Journal of Theoretical and Applied Information Technology
30th April 2019. Vol.97. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONTOLOGY2RDB FOR STORING ONTOLOGY IN RELATIONAL DATABASE

[1]RAWAN ABO ZIDAN, [2]MUHAMMAD MAZEN ALMUSTAF, [3]MOHAMMAD TAHAWI

[1,2]Syrian Virtual University, Web Science program, Syria

[3]Cern, Switzerland

E-mail: [1]rawan_684969@svuonline.org, [2]t_mmustafa@svuonline.org, [3]Mhd.tahawi@gmail.com

ABSTRACT

With the large expansion of data volume found on the web the majority of it is stored in relational format which provide maturity, performance, robustness, reliability, and availability but they lack the expressiveness and process-ability of semantic data and ontologies. The semantic web has a significant impact in various domains, which has led to the need of finding a formula for communicating with relational databases. In this paper we present a tool that acts as a bridge between the relational database and semantic data in which we provide a fully automated and effective approach that converts between a populated ontology and a relational database without losing the ontological structure while preserving the knowledge relations between the entities. Finally, we tested this approach to convert a weather ontology into relational database, by converting its ontological constructs to their corresponding entities in relational format, then maps the converted ontology with pre-existed relational database so that it could be queried effectively, as a result we were able to get a relational database with knowledge incorporated relations that was quite descriptive of the stored knowledge.

Keywords: *Relational Database, Semantic Web, Ontology, OWL, WordNet.*

1. INTRODUCTION

The current web is rapidly growing, and this leads to a huge amount of information and techniques used to process this information to facilitate its access and use [1] [2].

The web needed to be extended to the web with semantics, this is what led to sematic web appearance.

Semantic Web provides technology to capture, share, and reuse structured and machine-readable domain specific knowledge and makes it available on web.

Relational databases have serve tremendous role in supporting data in various websites from storing to querying data, but they lack the ability to add semantics to data.

On the other side, ontology is a knowledge base system that contains a vocabulary of basic terms concerning a particular domain and semantic interconnections between those terms [3].

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 [4] [5] [6].

Improving and sharing domain specific knowledge that resides in database is one of our motivations to create the mapping with ontologies, so we can share information, and new knowledge can be inferred and querying richer representations will be possible [7].

Querying the system will be more advanced, robust and optimized. A common goal is the consolidation of distributive information in the form of common vocabulary.

Ontology2RDB provides relational database access through ontologies, by converting the domain ontology into RDB. In this approach, data access is enabled by defining links between ontology concepts and relational data by using WordNet library.

This new and improved way of storage will not only improve the storage method but also helps in managing the OWL data in an efficient manner [8].

Previous researchers have worked on mapping of RDF/OWL concepts into relational database. But these mapping approaches have certain problems like loss of structure, loss of data and perform only initial mappings i.e. tables to classes and columns to properties. Most of transformation tools are semi-automatic and need human intervention [9].

The majority of existing applications need integration among these systems. Through this mapping ontological data can be accessed from existing relational database applications. The relational database was chosen to store the ontology because of:

- Legacy data: When stored in relational databases, ontologies can interoperate with a large amount of data in existing relational databases.
- Legacy applications: When stored in relational databases, ontologies can be accessed from within existing relational database applications.
- Large scale ontologies: The ability of relational databases to store a large amount of data proves that the relational databases are also suitable for storing large scale ontologies that can contain millions of instances [10].

We have proposed a tool which is called Ontology2RDB that is fully automatic in mapping OWL ontology to relational database format then binds pre-exists relational database semantically with the converted ontology using WordNet.

Through mapping we can share information, querying the system will be more advanced, robust and optimized. A common goal is the consolidation of distributive information in the form of common vocabulary.

The rest of the paper is organized as follows. Section 2 describes briefly the motivation to build Ontology2RDB, in section 3 previous approaches for storing ontologies and their drawbacks have been provided. System architecture is discussed in Section 4 in which it gives an overview of the proposed tool Ontology2RDB and used libraries. Section 5 presents the experiments and results of application of the rules defined in Section 4 as well as a comparison between other tool called OWLMap and the current work. The paper ends with a conclusion and future work in Section 6.

2. MOTIVATION

Ontologies are important to application integration solutions because they provide a shared and common understanding of data that exist within an application integration problem domain [15].

But query facility in ontology management system is not as efficient as in relational database system [23].

With the large use of relational database systems and the fame of ontologies, we require a tool for converting the ontology into relational database to improve information seeking, recovery and query facility.

Because of large size and complex ontologies like OWL Full Ontologies, they need to be stored in efficient way so it could be queried effectively, so they need to be stored in relational databases.

Storing ontology in relational database facing a series of challenges, like automated or semi-automated mapping, is mapping based on rules is a good solution or not.

3. RELATED WORK

Various tools and algorithms have been developed to support the conversion of owl ontologies into relational database.

-Previous ontology to relational database transformation approaches e.g. OWL to ER and ER to OWL use conceptual graphs. They perform step-wise transformations where first step is to transform the OWL ontology to ER and second step is to transform ER to relational database [11]. Oracle Semantic data storage is also used but most OWL constructs are missing [12].

-**Rule based transformation:** presented by [13] and [10] are based on “mapping rules”. The short comes of these approach is that few constructs are missed during transformation. Few sub-properties and few constructs of OWL ontology are not considered e.g. property restrictions.

-Reference [14] developed a tool named “OntoRel” for transformation. The disadvantage of “OntoRel” was that it only selects few main OWL constructs for transformations.

-**Edge Approach:** Store all the attributes information (object identities, name, and flag) in a single table called Edge table.

-**Attribute approach:** Attributes with the same name grouped into one table. Conceptually, this approach corresponds to a horizontal partitioning of the Edge table used in the Edge approach, using name as the partitioning attribute. Thus, as many attribute tables are created as for different attribute names in the XML document, and each Attribute table has the following structure:

Aname(source, ordinal, flag, target)

The key of the Attribute table is source, ordinal, and all the fields have the same meaning as in the Edge approach [15].

-**Universal approach:** stores all attributes with separate columns for each attribute present in XML document.

Journal of Theoretical and Applied Information Technology
30th April 2019. Vol.97. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

-**Normalized universal approach:** Introduces separate overflow tables for multi valued attributes.

-**Basic in lining approach:** It maps the XML DTD into relations [9].

-**Reversible lossless transformation from OWL 2 Ontologies into relational database**: proposed OWL2ToRDB transformation in QVT Relation (QVTR) language that is capable for defining bidirectional transformations, implement transformations in both directions [16].

-**Storing ontology includes fuzzy data types:** deals with the need for managing large amounts of fuzzy data in the context of the Semantic Web. A schema to store ontologies with fuzzy datatypes into a database is presented as part of a framework designed to perform tasks of fuzzy information extraction and publishing [17].

-**Mapping of OWL ontology concepts to RDB schema Approach:** Use some principles and algorithm, proposed the prototype tool as a plug-in for an ontology editor protégé.

-The most commonly used platforms to enable the persistent storage and query of ontologies in relational databases are: AllegroGraph, Jena API, Open Anzo, Minerva [18] and Sesame [19]. AllegroGraph is currently available only in the form of trial versions [6]. 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.

Our tool Ontology2RDB can convert new ontology constructs that have not previously converted in previous research such as unionOf, intersectionOf and one of, automatically without human intervention and create a mechanism for linking with relational databases to integrate with different existing relational databases.

4. SYSTEM ARCHITECTURE

The paper presents a process for converting an OWL ontology into relational database automatically and without loss of data, then links the converted ontology to a pre-existing relational database.

In this section, we will define ontologies, how to store an OWL ontologies and the libraries that have been used.

4.1. Ontology

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related [20].

W3C offers a large palette of techniques to describe and define different forms of vocabularies in a standard format. These include RDF and RDF Schemas, Simple Knowledge Organization System (SKOS), Web Ontology Language (OWL), and the Rule Interchange Format (RIF) [21].

The Web Ontology Language OWL extends RDF and RDFS. Its aim is to bring the expressive and reasoning power of description logic to the semantic web.

An OWL ontology may include descriptions of classes, properties and their instances.

4.2. Storing Ontology

There are two main methods for storing ontologies using:

1) File systems for storing ontologies in flat files. The main problem with this technique is that file systems do not provide scalability, share-ability, or any query facility [10].
2) Database Systems, in particular relational database management systems if ontology is stored in relational format then it can easily interoperate with large amount of existing web data.

We found that relational databases systems are better for storing ontologies, that why we choose it in Ontology2RDB tool.

4.3. Used Libraries

1) DotNetRDF

The goal of this project is to create an open source library using the latest framework of .NET.

DotNetRDF provides developers with powerful and easy to use API for working with RDF and SPARQL in .NET environments.

A complete library for parsing, managing, querying and writing RDF.

Free and open source under a permissive MIT license [22].

2) WordNet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations [4].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.4. System Components

Ontology2RDB provides automatic conversion of OWL ontology to relational database format.

As Figure 1 shows, after the domain ontology identified, the ontological constructs were extracted using DotNetRDF library by extracting the classes, properties with its two kinds object and datatype and other constructs.

The proposed conversion mapping rules is applied, then links will be identified between the relational database and the converted ontology using WordNet library.

As a result, we get a new relational database.

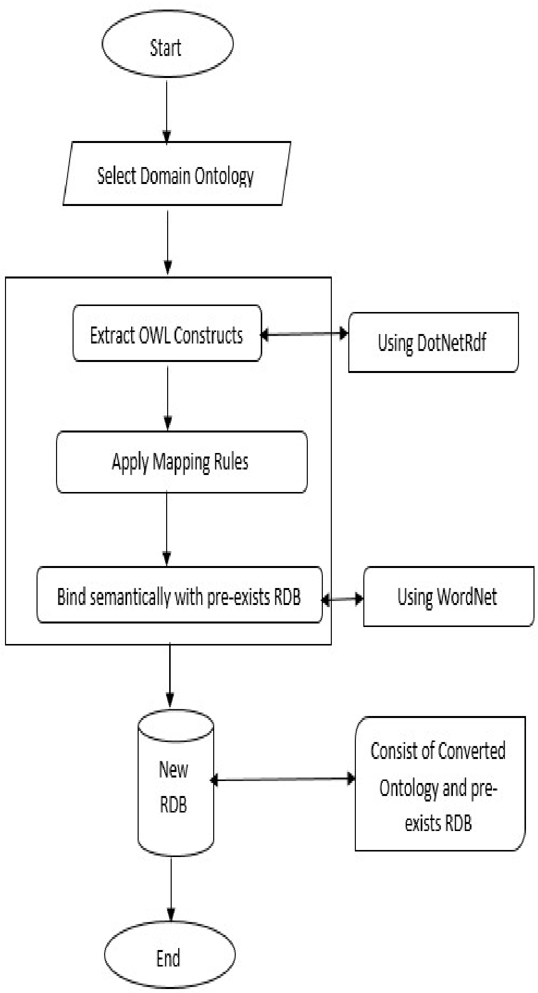

Figure 1 System Architecture

In figure 1 first we identify the ontology file, then extract the ontology constructs using DotNetRDF library like classes, properties with its two kinds object and datatype and other constructs.

Apply the proposed Mapping Rules, then map semantically between the converted ontology and pre-exists relational database using WordNet library.

The result is new relational database contains the converted ontology and pre-existing relational database and the mapping between them.

4.5. Mapping Rules

The mapping rules of conversion an OWL ontology into relational database follows certain rules/principles, by extracting the classes, properties and the ontological construct using the library DotNetRDF, then convert them into their corresponding in relational database.

The file will be processed to extract root class, super classes and subclasses.

Classes Conversion Rule:

Convert all the above classes into tables then bind subclass with his parent class using One-One relation.

Individuals Conversion Rule:

Extract individuals from ontology file and convert them as values inside the above classes.

OWL properties Conversion Rule:

Extract OWL properties object and data type properties and define its Domains and Ranges for these properties.

- If the property has subproperty it will be converted into table.
- If the property type owl:functional It will be converted into column inside the class that exists in the domain and bind the added column with the class as foreign key to the primary key that exists in the range class, If the property does not have an owl: functional it will be converted into table.
- If the property is Object type: the values come from the values that exists in the range of another table.
- If the property is Datatype: the values that exist in the range are values for this column and define the column type from the property data type.
- Add the domain of owl:inverseOf property as a range to the other one and the range will be added as a domain to the other.
- The same individuals will be added using the relation owl:sameAs.
- Set the column representing owl:inverseFunctionalProperty as unique.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Data Type Conversion of Data type properties:

During transformation of data type properties, we have to transfer data type of data type properties, in table1 we present a list of commonly used XSD types along with their corresponding SQL data types.

Table 1Common types of converting Data Type of Data Types Properties

XSD	SQL
Short	Smallint
Integer	Integer
Long	Integer
Decimal	Decimal
Float	Float
Double	Double precision
String	varchar
Normalized String	varchar
Language	varchar
Time	Time
Date	Date
Datetime	TimeStamp
Boolean	Bit

OWL Restrictions Rule:

Create tables to store all OWL Restrictions.

Union and Intersection Conversion Rule:

Create a view representing owl:intersectionOf by using select inner join of each one of the classes.

Create a view representing owl:unionOf by using select full join of each one of the classes.

Enumerations Conversion Rule:

Create a table representing owl:oneOf and inserting each one of its instances as tuple in this table.

4.6. Bind Semantically between the Relational Database and the Converted Ontology

After converting the ontology into relational database, the next step is to link the tables of relational database derived from the ontological conversion process with already exists relational database, the links will be obtained based on semantic of the relations.

Given a relational database resulting from ontology conversion A, and already exists Relational Database B, the links will be generated as the following:

- Extract schema of each table in B, then find synonyms of each table using WordNet.
- Compare each one of the synonyms with A tables to find matching tables.

The result is new relational database that contains the pre-exists relational database and the converted ontology and the links between them, so it could be queried effectively.

5. EXPERIMENT

5.1. Dataset

A weather ontology was used as the input ontology in the experiments in this study, you can download it and explore it, after performing transformation all ontological constructs were saved in RDB schema.

Another input is SQL server relational database about weather, chosen SQL Server 2016 to store this information in relational database format.

Table 2 presents the Weather database schema in a tabular format including all the tables, columns and relationships that exist in the database.

The assumption for this experiment that the domain ontology of interest is defined before, and there is an already existing relational database.

The tables of the weather relational database are Station, Stats, WeatherState, WeatherHourlyForecast, WeatherDailyForecast and AirPollution, total number of records in the relational database is 1500 record.

Table 2 Weather Relational Database

In figure 3 represent weather relational database using MS SQL Server 2016

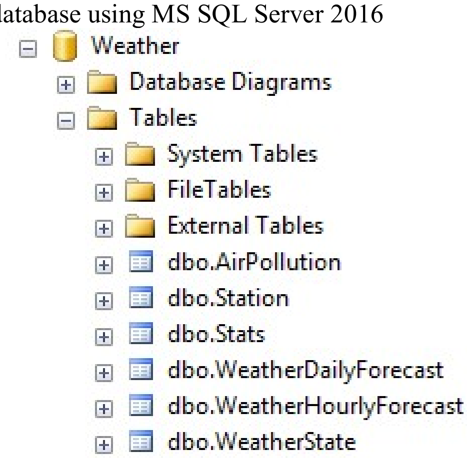

Figure 2 Weather Relational Database

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.2. Testing Environment

Table 3 represents some important specifications of our machine are given, in order to analyze and evaluate the performance of Ontology2RDB.

Table 3 Machine Specifications

Processor	CPU speed	OS	Memory	System type
Intel core i5	2.40GHz	Windows 10	8GB	64bit operating system

The time Ontology2RDB takes to process an OWL ontology is 18 seconds, it is tested on different ontologies sizes.

5.3. Evaluation Matrix

To test the performance of our tool, we take a Weather ontology and converted into relational database, to evaluate the loading time of ontology into relational database.

The performance and results of Ontology2RDB are compared with OWLMap tool, various metrics including (1) number of converted classes, (2) number of converted subclasses and their relationship, (3) number of converted object properties, (4) number of converted datatype properties and their datatype, (5) restrictions, (6) number of converted unions, (7) number of converted intersections, (8) number of converted enumerations.

5.4. Experimental Results

Figure 3 shows the tables of OWL ontology constructed in Ms Sql Server database.

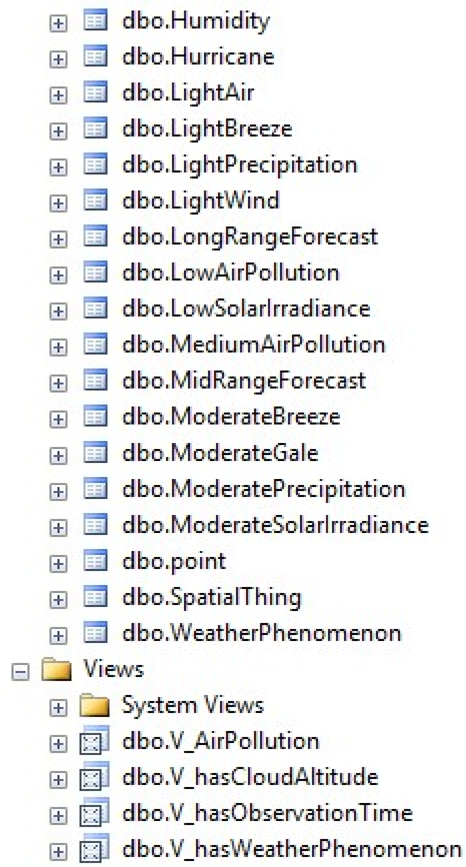

Figure 3 Tables of the OWL ontology including enumeration tables with the views constructed from owl:intersectionOf and owl:unionOf

Figure 3 is the result of converting ontology classes, enumeration and properties with its two kinds object and datatype, and the views are the result of converting union and intersection.

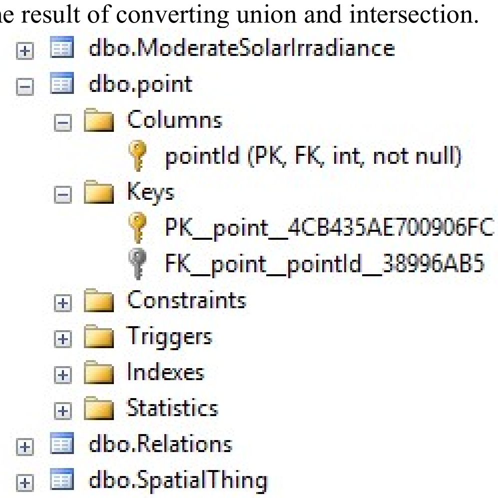

Figure 4 Converting Super and Sub Classes into RDB

Journal of Theoretical and Applied Information Technology
30th April 2019. Vol.97. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

As shown in figure 4 the class Point is a subclass of class SpatialThing, after applying the classes conversion rule, the two classes are linked with One-One relation.

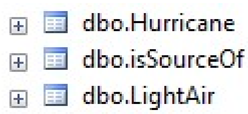

Figure 5 Converting Property has Subproperty into table

As shown in figure 5 the property isSourceOf has subproperty topObjectProperty, after applying the owl properties conversion rule, the property is converted into SQL table.

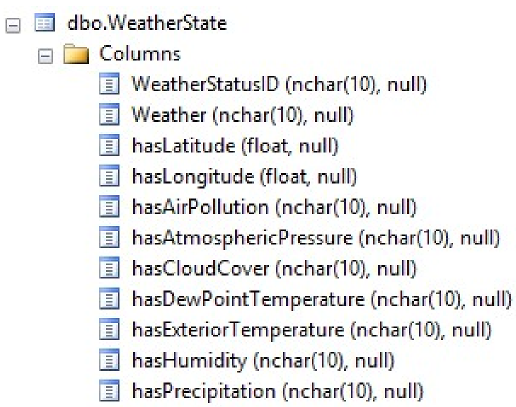

Figure 6 Converting owl:functionalproperty into column in table WeatherState

Figure 6 represents the columns of table WeatherState, which come from converting owl:functionalproperty into columns in the domain class.

	WeatherCondition
1	Snow
2	Fog
3	Cloud
4	PartlyCloud
5	Rain
6	LightCloud
7	Sleet
8	Thunder
9	Hail
10	Sun
11	LightRainSun

Figure 7 Appling enumerations conversion rule

Figure 7 represents table of WeatherCondition after applying enumeration conversion rule and inserting each instance of owl:oneOf as a tuple in the table.

The next step is performing the binding semantically between the relational database and the converted ontology using WordNet, by finding the synonyms of the name of each table in the already existing relational database, then matching these synonyms with the names of the converted ontological tables, then creating a table that includes the names of both tables from the relational database and corresponding table name of the ontology.

	MappingId	RDB_Tables	Ontology_Tables
1	1	WeatherState	AiringWeatherState
2	2	AirPollution	AirPollution

Figure 8 Binding between pre-exists weather RDB and the converted ontology

In figure 8 is the result of using WordNet library. Shows that, WeatherState table exists in relational database matches AiringWeatherState that exists in the ontology.

Both tables of AirPollution that exist in relational database and ontology match.

Thus, we have introduced an automated method of linking the ontology and pre-exists relational database without human intervention.

The total number of tables that Ontology2RDB created in the relational database is 200 table, including table represent enumerations, and the total number of created views is 114 view that represent intersections and unions of the OWL ontology file.

Since a relational model does not support all constructs of an ontological model, so there are some limitations in Ontology2RDB like converting class complement and OWL ontologies may contain properties without any specific domain/range, meaning that they could be associated with individuals of any class.

5.5. Comparing with OWLMap Tool

OWLMap is a tool provides automatic and lossless approach for transformation of ontology into relational database format.

OWLMap is based on approach that is lossless as well as it does not loose data, data types and structure by defining mapping rules to transform ontology in to a database format [23].

It is observed that some ontological constructs are not transformed in OWLMap such as intersection, union and enumeration.

We compare the ontological constructs that OWLMap used and our proposed tool.

Table 4 Comparing with OWLMap Tool

Constructs that have been converted	OWLMap	Ontology2RDB
Classes	Yes	Yes
Subclass	Yes	Yes
Object properties	Yes	Yes
Data type properties	Yes	Yes
Restrictions	Yes	Yes
Union	No	Yes
Intersection	No	Yes
Enumeration	No	Yes

In table 4 we found that in Ontology2RDB has converted more constructs than OWLMap.

we have converted more constructs has not been converted in other researches like Union, Inersection and Enumeration.

Ten ontologies from standard web site of Stanford University were taken to perform the testing.

TABLE 5 EVALUATION OF THE ONTOLOGIES ON OWLMAP AND OUR TOOL

Table 5 presents the result of testing on Ontology2RDB compared with the result of OWLMap tool.

When converting Pizza ontology in Ontology2RDB in line (2), it has converted 96% from the total number of ontology classes, while in OWLMap (1) has converted 94% of total classes.

On the other hand, when converting subclasses and object properties, OWLMap tool is better than Ontology2RDB, where it has more constructs converted.

In both tools converted the ontology restrictions to their corresponding tables.

The result of converting Edu ontology in line3 and line4, OWLMap tool is better than Ontology2RDB when converting ontology classes, but when converting subclasses Ontology2RDB is converting more subclasses than OWLMap.

Both tools the same when converting object properties and restrictions.

Ontology2RDB converts all union constructs to their corresponding views in Edu ontology and converts 90% of intersections in the ontology file.

Ontology2RDB converts all enumerations that exist in ontology file.

The Null value represents that the constructs hasn't been converted.

As a result of the comparison we found that Ontology2RDB has added more constructs, the previous tools have not converted, which are Union, Intersection and Enumeration.

6. CONCLUSION

The Semantic Web is an effective concept for knowledge management.

For its best use, it requires the use of ontologies because it is a mechanism for representing concepts and link them by meaningful relationships.

In order to increase the effectiveness of the semantic Web, it has to convert its ontologies into a relational database to be easily queried and to facilitate search and retrieval.

Our research aimed to improve the storage model of OWL by mapping all ontology information in to relational database so that it can be queried easily.

Ontology2RDB is fully-automatic and can handle wider range of OWL constructs.

Eleven ontologies were loaded into Ms SQL Server relational database and compared with OWLMap tool.

Results from comparison with other tools show that our tool provides improved performance.

The study found the following results:

1. Find a mechanism for automatic conversion of the ontological file.
2. Map constructs of OWL ontology into RDB like "intersection, union and Enumeration" that didn't handle in any transformation approach.
3. Linking the converted ontology with an existing relational database using WordNet to obtain new semantics that did not exist.

Our future work in this domain is to focus on transferring other ontology construct to relational database like class complement and properties without any specific domain/range to enable researchers to convert various ontologies to relational database to increase query and search efficiency.

REFRENCES:

[1] Breslin, J. G., O'Sullivan, D., Passant, A., & Vasiliu, L, "Semantic Web computing in industry. Computers in Industry," vol. 61,

no. 8, pp. 729-741, 2010.

[2] Berners-Lee, T., Hendler, J., & Lassila, O, The semantic web. Scientific american, 2001.

[3] Fonou-Dombeu, J. V., & Kwuimi, R, "The Underpinnings of Ontology Storage in Relational Databases: An Empirical Study," in *In 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, 2018.

[4] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J, " Introduction to WordNet: An on-line lexical database," *International journal of lexicography,* vol. 3, no. 4, pp. 235-244, 1990.

[5] Morsey, M., Lehmann, J., Auer, S., Stadler, C., & Hellmann, S, " Dbpedia and the live extraction of structured data from wikipedia," vol. 46, no. 2, pp. 157-181, 2012.

[6] Stegmaier, F., Gröbner, U., Döller, M., Kosch, H., & Baese, G, "Evaluation of current RDF database solutions. In Proceedings of the 10th International Workshop on Semantic Multimedia Database Technologies (SeMuDaTe)," in *4th International Conference on Semantics And Digital Media Technologies (SAMT)*, 2009.

[7] Zemmouchi-Ghomari, L., Djouambi, A., & Chabane, C, "Proposal for a Mutual Conversion Relational Database-Ontology Approach," *International Journal of Modern Education and Computer Science,* vol. 10, no. 7, p. 13, 2018.

[8] ALAmri, A, "The relational database layout to store ontology knowledge base," *In 2012 International Conference on Information Retrieval & Knowledge Management,* pp. 74-81, 2012.

[9] Humaira, A., Tabbasum, N., & Ayesha, S, "A survey on automatic mapping of ontology to relational database schema," *Research Journal of Recent Sciences ,* 2015.

[10] Astrova, I., Korda, N., & Kalja, A, "Storing OWL ontologies in SQL relational databases," *International Journal of Electrical, Computer and Systems Engineering,* vol. 1, no. 4, pp. 242-247, 2007.

[11] Tirmizi, S. H., Sequeda, J., & Miranker, D, "Translating sql applications to the semantic web," in *In International Conference on Database and Expert Systems Applications*, 2008.

[12] Wu, Z., Eadon, G., Das, S., Chong, E. I., Kolovski, V., Annamalai, M., & Srinivasan, J, " Implementing an inference engine for RDFS/OWL constructs and user-defined rules in Oracle," in *In 2008 IEEE 24th International Conference on Data Engineering*, 2008.

[13] Zina, N., & Kaouther, N, " Automatically building database from biomedical ontology," in *In International Work-Conference on Bioinformatics and Biomedical Engineering*, 2014.

[14] de Brum Saccol, D., de Campos Andrade, T., & Piveta, E. K, "Mapping owl ontologies to relational schemas," in *in Information Reuse and Integration (IRI)*, 2011.

[15] Gali, A., Chen, C. X., Claypool, K. T., & Uceda-Sosa, R, "From ontology to relational databases," in *In International Conference on Conceptual Modeling*, Berlin, Heidelberg, 2004.

[16] Vyšniauskas, E., Nemuraitė, L., Butleris, R., & Paradauskas, B, "Reversible lossless transformation from OWL 2 ontologies into relational databases," *Information technology and control,* vol. 40, no. 4, pp. 293-306, 2011.

[17] Barranco, C. D., Campaña, J. R., Medina, J. M., & Pons, O, " On storing ontologies including fuzzy datatypes in relational databases," in *In 2007 IEEE International Fuzzy Systems Conference* , 2007.

[18] Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y., & Pan, Y, "Minerva: A scalable OWL ontology storage and inference system," in *In Asian Semantic Web Conference*, Berlin, Heidelberg, 2006.

[19] Broekstra, J., Kampman, A., & Van Harmelen, F, 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, 2003, pp. 197-222.

[20] McGuinness, D. L., & Van Harmelen, F, "OWL web ontology language overview," *W3C recommendation,* vol. 10, no. 10, 2004.

[21] "VOCABULARIES," [Online]. Available:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

https://www.w3.org/standards/semanticweb/ontology.

[22] Barbur, G., Blaga, B., & Groza, A, "OntoRich-A support tool for semi-automatic ontology enrichment and evaluation," in In 2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing, 2011.

Table 2 Weather Relational Database

Relation	Primary Key	Foreign Key
Station(StationID, CITY, State, Lat_N, Long_W)	StationID	Null
Stats(StatsID,StationID,Month, Temp, Rain)	StatID	StationID refers to StationID in Station
WeatherState(WeatherStateID, WEATHER)	WeatherStateID	NUll
WeatherHourlyForecast(ID,WeatherStateId, WinfSpeed,WindDirection, AirPollutionId)	ID	WeatherStateId refers to WeatherStateID in WeatherState AirPollutionId refers to ID in AirPollution
WeatherDailyForecast(ID, WeatherStateId,SunsetTime,SunriseTime)	ID	WeatherStateId refers to WeatherStateID in WeatherState
AirPollution(ID, AirPollutionPercent)	ID	Null

Table 5 Evaluation of the ontologies on OWLMap and Ontology2RDB

	Ontologies	Classes	Subclasses and their relationship	Object properties	Datatype properties and their data types	Restrictions	Union	Intersection	Enumeration
1	Pizza/ OWLMap	94%	Yes, 40%	98%	Yes, all	Yes	Null	Null	Null
2	Pizza in Ontology2RDB	96%	Yes,38%	95%	Yes, all	Yes	98%	96%	100%
3	Edu/OWLMap	100%	Yes, 60%	100%	Yes, all	Yes	Null	Null	Null
4	Edu in Ontology2RDB	90%	66%	100%	100%	Yes	100%	90%	100%
5	Trade/OWLMap	100%	Yes, 60%	98%	Yes, all	Null	Null	Null	Null
6	Trade in Ontology2RDB	90%	60%	97%	90%	Yes	90%	100%	100%
7	Travel/OWLMap	98%	Yes, 60%	94%	Yes, all	Yes	Null	Null	Null
8	Travel in Ontology2RDB	95%	80%	100%	90%	Yes	100%	95%	100%
9	Event/OWLMap	100%	Yes, 60%	60%	60%	Null	Null	Null	Null
10	Event in Ontology2RDB	100%	66%	50%	70%	Yes	Null	Null	Null
11	Delegation/OWLMap	96%	Yes, 60%	100%	Yes, all	Yes	Null	Null	Null
12	Delegation in Ontology2RDB	100%	80%	60%	100%	Yes	99%	Null	Null
13	Education/OWLMap	100%	Yes, 60%	20%	70%	Null	Null	Null	Null
14	Education in Ontology2RDB	100%	66%	40%	60%	Yes	90%	100%	100%
15	Car Advertising/OWLMap	100%	Null	100%	Yes, all	Null	Null	Null	Null
16	Car Advertising in Ontology2RDB	98%	30%	80%	100%	Yes	100%	100%	90%
17	Wine/OWLMap	98%	Yes, 40%	100%	Yes, all	Yes	Null	Null	Null
18	Wine in Ontology2RDB	93%	50%	96%	100%	Yes	100%	98%	100%
19	Camera/OWLMap	100%	Yes, 80%	100%	Yes, all	Yes	Null	Null	Null
20	Camera in Ontology2RDB	100%	87%	95%	93%	Yes	Null	100%	100%