic Web. Various techniques for mapping ontology features to that of relational database to enable the persistent storage of ontologies into RDB are presented in [14] [15] [16]. Three ontologies were stored and queried in MySQL databases via Jena API in [17]; the authors drew a similar conclusion as that of this study with regards to the scalability of Jena API. In [44] system properties of Jena Against sesame are provided. The authors describe the main difference between Jena and Sesame in terms of the properties that they are sharing and those which are different. [45] Provides similar analysis as in [44] but both do not provide an empirical analysis of the two platforms in terms of the performance. Several RDF databases solutions are reviewed in [8] and [10]. In [8], an evaluation of selected platforms including Sesame and Jena was carried out. However, not only was the study limited to RDF ontologies, but, the evaluation also was limited to the query response times only. In [10], ontology storage models such as generic and ontology specific schema as well as the functionalities of an RDF middleware and RDF query languages are discussed in detail.

VI. CONCLUSION AND FUTURE WORK

In this research, five ontologies were loaded into MySQL relational databases using two popular Semantic Web platforms, namely, Sesame and Jena API. Three metrics were used to measure and compare the performances of both platforms in terms of speed and scalability. The experiments showed that both platforms are scalable and could successfully parse and load ontologies of different sizes into relational database and that Sesame loads bigger ontologies faster than Jena API

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-420-6

into relational databases. Experiments also show that Sesame provides quicker responses to SPARQL queries compared to Jena API.

The future direction of the research would be to extend the study with more platforms so as to provide a more comprehensive performance evaluation of existing Semantic Web platforms for storing and querying ontologies in relational databases.

REFERENCES

[1] A., Lally, J. M. Prager, M. C. McCord, B. K. Boguraev, S., S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll, "Question analysis: how Watson reads a clue", IBM Journal of Research and Development, Vol. 56, No. 3/4, May-June 2012, pp. 1-14.

[2] A. Gali, C. X. Chen, K. T. Claypool, and R. Uceda-sosa, "From ontology to relational databases", ER Workshops, LNCS, Vol. 3289, 2004, pp. 278-289.

[3] M. Laclavík, "Ontology and agent based approach for knowledge management", Doctoral Thesis, Slovak Academy of Sciences, 2005.

[4] X. Lili, S. Lee, and S. Kim, "E-R model based RDF data storage in RDB", In Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu, China, 9-11 July 2010, pp.258-262.

[5] G. De Melo, F. Suchanek, and A. Pease, "Integrating YAGO into the suggested upper merged ontology, tools with artificial intelligence", In Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence. ICTAI'08. Dayton, OH, 3-5 November, 2008, pp.190-193.

[6] A.M. George, B. Richard, F. Christiane, G. Derek, and M. Katherine, "Introduction to WordNet: an on-line lexical database", International Journal of Lexicography Vol. 3, Issue 4, 1990, pp.235-244.

[7] M. Morsey, J. Lehmann, S. Auer, C. Stadler, and S. Hellmann, "DBpedia and the live extraction of structured data from wikipedia", Program: electronic library and information systems Vol.46, No. 2, 2012, pp. 157-181.

[8] F. Stegmaier, U. Gröbner, M. Döller, H. Kosch, and G. Baese, "Evaluation of current RDF database solutions". In Proceedings of the 10th International Workshop on Semantic Multimedia Database Technologies in conjunction with the 4th International Conference on Semantic and Digital Media Technologies (SAMT 2009),Graz, Austria, 2-4 December 2009 ,pp.1-17.

[9] D. HuiJun, W. WenGuo, Y. Jian, "A B-Tree algorithm for partitioned index based on CIDR list". International Conference of Information Technology, Computer Engineering and Management Sciences, Nanjing, Jiangsu, 24-25 September, 2011, pp. 124-128.

[10] A. Hertel, J. Broekstra, and H. Stuckenschmidt, "RDF Storage and Retrieval Systems", International Handbooks on Information Systems, 2009, pp.489-508.

[11] F. Dieter, V.H. Frank, K. Michel, and A. Hans, "On-To-Knowledge: ontology-based tools for knowledge management", In Proceedings of the eBusiness and eWork (EMMSEC 2000) Conference, Madrid, Spain, 18-20 October, 2000, pp. 1-7.

[12] D. Fensel, J. Hendler, H. Lieberman, W. Wahlster, and T. Berners-Lee, "Sesame: an architecture for storing and querying RDF data and schema information, spinning the semantic web: bringing the World Wide Web to its full potential", MIT Press, 2005, pp. 197-222.

[13] S. Harris, "SPARQL query processing with conventional relational database systems", In Proceedings of the 6th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS), New York City, USA, November 2005, 20-22 pp.235-244.

[14] W. Teswanich, and S. Chittayasothom, "A transformation from RDF documents and schemas to relational databases", In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, Canada, 22-24 August 2007, pp.38-41.

[15] S. Ramanujam, A. Gupta, L. Khan, S. Seida, and B. Thuraisingham, "R2D: A bridge between the semantic web and relational visualization tools", In Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC), Berkeley, CA, USA, 14-16 September 2009, pp.303-311.

[16] A. Alamri, "The relational database layout to store ontology knowledge base", In Proceedings of the International Conference on Information Retrieval & Knowledge Management (CAMP), Kuala Lumpur, Malaysia, 13-15 March, 2012, pp.74-81.

[17] J. V. Fonou-Dombeu, N. Phiri and M. Huisman, "Persistent storage and query of e-government ontologies in relational databases", In Proceedings of the 3rd International Conference on Electronic Government and the Information System Perspective, Munich, Germany,1-4 September, 2014, pp.118-132.

[18] J. Zhou, L. Ma, O. Liu, L. Zhang, Y. Yu, and Y. Pen, "Minerva : a scalable owl ontology storage and inference system", In Proceedings of the First Asian conference on The Semantic Web, Beijing, China,3-7 September 2006, pp.426-443.

[19] G. Yuanbo, P. Zhengxiang, and J. Helfin, "LUBM a benchmark for OWL knowledge base systems", In Proceedings of the 3rd International Conference on Semantic Web (ICSW),Hiroshima, Japan, 7-11 November, 2004, pp.158-182.

[20] K. Taha, "GOseek: A gene ontology search engine using enhanced keywords", In Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3-7 July 2013 pp.1502-1505.

[21] X. Qing-Wei, L. Qiang, L. Qian, L. Qing-Ming, and L. Yi-Xue, "Semantic query and reasoning among gene ontology annotations", In Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering, ICBBE, Wuhan, China, 6-8 July, 2007, pp. 298-301.

[22] M. Gan , and R. Jiang, "Inferring semantic similarity through correlating information contents of gene ontology terms", In Proceeding of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) ,Shanghai, China, 18-21 December, 2013, pp.1-5.

[23] C. Chien-Ming ,C. Chia-Sheng, H. Zhen-Li, and P. Tun-Wen, "Discover significant associations of orthologous simple sequence repeat patterns with gene ontology terms", IEEE International Conference on Bioinformatics and Biomedicine Workshop, Washington, USA, 1-4 November, 2009,pp.209-213.

[24] L. Jing, Ng, M.K.;Y. Liu , "Construction of gene networks with hybrid approach from expression profile and Gene Ontology", IEEE Transactions on Information Technology in Biomedicine, Vol.14 , Issue: 1, 2009, pp.107-118.

[25] Parrot online tool, Available at http://ontoruleproject.eu/parrot.html [Retrieved: July,2015]

[26] Gene Ontology Consortium, "The Gene Ontology (GO) database and informatics Resource", Nucleic Acids Research Journal, Jan, Vol. 32, No. 1, 2004, pp. 258-261.

[27] B. Smith, J. Williams, and S. Schulze-Kremer, "The ontology of the Gene Ontology", In Proceedings of the Annual Symposium of the American Medical Informatics Association, Washington DC, USA, November, 2003, pp.609-613.

[28] CGOV: Central Government Ontology. An Ontology of the UK Central Government, Available at

http://reference.data.gov.uk/def/central-government [Retrieved: May,2014]
[29] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "WordNet: An on-line lexical database", International Journal of Lexicography, Vol 3, No.4, 1990, pp.235-244.
[30] M. Van Assem , A. Gangemi, and G. Schreiber, "Conversion of WordNet to a standard RDF/OWL representation", In Proceedings of the Language Resources and Evaluation conference, Geneo, Italy, 24-26 May, 2006, pp.1-6.
[31] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::similarity - measuring the relatedness of concepts", In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, CA,USA, July 25-29, 2004, pp. 1024-1025.
[32] J.V. Fonou-Dombeu, and M. Huisman, "Combining ontology development methodologies and semantic web Platforms for E-government Domain Ontology Development. International Journal of Web & Semantic Technology (IJWesT) Vol.2, No.2, October, 2011, pp.12-25.
[33] S. Schulz, E. Beisswanger, J. Wermter, and U. Hahn, "Towards an upper-level ontology for molecular biology", In Proceedings of the AMIA annual symposium, 2006, pp.694-698.
[34] E. Beisswanger, S. Schulz B., H. Stenzhorn , and U Hahn, "BIOTOP: An upper domain ontology for the life sciences :a description of its current structure, contents, and interfaces to OBO ontologies", Journal of Applied Ontology- Towards a meta ontology for the Biomedical Domain, Vol. 3, No. 4,2008, pp.205-212.
[35] Sesame online documentation, available at http://openrdf.callimachus.net/sesame/2.7/docs/users.docbook ?view [Retrieved: September, 2014]

[36] X. Huang, and C. Zhou, "An OWL-based WordNet lexical ontology", Journal of Zhejiang University Science A, Vol.8, No. 6, May 2007 pp.864-870.
[37] S. Heymans, L. Ma, D. Anicic, Z. Ma, N. Steinmetz, Y. Pan, J. Mei, A. Fokoue, A. Kalyanpur, A. Kershenbaum, E. Schonberg, K. Srinivas, C. Feier, G. Hench, B. Wetzstein, and U. Keller. "Ontology reasoning with large data repositories, Semantic web and beyond computing for human experience", Vol.7, 2008, pp.89-128.
[38] Y. Yuang, G. Li, and X.Wang, "Semantic web rough ontology: definition, model and storage method", In Proceedings of the 6th International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), Xi'an, China, 23-24 November, 2013, pp.104-107.
[39] Z. Zhou, and X. Yongkang, "A study on ontology storage based on relational database", china, IEEE Conference Anthology, 1 8 Jan, 2013, pp.1-5.
[40] R. Agrawal A. Somani, and Y. Xu, "Storage and querying of e-commerce data", In Proceedings of the 27th International Conference on Very Large Data Bases, Rome, Italy, September 11-14, 2001, pp.149-158.
[41] J. Lu, M. Li , Z. Lei , B. Jean-Sébastien , W. Chen , P. Yue , and Y. Yong, "SOR:A practical System for ontology storage, reasoning and search", In Proceedings of the 33rd international conference on very large databases, Vienna, Austria, September 23-28, 2007, pp.1402-1405.
[42] J. Zhou, L. Ma, Q. Liu, L. Zhang, Y. Yu, and Y. Pan, "Minerva a scalable owl ontology storage and inference system", In Proceedings of the first Asian semantic web conference, Beijing, China, 3-7 September, 2006,.pp 429 443.
[43] I. Astrova, N. Korda, and A. Kalja, "Storing OWL Ontologies in SQL Relational Databases", International Journal of Electrical, Computer, and Systems Engineering, Vol. 1, No 4, 2007.
[44] http://blog.sparna.fr/2012/05/08/rdf-sesame-jena comparaison-des-fonctionnalites/ [Retrieved: July, 2015]
[45] http://db-engines.com/en/system/Jena%3BSesame [Retrieved: July, 2015]