

OWLMap: Fully Automatic Mapping of Ontology into Relational Database Schema

Humaira Afzal

Department of Computer Science
COMSATS Institute of Information
Technology,
Lahore, Pakistan

Mahwish Waqas

Department of Computer Science
COMSATS Institute of Information
Technology,
Lahore, Pakistan

Dr. Tabbassum Naz

Department of Computer Science &
IT
University of Lahore,
Lahore, Pakistan

Abstract—Semantic web is becoming a controversial issue in current research era. There must be an automated approach to transform ontology constructs into relational database so that it can be queried efficiently. The previous research work based on transformation of RDF/OWL concepts into relational database contains flaws in complete transformation of ontology constructs into relational database. Some researchers claim that their technique of transformation is entirely automated, however their approach of mapping is incomplete and miss essential OWL constructs. This paper presents a tool called OWLMap that is fully automatic and provides lossless approach for transformation of ontology into relational database format. Number of experiments have been performed for ontology to relational database transformation. Experiments show that proposed approach is fully automatic, effective and quick. Our OWLMap is based on an approach that is lossless as well as it does not loose data, data types and structure.

Keywords—*Semantic Web; Ontology; Database; Mapping; OWL; Jena API*

I. INTRODUCTION

The concept of semantic web is the extension of current web from human readable form to machine processable form by adding semantics. By applying structured information in semantic web, machines are capable to search, process, integrate and present the information in a meaningful and intelligent manner. Conventional search engines dissatisfy users by retrieving inadequate and inconsistent results because they work on predefined standards, terms that work in centralized environment. By semantic and ontology users are able to develop new facts and use their own keywords in different environment [1]. There are different techniques for storing ontology. Ontology can be stored in flat files [2]. But this technique does not provide scalability, query and other functionalities that database system can provide. Ontology repositories are used to hold ontology saved by Ontology management system [3]. But query facility in ontology management system is not as efficient as in relational database system. Relational database system has many advantages as compared to ontology management system like performance, robustness, maturity, reliability and availability. If ontology is stored in relational format then it can easily interoperate with large amount of existing web data. By using SQL, it is easy to retrieve information provided by ontology. If ontology is transformed into relational database then it will make semantic web more useful.

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 [4].

We attempt to explain these problems and provide a solution in the form of OWLMap. The structure of this paper is as follows. In Section 2, previous approaches and their drawbacks have been provided. Section 3, describes proposed methodology. In Section 4, explains the implementation of our approach with the help of case study. Section 5; describe the important phase of testing. Section 6, concludes the main points of paper and give some future directions.

II. RELATED WORK

Reference [5] has purposed an approach for transformation of OWL to ER and vice versa by using conceptual graphs. The transformation is performed step by step, where the first phase is to transform the OWL ontology to ER and second phase is to transform ER to relational database. Reference [6] used “Oracle Semantic data storage” approach for transformation, but most OWL constructs are missing in this approach. Reference [7] suggested the “Storing ontology includes fuzzy data types” approach. Reference [8] purposed “large scale ontology management” approach that covers some constructs of OWL and transformation tool is not fully automatic.

Rule based transformation presented by [9] 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. OWL2DB algorithm is another approach to map OWL documents into relational tables without any human intervention [11]. The transformation is incomplete and it only saves class instances in relational format. Reference [12] proposed an approach for mapping of ontology to relational database. This approach is tested on ontology selected from product configuration domain. This approach covers only a few part of OWL DL syntax. In “Mapping of OWL ontology concepts to RDB Schemas” approach purposed by [13], authors purposed some mapping principles and algorithm. The prototype tool has been added as plug-in for an ontology editor named protégé.

It lacks some mappings like intersection, class complements, union, and property relations.

Reference [4] provides the state of the art for tools in the domain of automatic mapping of ontology into relational databases and highlights the need of fully automatic tool for transformation of ontology in to relational database.

Reference [14] purposed that it is required to have machine learning techniques for semantic mappings. Reference [15] developed a tool named “OntoRel” for transformation. The disadvantage of “OntoRel” was that it only selects few main OWL constructs for transformations. Reference [16] proposed a hybrid approach for reversible and lossless transformation. To improve query capabilities of the thus approach, more research is required.

III. METHODOLOGY

We have proposed a tool called OWLMap that is fully automatic in mapping ontology (OWL) to relational database format. In proposed system for transforming ontology to relational database format, initially a user will select an ontology file, and then information will be extracted about ontology constructs. After extracting this information, proposed mapping rules (given in section C) will be applied automatically to ensure lossless transformation.

A. System Architecture

In suggested approach for automatically convert OWL ontology to relational database format. Fig 1, explains an approach for transforming ontology to relational database format. As the figure depicts, initially, select an ontology file, then information is extracted about ontology constructs using Jena API. Mapping rules are defined to ensure lossless transformation. Based on the mapping rules, transformation of ontological constructs into relational database takes place. The main focus is to develop a tool from OWL ontology to relational database that is fully-automatic and can solve various problems from the previous approaches.

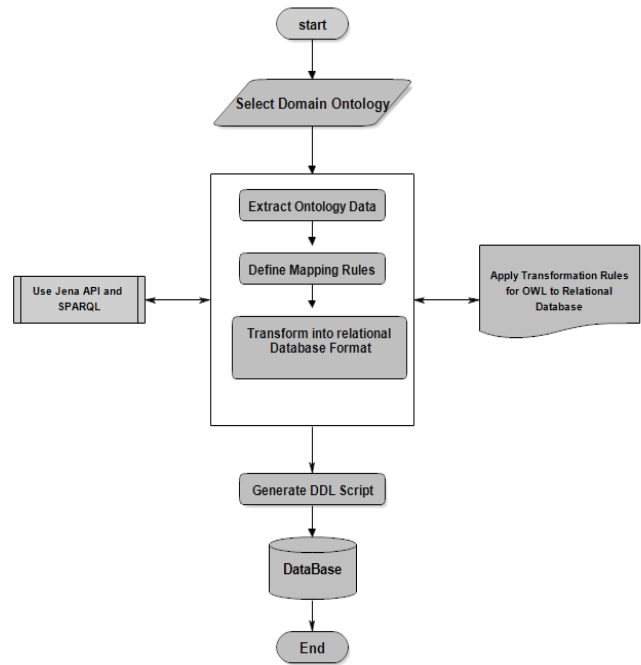


Fig. 1. System Architecture for Ontology Transformation to Relational Database (OWLMap)

B. Transformations Process for OWL to RDB

An algorithm has been developed to map ontology constructs into relational database format. Following that algorithm the transformation process is given below.

- 1) The given ontology or OWL file is first parsed to get Classes. Root class, super classes and subclasses are extracted.
- 2) Next Jena methods are applied to get two types of OWL properties i.e. object and data type. Data types of properties and restrictions are extracted as well.
- 3) Build database connection and transform this information into relational database format.

4) Classes and subclasses are transformed into separate tables and create one-to-one relationship among association classes according to mapping rules.

5) Then map properties as attributes of tables associated with corresponding class according to mapping rules.

6) Some properties are transformed into separate tables like multi-valued properties and properties having sub-properties.

7) Create separate metadata tables to store information about property restrictions.

8) Finally, ontology constructs are transformed into relational database.

C. Mapping Rules

Mapping rules are used to transform ontology to relational database. This section provides all the mapping rules used in OWLMap to transform ontology in to a database format. According to these rules, classes in ontology are transformed into relational database tables. The object type properties in selected ontology are transformed into columns or tables according to their relationship. Ontology data type properties are also transformed into columns or tables according to their values (single value or multi value).

1) OWL Classes:

Rule 1: Each OWL Class (Subclasses and association classes) will be transformed into a table in Relational database. Class name will become Table name. Table will be allocated a primary key. A table that relates to subclass is assigned a primary key a foreign key that reference to its "Super table" (one to one relationship between tables in relational database)

2) OWL Properties:

Rule 2: Single valued and functional object type property will be mapped into a foreign key in the table. The domain of the object property becomes the table. The range of the object property becomes another table. The name of the object property will be the name of foreign key linking two tables

Rule 3: Single valued and inverse of object type property will be mapped into a foreign key in the table that relates to range of object property and this key reference to primary key in the table that corresponds to the class specified as domain of object property. The name of the inverse object property will be the name of foreign key.

Rule 4: If Object type properties are multi-valued then they will be mapped into a separate table and will be assigned a primary key that's a combination of two foreign keys. One foreign key references the primary key of domain table and other to range table. The name of the object property will be the name of Table.

Rule 5: If object type properties are further divided into sub properties then they will be transformed into a table and their sub properties will be mapped into columns of that table. The name of the super property will be the name of table by adding prefix Prop_.

Rule 6: Single-valued data type property will be mapped into a column in the table that relates to the domain of data

type property. The data type property's name will become the name of the column.

Rule 7: Multi-valued data type property will be mapped into a table and will be assigned a primary key that is a combination of corresponding column and the foreign key that reference to the domain table of data type property. The data type property's name will become the name of the Table

Rule 8: If data type properties are further divided into sub properties then they will be transformed into a table and their sub properties will be mapped into columns of that table. The name of the super property will be the name of table by adding prefix Prop_.

Rule 9: Data Type Conversion of Data type properties: We have converted data types of data type property from XSD to SQL, because OWL uses XSD data types. TABLE 1, shows how to convert different data types from XSD to SQL.

TABLE I. CONVERSION OF DATA TYPES TO DOL FROM XSD

XSD data Types	SQL Data Types
Short	SMALLINT
Integer	INTEGER
Negative Integer	INTEGER
Nonnegative Integer	INTEGER
Unsigned Int	INTEGER
Integer	INTEGER
Negative Integer	INTEGER
Nonnegative Integer	INTEGER
Unsigned Int	INTEGER
Long	INTEGER
Unsigned Long	INTEGER
Decimal	DECIMAL
Float	FLOAT
Double	DOUBLE PRECISION
String	CHARACTER VARYING
Normalized String	CHARACTER VARYING
Token	CHARACTER VARYING
Language	CHARACTER VARYING
NMTOKEN	CHARACTER VARYING
Name	CHARACTER VARYING
NC Name	CHARACTER VARYING
Time	TIME
Date	DATE
Datetime	TIMESTAMP
gYearMonth	DATE
gMonthDay	DATE
gDay	DATE
gMonth	DATE
Boolean	BIT
HexBinary	CHARACTER VARYING
AnyURI	CHARACTER VARYING

3) OWL Restrictions:

To preserve all information about ontological constraints, this information is stored in Meta data tables. Every type of restriction has its own table.

Rule 10: Some values from restriction maps to table having columns, restriction class (this column points to the table of the related restriction resource class), property (includes the property concerned), domain class and range class of property. All values from restriction maps to table having columns, restriction class (this column points to the

table of the related restriction resource class), property (includes the property concerned), domain class and range class of property. Has value restriction maps to table having columns, restriction class (this column points to the table of the related restriction resource class), oN property (includes the property concerned), domain class and range class of property. In case of “Has value Restriction” Meta data table, a column “value” is added for storing the value of restricted resource of related property.

Rule 11: Inverse functional property will be mapped to unique constraint on the corresponding column. And required Property will be mapped on the corresponding column as Not Null Constraint.

IV. EXPERIMENT

Number of experiments has been performed for ontology to relational database transformation using OWLMap. Different ontologies from multiple domains are presented to our OWLMap tool for transformation. Experiments show that proposed approach is fully automatic, effective and quick. This approach is lossless as well and performs the transformation successfully.



Fig. 2. Class hierarchy available in pizza ontology

In this section, we have presented automatic transformation process taking famous “Pizza Ontology” as an input. Pizza ontology is downloaded from standard Website of Stanford University. Pizza ontology is developed at

Manchester. It has often been considered as important ontology for learning basic concepts of ontology and OWL language [18] and [19]. This ontology has been chosen because pizzas are widely understood in all cultures or across the world. The pizza ontology includes most of OWL features. These ontology concepts are used to present main components of pizza domain, as illustrated in fig 2. This ontology has number of Object type properties, data type properties and their sub properties, shown in fig 3 and 4.

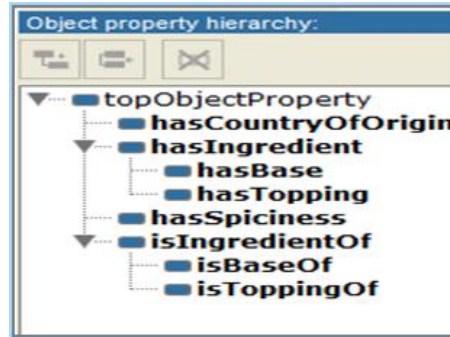


Fig. 3. Object type properties in pizza ontology

First, download the pizza ontology, explore it with the help of Protégé and check its consistency by using reasoner. Next, with the help of Jena API, extract all the information about ontological constructs e.g. classes, sub classes, object properties, data type properties, their domain and range, data types, restrictions etc. according to mapping algorithm.

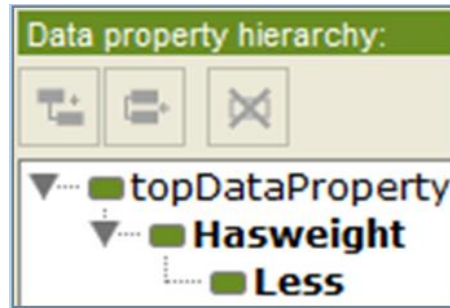


Fig. 4. Data type properties in pizza ontology

In the next step, choose SQL Server 2008 R2 to store this information in relational database format. Database is created and connection with the database is established. Then implement defined mapping rules to transform ontology into relational database format. According to rule 1, transform all ontology classes into tables in RDB and class name become table name, assign a primary key as shown in fig 5. A table that relates to subclass is assigned a primary key and a foreign key that reference to its “Super table” (one to one relationship between tables in relational database) as shown in fig 6.

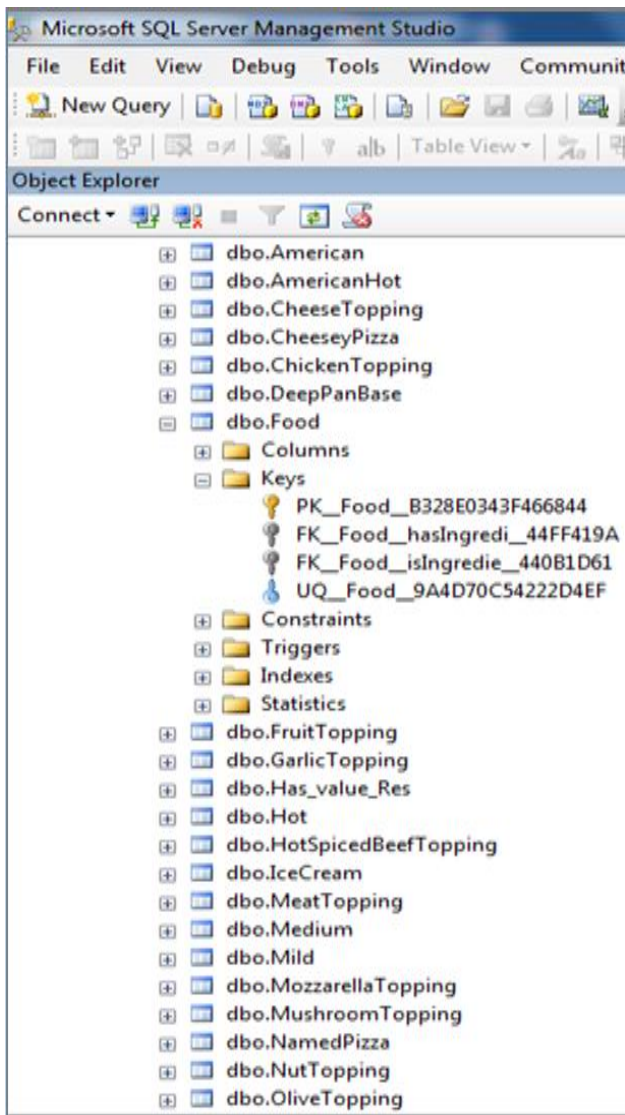


Fig. 5. Conversion of ontology classes into tables in RDB

Fig 6, explains the one to one relationship between class and its related subclass. In Pizza ontology “Pizza class” is constructed as subclass of “food class”. So when Pizza class is transformed into relational database format, a Pizza table is created with one to one relation to its super class table called Food. This mapping rule is applied in all association classes while transforming into relational database format.

After mapping classes and subclasses into relational database format, transformed object type properties according to defined mapping rules. According to mapping rules, transform single valued and functional object type property into foreign key in the table that relates to domain of object type property and this key reference to primary key in the table that relates to the range class of object type property as illustrated in fig 7 and 8.

In fig 7, “Has Spiciness” is functional property and has specified “Spiciness” class as domain and range of this property. The object type property transforms to a foreign key in the table. That table represents a class specified as the domain of the object property. In fig 8, the key that reference the primary key in the table is related to the class specified as the range of object property. We have transformed object type property that is “inverse functional” and single valued into a foreign key.

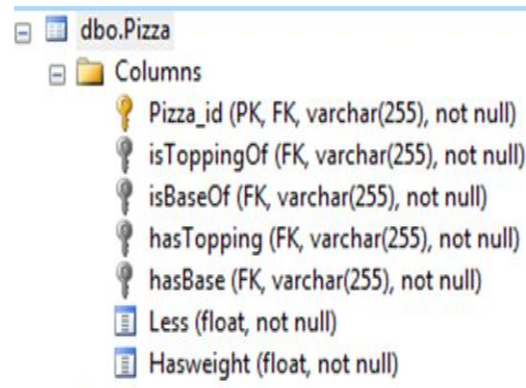


Fig. 6. Conversion of ontology subclass into table in RDB

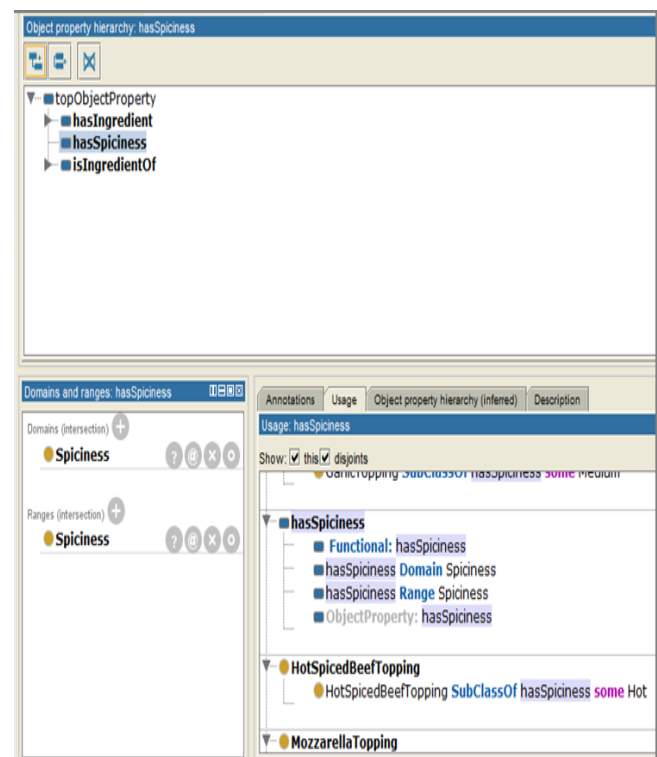


Fig. 7. Functional object type property in pizza ontology

The table that relates to range class of object type property and this key reference to primary key in the table that relates to domain of object type property as illustrated in fig 9 and 10.

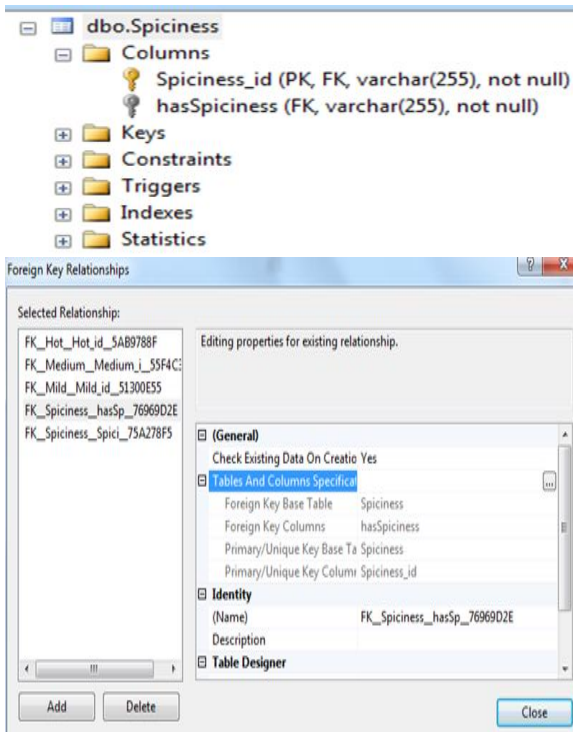


Fig. 8. Transformation of single valued and functional object property

In fig 9, “is Base of” object type property is inverse of “has Base” object type property. If object property has sub properties e.g. “has Ingredient” has two sub properties “has Base” and “has Topping” then super property maps to a table in relational database and its sub properties maps to a column in corresponding table as shown in fig 11.

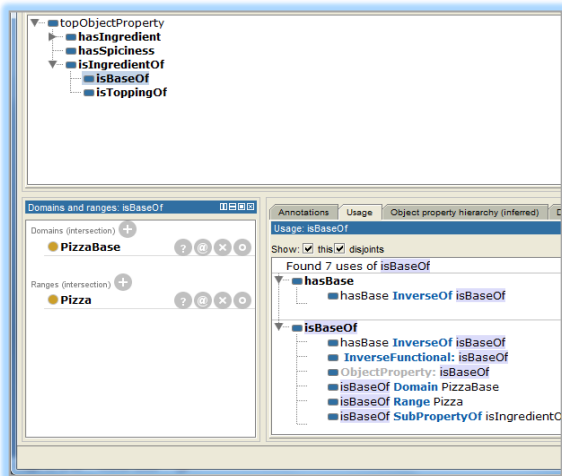


Fig. 9. Inverse functional and single valued object property

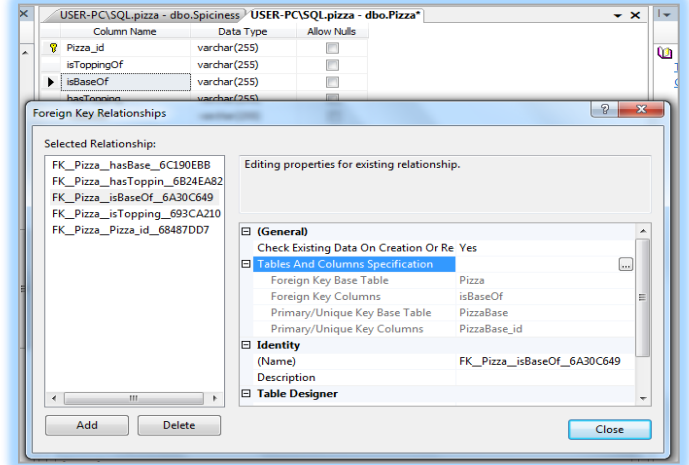
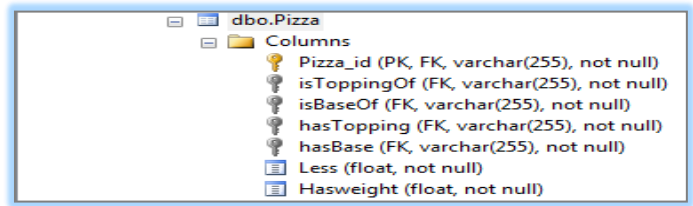


Fig. 10. Conversion of object function property

After mapping object type properties, OWLMaptransfers data type properties and their data types into relational data base format. If data type property is single valued e.g. In Pizza ontology “has Price” is a data type property and has specified “Pizza” class as its domain and has data type “float”. In fig 12, the property in ontology maps to a column in the table that relates to the class specified as domain of data type property. In table 1, the column data type which is specified as range of data type property converted from XSD data type to SQL data type. If data type property has sub properties e.g. in pizza ontology “has Weight” data type property has sub property “Less”, so this property maps to a table in relational database and its sub properties maps to a column in related table as shown in fig 13. While converting OWL ontology into relational database format, we want to preserve all information of ontological constraints. For this purpose we have saved this information in special Meta data tables. Each type of restriction has its own table as explained above in mapping rules.

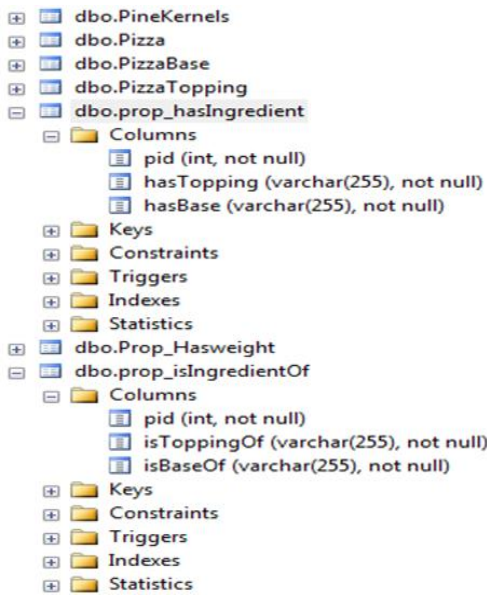


Fig. 11. Conversion of object type property that is further divided into sub properties

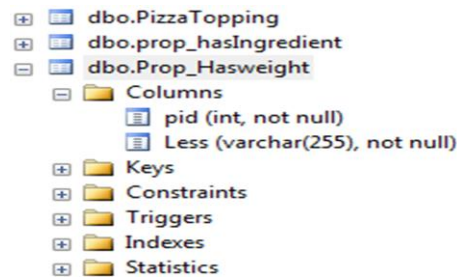


Fig. 13. Conversion of data type property that is further divided into sub properties

Some_id	ResClass	Prop	Propdomain	PropRange
1	1	Pizza	hasBase	Pizza
2	2	OliveTopping	hasSpiciness	Spiciness
3	3	GarlicTopping	hasSpiciness	Spiciness
4	4	NutTopping	hasSpiciness	Spiciness
5	5	MushroomTopping	hasSpiciness	Spiciness
6	6	ChickenTopping	hasSpiciness	Spiciness
7	7	AmericanHot	hasTopping	Pizza
8	8	MozzarellaTopping	hasSpiciness	Spiciness
9	9	IceCream	hasTopping	Pizza
10	10	American	hasTopping	Pizza
11	11	HotSpicedBeefTopping	hasSpiciness	Spiciness
12	12	SultanaTopping	hasSpiciness	Spiciness

Fig. 14. Meta data table “some values from restrictions” after conversion

Fig 14, shows some values from the restrictions are mapped to table having columns restriction class (this column points to the table of the related restriction resource class), property (includes the property concerned), domain and rRange. In case of “Has value Restriction” when we have created Meta data table, a column “value” is added for storing the value of restricted resource of related property.

V. TESTING

To test the performance of OWLMap tool, ten different ontologies have been taken from standard web site of Stanford University. These ontologies have different sizes, and from different domains. Testing of developed tool with different types of ontologies increases its efficiency and reliability. In table 1; some important specifications of our machines are given. These components play vital role in process of testing.

TABLE II. MACHINE SPECIFICATIONS

Processor	CPU speed	OS	Memory	System type
Intel core i5	2.40GHz	Windows 7	4 GB	64 bit operating system

We have also observed the time required for conversion from ontology to database by developed tool. It is observed that conversion time is different for large and small ontologies. For large ontologies, it takes about 25 seconds and

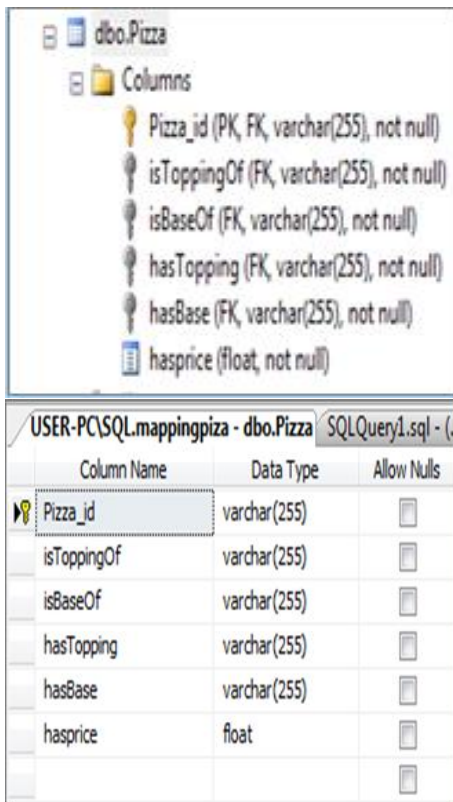


Fig. 12. Conversion of single valued data type property and its data type

for small ontologies it takes only 10 seconds in transformation. In table 2, detailed information about sample ontologies and their components are given. This information plays an important role while testing of given approach. It

becomes very easy to check the efficiency of developed tool by comparing this information.

TABLE III. DETAIL ABOUT SAMPLE ONTOLOGIES

Sample Ontology	Description	Number Of Classes	Object Type Properties	Data Type Properties	Restrictions
Pizza	About pizza and its components	>80	7	3	Yes
University	About university domain	29	6	2	Yes
Camera	About camera and its components	11	7	8	Yes
Trade	Define trade system	>70	20	3	No
Travel	Defines tourism	33	6	4	Yes
Event	About event handling	>60	7	10	No
Delegation	About management	18	5	3	Yes
Education	Defines education domain	36	12	>30	No
Car Advertising	About car advertising	10	3	8	No
Wine	About wine components	>15	3	1	Yes
Db1	About Library system	7	3	14	No
Docdb	About hospital	4	4	>10	No

TABLE IV. EVALUATION OF SUGGESTED TOOL WITH THE HELP OF SAMPLE ONTOLOGY

Converted ontology concepts	Classes	Subclasses and their relationship	Object properties	Data type properties and their data types	Restrictions
Pizza Ontology	94%	Yes, 40%	98%	Yes, all	Yes, converted to Metadata tables
Edu Ontology	100%	Yes, 60%	100%	Yes, all	Yes, converted to Metadata tables
Trade Ontology	100%	Yes, 60%	98%	Yes, all	Null
Travel Ontology	98%	Yes, 60%	94%	Yes, all	Yes, converted to Metadata tables
Event Ontology	100%	Yes, 60%	60%	60%	Null
Delegation Ontology	96%	Yes, 60%	100%	Yes, all	Yes, converted to Metadata tables
Education Ontology	100%	Yes, 60%	20%	70%	Null
Car Advertising Ontology	100%	Null	100%	Yes, all	Null
Wine Ontology	98%	Yes, 40%	100%	Yes, all	Yes, converted to Metadata tables
Camera Ontology	100%	Yes, 80%	100%	Yes, all	Yes, converted to Metadata tables

VI. CONCLUSION AND FUTURE WORK

As the semantic Web is gaining importance, there is a need of an efficient approach to map all ontology information into relational database so that it can be queried easily. Therefore, we have developed a tool OWLMap for automatic and lossless transformation of ontology into relational database. This transformation approach can map all the constructs of ontology including “sub properties” that are not handled before in any transformation approach. For lossless transformation of ontology constructs into relational format, mapping algorithm and rules are defined. According to these mapping rules, ontology classes should be transformed into tables, object type properties should be mapped into columns or tables, data type properties should be mapped into columns or tables according to mapping rules and restrictions must be stored into meta data tables.

OWLMap can help in mapping ontological data into relational databases that can be further utilized in different applications. It is easy to access heterogeneous and distributed information. This approach will play an important role in Advance Querying / Query Optimization. Other benefits of this approach are quick retrieval of data and schema Integration. This approach is capable to automatically transform most of ontology constructs into relational structure. In future, it is required to transform some other ontological information i.e. Class complements, comments, and enumerated or intersection classes. Reflexive and Irreflexive properties of OWL also need some attention in future.

REFERENCES

- [1] V. Jain, and S. V. A. V. Prasad, “Mapping Between RDBMS And Ontology: A Review”, International journal of scientific & technology research volume 3, issue 11, 2014.
- [2] D. Moldovan, M. Antal, D. Valea, C. Pop, T. Cioara, I. Anghel, and I. Salomie, “Tools for Mapping Ontologies to Relational Databases: A Comparative Evaluation”, In Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference, pp. 77-83.
- [3] I. Astrova, and A. Kalja, “Storing OWL Ontologies in SQL3 Object-Relational Databases”, In AIC’08: Proceedings of the 8th conference on Applied informatics and communications, pp. 99-103.
- [4] H. Afzal, T. Naz, and A. Sadiq, “A Survey on Automatic Mapping of Ontology to Relational Database Schema”, Research Journal of Recent Sciences, Vol. 4(4), 66-70, 2015.
- [5] S. H. Tirmizi, J. Sequeda, and D. Miranker, “Translating SQL Applications to the Semantic Web”, In Proceedings of the 19th International Conference on Database and Expert System Application: 450-464, 2008.
- [6] Z. Wu, G. Eadon, S. Das, E. I. Chong, V. Kolovski, M. Annamalai, and J. Srinivasan, “Implementing an Inference Engine for RDFS/OWL Constructs and User-Defined Rules in Oracle”, In Proceedings of IEEE 24th international Conference on Data Engineering: pp. 1239-1248. Mexico, Cancun, 2008.
- [7] C. D. Barranco, J. R. Campana, J. M. Medina, and O. Pons, “On storing ontologies including fuzzy datatypes in relational databases”, In Proceedings IEEE International Conference on Fuzzy System, July 23-26, 2007: 1-6.
- [8] R. Goodwin, and J. Y. Lee, “Ontology Management for Large Scale Enterprise”. Electronic Commerce Research and Applications5 (1): 2-15, 2006.
- [9] N. Zina, and N. Kaouther, “Automatically building database from biomedical ontology”
- [10] I. Astrova, N. Korda, and A. Kalja, “Storing OWL Ontologies to SQL Relational Databases”, International Journal of Electrical, Computer, and Systems Engineering 1(4), 2007.
- [11] A. Gali, C. X. Chen, K. T. Claypool, and R. Uceda-Sosa, “From ontology to relational databases”, In Proceedings of International Workshop on Conceptual-Model Driven Web Information Integration and Mining: 278-289. Shanghai, China, 2005.
- [12] E. Vysniauskas, and L. Nemuraite, “Transforming Ontology Representation from OWL to Relational Database”, Information Technology and Control, 2006, 35(3A): 333-343.
- [13] E. Vysniauskas, and L. Nemuraite, “Mapping Of OWL Ontology Concepts To RDB Schemas” In proceedings of the 15th International Conference on Information and Software Technologies Kaunas, Lithuania, April 23-24, 2009: 317-327.
- [14] W. Hu, and Y. Qu, “Discovering Simple Mappings between Relational Database Schemas and Ontologies”, In Proceedings of the 6th international the semantic Web and 2nd Asian conference on Asian semantic Web conference: 225-238, 2007.
- [15] D. D. B. Saccol, T. D .C. Andrade, and E. K. Piveta, “Mapping OWL Ontologies to Relational schemas”, In proceeding of IEEE International Conference on Information Reuse and Integration (IRI), 2011.
- [16] E. Vysniauskas, L. Nemuraite, R. Butleris and B. Paradauskas, “Reversible Lossless Transformation from OWL 2 Ontologies into Relational Database”, Information Technology and Control (4), 2011.
- [17] G. Antoniou, P. Groth, F. V. Harmelen, and R. Hoekstra, “Semantic Web Primer”, MIT press/London, 2012.
- [18] M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H. H. Wang, “The Manchester OWL Syntax”
- [19] R. Sivakumar, and P.V. Arivoli, “Ontology Visualization Protégé Tools-A Review”, International Journal of Advanced Information Technology 1, no. 4, 2011.