Semantic Enrichment of Relational Databases

Hamaz Kamal
LIRE Laboratory, dept of computer sciences
Mentouri University
Constantine, Algeria
hamaz.kamal@gmail.com

Benchikha Fouzia
LIRE Laboratory
Mentouri University
Constantine, Algeria
f_benchikha@yahoo.fr

***Abstract*— With new technologies the semantic representation started to be highly requested especially for the stored data. Lately works to integrate ontologies within databases have appeared in order to store data with their semantics. In this paper we present a reverse engineering process that transforms relational databases to ontologies. These last are enriched under an enrichment process that uses external domain ontology. Our goal is to have a well representation of the domain semantics to create a more complete Ontology-Based Database.**

Keywords- relational databases; reverse engineering; ontologies; ontology-Based Database; Enrichment.

I. Introduction

Information systems play an important role in the development of enterprises and organizations. Lately most of enterprises want to upgrade their information systems with new technologies to provide better results and better interactions. Databases play one of the principal roles in the development of every information system. Nowadays the most used databases are still in the relational model, which appeared in the 70'. However, in the last decade the semantic representation of the stored data has started to be highly requested [9], especially for enterprises that stand to have well defined semantics within heterogeneous and distributed environments. Unfortunately, the relational model until then lacked a semantic representation of the stored data [10].

The Entity-Association 'E/A' model represents semantics, which are unfortunately lost once we represent it as a relational model. Other models have appeared after the relational one as the Object-Relational and Object-Oriented models, which allow representing some of the real world semantics.

The emergence of ontologies has given a new view of information representation with its semantics [14]. Ontologies provide vocabularies that can semantically and rigorously represent a real world domain. Hence they have been adopted within databases and allow many existing database's research areas to ameliorate the use of databases. We call a database that stores ontology an Ontology-Based DataBase (OBDB).

In this paper we present a process of relational databases reverse engineering to ontologies which use external domain ontology. Our goal is to provide an OBDB that gives better user' requests results.

The rest of the paper is organized as following: section two discusses the integration of ontologies in databases and their use in the process of databases reverse engineering. Some related works are presented. Our approach is detailed in section three. Section four presents a comparison study between approaches followed by a conclusion in section five.

II. Using ontologies in relational databases reverse engineering

In this section we present the concept of ontology and its integration in the databases field. In particular, ontologies receive a wide speed attention in the databases reverse engineering process. Some related works are discussed before presenting our proposed approach.

A. Ontologies in databases

An ontology is defined as a specification of a conceptualization [14]. It defines a hierarchy based on structured vocabularies, grouping together concepts/classes with their relationships, instances and eventually axioms.

Ontologies provide many advantages in comparison with the relational, conceptual and object models [8]. The main advantages can be highlighted as following: rigorous formulation of the conceptual schemas, the possibility to do inferences over the ontology in order to get more useful results, the possibility to use online dictionaries and vocabularies as WordNet. There are also possibilities to do tests to validate the ontology. Many works on using ontologies within databases have started to appear [6], [8], assuming that using ontologies will give more semantically efficient databases.

B. Related Work

The reverse engineering is generally defined as the process of reconstruction or restructuration of a model into a higher level model. For databases, it is mainly considered as the process of enrichment of a database source to give it the ability to discover more semantics. Many models were defined as the source-to-target of this process, passing from the conceptual model to the ontological model.

The existing work on reverse engineering of relational databases to ontologies can be classified into three categories.

1) Approach Based On Analysis of the Relational Schema: Most of existing works are based on this approach, and are mainly based on using the relational schema for creating the ontology, or for mapping it into an existing ontology. The process of Stojanovic et al [13], supposes that the ontology exists and follows rules as a process of reverse engineering to create mapping between the relational schema and the ontology to provide expected annotation. This approach aims to help on migrating static websites to the semantic web, and thus

978-0-7695-4845-6/12 $26.00 © 2012 IEEE
DOI 10.1109/IWAISE.2012.19

annotations will be needed. In [1], the proposed approach analyses the SQL-DDL code of the database and the correlations between keys, between attributes and data to discover more semantics and more dependencies. The process divides the relational schemes on base relations, composite relations and dependant relations. An extension of this work has succeeded in [2] and represents the resulted ontology on OWL. It also takes into account the representation with OWL of the database constraints as CHECK and UNIQUE. Similarly, Buccella et al [5] tries to create ontology from SQL-DDL code only following a set of rules.

2) Approach Based On Analysis of User Queries: The most representative of this approach is Vipul Kashyap's approach [11], which shows that it's possible to use user queries for the creation of new concepts for an ontology. The approach creates a first ontology from the relational schema and enriches it after analyzing different user queries. The approach is oriented to specific domains to decrease the number of conflicts.

3) Approach Based On Analysis of HTML Pages: In this category we find the work of Astrova et al [3], and the one of Benslimane et al [4]. This last uses HTML pages that communicate with the database and analyses them to discover more semantics. The main purpose of this approach consists on the extraction of the information found on HTML pages, and organizes them in a way that permits to discover relevant concepts, constraints and dependencies between data. The process takes also sub-relational schemas and transforms them to sub-conceptual schemas based on UML, after this all sub-conceptual schemas will be transformed to one global OWL ontology.

III. The Proposed Approach

In this section we present our approach of relational databases reverse engineering to ontologies.

A. General Architecture

The architecture of our reverse engineering process shown in figure 2 is divided in three steps: transformation process, enrichment process and data migration.

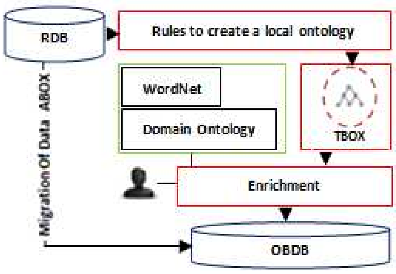


Figure 1. Reverse engineering approach.

B. Reverse Engineering Process

In this paper we present the two first steps of our reverse engineering process. The migration of data will be detailed in future work. We suppose that the relation schema is in the 3NF, and that we have the set of the candidate keys of the relational schema. Table 1 presents an example of a relational schema, which will be used to show some results on the application of rules.

TABLE I. Relational Schema.

Relation Schemes:
Person(Id, Name, PhoneNum, Address, Country) Student(StudentID, Deprtn) Professor(ProfessorID, Salary) Article(Idart, Author, Title, Domain, Event) Department(IdDep, Name, Organisation) Prepares(StudentID, Idart) Publication(Idpub, Publisher, Title, ThesisId)
Foreign keys/Candidate keys:
StudentID references Id of Person, Deprtn references IdDep of Department. ProfessorID References Id of Person. StudentID references Sudent, Idart references Idart of Article. Idpub references Idart of Article, Publisher (is a candidate key) & references StudentID of Student. ThesisId is a candidate key.

1) The Transformation Process: This process is composed into four sets of rules for the creation of classes, hierarchy, properties and attributes of classes. We use OWL to express results of transformation.

Rule1: Create a class for each relation in the database where the primary key isn't composed exclusively of foreign keys

e.g: <owl:Class rdf:about="Person"/>

Rule2: Create a generalization relationship between two classes if the relations related to the classes have the same primary key, and/or the values of one of the primary key are a subset of the values in the second primary key.

e.g: <owl:Class rdf:about="Person">
<rdfs:subClassOf rdf:resource="Student"/>
</owl:Class>

Rule3: Create a generalization relationship between two classes if: one of the two relations related to the classes has its values of the primary key included in the candidate key of the second relation.

e.g: <owl:Class rdf:about="Publication">
<rdfs:subClassOf rdf:resource="Article"/>
</owl:Class>

Rule4: Create an 'Object Property' if the primary key of a relation is used as a foreign key in a second relation and not participant to its primary key. The domain and range are the classes related to the relations. For this case the cardinality will be considered as <1:N> and in other cases <0:N>.

e.g: <owl:ObjectProperty rdf:ID="Deptn">
<rdfs:domain rdf:resource="Department"/>
<rdfs:range rdf:resource="Student"/>
</owl:ObjectProperty>

Rule5: Create an 'Object Property' if the primary key of a relation is used as a foreign key in a second relation and participating to its primary key. The domain and range are the classes related to the relations. In this case the cardinality will be <1:1>.

Rule6: Create an 'Object Property' if a relation has a primary key which is composed of two foreign keys. The domain and range are the classes related to the relations that references those foreign keys. In this case the cardinality will be <M:N>.

e.g:<owl:ObjectProperty rdf:ID="Prepares">
<rdfs:domain rdf:resource=" Student"/>
<rdfs:range rdf:resource="Article"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Preparedby">
<owl:inverseOf rdf:resource="Prepares"/>
</owl:ObjectProperty>

Rule7: Create a 'Functional Property' if a relation has a primary key composed of more than two foreign keys. Such a case can't be represented in OWL with ObjectProperty, hence we will represent it in the same way as in [5], so we represent the relation as a class and foreign keys as Functional Properties.

Rule8: For each class, create the same attributes as those that exist in the related relation, in the exception of foreign keys

e.g:<owl:DatatypeProperty rdf:ID="ProfessorID">
<rdfs:domain rdf:resource="Professor"/>
<rdfs:range rdf:resource="&xsd;Integer"/>
</owl:DatatypeProperty>

2) The Enrichment Process: The enrichment process allows enhancing the first result of the transformation process using another set of rules and an algorithm that helps adding more classes. We will use the domain ontology that we call DOnto found in [7], which a part of it is shown below. We use also WordNet to get synonyms of classes.

Figure 2. Part of the domain ontology.

- Rules for the case of candidate keys: A relational database can have many candidate keys, which they can be seen as relevant entities or hidden relations as shown in [10]. Hence it is possible to represent them as classes. It occurs that a candidate key do not represent a relevant object, that's why we compare it with classes of the domain ontology. We propose the following rules for the case of candidate keys.

Rule9: It consists of 4 steps:

1 - Using the Stemming technique to obtain the root word of the candidate key.
2 - Using WordNet to find all possible synonyms.
3 - Comparing the results with the classes of the domain ontology.
4 - Creating a class once there is a similarity with one of the classes of the domain ontology.

e.g: The candidate key 'Thesis' is created as a new class:
<owl:Class rdf:about="Thesis"/>

Rule10: Create a generalization relationship between the class related to the candidate key and the class related to the relation that included the candidate key (if the inheritance exists in the domain ontology).

Rule11: Create an 'Object Property' between the class related to the candidate key and the class related to the relation that includes the candidate key. In this case the cardinality will be always <1:1>; we represent this detail on the more general class as following:

e.g: <owl:Class rdf:about="Thesis">
<rdfs:subClassOf rdf:resource="Publication"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="hasThesis"/>
<owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasThesis">
<rdfs:domain rdf:resource="Publication"/>
<rdfs:range rdf:resource="Thesis"/>
</owl:ObjectProperty>

Rule12: Create for each class related to the candidate key an attribute with the same name of the class, and delete this attribute from the class related to the relation that included the candidate key.
e.g: The class Thesis will have 'Thesis' as an attribute Thesis {Thesis}.

- **Algorithm of Enrichment:** Our proposed algorithm is inspired in some points from the algorithm in [12]. The main steps that the algorithm treats are shown in the following. (LO: Local Ontology, DOnto: Domain Ontology).

Algorithm:

Prior Step. Organize classes of LO from top to down. Organize classes of DOnto from bottom to top.

Step1. Compare every class *i* in LO 'starting from top' with classes of DOnto 'starting from bottom':

- Syn= WordNet(Stemming(i)), compare Syn with classes of DOnto. // Retrieving synonyms before comparing.

Step2. if a concept *i* in LO is similar to a concept *j* in DOnto:

- Calculate the degree of similarities
***Add subclass(es) of* j *to* i**
if i do not have a super class:
***Add the super class of* j *to* i**

Step3. if a class* 'k' *is a subclass of* 'i' *and also a subclass of* 'x' *in DOnto (we call* x *any class added from* j *to* i*):

- Delete in LO the subsumption between (*i *&* k*) and create the new subsumption between (*x *&* k*)

The degree of similarities that is being calculated in the algorithm is based on analyzing properties of concepts. The degree of similarities can help the expert to decide whether a concept will be retained or deleted. Figure 3 shows the three cases of the algorithm.

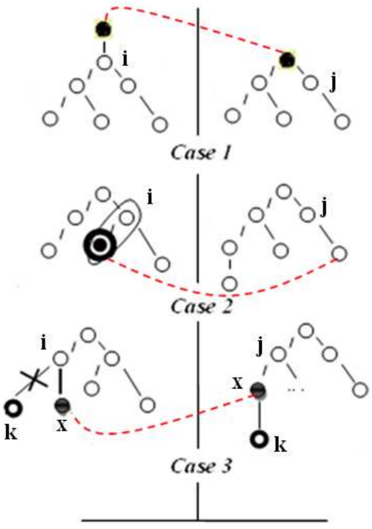
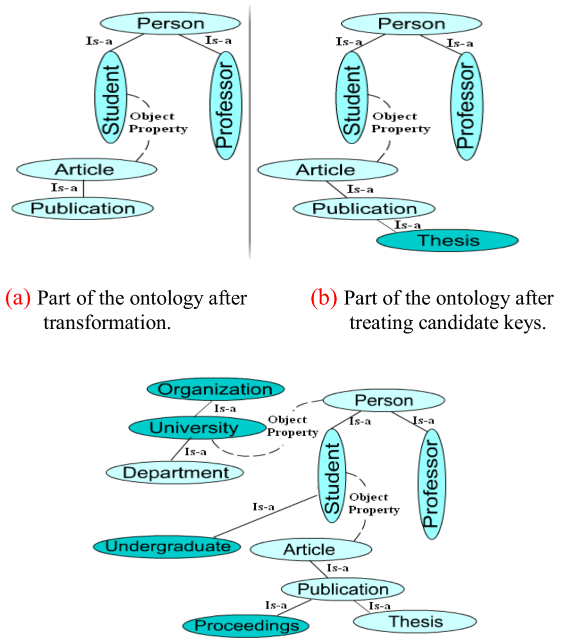


Figure 3. Part of the domain ontology.

Figure 4 shows a part of the resulted ontology after the transformation process, and how it is enriched after applying the enrichment process.



(a) Part of the ontology after transformation.

(b) Part of the ontology after treating candidate keys.

(c) Part of the ontology after applying the algorithm

Figure 4. Results of reverse engineering a relational database with enrichment.

The resulted ontology can be validated by an expert using any platform or tool as Protégé.

IV. COMPARISON STUDY

The following table summarizes a comparison between related work and our approach.

TABLE II. COMPARISON STUDY

Approach / Characteristics	Vipul Kashyap	Astrova et al	Benslimane et al	Our Approach
Create ontology from scratch	*yes*	*yes*	*yes*	*yes*
Automatic process	*no*	*yes*	*no*	*no*
Analysis key correlations	*yes*	*yes*	*yes*	*yes*
Analysis attributes correlations	*no*	*yes*	*no*	*no*
Analysis data correlations	*no*	*yes*	*no*	*yes*
Enrichment of the resulted ontology	*yes (user queries)*	*no*	*yes (HTML pages)*	*yes (Domain Ontology)*

This comparison shows the major differences that exist between processes of the related work. However the process of our approach is semi-automated because of the enrichment process that requires interactions of an expert.

V. CONCLUSION

In this paper we have proposed a reverse engineering process that aims to transform a relational database to an Ontology-Based Database. We attempt to add additional semantics behind applying the enrichment process. Our future works consist essentially to add more semantics, and to validate the full proposed process in a tool. Another important step is to choose one of the specific representations that exist for the OBDB to store our ontologies with data.

VI. REFERENCES

[1] I.Astrova, “Reverse Engineering of Relational Databases to Ontologies”, Proceedings of the 1st European Semantic Web Symposium (ESWS), Heraklion, Greece, LNCS, 3053, pp. 327-341, 2004.

[2] I.Astrova and A.Kalja, “Mapping of SQL Relational Schemata to OWL Ontologies”, Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18-20, 2006 (pp375-380).

[3] I.Astrova. and B.Stantic, “An HTML Forms Driven Approach to Reverse Engineering of Relational Databases to Ontologies”, Proceedings of the 23rd IASTED International Conference on Databases and Applications (DBA), Innsbruck, Austria, pp. 246-251, 2005.

[4] S.Benslimane, M.Malki, M.Rahmouni, and A.Rahmoun, “Towards Ontology Extraction from Data-IntensiveWeb Sites: An HTML Forms-Based Reverse Engineering Approach”, The International Arab Journal of Information Technology, Vol. 5, No. 1, January 2008.

[5] A.Buccella et al, “From Relational Databases to OWL Ontologies”, Proceeding of the 6th National Russian Research Conference. Pushchino, Russia. 2004.

[6] H. Dehainsala, G. Pierra, and L. Bellatreche, “Ontodb: An ontology-based database for data intensive applications”, In DASFAA, pages 497–508, April 2007.

[7] Domain Ontology, http://ontoware.org/swrc/swrc_v0.3.owl

[8] C. Fankam, “OntoDB2: Un système flexible et efficient de base de données à base ontologique pour le web sémantique et les données techniques”, Thèse de doctora, France.2009.

[9] S. Jean, Y. Aït Ameur, and G. Pierra, “Querying ontology based databases the ontoql proposal”, Proceedings of the 18th International Conference on Software Engineering & Knowledge Engineering (SEKE’2006), pages 166–171, 2006.

[10] P.Johannesson, “A Method for Transforming Relational Schemas into Conceptual Schemas”, Proceedings of the Tenth International Conference on Data Engineering IEEE Computer Society Washington, DC, USA 1994. ISBN:0-8186-5400-7.

[11] V.Kashyap, “Design and creation of ontologies for environmental information retrieval”, Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management. Alberta, Canada. 1999.

[12] C.R. Rene Robin, “A Novel Algorithm for Fully Automated Ontology Merging Using Hybrid Strategy”, European Journal of Scientific Research. ISSN 1450-216X Vol.47 No.1 (2010), pp.074-081.

[13] N.Stojanovic, L.Stojanovic, and R.Volz, “Migrating data-intensive Web Sites into the Semantic Web”, Proceedings of the 17th ACM symposium on applied computing. 2002.

[14] G. Thomas R, Formal ontology in conceptual analysis and knowledge representation. Chapter: “Towards principles for the design of ontologies used for knowledge sharing,” Kluwer Academic Publishers. 1993.