The Approach for Migrating Data Intensive Web Application into Semantic Web Application

Munteha Riaz and Amjad Farooq

Computer Science and Engineering Department,
University of Engineering and Technology Lahore, Pakistan

***Abstract*— Upcoming revolution in history of web is semantic web. Apparently solution of many drawbacks of current web is Semantic web. Our goal is to transform Data Intensive Web to Semantic web by applying set of rules. After study and evaluation of different approaches, a rules set is been presented, in consolidated and more generalized form, which is integrated and also semi automatic. Also some relationship rules are proposed to map relationship entities onto Ontology classes, which eventually assist in transforming to Semantic web.**

Keywords – *Web Applications, Web-Semantic and Data Intensive*

I. INTRODUCTION

This paper focuses on the reverse engineering technique of migration data Intensive Web into Semantic Web by applying a specific rule set on RDBMS. This technique starts with conversion of source relation database schema to equivalent ontology classes. Then generated ontology classes using for mapping database contents to ontology based knowledge base which is converted into RDF and available over the web. After publishing these RDF on the web means content of web page is available and also understandable for machine which is perquisite of semantic web.

As there are a large number and variety of rules described, to accommodate different scenarios, which have their impact and benefits in those specific terms. So some common rules have been collected and generalized to assist in ontology creation of most of the data intensive relational database into Ontologies, and eventually Semantic Web.

Moving on, some generic relationship rules are also been proposed that can facilitate these relationships to transform from a relational database to ontology.

II. SEMANTIC WEB AND ONTOLOGY

Now-a-days the most popular source of information is World Wide Web. The number of users and the attention it attracts speaks for themselves but this great source of information is only understandable to human beings. Moreover the World Wide Web is a huge group of documents, images, and even sounds that put burden on the user for extracting and interpreting relevant information. So, a need arose for a modification in traditional web so that it is understandable by machines, this is where semantic web comes into action.

The Semantic Web is about two things.

- Integration formats which are common
- Grouping of data drawn from various sources

On the other side, web is all about documents interchanging and languages that helps in how data and real world objects relates to each other. This helps a user/machine, to start from one database and move to never-ending set of database because all databases are linked on the bases of having same thing.

The W3C gives the following definition for the Semantic Web:

"*The Semantic Web is an extension of the current Web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation*"

Let's assume we are on the web for some information gathering, for instance we need to know the best Hotel in different places of the world so that we can organize our summer trip at our best. For this we have to consult different sites which may provide area specific information, some relative information, or may differ in style, purpose or language which may come out to be useless and redundant for us. So what we need is to mentally or manually formalize all such information at our end then integrate it, filter it out for our purpose, acquire relevant information and spend a lot of time to actually get what we want.

So the problem at end now is that we only have access to the form of data exposed to us by web designers. There are some sites which gather data for us like Expedia and Trip Advisor which present us gathered information, but it is the way designer shows us data, still we cannot access the data we want.

What about the idea of accessing the original data and then combining it the way we need to acquire i.e. perceive data from web in the same manner as we do from documents so that we could be able to link to data and use the data according to our needs.

Or in other words, we would like to "*extend"* the data of current Web to a "*Web of data" i.e.* Permit for applications to utilize the data directly.

It can be said that mash up sites are already doing same sort of work, but mash up sites enforce some unplanned tasks but Semantic web is "web of data" and "Semantic Web Technologies" is a group of standard technologies to realize a Semantic web. The basic component of semantic web is

Munteha Riaz and Amjad Farooq are with the Computer Science and Engineering Department, University Of Engineering and Technology Lahore, Pakistan (Email: amjadfarooq@uet.edu.pk).

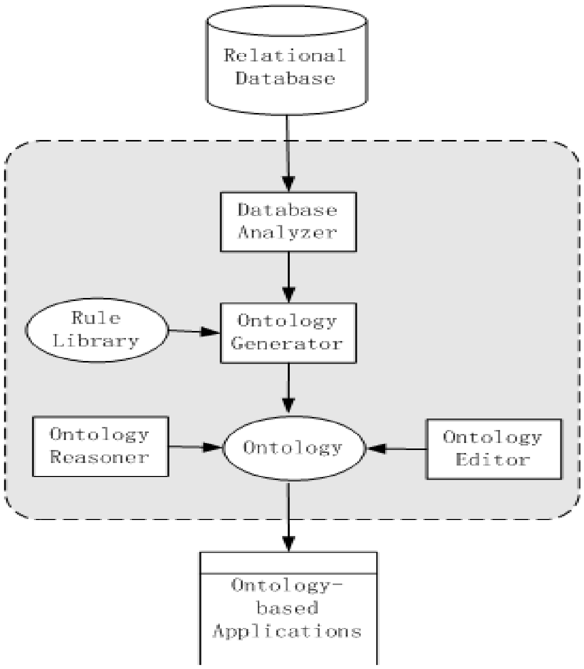

Fig. 1. Ontology learning Framework

ontology. Ontolgies present such domain model which is shared and understandable for both machine and human. By ontologies, content of a web become suitable for machine use and machine perceive content in a same way as human do. But in case of data intensive current web pages, contents are just for human utilization. Fig. 1 shows the Ontology learning Framework [1].

After ontology next step is to define semantics of web pages and this step requires defined concepts, relations and properties of these concepts. Objective of Semantic web is expressing web page meaning and to achieve this objective semantic web is divided into layers structure (Fig. 2).

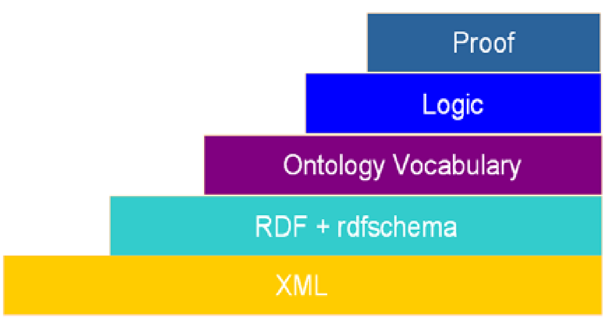

Fig. 2. Semantic web layers structure

- ***XML layer:***
  This layer is used as layer for syntax. It gives arbitrary structure and does not produce any semantics.
- ***RDF layer:***
  A data layer where data is presented in a uniform way in RDF Schema.
- ***Ontology layer:***
  This layer indicates data semantic.
- ***Logic layer:***
  This layer gives rules and logic that facilitate in making of intelligent analysis.
- ***Proof layer:***
  This layer is used to support communication for proofs exchanging.

As semantic web have matchless benefits so in web 3.0 it is better to transform data intensive sites to semantic web. In this paper, achieving semantic web by different techniques is going to address. How semantic web replaced data intensive web, in which web pages are fixed and static, with such web pages in which data is generated at user request. Relational database and reverse engineer is one of solution to achieve semantic web and following are the benefits:

- Getting metadata is automated and fast.
- Dynamic web page's content is machine understandable
- Content of dynamic web pages visible to such search engines which are specialized
- There was a problem of updating metadata on basis of dynamically changes in data intensive web. It is resolved in semantic web.
- Different community member's information can be exchanges on web 3.0.

Database is major part of data intensive web applications due to which data bases is very heavy and large in term of contents. As we transform data-intensive web applications into semantic web, application data is essential target, so here we are going to use reverse engineering approaches and relation data base as input. One of initial and basic technologies regarding semantic web is known as Ontologies. Database schema of web is explained in machine annotation form then defining terms utilized in such annotation must be fixed in ontology that can be shared. After analyzing reverse engineering approaches, following are those that targets Ontologies (Table 1) as output [3-8].

- Kashyap's Approach
- Rubin et al.'s Approach
- Stojanovic et al.'s Approach
- Astrova's Approach

After ontology next step is to define semantics of web pages and this step requires defined concepts, relations and properties of these concepts. Objective of Semantic web is expressing web page meaning and to achieve this objective semantic web is divided into layers structure.

In process of converting data- intensive web applications to semantic web Ontology creation is basic step. Various steps and rules have been proposed for this type of Ontology creation. In this thesis, Ontology acquisition from relational database (OARDB) and transformation rules taken, generalized and consolidated from [1] and [2] are presented and discussed. These are the set of rules which give more precise results and provide more depth in Ontology generation collectively and in presented order rather than applying them on specific data set, which may give limited results in some

Table 1: Analyzing Reverse Engineering Approaches

	Kashyap's Approach	**Rubin *et al.*'s Approach**	**Stojanovic *et al.*'s Approach**	**Astrova's Approach**
Ontology Language	N/A	n/a	Frame logic	n/a
Axiom creation	X	✓	✓	✓
Automatic ontology population	X	✓	Semi-automated	X
Semantic character analyzed	Not always	✓	n/a	n/a
Ontology required	X	✓	X	X
XML schema required	X	✓	✓	X
XML translator required	X	✓	X	X
Auxiliary information required	Not always	X	X	✓
Identify inheritance relationship	✓	N/A	✓	X
Domain specific approach	✓	X	X	X
ER model involved	✓	X	X	✓
Ontology Refinement	✓	N/A	✓	X
Disadvantages:	Does not create axioms, which are part of the ontology	This approach needs several components: ontology, XML schema, and XML translator	All behavioral parts of SQL, built in function and queries cannot be mapped.	It does not support inheritance relationship

cases. Following are the three major stages of acquiring ontology, by reverse engineering Relational Data base.

1. Extracting RDMS Information
2. Create Ontology
3. Data Migration

1). Extracting RDMS Information

Following are the RDBMS information needs to extract for Ontology creation:

i. Relation names
ii. Attribute names
iii. Primary keys
iv. Foreign keys
v. Integrity Constraints.

2). Create Ontology

- *Class creation (concept) rules*

Rule 1: For all relations in database, if in relations R1,R2,…,Rn, we have primary keys PK1,PK2,…,PKn. And in any relations i.e. R1 and R2, we have PK1 = PK2, then all the relations with same primary keys must be merged into one ontological class.

Rule 2: In database, a relation R1 can be converted into an Ontology class if, Rule 1 is not satisfied, R1 has must have only one primary key, R1's foreign keys should not greater than one and there exists an attribute A where $A \in pkey(R)$ and $A \notin fkey(R)$.

- *Property creation rules*

Rule 3: For all relations in a database, if there are two relations i.e. R1 and R2, and R1 contains set of attributes $A1, A2, \ldots, A_i$ which are also exist in R2, i.e. $R1(A_i) \subseteq R2(A_i)$, but set of attributes in not primary key of R1 i.e. $A_i \not\subset pkey(R1)$, then on base of $A_{i.,}$ P12 (an object property) can create. Assume there are two classes C1 and C2 of relation R1 and R2, so C1 and C2 are domain and range of object property P12and C1 and C2 are domain and range of object property P21. At the same time, P12 and *P21* are two inverse object properties.

Rule 4: If Rule 3 does not follows and any attribute is not created as 'Object Property', then attribute can be used to build "Data type Property" of respective Ontology class C.

- *Cardinality rules*

Rule 5: For a relation R, if an attribute $A \in R$ is PK or FK of R, then the both cardinalities (i.e. minCardinality, maxCardinality) of the property P equivalent to A is 1.

Rule 6: For a relation R, if an attribute $A \in R$ and is set NOT NULL, 1 is minCardinality of P corresponding to attribute A.

Rule 7: For a relation R, if an attribute $A \in R$ and is set UNIQUE, 1 is maxCardinality of P correspond to attribute A.

- *Class relationship rules*

Rule8: Rule 8 is rule of association and use following relationships:

- One-to-one
- One-to-many
- Many-to-many

Rule9: For Relations R1, R2 in database, if the primary key of R1 consists of only one foreign key referring to R2, then R2 is said to be sub class of R1.

3). Data Migration

After creation of ontology next step is migration of data. Objective of this step is use tuples of relation database and build ontology on their bases. Ontological structure is achieved by using above mentioned rules. Now to get ontological instances from relational database tuples we have some more rules to follow.

Rule 10: For any relation R, if class C is for relation R, then each tuple of relation R can map to ontological instance associate with unique identifier and that unique identifier can be formulate by appending primary key value to the name of relation R.

Rule 11: For a relation R, the values of the tuple can be mapped to the values of the corresponding property of ontological instance.

Rule 12: Relation R has foreign key F. Then foreign key value can be mapped to ontological instance and that ontological instance can mapped on the value of foreign key's object property.

III. CONCLUSIONS

The In this paper, following relationship rules are proposed: As in RDBMS, entities are directly mapped on to classes, so if these rules are directly applied to database entities, instead of first mapping them on classes and then converting these classes to Ontology classes, that would be beneficial to directly map these entities on to Ontology classes. By applying them on relational database schema, Ontologies can be refined in more precise way.

***Aggregation* –** Directional association among different objects is Aggregation. Two objects can be directional aggregated if one object has another object. Direction between them specified which object contains the other object. Aggregation is also called a "Has-a" relationship.

Aggregation Rule: "For all relations in data base, if relation R1 has primary key which is composite (unique key plus foreign keys) then R1 is said to have aggregation relation with the table(s) of these foreign keys."

***Composition* –** When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition. It is also known as restricted form of aggregation.

Composition Rule: "For all relations in data base, if R1 relation has composite primary key and there is no other attribute except all foreign keys, then R1 is said to have composition relation with the table(s) of these foreign keys."

IV. FUTURE WORK

For Future work, the rest of Relationship rules including Abstraction, Realization and Dependencies have been targeted. Once all the relationships are well ruled, it would be very efficient to convert those entities into Ontological classes. As in RDBMS, entities are directly mapped on to classes, so if these rules are directly applied to database entities, instead of first mapping them on classes and then converting these classes to Ontology classes, that would be beneficial to directly map these entities on to Ontology classes. By applying them on relational database schema, Ontologies can be refined in more precise way.

REFERENCES

[1] Learning Ontology from relational Database by MAN LI, XIAO-YONG DU, SHAN WANG School of Information, Renmin University of China, Beijing 100872, China.

[2] Chiang R., Barron T., Storey V. (1994) Reverse engineering of relational databases: Extraction of an EER model from a relational database. Journ. of Data and Knowledge Engineering, Vol. 12, No. 2, March 1994, pp. 107–142., 1994.

[3] Kashyap V. (1999) Design and creation of ontologies for environmental information retrieval. 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99), Banff, Canada, October 1999.

[4] Rubin, D.L., M. Hewett, D.E. Oliver, T.E Klein and R.B. Altman (2002). Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. In R.B. Lihue et al. (Eds.), Proceedings of the Pacific Symposium on Biology.

[5] Stojanovic, L., N. Stojanovic and R. Volz (2002). Migrating data-intensive web sites into the semantic web. In Proceeding of the 17th ACM Symposium on Applied Computing (SAC'2002), Madrid, Spain

[6] Astrova, I. (2004). Reverse engineering of relational databases to ontologies. In Proceeding of the 1st European Semantic Web Symposium (ESWS), Heraklion, Greece. LNCS, vol. 3053

[7] Tijerino, Y.A, D.W. Embly, D.W. Lonsdale, Y. Ding and G. Nagy (2005). Towards ontology generation from tables. 8(3)

[8] Astrova, I., and B. Stantic (2005). An HTML forms driven approach to reverse engineering of relational databases to ontologies. In Proceeding of the 23rd IASTED International Conference on Databases and Applications (DBA), Innsbruck, Austria.