A Framework for OWL DL based Ontology Construction from Relational Database using Mapping and Semantic Rules

C.Ramathilagam
Assistant Professor
Adithya Institute of Technology
Coimbatore, TamilNadu, India

M.L.Valarmathi, Ph.D
Associate Professor
Government College of Technology
Coimbatore, TamilNadu, India

ABSTRACT

World Wide Web (WWW) is capable of retrieving the results only through location finding of search terms, where as Semantic web retrieves by analysing the keywords because of the machine accessible nature of semantic data. One such Semantic data format is Ontology. Ontologies are the web documents generated by Web Ontology Language to provide more precise web content, thus by improving the performance of information retrieval. Semantic web requires data either in terms of manual creation or through conversion from existing data. Ontologies lack semantically rich data because the manual construction of these documents are time consuming and also the domain experts need to understand the syntax and semantics of Ontology development languages. One alternate to compensate for the rich set of data is to take the contents of Relational Data Base (RDB) for domain related applications in the Semantic Web. This is possible by mapping the RDB constructs into Ontology constructs. This paper proposes an RDB to Ontology mapping system framework which can generate an Ontology based on the proposed Mapping Rules for a Banking domain. The Mapping rules are generated both for, 1) direct mapping 2) integrity constraints mapping. Direct mapping is, RDB components like table, attribute, data are mapped to the corresponding Ontology components. Integrity constraints mapping covers primary key, foreign key and column constraints mapped to the relevant Ontology components. This is the direct translation of RDB structure. This paper also proposes additional Semantic Rules for the generated Ontology to provide richer semantics. This in turn provides more expressiveness in Ontology representation and also efficient reasoning power.

Keywords

Semantic web, schema, mapping, Ontology, RDF, RDFS, OWL, DL

1. INTRODUCTION

Database, the collection of data as meaningful information. Relational database, the related information collection. Knowledge, the collection of derived information. Ontology, the derived information collection from any domain of discourse. World Wide Web (WWW), is the service provided through the Internet from the scattered heterogeneous data. The information available in the Internet is in any form of structured data like databases, XML data, data warehouses, enterprise systems (CRM, ERP etc.) or unstructured data like Excel spreadsheets, Word documents, Email messages, RSS feeds, audio files, video files etc. In the web, these forms of data can provide only the information as it contains.

The main difference between the conventional web and Semantic web can be considered in terms of the data format, the way of extracting that data and the outcome of the web search. In the Semantic web, the pattern of data is provided in terms of knowledge. The knowledge, however is derived information from given set of data. Ontology is one such document to define or represent knowledge. The Semantic Web not only contains Ontological documents, also RDF documents, XML data. The hierarchy of knowledge representation formats is given (see Figure 1). XML (eXtensible Markup Language) provides only the syntactic interoperability among the data. XML Schema is the XML document definition with sophisticated set of data types. Resource Description Framework (RDF) has the language format with (Object, Property, Value) triplet. Every Resource or Object defined in this format is a triplet. The knowledge representation formats has a language to develop Semantic Web documents. Upper the layer, more the preciseness in the representation and the lower layers provide less precise representation. In spite of that, the higher level languages include the lower level language syntax also to combine all representations. Among the given levels, OWL (Web Ontology Language) document contains more expressive power, efficient reasoning support, well defined syntax, formal semantics and the convenience of expression than the lower layers of representation. Since WWW contains domain oriented database files in a large quantity, it does not suffer from lack of data, but lacks with efficient information retrieval. But semantic web lacks such a huge repository of data. This means that Semantic web requires data either (1) in terms of manual creation or (2) through conversion from existing data. The tedious and boredom work of new Ontology creation and the necessity for domain experts to understand Ontology development language syntax & semantics leads a way to the mapping process of Relational Databases to the Ontology.

RDB to Ontology mapping process or the classification of RDB to Ontology mapping can be done in any one of the following ways. In all the cases vice versa ie., Ontology to RDB mapping is also possible.

The classification of RDB to Ontology Mapping

- Constructing new Ontology by mapping with existing RDB or constructing RDB with existing Ontology
- Mapping or creating set of correspondences between existing RDB constructs and Ontology constructs
- Ontology learning from RDB
- Extraction of Ontology from RDB
- Integrating RDB with Ontology
- Maintaining the semantics of Ontology in RDB
- Discovering relationships between RDB and Ontology
- Semantic based Querying using Ontology in RDB
- Supporting Ontology based Semantic matching in RDB

etc.,

RDB mapped to Ontology means that the RDB constructs already exist and the new Ontology is developed based on the RDB constructs to provide rich set of data for the Ontology. Vice versa is also possible, that is, Ontology mapped to RDB means that Ontology already exist and the RDB is newly constructed based on Ontology constructs to provide more reasoning and expressive power of Ontology.

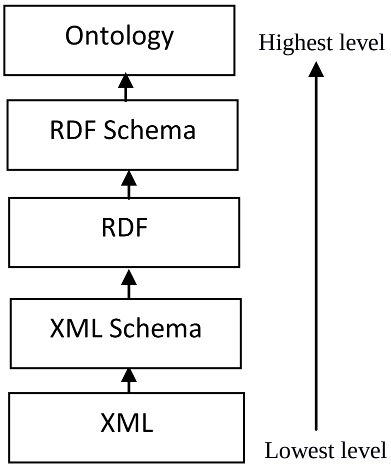

Fig 1: Different Knowledge Representation Formats in Semantic Web

Several works have been done on Ontology to RDB mapping also. If either one of the above two mappings is done then it is known as Forward direction of mapping. Both forward and backward direction of mapping is known as Reversible Mapping process. This reversible process [15] means that RDB is mapped to Ontology and again the resultant Ontology is mapped to original RDB in order to check whether the Ontology has been generated purely based on all of the constructs of RDB. But this is a tedious task. This paper focuses on forward direction of mapping, ie., RDB to Ontology mapping alone. The paper is organised as follows. Section 1 gives introduction. Section 2 explains the Relational and Ontology data models. Section 3 describes about the RDB to Ontology mapping process with mapping rules and semantic rules. Section 4 deals with the framework for the implementation of this model. Section 5 explains about the related work. Section 6 gives conclusion and the future enhancements of this work.

2. DATA MODEL

A database model is a type of data model that determines the logical structure of a database and fundamentally determines the manner in which the data can be stored, organized, and manipulated. The most popular example of a database model is the Relational model, which uses a table-based format. The Ontology model is also described in this section which uses the basic construct as class or concept.

2.1 Relational Data Base model

Relational database is formalised by First Order Logic (FOL). FOL is a formal system used in mathematics, philosophy and computer science etc. It models the world in terms of objects, properties, relations and functions. Objects are things with individual identities. Properties of objects that distinguish them from other objects. Relations that hold among sets of objects. Functions which are a subset of relations where there is only one value for any given input. Each object terms a table or relation in relational database. The relation between FOL and RDB is given in Table 1.

Table 1. Relationship between FOL and RDB

FOL	RDB
Objects	Table /relation
Properties	Attributes
Relations	Relationship among tables (Keys)
Functions	Subset of Relations

The properties are the attributes of table and the relations indicate the relationship of table with other table. In relational model this kind of relationship is achieved through ‘keys’. Here the tables use an attribute or part of the attributes as key. In this system, a Banking domain is provided and its relational database constructs are mapped to Ontology constructs. Objects in banking system includes account, customer, branch, depositor, loan and borrower etc. The schema diagram of sample banking database system is given. (see Figure 2). Schema diagram represents the data base or table design of the particular domain. It contains the properties of each object and the directed lines denote the relationship of objects with other objects. The tables branch, account, customer and loan contains branch name, account number, customer name and loan number as primary keys respectively. The remaining attributes in these tables are called non-key attributes. The primary key fields are underlined. Depositor and borrower tables contain both the fields as primary key.

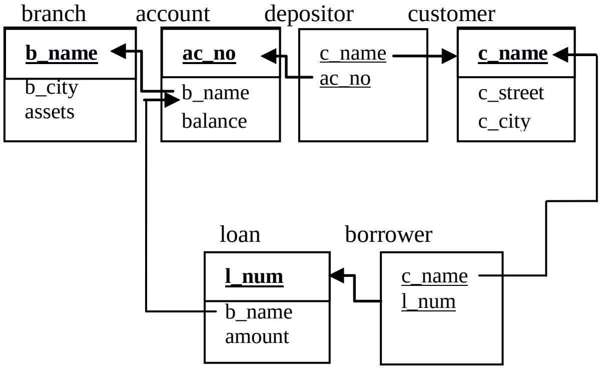

Fig 2: Schema diagram of sample Banking domain

The direction of line indicates the foreign key relation with the parent table. Branch name of account table is the foreign key of branch name of branch table. Account number of depositor table is the foreign key of account number of account table. Customer name of depositor table is the foreign key of customer name of customer table. Customer name of borrower table is the foreign key of customer table customer name. Branch name of loan table is the foreign key of branch name of account table. Here, foreign key has two representations. The foreign key may be either a non-key attribute or part of the primary key.

2.2 Ontology model

Ontology is formalised with Description Logic (DL) which is the subset of FOL. Ontology is more expressive and has more efficient decision problems than FOL. It provides logical formalism for Ontology. It models the real world in terms of concepts, roles and individuals and their relationships. In spite of the implementation of DL in the Ontology or in OWL language the FOL terminology is mostly used in OWL language. DL and OWL have commonalities and it is given in Table 2.

Table 2. Relation between DL and OWL

DL	OWL
Concept	Class / predicate
Role	Property
Individual	Instance / constant
Relationships	Class & Subclass relation, property & Subproperty relation

Ontology is developed with Web Ontology Language (OWL). OWL is available in three flavours.

1) OWL Full - OWL Full allows free mixing of OWL with RDF Schema
2) OWL DL - puts constraints on the mixing with RDF and requires disjointness of classes, properties, individuals and data values.
3) OWL Lite - supports only a subset of the OWL language constructs.

OWL DL

This system follows the syntax from OWL DL for Ontology development. OWL DL provides various elements which gives the expressiveness with more reasoning support than the other two. OWL DL is a richer vocabulary description language for describing the class elements with subclass, equivalent class and disjointness, property elements with data type property, object property, value properties, minimum and maximum cardinalities. The instances are defined using RDF syntax. The special properties such as transitive, symmetric, functional and inverse functional concepts are included. The enumerations concept and Boolean combinations such as complement, intersection and union are provided in OWL vocabulary. The property relationships and restrictions are the RDF Schema representations and are used in OWL DL. The property elements to define relationships are Type, subClassOf and subPropertyOf. The elements to define property restrictions are Domain, Range, ConstraintResource and ConstraintProperty.

Examples for basic OWL elements.
Concept or class names e.g., Cat, Animal, Doctor, Equivalent to FOL unary predicates.
Role or property names e.g., sits-on, hasParent, loves, Equivalent to FOL binary predicates.
Individual names e.g., Raman, Coimbatore, Delhi, Equivalent to FOL constants.

3. RDB TO ONTOLOGY MAPPING PROCESS

Mapping the constructs of RDB to Ontology means that the RDB tables are mapped to Ontology classes, attribute names and data type are mapped to Ontology Data type properties and data or record are mapped to Ontology instances or individuals. This is known as direct mapping. The integrity constraints mapping includes the primary key is mapped to Ontology Inverse functional property, foreign key to Object property and RDB column constraints mapping is done with OWL Property restrictions. The basic RDB constructs and its corresponding Ontology constructs based on the proposed mapping rules is given in Table 3. This mapping is considered as direct mapping in the sense that whatever RDB constructs possess they are mapped to Ontology components. Apart from this direct mapping, this system also proposes additional rules to provide a semantic based Ontology. These rules include the Ontology class, subclass relation, property, sub property relations and restrictions and special properties. These additional OWL Description Logic elements are used to enhance the semantics of generated Ontology. The Entity Relationship (ER) diagram for the banking domain is considered for this semantics based Ontology development. The relation between objects through sample ER diagrams are given. (see Figures 3 and 4).

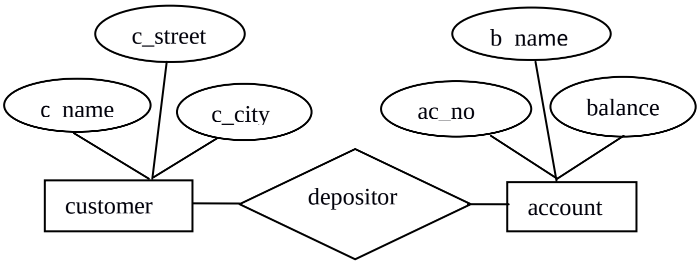

Fig 3: ER diagram for customer and account relationship

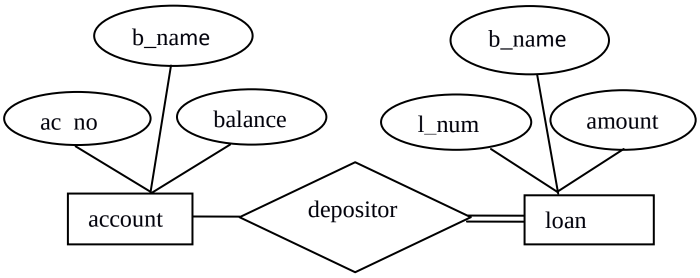

Fig 4: ER diagram for account and loan relationship

ER diagram is the advanced data model consist of objects called entities and their relationships. Figure 3 indicates that the customer who has account is a depositor. This means that customer and depositor are meaning the same object. Figure 4 indicates that loan object has total participation with depositor relation in the relationship set. That is, each loan must have at least one associated account. Based on this Entity relationship concept, the additional Semantic Rules are imposed for the generated Ontology.

Table 3. RDB and Ontology Basic Constructs

RDB	Ontology
Table	Class / concept
Attribute	Data type Property
Record	Instances / individuals
Primary key	Inverse functional property
Foreign key	Object property
Column constraint	Property restrictions

3.1 RDB to Ontology Mapping Rules

The Mapping rules proposed below refers the method of generating Ontology from the Banking domain RDB. For each rule, a sample RDB component is given. The corresponding OWL DL syntax is given inside the box.

3.1.1 Direct mapping

3.1.1.1 Table to Class mapping

For all the tables of banking system, classes are defined in OWL DL. Tables are mapped to classes based on the rule.

Rule 1 : "Create a class in Ontology for the RDB table if it has at least one non-key attribute"

RDB component (account table structure)

create table account (ac_no integer primary key, b_name char(15) references branch (b_name), balance float check (balance > 1000)

account class in Ontology

```
<owl:class rdf:ID = "account"/>
```

3.1.1.2 Attribute to Data Type Property mapping

Rule 3 : "Create Ontology Data type Property for each RDB attribute and its data type"

RDB Component

attribute names and data types (account table structure)

Ontology

```
<owl:DatatypeProperty rdf:ID = "ac_no">
     <rdfs:domain rdf:resource = "account"/>
     <rdfs:range rdf:resource = "&xsd;positiveInteger"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID = "#b_name">
     <rdfs:domain rdf:resource = "account"/>
     <rdfs:range rdf:resource = "&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID = "#balance">
     <rdfs:domain rdf:resource = "account"/>
     <rdfs:range rdf:resource = "&xsd;float"/>
</owl:DatatypeProperty>
```

3.1.1.3 Data to Instance mapping

Rule 3 : "Map each row of data with Ontology instances of classes as declared in RDF syntax."

RDB component

Insert into account (ac_no, brance_name, balance) values (36883, Saravanampatty, 200000)

Ontology instance

```
<account>
  <ac_no rdf:datatype="&xsd:integer">36988</ac_no>
  <b_name df:datatype="&xsd:string">SVPatty</b_name>
   <balance rdf:datatype ="&xsd:double">20000</balance>
</account>
```

3.1.2 Integrity Constraints Mapping

3.1.2.1 Primary key to Inverse Functional Property mapping

Rule 4 : "Create Inverse Functional Property and set minimum Cardinality value as "1" for an RDB column for which two different rows cannot have the same value and for setting Not Null constraint respectively"

For example, one account number is assigned to one person only. A column constraint primary key is mapped to both an inverse functional property and a minimum cardinality of 1.

RDB Component

Primary key ac_no (account table structure)

Ontology

```
<owl:InverseFunctionalProperty rdf:ID = "ac_no"/>
<owl:class rdf:ID = "account">
   <rdfs:subClassOf>
      <owl:restriction>
        <owl:OnProperty rdf:resource = "#ac_no"/>
             <owl:minCardinality rdf:datatype =
                 "&xsd:nonNegativeInteger"1/>
      </owl:restriction>
   </rdfs:subClassOf>
</owl:class>
```

3.1.2.2 Foreign key to Object Property mapping

Rule 5 : "Create Ontology Object Property with domain and range constraints to set a foreign key, if foreign key is not a primary key or part of a primary key "

RDB Component

Branch name of account table is the foreign key for branch name of branch table and it is not a primary key or part of primary key.

Ontology

```
<owl:ObjectProperty rdf:ID = "b_name">
    <rdfs:domain rdf:resource =     "#account"/>
    <rdfs:range        rdf:resource        =        "#branch"/>
</owl:ObjectProperty>
```

Rule 6 : "If foreign key is a primary key or part of a primary key in RDB, create Object Property with domain and range constraints accompanied by a cardinality of 1"

RDB Component

Create table borrower(c_name char(15) references customer (c_name), l_num float primary key(c_name, l_num))

Ontology

```
<owl:ObjectProperty rdf:ID = "c_name">
   <rdfs:domain rdf:resource = "#borrower"/>
   <rdfs:range rdf:resource = "#customer"/>
</owl:ObjectProperty>

<owl:Class rdf_ID="#borrower">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resourse="#c_name"/>
         <owl:cardinality
                rdf:datatype="&xsd:nonNegativeInteger"1/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

3.1.2.3 Column Constraints to Property Restrictions mapping

RDB Component

balance > 1000 condition   (account table structure)

Rule 7 : "Create Ontology property restriction for Cardinality to map constraints at column level ie., for the restrictions on how many distinct values a property may or must take"

Ontology

```
<owl:Class rdf:about="#account">
  <rdfs:subClassOf>
     <owl:Restriction>
         <owl:onProperty rdf:resource="#balance"/>
             <owl:minCardinality
               rdf:datatype="&xsd;nonNegativeInteger">1000
             </owl:minCardinality>
     </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

3.2 Semantics Rules

The direct and integrity constraints mapping allows whatever RDB structure possesses, they are directly mapped to Ontology structures. This structure provides sufficient information retrieval as in the RDB. The generated Ontology can provide sufficient reasoning power only if the special type of rules are applied. Semantics based rules and its corresponding OWL DL syntax are given inside the boxes which can add more expressiveness to the developed Ontology.

Rule 8 : "Create equivalence class element for any two Ontology classes if those two classes denote the same object"

Customer and depositor both terms denote customer.

```
<owl:Class rdf:ID="customer">
   <owl:equivalentClass rdf:resource="#depositor"/>
</owl:Class>
```

Rule 9 : "Create All Values From property for any two classes both of which should depend on each other"

The borrower who applies loan must be a depositor. The corresponding Ontology is,

```
<owl:Class rdf:about="#borrower">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#with_loan"/>
     <owl:allValuesFrom rdf:resource="#depositor"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Rule 10 : "Create Inverse Of property for any two object relations which are inversely related"

Depositor owns account and in turn the account is owned by a depositor. For the depositor and account objects, the relations owns and owned by are inversely related.

```
<owl:ObjectProperty rdf:ID="owns">
  <rdfs:domain rdf:resource="#depositor"/>
  <rdfs:range rdf:resource="#account"/>
  <owl:inverseOf rdf:resource="#ownedby"/>
</owl:ObjectProperty>
```

Rule 11 : "Create some Values From property for the class which should have at least one value in another class with the relevant relation"

The borrower must have at least one account in the bank.

```
<owl:Class rdf:about="#borrower">
 <rdfs:subClassOf>
   <owl:Restriction>
     <owl:onProperty rdf:resource=#owns"/>
     <owl:someValuesFrom rdf:resource="#account"/>
  </owl:Restriction>
<rdfs:subClassOf>
<owl:Class>
```

Rule 12 : "Create sub Class element for two classes in which one is the sub class of the other"

Borrower has the subclass relation with depositor

```
<owl:Class rdf:ID="borrower">
  <rdfs:subClassOf rdf:resource=#depositor/>
</owl:Class>
```

Rule 13 : "Create disjoint With class element if two classes are disjoint with each other"

Account_number is disjoint With Loan number

```
 <owl:Class rdf:about=”#account-no”>
  <owl:disjointWith rdf:resource=”#loan-no”/>
</owl:Class>
```

Rule 14 : “Create has Value property if any property has some specific value”

The person with maximum loan amount is given by this element

```
<owl:Class rdf:about="#borrower">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#with_max_loan_amt"/>
     <owl:hasValue rdf:resource="#36885”/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Rule 15 : “Create minimum or maximum Cardinality, if any property has a minimum or maximum value otherwise at least or at most values respectively”

The person who owns account should maintain minimum of Rs.1000 as balance

```
<owl:Class rdf:about="#depositor">
  <rdfs:subClassOf>
   <owl:Restriction>
        <owl:onProperty rdf:resource="#min_balance"/>
         <owl:minCardinality rdf:datatype=”&xsd:float”>1000
        </owl:minCardinality>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

4. RDB TO ONTOLOGY MAPPING PROPOSED SYSTEM FRAMEWORK

The system provides different modules. RDB schema is extracted from the data dictionary using JDBC engine. The Mapping engine process includes for each RDB component, the suitable mapping rule that is either direct mapping rule or integrity constraint mapping rule is applied and the corresponding Ontology component is generated. The generated Ontology is placed in OWL file. This process is repeated until all RDB components are extracted from the data dictionary. The Semantic Rules are then applied for the generated Ontology. The new Semantic based Ontology is placed in the OWL Ontology file. The final OWL file holds the newly generated Ontology with more expressiveness and reasoning power. The framework of the system is given in Figure 5.

Proposed algorithm for mapping process

```
Algorithm : OWL DL based mapping of RDB to Ontology
Input : RDB components
Output : OWL DL based Ontology
Begin
        OWL file ={}
        Connect with JDBC engine
        Extract database components one by one from the
                                   Data Dictionary (DD)
        Repeat until DD is empty
                  Generate Ontology components based on
                   the Direct Mapping Rule for each RDB
                   component
                   OWL file = OWL file +  generated Ontology
        Apply Semantic Rules on the generated Ontology
        OWLfile = OWLfile +  Semantic rules based Ontology
End
```

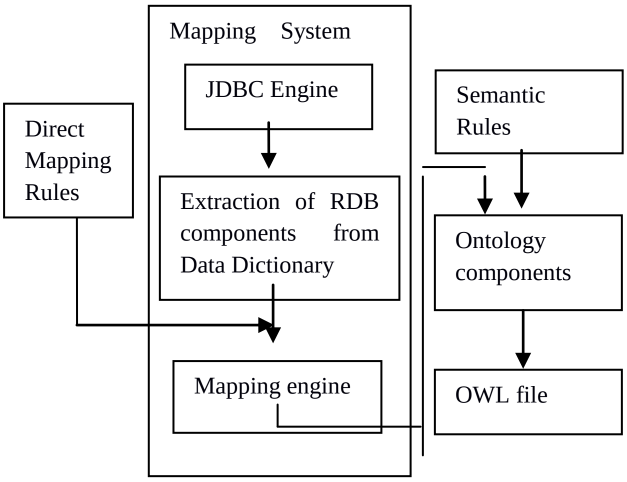

Fig 5: RDB to Ontology Mapping System Framework

5. RELATED WORK

Several works are done in Forward direction of mapping from RDB to Ontology. Shufeng Zhou et al [3] creates an Ontology generator from RDB by extracting metadata information from RDB with reverse-engineering and analyses corresponding relationship between RDB and Ontology, then presents Ontology generation. Irina Astrova et al.,[4] generates the Ontology based on the given SQL scripts of any RDBMS by providing SQL constructs and its corresponding Ontology constructs. Quang Trinh et al., [5] describes a formal algorithm to use the relational database RS meta-data and structural constraints to construct its OWL Ontology while preserving the structural constraints of the underlying relational database system. Syed Hamid Tirmizi et al., [6] defines a system for automatic transformation of SQL DDL Schemas into OWL DL Ontologies which represents the First Order Logic(FOL) based translation of SQL applications to the Semantic Web. The RDB constructs are based on FOL, this paper tries to provide a system which has the expressive and reasoning power of FOL into the Ontology. Zdenka Telnarova [7] focused on the principles of automatic conversion of constructs of Ontology and transfer of relational data model to constructs of OWL Ontology and transfer of relational data to Ontology instances. Lei Zhang, Jing Li [8] provided an approach for automatic generation of Ontology based on database by analysing the Ontology and database by constructing rules of Ontology elements based on RDB and provides a system to generate Ontology automatically. Ouyang Dan-tong et al., [9] presents a set of constraint axioms called IC-mapping axioms, based on these a special Ontology with integrity constraint, which is adapted to map Ontology knowledge to data in relational databases. Ernestas Vysniauskas et al., [10] proposed an algorithm and generates a tool for transformation of domain Ontology described in OWL to RDB. Guntars Bumans [11] provides a simple and elaborate example of how mapping information stored in relational tables can be processed using SQL to generate RDF triples for OWL class and property instances. Irina astrova et al., [12] proposed an approach for automatic transformation of Ontologies to relational databases where the quality of transformation is also considered. Raji Ghawi, Nadine Cullot [13] proposes a general interoperability architecture that uses Ontologies for explicit description of the semantics of information source, web services to facilitate the communication between the different components of the architecture. Souripriya das et al.[14] addresses three issues by allowing OWL Lite and OWL DL based Ontologies to be stored in Oracle RDBMS and by providing a set of SQL operators for Ontology-based semantic matching. Ernestas Vysniaskas et al.,[15] defines a reversible information preserving transformation from OWL2 Ontologies into relational databases using the proposed hybrid approach. Ayesha Banu et al., [17] builds a wrapper to act as a translator from the semantic queries issued to the system into the syntactic data. Dmitry V.Levshin [18] presents an approach to translation of data schema and most important constraints from relational databases into the Semantic web without any extensions to the Semantic Web Languages.

In any of the above works the Semantic rules are not defined or implemented for the generated Ontology. This paper covers various Semantic rules to promote the developed RDB based Ontology with more reasoning support. This in turn provides efficient information retrieval compared with direct transformation of RDB to Ontology with only the RDB based components.

5. CONCLUSION AND FUTURE WORK

In this paper, a framework is given for mapping a Relational Data Base to Ontology by providing necessary mapping rules for direct mapping and integrity constraints mapping. Direct mapping includes mapping the RDB tables, attributes and data to the corresponding Ontology components class, properties and instances. Integrity constraint mapping includes mapping the primary key, foreign key and column constraints to Ontology Inverse Functional property, Object property and Property Restrictions. Here, whatever RDB constructs possess, are directly generated in Ontology using OWL DL language. This provides sufficient information retrieval in the Ontology as that of RDB. This system also proposes additional Semantic rules so that the retrieval process is more efficient with the expressiveness provided through the rules. As the future work, the generated Ontology can be checked for efficient information retrieval with appropriate tools. Also, the complete system for banking can be considered with benchmark data, the corresponding Ontology can be developed with more reasoning power.

6. REFERENCES

[1] Grigoris Antoniou and Frank Van Harmelen, 2004, Semantic Web Primer

[2] Silbercharz, Korth, Sudarshan, 2005, Database System Concepts, 4th edition

[3] Shufeng Zhou, Haiyun Ling, Mei Han, Huaiwei Zhang, 2010, Ontology generator from Relational Database Based on Jena, Computer and Information Science

[4] Irina Astrova, Nahum Korda, and Ahto Kalja, 2007, Rule based Transformation of SQL Relational Databases to OWL Ontologies, , World Academy of Science, Engineering and Technology

[5] Quang Trinh, Ken Barker, Reda Alhajj, 2006, RDB2ONT : A tool for generating OWL Ontologies from Relational Database systems, , IEEE

[6] Syed Hamid Tirmizi, Juan Sequeda, and Daniel Miranker, 2008, Translating SQL applications to the Semantic Web, Springer

[7] Zdenka Telnarova, 2010, Relational database as a source of Ontology creation, IEEE

[8] Lei Zhang, Jing Li, 2011, Automatic Generation of Ontology based on Database, Journal of Computational Information Systems

[9] Ouyang Dan-tong, CUI Xian-ji, YE Yu-xin, 2010, Mapping integrity constraint Ontology to Relational databases, The journal of China Universities of Posts and Telecommunication, Elsevier

[10] Ernestas Vysniauskas, Lina Nemuraite, Bronius Paradauskas, 2012, Preserving semantics of OWL2 Ontologies in Relational Databases using Hybrid approach, ISSN 1392, 124X Information Technology And Control, 2012, Vol 41, No.2

[11] Guntars Bumans, 2010, Mapping between Relational Databases to OWL Ontologies: An example, Computer Science and Information Technologies

[12] Irina Astrova, Ahto Kalja, Nahum Korda 2007, Automatic Transformation of OWL Ontologies to SQL Relational Databases, IADIS European Conference on Data Mining

[13] Raji Ghawi, Nadine Cullot, 2007, Database –to-Ontology mapping generation for Semanitc Interoperability, VLDB'07, ACM

[14] Souripriya Das, Eugene Inseok Chong, George Eadon, Jagannathan Srinivasan, 2004, Supporting Ontology based Semantic matching in RDBMS, VLDB'04

[15] Ernestas Vysniauska, Lina Nemeraite, Rimantas Butleris, Bronius Paradauskas, 2011, Reversible Lossless transformation from OWL2 Ontologies into Relational Databases, Information Technology and Control, Vol 40, ISSN 1392,No 4

[16] Carmen Martinez-Cruz, Ignacio J.Blanco, M.Amparo Vila, 2011, Ontologies versus relational database: are they so different? A comparison, Springer

[17] Ayesha Banu, Syeda Sameen Fatima, Khaleel Ur Rahman Khan, 2011, Semantic-based Querying using Ontology in Relational database of Library Management System, International Journal of Web & Semantic Technology (IJWesT), Vol2, No.4

[18] Dmitry V.Levshin, 2009, Mapping Relational Databases to the Semantic Web with Original meaning, International Journal of Software Informatics Vol 4, No.1

[19] Yuan An, Thodoros Topaloglou, 2008, Maintaining Semantic Mappings between Database Schemas and Ontologies, Springer LNCS, Vol 5005

[20] Wei Hu, Yuzhong Qu, 2007, Discovering Simple Mappings between Relational Database Schemas and Ontologies, Springer LNCS, Vol 4825