An Enhanced Technique for Incorporating Schema Changes into Ontology

Waqas Ahmed, Muhammad Ahtisham Aslam

COMSATS Institute of Information Technology Lahore

wqs.ahmed@gmail.com , ahtisham@ciitlahore.edu.pk

Abstract

Semantic Web is future of the current web which will be equally understandable for both human and computers. Ontology is specification of shared conceptualization. However, ontology creation from scratch is very laborious, time consuming and requires intensive domain knowledge. Most of the industrial data is present in relational databases. A relational database schema represents the domain model. An ontology constructed from this schema can represent the concepts in the domain of discourse. But research and public databases are not static. Their schema evolves over time. Once a database schema is changed, these changes in schema must also be incorporated in database ontology. To do so, we present a framework for incorporating schema changes into ontology which will help us to generate and synchronize ontology with database schema.

I. INTRODUCTION

Web semantics are the representation of contents in machine process able form. These semantics and meanings are required to be expressed in a particular way to make them useable by machines. Traditional World Wide Web presents contents for human whereas focus of Semantic Web is to mange and present contents for machines to manipulate them [8].

There are more than 850 public biological research databases [4] and they can be integrated together to retrieve specific information [9]. All these public databases are dynamic in nature. These databases are not static and their schema keeps on changing. An ontology which is a formal, explicit specification of a shared conceptualization [7] represents the abstract domain model. This domain model can be used to infer information not presented explicitly.

A relational database schema also represents the domain model by representing the entities present in the domain, the attributes associated to these entities and relationships among these entities.

Therefore ontology can be generated automatically from relational database using schema information. This automatic ontology generation from relational database schema not only makes it possible to construct ontology without detailed domain analysis but also is very effective in term of time and cost. Various techniques of database to ontology generation are discussed in [2, 3, 5].

Once ontology is generated from database, it may go through manual annotations and at the same time the database schema, from which ontology was generated, may also change. This change in database schema refers to a change in domain model. This change in domain model should also be represented into ontology that was generated from database. In this paper we present a framework for ontology generation and change synchronization from database schema.

Organization of this paper is as follow: in coming section we discuss some related work and their shortfalls. In section 3 we present ontology generation method from database. Section 4 goes into details on how the changes in database schema will be detected and implemented. Section 5 compares our approach with existing work and we conclude with some implications of our work and future work.

II. RELATED WORK

Justas Trinkunas and Olegas Vasilecas [3] presented a reverse engineering approach for ontology construction from relational databases. Their approach first transforms a database schema into conceptual ER model. Then this conceptual ER model is transformed to ontology using transformation rules. Both, the model and ontology, are expressed using directed graphs .This transformation can be represented as G_{ER} (Model) $\rightarrow G_O$ (OWL Ontology) where G_{ER} is graph representing ER model and G_O is Ontology graph. During transformation process each entity element of conceptual graph is transformed to OWL class element. An attribute is transformed to OWL data type property. Relationships in conceptual model are transformed to OWL Object properties. Following transformations are performed

1. G_{ER}(Model) $\rightarrow G_O$(OWL Ontology)
2. G_{ER}(Attribute) $\rightarrow G_O$(Data Property)
3. G_{ER}(Relationship) $\rightarrow G_O$(Object Property)

Ontology construction approach presented in [2] generates RDF ontology from existing relational database schema. This approach works by generating an RDF class for each database table. The instances of database table become the instance of mapped class in ontology. Each column in table becomes a property. The column representing a foreign key has domain set to the class name in which it is foreign key and range as the name of class in which it is a primary key. A class structure that represents the class subclass relationship is specified manually.

DB2OWL [5] is an automatic tool that generates ontology from relational database. This generated ontology is expressed in OWL-DL. The tool uses predefined mapping rules for database schema to ontology concepts and constructs. Mapping process maps each database table to concept in ontology and columns are mapped as properties. Joining or bridging tables which represent a many-to-many relationship between two tables are not exactly mapped as classes in ontology. This many- to-many relation is expressed using object properties. Mapping process also takes care of class /subclass relation as well. One -to- many relation constraints between two tables is mapped as an object property expressing relations between corresponding classes in ontology.

In their work "A Coevolution Approach for Database Schemas and related Ontologies" Adreas Kupfar et al. [1] has presented an approach to incorporate schema changes into related ontology. Firstly ontology is generated from database schema. Ontology is implemented with small subset of OWL called OWL-Lite. During ontology generation process concepts for terms Database, Relation, Attribute and an object property ConsistOf are created to express hierarchy. Ontology generation process automatically extracts database schema and generates instances of these concepts. Some of the information is not available in database schema but can be very helpful if added to ontology. A conceptual model may contain this vital information. If a conceptual model is available then some relations are also added manually from it. Now in database there can be following schema changes

- Creation of new table
- Deletion of an existing table
- Alteration of existing table

Addition of new table into database is mapped into ontology as addition of an instance in concept Relation. Deletion is bit complex in nature. When a database table is deleted it is necessary to delete all links as well. All renaming operations are treated as additions or deletions. If a column is renamed, a new instance of attribute is added to ontology and previous one is removed.

Table 1. summarizes some related work regarding database to ontology construction.

III. ONTOLOGY GENERATION AND CHANGE SYNCHRONIZATION

The solution consists of two phases. Ontology from database schema is prerequisite for our framework. Therefore first we will construct domain ontology from database schema using reverse engineering. Once ontology is generated, a change synchronization mechanism between database and ontology will be developed.

A) Phase One: Ontology Generation

The database schema has the information of physical implementation of the database. It contains the information of entities and relationships among them. These entities usually represents the terms in the domain of discourse. Ontology is the collection of terms present in domain of discourse and relationships present among these terms.

TABLE 1

Summary of Related Work

Title	Technique	Limitations
A Technique for Automatic Construction of Ontology from Existing Database to Facilitate Semantic Web	Transformation rules from direct database schema to ontology constructs are predefined and schema is directly converted to ontology. Every table becomes an RDF class in ontology and all instances of the table become instances of the class corresponding table name. Attributes of table are converted to properties	The technique also converts bridging tables into classes where as the relations they are specifying is required to be specified as restrictions in related classes. Another issue with this technique is that class/subclass relationship has to be specified manually
DB2OWL: A Tool for Automatic Database-to Ontology Mapping	This technique converts a database schema into new owl ontology. The mapping takes place according to predefined table cases. This case set includes relation between tables, class/subclass relationship translation. During mapping process an R2O document is automatically generated to support query interoperability	The approach directly converts physical schema into ontology and does not specify what to do with table instances
Building Ontologies from Relational Databases Using Reverse Engineering Methods	A relation to conceptual modeling is performed. The conceptual model is presented in the form of graph. This conceptual graph is then transformed to ontology graph using mapping rules	The process is semi automated. The database schema to conceptual modeling is performed by third party tool. The graph is then transformed to ontology

Therefore keeping this similarity in mind, ontology can be constructed from existing database.

Every entity in the database corresponds to terms or classes in the ontology. The relationships among these entities can be the object properties in ontology. Different classes in ontology can be related using these relations for example the university database structure provided in [10] the entity Alumnus can be represented as OWL class as follow

```
<Declaration>
 <OWLClass
URI="&Ontology1219867591494;
Alumnus"/>
 </Declaration>
```

Above mentioned snippet declares an OWL class named as Alumnus. Every term has a unique identifier which is unique among all terms present in ontology. This unique resource identifier (URI) is also mentioned in snippet.

Relational databases use a process of specialization in which there exist class/subclass relationship between entities. This inheritance relationship is very important for ontology. In University database example the Alumnus is subclass of class Person. This relation in ontology can be specified as follow

```
  <SubClassOf>
    <OWLClass
URI="&Ontology1219867591494;Alum
nus"/>
<OWLClass
URI="&Ontology1219867591494;Perso
n"/>
```

The code segment mentioned above declares that class Alumnus is a subclass of class Person. This assertion creates a class hierarchy that then can be used to infer implicit information.

i. Physical to Conceptual Graph

Our new technique will automatically transform physical schema into conceptual model. This conceptual model will be represented by a directed graph [3]. Each entity will be represented by a node N. Each edge will be written as triplet (N1, α, N2) where n1 and n2 are nodes and α is label of the edge. For our example of university database, the entities Person and Alumnus will become nodes. Each node will have its associated attributes. The edges will represent relationship between nodes. The relationship between Person and Alumnus is of class/subclass therefore the edge will look like this

N1 = Person

α = hasSubClass

N2 = Alumnus

Class/Subclass relationship will be determined from physical schema along with other relations using conceptual to physical mapping rules. One way to represent class/subclass relationship in physical schema is using the same attribute as primary key in both super and sub class. This relationship can be determined and then converted to conceptual graph model.

ii. Ontology Generation

In next step, this graph will be transformed to ontology using owl specifications. Different relations other than class/subclass will be represented using object properties. All attributes except foreign keys of a table will become the data properties. Other properties of relations will be created according to the nature of relationship that exists in the database. Fig. 1 represents phase one of solution.

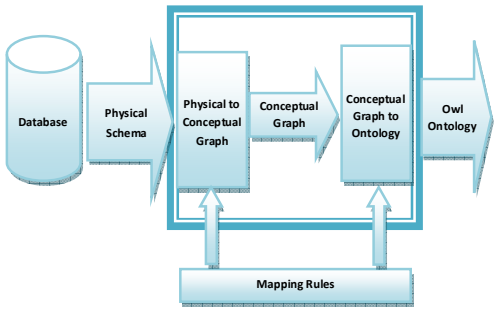

Fig 1. Phase one: Ontology Generation

B) Phase Two: Change detection and Implementation

Once ontology is generated from database schema there is possibility that database schema is changed. This change can be addition of new tables, deletion of existing tables, addition or deletion of columns or new relationships. These changes must also be made to ontology to maintain a consistency between both ontology and database. Generally once created, ontology also goes through manual maintenance which could cause changes in it as well.

i. Schema change Detection Technique

Now to incorporate schema changes into ontology constructed from this schema, the knowledge of difference between ontology and database schema is mandatory. In order to determine this difference changed schema will be transformed to conceptual graph as it was done during ontology generation process. Ontology will also be transformed into conceptual graph using ontology to conceptual model mapping rules. Now both graphs will be compared and difference will be the changes in ontology and database schema. This difference will be computed during change set computation phase. These schema changes from database perspective will then be made to ontology using schema change mapping rules. Fig. 2 represents this phase

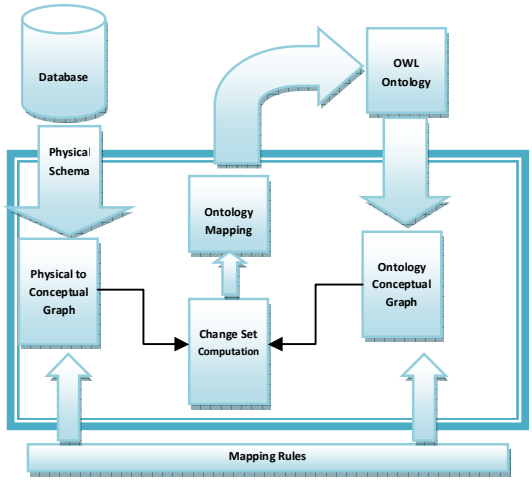

Fig 2. Phase Two: Change Detection and Implementation

IV. Comparison with Existing Work

In [1] a mapping scheme for change implementation into ontology is presented. This work is based on ontologies created using OWL-Lite which is least expressive ontology language and has no support for relations and other constraints. We will use OWL-DL [6] for ontology generation and then changes will also be made to these ontologies using new change mapping rules for relations.

Our database schema chance identification approach is novel. In work cited above it is assumed that schema changes will be obtained from database schema change logs. These change logs are not available and not updated most of the time. Moreover this change detection approach does not consider the changes made into ontology. For example is a new table is added to database and this change is also made to ontology manually. Existing technique is not able to identify that this concepts already exist in ontology as it does not consider ontology during change identification process**.**

V. IMPLICATIONS

As discussed above, the public research databases are not static. Their schema evolves over time. ontologies are used to represent domain of discourse and make inference not possible by databases. For better results and correct representation of domain model, database schema and related ontologies must correlate with each other. Changes in database base schema should also be made to ontology to gain maximum accuracy. Our approach will provide mechanism for simultaneous evaluation of both, database schema and related ontology.

Another benefit of proposed approach is that it detects changes made to both the database and ontology resulting better ontology. In our framework we presented scheme to map database schema changes to ontology. Changes in ontology can also be mapped to database in future.

VI. Conclusion and Future Work

We presented a two phased process of generating ontologies from database schema and then detecting changes in database schema and implementing these changes into ontology. In future plans are to determine mapping rules and implement our presented approach as protégé plug-in.

REFERENCES

[1] A. Kupfer; S. Eckstein; K. Neumann; B. Mathiak, "A Coevolution Approach for Database Schemas and Related Ontologies," in *19th IEEE International Symposium on. Computer-Based Medical Systems, 2006. CBMS 2006.*, pp.605-610, 2006

[2] Mukhopadhyay, D.; Banik, A.; Mukherjee, S., "A Technique for Automatic Construction of Ontology from Existing Database to Facilitate Semantic Web," in *10th International Conference on Information Technology (ICIT 2007).* , pp.246-251, 17-20 Dec. 2007

[3] Trinkunas, J. and Vasilecas, O. 2007. "Building Ontologies from Relational Databases Using Reverse Engineering Methods". In *Proceedings of the 2007 international Conference on Computer Systems and Technologies* (Bulgaria, June 14 - 15, 2007). B. Rachev, A. Smrikarov, and D. Dimov, Eds. CompSysTech '07, vol. 285. ACM, New York, NY,16.

[4] M. Y. Galperin. The Molecular Biology Database Collection:2006 update. *Nucl. Acids Res.*, 34(suppl1):D3–5, 2006.

[5] Cullot N., Ghawi R., and Yétongnon K. DB2OWL: A Tool for Automatic Database-to-Ontology Mapping. In *Proceedings of the 15th Italian Symposium on Advanced Database Systems*, 2007.

[6] S. Bechhofer, F. v. Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Stein.

OWL web ontology language reference, Feb. 2004. World Wide Web Consortium.

[7] Studer R., Benjamins V. R., Fensel D, "Knowledge Engineering. Principles and Methods", In IEEE Transactions on Data and Knowledge Engineering 25 (1998) 1-2, p. 161-197.

[8] Berners-Lee T., Hendler J., and Lassila O. The Semantic Web. *Scientific American*, May 2001.

[9] Z. Lacroix and T. Critchlow. *Bioinformatics – Managing Scientific Data*. Morgan Kaufmann, 2003.

[10] R. Elmasri, S. B. Navathe, *Fundamentals of Database Systems*, 3rd ed, International Edition: Addison –Wesley, 2000, pp. 84