Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Fully automatic OWL generator from RDB schema

Tabbasum Naz [1,*], Maham Shuja [2], Syed Khuram Shahzad [3], Muhammad Atif [1]

[1]Department of Computer Science and Information Technology, The University of Lahore, Lahore, Pakistan
[2]Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan
[3]Department of Computer Science and Information Technology, The Superior University, Lahore, Pakistan

ARTICLE INFO

Article history:
Received 14 November 2017
Received in revised form
29 January 2018
Accepted 10 February 2018

Keywords:
OWL generator
Semantic web
Schema mapping
DB to OWL

ABSTRACT

Use of ontologies in information systems and artificial intelligence has been emphasized in the recent years. Other than standardizing the vocabulary across a domain, ontologies enable the sharing of information between disparate systems within the same domain. Ontology engineers spend a lot of efforts in developing ontologies. A large amount of data on the Web is stored in the relational databases. In this paper, we have proposed and developed a tool that can fully automatically develop OWL ontology from a relational database. The main focus of our research is to develop a transformation process and to create rules for mappings between RDB and OWL constructs. Existing approaches have drawbacks that they are not fully automatic, performed mapping at a very basic level, outdated and are not easily accessible. In case of a large database, the existing tools fail to perform conversions efficiently. Our proposed tool is evaluated on different relational databases and can successfully perform the transformation with new mapping rules. Our tool is able to develop sub-data-properties and sub-classes which was never available before.

1. Introduction

World Wide Web contains a lot of information. Some of this information is useful where as other is not required. Now extracting only the required information from this big pool, we call web is a very difficult task. Available search engines can provide some information but user still has to go through all the data manually to get the required information. The reason behind it is that, the data on the web is not stored in a formal way. W3C introduced the concept of semantically arranged data to overcome the above problem.

In Semantic web, we can organize the data in machine understandable format that makes aggregation and combination of existing web information very easy. By using ontology, it is easy to capture the knowledge of a specific field and provides a common understanding of field knowledge. Ontology defines a domain; it comprises a list of classes, subclasses, and their relationships. Manual ontology engineering is extremely labor-intensive task. Xiang et al. (2015) used ontology design patterns for automatic generation of ontology terms, annotations and axioms. Erling and Mikhailov (2006) discussed the importance of Meta schema language for mapping SQL Data to RDF Ontologies. Through Virtuoso's declarative, they have developed a process that results in RDF Data sets and optimized data access without physical regeneration of RDF Data Sets from SQL Data.

The existing data stored on the web is mostly stored in Relational Databases format. To get structured and semantically arranged information, we need ontologies. The problem arises in conversion of relational database to ontology is that how to map database schema to ontology's constructs. There are problems in existing manually mapping approaches because it is time consuming process and requires oncologists. Ontologists have to spend a lot of time in database to ontology transformation. Recent research has shown a number of approaches and proto-type tools in this domain, but they have some problems. Some of these are semi-automatic which need human effort for conversion and some are automatic but conversion process is incomplete. To overcome the above given problem and get the advantage of semantically arranged data in the form of ontology, we purpose a mapping approach based on detailed mapping rules so that there is no loss of data in conversion procedure. To reduce the manual effort, we have

* Corresponding Author.
Email Address: tabbasum.naz@cs.uol.edu.pk (T. Naz)
https://doi.org/10.21833/ijaas.2018.04.010

developed a fully automatic tool for Relational Databases to ontology conversion.

This paper is structured into subsequent sections: Section 2 describes the existing work; Section 3 explains the fundamentals of domain. System architecture and algorithm is explained in Section 4. Section 5 explains mapping rules. Section 6 contains a case-study. Testing results are provided in Section 7. Finally, section 8 concludes our work and provides some future recommendations.

2. Related work

Understanding the importance of database to ontology conversion, a lot of research work is in progress. Recent research has shown a number of approaches and proto-type tools, having some drawbacks. Some approaches are automatic and others are semi-automatic which involve human effort. Some of the existing approaches are discussed in subsequent section.

Banu et al. (2011) has suggested an approach on ontology extraction and online query retrieval. But the proposed rules are of very basic level. Cerbah (2008) has implemented a semi-automatic ontology generation tool. But mapping rules and working of the tool is not discussed in detail. Tirmizi et al. (2008) has proposed a conversion approach that uses SQL DDL language. By using this approach, schema is automatically extracted for ontology building. But according to them for an accurate ontology generator human interaction is important, which makes the approach manual. Zhang and Li (2011) have proposed an approach to automatically generate ontology from relational database. But there exist a difference between the automatically generated and manually generated ontology. O'Connor et al. (2010) has proposed a conversional approach by using spreadsheets. The tool named "Mapping master" is implemented as a protégé plug-in and the conversion procedure is executed in three steps but it is not complete transformation. Ra et al. (2012) has proposed the methodology that is a combination of two approaches; therefore it is named as "Mixed Ontology Building Methodology (MOBM)". The drawback of this approach that mapping is not automatic and needs a lot of human effort to retrieve information. Saleh (2011) has suggested offline ontology extraction and online query issuing. In his proposed method, there is an issue in query issuing because it do not support SPARQL syntax and maps very limited data. Cullot et al. (2007) has implemented a tool DB2OWL. But developed applications only map tables and columns. Gherabi et al. (2012) has proposed another tool to migrate database to ontology and has suggested a method that is divided into three phases. Again the proposed method has limitations. Another problem with few already implemented prototypes e.g., "Mapping master", "RDBtoOnto" and "relational-owl" are that, these tools are not accessible. Zhou et al. (2011) has proposed a semi-automatic method of converting database schema to ontology, and used "Word Net" to extend the extracted ontology data. Again main issue is that it is not automatic.

3. Preliminaries

This section briefly introduces the relational databases and semantic web, which will be helpful in understanding the rest of the paper.

The relational databases are the most popular storage tools and are widely used in all the fields. Data is stored and accessed in the form of tables; rows and columns. Data can be inserted, accessed, updated and deleted from the tables of RDB.

The table refers to relation, row refers to tuple and column refers to attribute in relational database. RDBs can store large amounts of data that is why most of the web data is stored in RDBs. The structured query language (SQL) is used for storing and accessing data from a relational database. Relational databases are easy to create access and extend. To ensure that the data in the RDB is accurate, referential integrity rules are applied.

Semantic web is an extension of ordinary web. It provides a standardized way of expressing the relationships between web pages, to allow machines to understand the meaning of hyperlinked information. Semantic Web refers to W3C's vision of the Web of linked data. Ontology is the basic concept of semantic web.

According to Antoniou and Van Harmelen (2004), *"Ontology is an explicit and formal specification of a conceptualization"*. Protégé is a free, open source ontology editor and knowledge-based framework. In Protégé ontologies can be developed in a variety of formats including OWL, RDF(S), and XML Schema.

4. Research methodology

In one of our paper, we have provided the survey of existing approaches for ontology to relational databases (Shujah et al., 2015). We have seen that ontology from RDBs can be developed in many ways. Some approaches first create databases and then create ontology, others uses a global ontology and converts existing databases into that global ontology. In our approach, we use an existing database, which is used to create an ontology automatically based on the database.

Fig. 1 shows the architecture of our tool, which automatically constructs ontology from an existing database. The main focus of our research is to create rules for mappings between RDB and OWL constructs. For constructing these rules, we first apply some mapping rules for database to ontology generation. Second, we construct ontology of our sample database.

4.1. Transformation process

For better understanding of our proposed tool, transformation process from database to OWL is explained below.

1. Sample database (MS Access) is browsed and selected.
2. Connection is established with the database.
3. After successful connection with database, database schema and meta-data is extracted and saved for further processing. Extracted schema contains tables, primary keys, foreign keys (exported keys, imported keys), corresponding parent tables of these imported and exported keys, columns and column data-types.
4. We have defined few mapping rules to transform database schema to ontology. Mappings are performed based on these rules. OWL API is used for database schema transformation to OWL ontology.
5. Tables from RDB schema are converted to corresponding ontology classes.
6. Primary keys are converted to corresponding functional data-type properties in ontology. Domain and range are set accordingly.
7. Foreign keys are converted to corresponding object properties in ontology. Domain and range are set accordingly.
8. Columns are converted to corresponding data-type properties in ontology. Domain and range are set accordingly.
9. Sub properties are established.
10. Sub-classes are established.
11. After applying the rules transformed OWL ontology is generated, which can be navigated in any OWL editor i.e. Protégé.

5. Mapping rules

As explained earlier, our main focus is on creating the fully automatic mapping rules for generation of OWL ontology from RDB. Therefore, in this section we will discuss the mapping rules in detail. An example database shown in Table 1 is used for better understanding of the mapping rules.

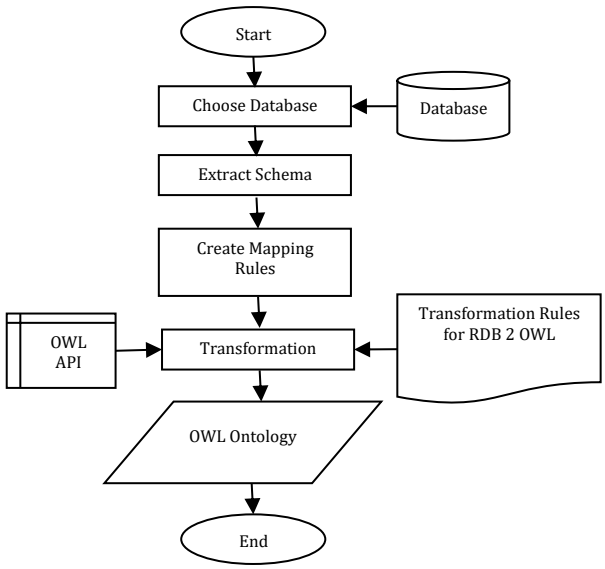

Fig. 1: System architecture to conver database to OWL ontology

5.1. Rule 1: Conversion of tables

Every table in relational database should be converted to class in ontology. Example: if we look in above database the tables Suppliers, Products, Categories, Employees, Customers and Orders are the tables, which will be converted to classes in ontology. Name of class remain same as that of the corresponding table.

5.2. Rule 2: Conversion of foreign keys

The foreign keys are converted to object type properties in ontology.

Table 1: Example database

Relation	Primary key	Foreign key
Suppliers (supplierID, companyName, address,...fax)	SupplierID	NA
Products (productID, categoryID, supplierID,quantityPerUnit)	ProductID	SupplierID (refers to suppliers), CategoryID (refers to categories)
Categories (categoryID, categoryName, description)	CategoryID	NA
Employees (employeeID, lastName, firstName,..., homePhone)	EmployeeID	NA
Customers (customerID, companyName,..., fax)	CustomerID	NA
Orders (orderID, customerID, employeeID,..., shipCountry)	OrderID	CustomerID (refers to customers), EmployeeID (refers to employees)
OrdersDetail(orderID, orderDid,.., quantityPerUnit)	OrderId	OrderID (refers to Orders)

The foreign keys in Table 1, which correspond to primary key of Table 2, are converted to object type properties in ontology. Class 2 corresponding to Table 2 is the domain, and class 1 corresponding to Table 1 is the range of this object property. Properties may have a domain and a range specified. Properties link individuals from the domain to individuals from the range (Antoniou and Van Harmelen, 2004). The table exporting the foreign key is usually the domain whereas the table which imports the foreign key is the range. Name of the property is same as that of the corresponding foreign key with a "has".

Example: If we look in above database the table Products contain two foreign keys i.e., SupplierID and CategoryID. The foreign key SupplierID corresponding to primary key of Supplier table is converted to object property has SupplierID. With class Supplier as its domain, and class Products as its range. As, table Suppliers is exporting foreign key SupplierID and table Products is importing foreign key SupplierID.

In the same way for CategoryID we will create object property has CategoryID, having Categories class as its domain and Products class as its range. As, table Categories is exporting foreign key CategoryID and table Products is importing foreign key CategoryID.

5.3. Rule 3: Conversion of columns

The columns are converted to data type properties in ontology.

All the columns in Table 1, which do not fulfill rule2, are converted to data type properties in ontology. Class1 corresponding to Table 1 is the domain, and data type of the column is range of the data type property. Name of the property is same as that of the corresponding column with a "has".

Example: If we look in above database and consider the table products. Other than SupplierID and CategoryID columns, which are converted to object properties, all the remaining columns are converted to data type property. The column ProductID of Table 1 is converted to data type property has ProductID, with class Products as its domain and integer as its range. In the same way column quantity PerUnit is converted to have quantity PerUnit data type property, having class Products as its domain and string as its range.

5.4. Rule 4: Conversion of primary keys

The primary keys are converted to functional data-type properties in ontology.

The primary key of Table 1 is converted to functional data type property in ontology. Class1 corresponding to Table 1 is the domain and data type of the corresponding column is its range.

Example: In above database consider the table Suppliers. Suppliers have a primary key SupplierID. This primary key is converted to functional data type property has FuncSupplier ID with class Suppliers as its domain and integer as its range.

5.5. Rule 5: Creating sub-data-type properties

The most common data-type properties will be used for creating sub-properties. Like region, city, country, street, are almost present in every database, will become sub-property of data-type property has Address.

Example: if we look in above database the table Employees contain lastName and firstName. They can be created sub-property of data-type property has Name.

5.6. Rule 6: Creating sub-classes

If Table 1 has foreign key FK1, that refers to an attribute in Table 2. And, Table 1, Table 2 corresponds to class1, class2 in ontology. Then, class1 will be sub-class of class2 only if the name of class1 contains substring from the name of class2.

Example: the table OrdersDetail contains foreign key OrderID that refers to the table Orders. Also "OrdersDetail" contains substring "Orders" that matches string from table Orders. Therefore we'll create OdersDetail sub-class, of class Orders.

6. Case study

This section explains the transformation procedure with the help of case study. The database we are taking as an example (Northwind) taken from sample templates of Microsoft Access. We preferred an Access sample database to avoid ambiguity, incompleteness and incorrectness of database.

A Northwind database demonstrates how MS Access can manage small business with tables i.e., Categories, Customers, Employees, Orders, Order Details, Products, Shippers and Suppliers. The Northwind database contains eight tables. In a relational model, data is stored in relations. Relation is another term used for table. Table 2 shows the meta-data for the Northwind database. The table is further divided into number of columns which are also called attributes. Each table has a primary key. A primary key is chosen by the database designer to identify tables uniquely within a database. There are rules to be kept in mind while creating a primary key.

In Fig. 2 relationship diagram shows how tables are related to each other. These tables use foreign keys to relate to other tables. A foreign key is an attribute or combination of attributes in a table that reference a primary key in another relation. The key connects to another table when a relationship is being established between two tables. A table may contain many foreign keys.

As the conversion starts, our tool extracts the meta-data of selected database (Northwind database in our case study). The meta-data is stored for further use. Extracted metadata contains all the table names, their primary keys, imported keys and table from which it is imported, exported keys and table to which it is being exported, columns and also the data type of these columns.

Fig. 3 shows the extracted meta-data by our tool. Next step is to use the extracted meta-data and convert it into OWL ontology. In OWL conversion part of our tool, the tables are first converted to ontological classes. All the tables from meta-data are converted to OWL classes, as shown in Fig. 4.

Object properties are important part of ontology. We can see in Fig. 5 that foreign keys are being converted to object properties of the ontology. Appropriate domain and range is assigned to the object properties. All the columns other than the foreign key attributes and primary key attributes of database are transformed into data type properties. As shown in Fig. 6, for table Suppliers, the columns Address, Country, Region etc. all are converted to data type property. If we see data type property has Address, the figure below shows its range is string and domain is Suppliers, Employees and Customers. As another example in Fig. 6, if we see data type

property has Discount, range is integer and domain is Order Details.

The data types from database to ontology are also handled carefully. If required, data type sub-properties are also created, by combining similar properties under one property. As shown in Fig. 6, has Address, has City, has Country and has Region are sub-properties combined under one property named has area.

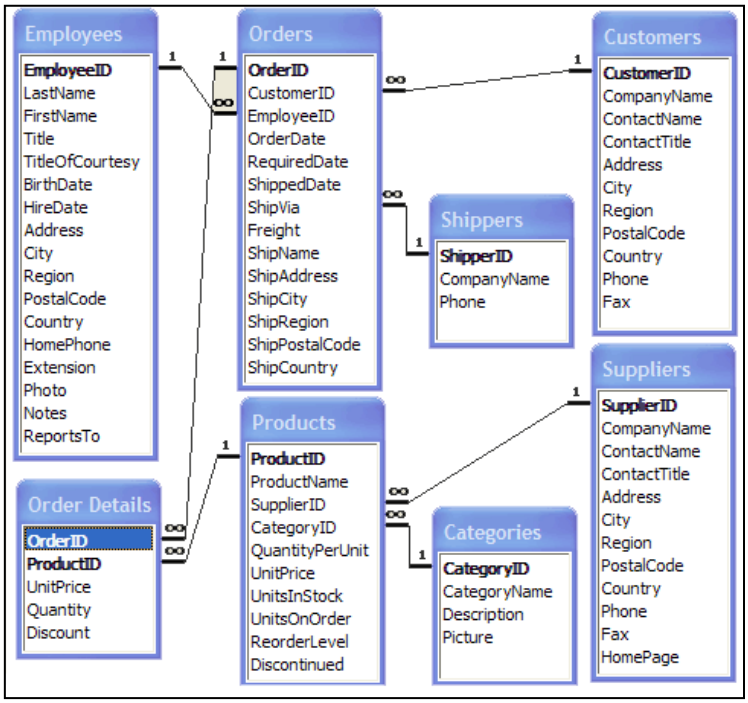

Fig. 2: Relationship diagram of northwind database of MS Access

Table 2: Meta data of the northwind database

Table name	No. of columns	No. of PKs	Primary key	No. of FKs	Foreign key
Categories	4	1	CategoryID	-	-
Customers	11	1	CustomerID	-	-
Employees	17	1	EmployeeID	-	-
Order Details	5	2	OrderID, ProductID	2	OrderID, ProductID
Orders	14	1	OrderID	3	EmployeeID, CustomerID, ShipVia
Products	10	1	ProductID	2	CategoryID, SupplierID
Shippers	3	1	ShipperID	-	-
Suppliers	12	1	SupplierID	-	-

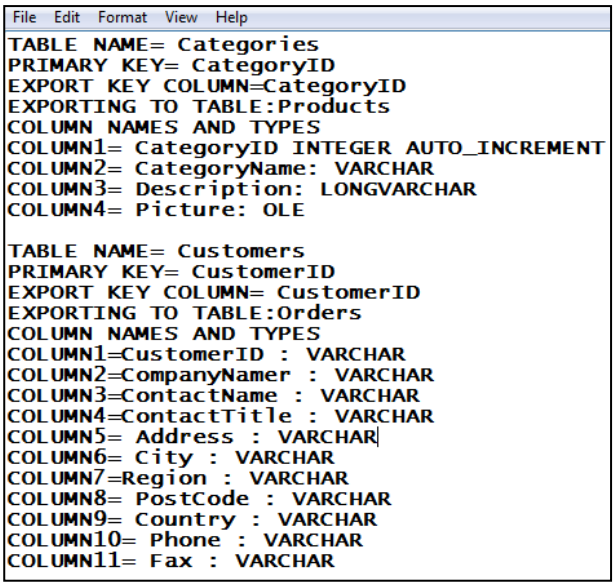

Fig. 3: Meta data extracted from the north wind database by our tool

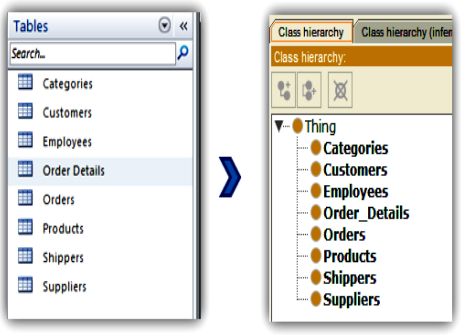

Fig. 4: Tables being converted to ontology classes

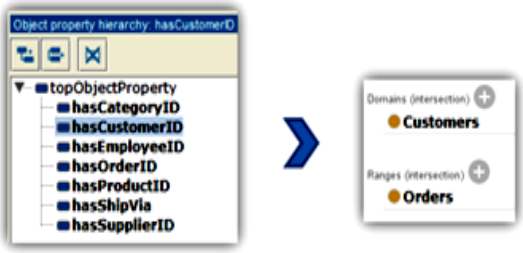

Fig. 5: Foreign keys being converted to object properties, also showing domain and range of object property CustomerID

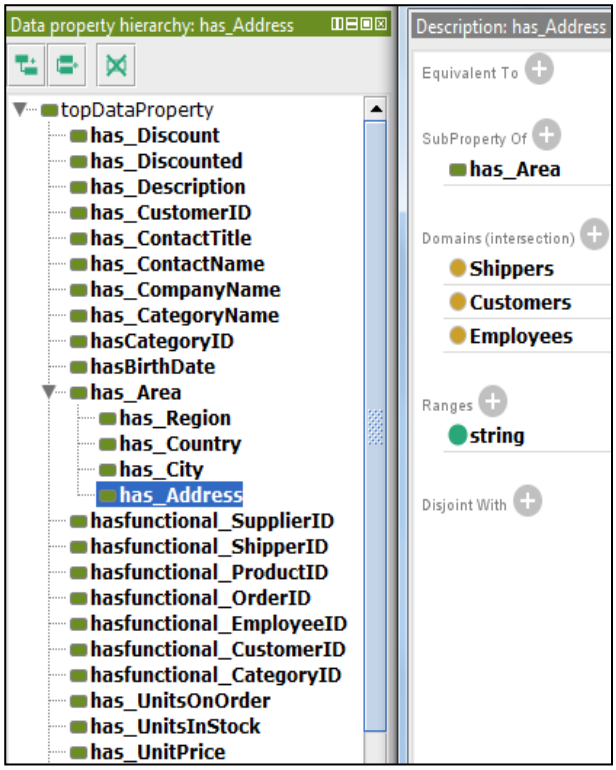

Fig. 6: Columns being converted to data type properties and sub-properties

Next step is to convert all the primary keys to functional data type properties. Functional data type properties refer to the individuals which can have only one possible value. They are also known as single valued properties (Antoniou and Van Harmelen, 2004). The primary key of Customers table is converted to functional property as shown in Fig. 7, Customers is its domain and string is its range.

The generated owl file is then opened in Protégé. Our newly developed OWL ontology from North wind database is shown in Fig. 8.

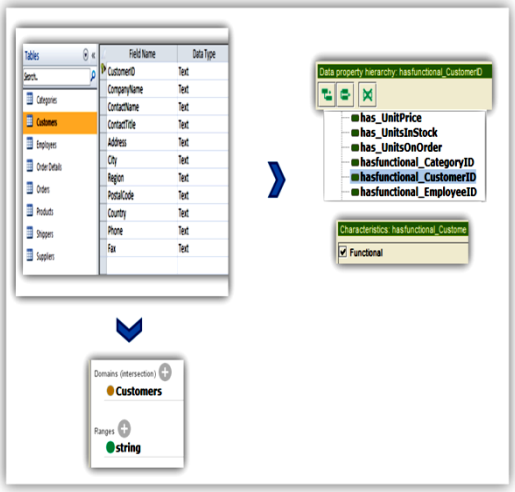

Fig. 7: Primary key being converted to functional data property, also showing domain and range of PK Customers

7. Testing

During the testing phase, twenty databases were used to evaluate the tool. Because of space issues we will show results of ten databases. Ten databases from different domains are taken. Some of these databases are large and some very small. Before going into details of testing of our tool, brief specifications of the machine used for testing are given in Table 3. Machine specifications play an important role while discussing the efficiency of the tool. Table 4 gives the detail of databases tested with our tool and converted to OWL ontology.

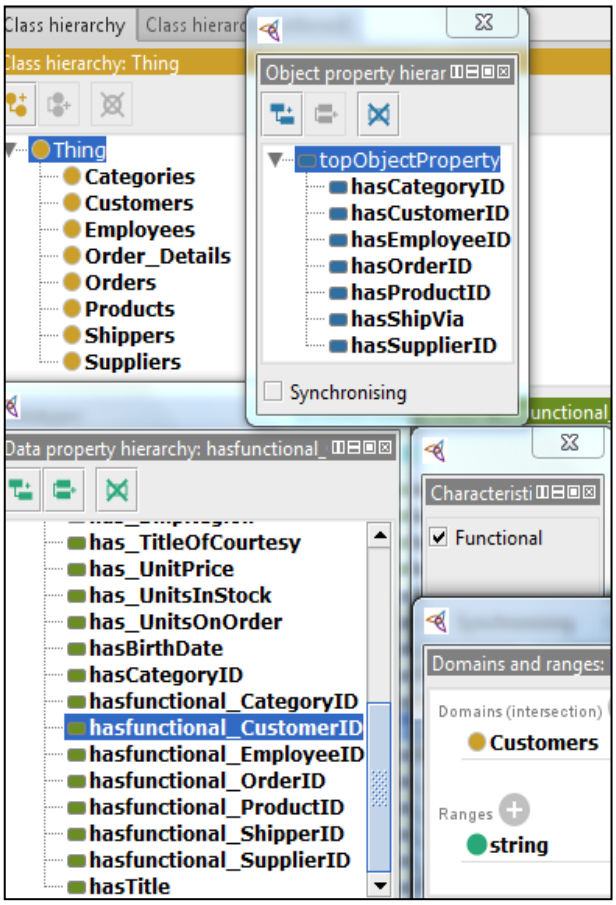

Fig. 8: Transformed north wind OWL ontology

Table 3: Machine specifications

Processor	Intel core i5
CPU speed	2.60 GHz
OS	Windows 7
Memory	8 GB
System type	64 bit system

Table 4: Sample database

Sample database	Domain
DB1[5]	Library database
DB2[6]	Factory database
DB3[5]	Book store database
DB4[5]	Doctor's database
DB5[5]	Product-sales database
DB6[11]	North wind database
DB7[6]	Sales database
DB8[6]	Vehicle's database
DB9[5]	Suppliers database
DB10[6]	Customer-employees database

Tables 5-7 show the results. These results are calculated manually. All the databases used were first manually converted to ontology and then compared to the ontology created from our tool. In the same way the percentages are calculated:

$$\text{Similarity Percentage} = \left(\frac{\text{Number of Classes}_{(\text{Extracted by our Tool})}}{\text{Number of Classes}_{(\text{Extracted Manually})}}\right) * 100\ \%$$

8. Conclusion and future work

Semantic web is considered as mature field these days. The existing data stored on the web is mostly stored in relational databases format. To get the benefits of semantically arranged data, we have to convert existing data into ontology. Ontologies help us in the integration of heterogeneous sources and knowledge management.

Researchers have done a lot of efforts in this domain and developed different tools. The major drawbacks in existing approaches are that they are not fully automatic, performed mapping at a very basic level, outdated and not accessible. In case of large database, the existing tools do not perform proper conversions efficiently.

Table 5: Database metadata

Database attributes	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10
Tables	7	4	7	4	7	8	21	7	8	13
F-Keys	3	2	6	4	4	7	6	6	5	11
P-Keys	3	5	8	4	7	9	20	8	11	14
Columns	26	18	34	23	30	76	154	66	36	68

Table 6: Transformed ontology attributes

Ontology Attributes	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10
Classes	7	4	7	4	6	8	20	6	8	13
Object Properties	3	2	6	4	4	7	6	6	5	11
Data Type Properties	18	13	30	14	21	52	121	53	26	51
Functional DataType properties	7	5	8	4	7	9	20	8	11	14
Sub-Classes	0	0	0	0	1	0	2	1	0	0
Sub-Data Properties	2	0	0	3	2	5	12	2	0	0

Table 7: Evaluation of tool using ten different databases

Attributes Transformed	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8	DB9	DB10
Classes	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Object Properties	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Data Type Properties	100%	99%	90%	90%	99%	99%	95%	95%	95%	80%d
Functional Data Type properties	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Object Property's Domain and Range	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Datatype Property's Domain and Range	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Sub-data Properties	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%	More than 60%
Sub-Classes	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%

We have introduced an approach, which automatically converts a relational database to OWL ontology. The tool is implemented in Java and uses OWL API. In this paper, our main focus is on mapping rules, so we over-come mapping limitations from the previous approaches. We have improved the tool by creating rules for sub-data-properties and sub-classes, which was never available in existing tools. Our tool also successfully converts primary keys to functional data-type properties, which is not implemented before. Our tool's efficiency is same for both small and large databases. The tool provides user friendly interface.

References

Antoniou G and Van Harmelen F (2004). A semantic web primer. MIT press, London, UK.

Banu A, Fatima SS, and Khan KUR (2011). Semantic-based querying using ontology in relational database of library management system. International Journal of Web and Semantic Technology, 2(4): 21-32.

Cerbah F (2008). Learning highly structured semantic repositories from relational databases. The semantic web: Research and applications, Springer, Berlin, Germany: 777-781.

Cullot N, Ghawi R, and Yétongnon K (2007). DB2OWL: A tool for automatic database-to-ontology mapping. In the Proceedings of the 15th Italian Symposium on Advanced Database Systems, Torre Canne di Fasano (BR), Italy: 491–494.

Erling O and Mikhailov I (2006). Mapping relational data to RDF in Virtuoso. Open Link Software. Available online at: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF

Gherabi N, Addakiri K, and Bahaj M (2012). Mapping relational database into OWL structure with data semantic preservation. International Journal of Computer Science and Information Security, 10(1): 42-47.

O'connor MJ, Halaschek-Wiener C, and Musen MA (2010). Mapping master: A flexible approach for mapping spreadsheets to OWL. In the International Semantic Web Conference, Springer, Berlin, Heidelberg: 194-208.

Ra M, Yoo D, No S, Shin J, and Han C (2012). The mixed ontology building methodology using database information. In the International Multi-Conference of Engineers and Computer Scientists, Hong Kong: 1-6.

Saleh ME (2011). Semantic-based query in relational database using ontology. Canadian Journal on Data, Information and Knowledge Engineering, 2(1): 1-16.

Shujah M, Naz T, and Sadiq A (2015). Approaches for Loss-less mapping from relational database to OWL Ontologies. Research Journal of Recent Sciences, 4(3): 91-99.

Tirmizi SH, Sequeda J, and Miranker D (2008). Translating sql applications to the semantic web. In the International Conference on Database and Expert Systems Applications, Springer, Berlin, Heidelberg: 450-464.

Xiang Z, Zheng J, Lin Y, and He Y (2015). Ontorat: Automatic generation of new ontology terms, annotations, and axioms based on ontology design patterns. Journal of Biomedical Semantics, 6(4): 1-10.

Zhang L and Li J (2011). Automatic generation of ontology based on database. Journal of Computational Information Systems, 7(4): 1148-1154.

Zhou X, Xu G, and Liu L (2011). An approach for ontology construction based on relational database. International Journal of Research and Reviews in Artificial Intelligence, 1(1): 16-19.