Mapping relational database into OWL Structure with data semantic preservation

Noreddine GHERABI Hassan 1 University, FSTS Department of Mathematics and Computer Science gherabi@gmail.com Khaoula ADDAKIRI
Department of Mathematics and Computer Science,
Université Hassan 1^{er}, FSTS, LABO LITEN Settat,
Morocco

Mohamed BAHAJ
Hassan 1 University, FSTS
Department of Mathematics and Computer Science
mohamedbahaj@gmail.com

Abstract— this paper proposes a solution for migrating an RDB into Web semantic. The solution takes an existing RDB as input, and extracts its metadata representation (MTRDB). Based on the MTRDB, a Canonical Data Model (CDM) is generated. Finally, the structure of the classification scheme in the CDM model is converted into OWL ontology and the recordsets of database are stored in owl document. A prototype has been implemented, which migrates a RDB into OWL structure, for demonstrate the practical applicability of our approach by showing how the results of reasoning of this technique can help improve the Web systems.

Keywords-component; RDB, RDF, OWL, Web ontology.

I. INTRODUCTION

The use of ontologies is rapidly growing since the emergence of the Semantic Web. To date, the platform of Web ontologies available continues to increase at a phenomenal rate. The requirement for the development of the current web of documents into a semantic web requires the inclusion of large quantities of data stored in relational databases (RDB). The mapping of these quantities of data from RDB to the Resource Description Framework (RDF) has been the focus of a large body of research work in diverse domains. Therefore, it is necessary to study the difference between Semantic Web applications using relational databases and ontologies.

There is a need for an integrated method that deals with DataBase Migration from RDB to Object-Oriented DataBase (OODB)/XML/RDF/OWL in order to provide an opportunity for exploration, experimentation and representation of databases in a Web data. With the current revolution in the use of the Web as a platform for application development, XML (eXtensible Markup Language) [1] was the first interest to many e-business applications.

Different researches are investigated in RDB migrations focusing on different domains. Most existing proposals are

restricted by a range of assumptions and characteristics such as the respect of the 3rd Normal Form and the integrity constraints [2].

Several approaches have been presented that directly map relational schemas to ontology languages [3]. Recently, the W3C RDB2RDF Working Group is developing a direct mapping standard that focuses on translating relational database instances to RDF [4].

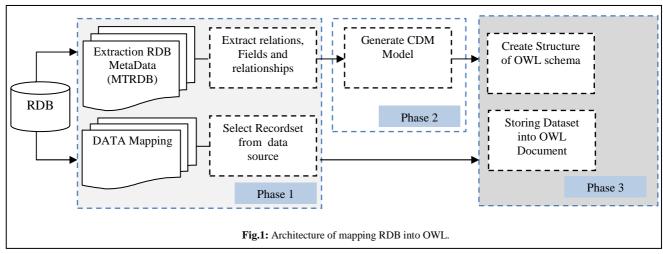
Furthermore, in our knowledge, there are some existing work raises the issue of constructing semantic mappings between relational schemas and ontologies.

In both Database and Semantic Web communities, more and more researchers have been aware of the importance for constructing semantic mappings

In our approach we have developed a tool to create ontology from a relational database. It looks for some particular cases of database tables to determine which ontology component has to be created from which database component. This prototype extracts the schema of the database (MTRDB) then transforms it into a canonical data model (CDM) to facilitate the migration process, after the system generates the structure of OWL file and the data of RDB is stored in an OWL document

II. OUR METHODOLOGY FOR MAPPING

In order to achieve flexible mapping and high usability, we presented our approach into three separate phases, as depicted in figure 1. The first phase consists to understand the structure of the relational database and its meaning. After, the Metadata of the relational schema (MTRDB) is extracted with the Recordset of the database and in the phase 2 we develop a Canonical Data Model (CDM) to facilitate the reallocation of field values in a class structure. Finally, in the phase 3 we describe the mapping process for generating the structure and data of OWL document. At the end we present our prototype for mapping RDB into OWL.



A. Mapping RDB into MTRDB

In this section, we present the proposed process for mapping the RDB into CDM.

1) Extracting MetaData of RDB (MTRDB).

Our process started by extracting the basic Metadata information about the RDB, including relations and fields properties.

In our approach an RDB schema is represented as a set of elements (Relation name (R_N) , set of fields (R_F) , Primary Keys (R_{PK}) , Foreign Keys (R_{FK}) and Unique Keys (R_{UK}))

$$MTRDB=\left\{R/R := R_{N}, R_{F}, R_{PK}, R_{FK}, R_{UK}\right\}$$

• R_N is the name of the relation and R_F describes the set of fields of the relation R is defined as a set of elements:

$$R_F = \{F | F := F_N, F_T, F_L, F_{NI}, F_D\}$$

Where:

- **F** is the field of the relation R.
- $\mathbf{F_N}$ is the name of F.
- **F**_T its type.
- $\mathbf{F}_{\mathbf{L}}$ is the data length of the field F.
- \mathbf{F}_{Nl} is nullable or not.
- **F**_D denotes the default value.
- R_{PK} denotes primary key of the relation (single valued key or composite key), .
- R_{FK} denotes the set of foreign key(s) of R, $R_{FK}(R) = \{FKn, R_{PK}(R')\}$, where FKn represents foreign key field name and $R_{PK}(R')$ name of an exporting (i.e., referenced) the second relation R' that contains the referenced R_{PK} .
- Relationships (RS): A relation R has a set of relationships RS. Each relationship (rel ε RS) between a relation R and another relation R' is defined as:

 $RS(R,R') := \{ \text{rel} \mid \text{rel} := (\ R_{PK}(R), R, R_{FK}\ (R),R',Ca) \}$ Where $R_{PK}(R)$ is the primary key of $R,\ R_{FK}\ (R)$ is the foreign key representing the relationship in R' and Ca the cardinality of the source relation R

Using the DatabaseMetaData interface for retrieving the structure of the database, the table in Figure 2 shows an overview of some instructions for extraction Metadata.

MTRDB: getMetaData							
R: TABLE_NAME							
F _N : COLUMN_NAME						RS	
F _T	$\mathbf{F}_{\mathbf{L}}$	F _{Nl}	F _D	R_{PK}	R_{FK}	R _{PK} (R)	R _{FK} (R)
TYPE_NAME	COLUMN_SIZE	IS_NULLABLE	COLUMN_DEF	getPrimaryKeys	getImportedKeys	PKCOLUMN_NA ME	FKCOLUMN_NA ME

Fig.2: the structure of MTRDB

2) Algorithm for extraction of MTRDB

This section presents the algorithm for extracting *MTRDB*, is used to extract the information about MetaData of RDB, which contains the names of the relations, fields and integrity constraints of all the relations extracted from an RDB. The input to the algorithm is an existing RDB and the output is the *MTRDB* structure as described in the Section A.1. The algorithm for extraction the MTRDB from RDB is as follows:

Algorithm Extracting _MTRDB (BD: RDB) return MTRDB

MTRDB: = null; // a set to store RDB relations

For each relation $r \in RDB$ do

Create element R for storing the prosperities of the relation r.

 $R.R_N := Extract name of (r)$

For each relation $R_N \, \varepsilon \, \, R$ do

 R_N , F_N :=ExtractFieldName(R_n)

 $R_{N.} F_{T} := ExtractFieldType(R_n)$

 $R_{N.}$ F_{L} :=Extractlengthofthefield (R_n)

 $R_{N.} F_{Nl}$:=ExtractBoolean (R_n) // (0 nullable /1 not nullable)

R_{N.} F_D:=ExtractFieldDefalutValue (R_n)

End For

 $R_N.R_{PK} := ExtractPrimaryKeys(R_n)$

 $R_N.R_{FK} := ExtractForeignKeys(R_n)$

 $R_N.R_U:=ExtractUniqueKeys(r)$

End For

For each set of relations (R, R') Create element RS for storing the prosperities of the relationships between R and R'.

 $RS.R_{PK}(R) := ExtractPrimaryKey(R)$

RS.R:= ExtractRlation (R)

 $RS.R_{FK}(R) := ExtractForeignKey(R')$

RS.R:= ExtractRlation (R')

End For

MTRDB: = MTRDB+ R // add the relation R to MTRDB

Return MTRDB

End algorithm

B. Generating CDM from MTDATA

The next step is to define the CDM based on a classification of relations, fields and relationships, which may be performed through data access.

The CDM model is based on three concepts: class, attribute and relationship. Attributes define class structure, whereas relationships define a set of relationship types. CDM classes are connected through relationships.

CDM Class is defined as a set of classes, is denoted as 3-tuple where the first element is the name of the CDM class, the second element is a list of attributes and the latest element is the relationships between classes:

$$CDM - Class := \{C|C := (C_N, C_A, C_R)\}$$

 C_N is the name of the class C, C_A is the list of attributes associated with this particular class:

$$C_A := \left\{ A \middle| A := (A_n, A_t, A_l, A_d) \right\}$$

Where A_n is an attribute name, A_t is its type, A_l is the length of this attribute and A_d is a default value if given.

 C_R describes the different types of relations that can exist between any pair of classes in the CDM.

$$C_R := \{ \operatorname{Re} lN, \operatorname{Re} lC, Cs, Cd \}$$

Where Re lN is the name of the relationship between the source class Cs and the destination class Cd and Re lC is

the Cardinality source of the class Cs, is represented by min...max notation.

C. OWL Structure

1) definition of OWL structure.

When the CDM has been obtained, the schema translation phase is started. Then, an appropriate set of rules is used to map the CDM constructs into OWL classes and create elements for storing OWL data

A class in OWL defines a group of individuals that belong together because they share some properties. Every individual in the OWL world is a member of the class owl:Thing. Thus each user-defined class is implicitly a subclass of owl:Thing.

Each class in CDM is translated to owl:class in the Web ontology, our class in OWL technology is represented as follows:

$$<$$
 owl : Class rdf : ID = "Class \in C_N"/ $>$

Each attribute A is translated into a owl:DatatypeProperty class and represented as:

```
< owl : DataTypePr operty rdf : ID = " A \in C_A " > < rdfs : domain rdf : resource = " \# C \in CDM - Class" / > < rdfs : range rdf : resource = "&xsd, Type \in A_t" / > < /owl : DatatypePr operty >
```

The relationship between two classes C1 and C2, the representation of the relationship in Web ontology is represented as follows:

```
< owl : ObjectProp erty rdf : ID = "RelN \in C<sub>R</sub>"> < rdfs : domain rdf : resource = "#C<sub>1</sub> \in CDM - Class" / > < rdfs : range rdf : resource = "#C<sub>2</sub> \in CDM - Class" / > < /owl : ObjectProp erty >
```

The cardinalities of a relationship are given by specifying minimum and maximum cardinalities.

For mapping the general cardinality we use:

<owl:Cardinality rdf:datatype="&xsd,nonNegativeInteger">
Cardinality \in RelC/owl:Cardinality>

And for mapping the maximal cardinality of each relationship we use this syntax:

<owl:maxCardinality
rdf:datatype="&xsd,nonNegativeInteger"> Cardinality
∈RelC</owl:maxCardinality>

2) Algorithm for Mapping CDM into OWL

Given a CDM Model as input, the algorithm goes through a main loop to classify CDM constructs and generate their equivalents in OWL.

The pseudo code of the mapping process is depicted in this Algorithm:

Input: The CDM model and Recordset of RDB

Output: The corresponding OWL schema and OWL Data

Step:

- **Step 1**: Translate each class in the CDM model into a Class in <OWL:class>.
- **Step 2**: Map each attribute and there proprieties in every CDM Class into <owl:DatatypeProperty> class.
- **Step 3**: Map the relationship between CDM classes into owl:ObjectProperty class .
- **Step 4**: Create an instance element of each recordset in RDB and translate the dataset of the recordset into instance.
- **Step 5:** Create an OWL schema for storing CDM structure and OWL data for storing dataset.

EndAlgorithm

III. EXPERIMENTAL STUDY

To demonstrate the effectiveness and validity of our method, a prototype has been developed. The algorithms were implemented using Java and Oracle/Mysql.

As an example, Figure 3 shows a relational database, PKs are bold and FKs are marked by "#".

- Product(**ProductID**, ProductName, ProductPrice)
- Customer(CustomerID, CustomerName, CustomerAdress)
- Employee(**EmployeeID**, EmployeeName)
- Order(OrderID, OrderDate, OrderQuantity, #CustomerID, #ProductID, #EmployeeID)
- Store(**StoreID**,StoreName)
- EmployeeStore(#EmployeeID,#StoreID

Fig. 3. Sample Relational database

The Conversion phase consists to converting existing RDB data to the text format defined by the target schema. Data stored as tuples in an RDB are converted into complex individuals in OWL document. We propose using CDM to guide the conversion process. Firstly, the RDB relations tuples are extracted using MetaDatabase instances. Figure 4 shows the RDB structure extracted from database. Secondly, these data are transformed to match the target format. Finally, the transformed data are stored into text files.

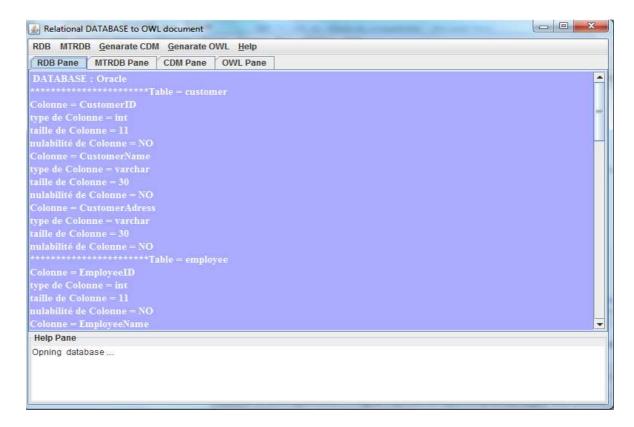


Fig.4: Extracting the RDB structure from database.

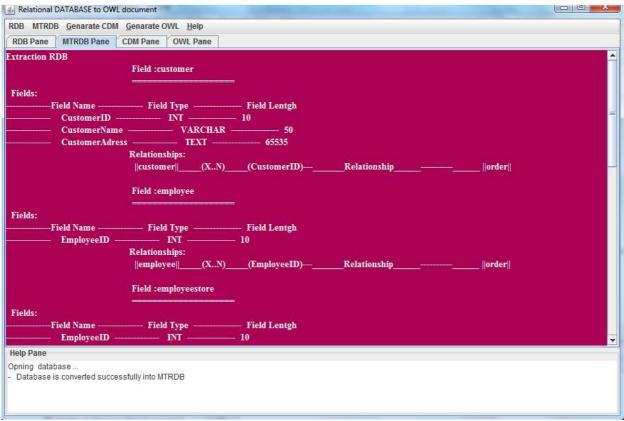


Fig.5: The MTRDB structure

The algorithm classifies each relation in the MTRDB by matching its attributes, primary key, foreign keys and its constraintes, and then maps the relation into CMD classes. Figure 5 shows the structure MTRDB extracted from RDB

During the mapping process, a CDM structure is automatically generated by the system to record the relationships between generated ontology components and the original database components, as shown in the platform of Figure 6

IV. RELATED WORK

In recent years, with the growing importance and benefits provided by Web semantic, there has been a lot of effort on migrating RDBs into the relatively newer technologies (XML/RDF/OWL) [5], [6], [7], [8]. Before applying a method for mapping relational database into web ontology, it must first extract the conceptual schema relational model. Extracting conceptual schema from a logical RDB schema has been extensively studied [9], [10]. Such conversions are usually specified by rules, which describe how to deduce RDB constructs (e.g., relations, keys), classify them, and identify the relationships. Fonkam et al [11] propose also an algorithm for converting RDB schemas into conceptual models

Blakeley [12] proposes a method for mapping RDB, this method consist to generate mappings between RDB and RDF with the RDB table as a RDF class node and the RDB column names as RDF predicates. Cullot et al [13]. use an efficient method for generating classes from tables and converts column to predicate, by using the specific relational database schema characteristics, after the mappings are stored in a R2O document.

V. CONCLUSION

In summary, the main achievements of this paper are listed as follows. Firstly, we have presented a new approach for mapping relational database into Web ontology. It captures semantic information contained in the structures of RDB, and eliminates incorrect mappings by validating mapping consistency. Secondly, we have proposed a new algorithm for constructing contextual mappings, respecting the rules of passage, and integrity constraints.

Finally, we have experimentally evaluated our approach on several data sets from real world domains. The results demonstrate that our approach performs well as compared to some existing approaches in average.

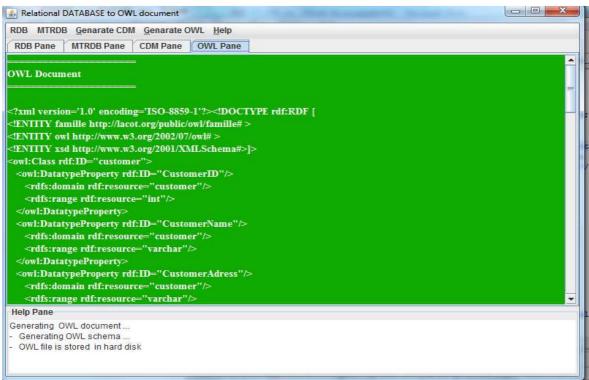


Fig.6: OWL data structure exported by the system.

VI. REFERENCES

- W. J. Pardi, XML in Action, Microsoft Press, Washington, 1999.
- [2] Fahrner, C. and Vossen, G.: Transforming Relational Database Schemas into Object-Oriented Schemas According to ODMG-93.In 4th Int. Conf. on Deductive and Object-Oriented Databases,pp. 429–446, 1995.
- [3] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker. Survey of directly mapping sql databases to the semantic web. Knowledge Eng. Review, To Appear 2012
- [4] M. Arenas, E. Prud'hommeaux, and J. Sequeda. Direct mapping of relational data to RDF. W3C Working Draft 24 March 2011, http://www.w3.org/TR/rdb-direct-mapping/.
- [5] Green, J., Dolbear, C., Hart, G., Engelbrecht, P., Goodwin, J."Creating a semantic integration system using spatial data", , in *International Semantic Web Conference* 2008 Karlsruhe, Germany
- [6] Noreddine Gherabi and Mohamed Bahaj. Robust Representation for Conversion UML Class into XML Document using DOM. International Journal of Computer Applications 33(9):22-29, November 2011
- [7] Cristian P´erez de Laborda and Stefan Conrad. Relational.OWL A Data and Schema Representation Format Based on OWL. In Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005), volume 43 of CRPIT, pages 89–96, Newcastle, Australia, 2005.
- [8] Tirmizi et al, "Translating SQL Applications to the Semantic Web", Tirmizi, S., Sequeda, J., Miranker, D., Lecture Notes in Computer Science, Volume 5181/2008 Database and Expert Systems Applications-(2008)

- [9] Wu, Z., Chen, H., Wang, H., Wang, Y., Mao, Y., Tang, J., Zhou, C., "Dartgrid: a Semantic Web Toolkit for Integrating Heterogeneous Relational Databases", Semantic Web Challenge at 4th International Semantic Web Conference (ISWC 2006), Athens, USA, 5-9 November 2006.
- [10] Alhajj, R.: Extracting the Extended Entity-Relationship Model from a Legacy Relational Database. Inf. Syst, vol. 28, pp. 597–618, 2003.
- [11] Fonkam, M. M. and Gray, W. A.: An Approach to Eliciting the Semantics of Relational Databases. In 4th Int. Conf. on Advanced Info. Syst. Eng., vol. 593, pp. 463–480, 1992.
- [12] Blakeley, "RDF Views of SQL Data (Declarative SQL Schema to RDF Mapping)", Blakeley, C., OpenLink Software, 2007.
- [13] Cullot, N., Ghawi, R., Yetongnon, K., "DB2OWL: A Tool for Automatic Database to Ontology Mapping", In Proc. of 15th Italian Symposium on Advanced Database Systems (SEBD 2007), pages 491-494, Torre Canne, Italy, 17-20 June 2007.