Automatic Constructing OWL Ontology from Relational Database Schema

Ines Fayech and Habib Ounalli

Department of Computer Sciences, Faculty of Sciences of Tunis, El Manar, Tunis, Tunisia

Abstract. In this paper we present a new tool, called DB_DOOWL, for creating domain ontology from relational database schema (RDBS). In contrast with existing transformation approaches, we propose a generic solution based on automatic instantiation of a specified meta-ontology. This later is an owl ontology which describes any database structure. A prototype of our proposed tool is implemented based on Jena in Java in order to demonstrate its feasibility.

1 Introduction

Most data on the web are still stored in relational databases which present limitation related to their semantic poverties. Indeed, traditional database management systems retrieve only data that exactly match query expressions. So, if the user has limited knowledge or no knowledge at all about the content of the database, he can't obtain results to his query. This gives rise to crucial needs for additional semantic support for a flexible and efficient database interrogation.

The concept of ontology, defined by Gruber [1] as "an explicit and formal specification of a conceptualization being the subject of a consensus", appears in a good place for the realization of these solutions.

In this paper we address the problem of constructing OWL ontology from relational database schema.

This problem was treated in the literature and different solutions were proposed in order to transform relational database (RDB) to ontology. We distinguish two main categories of database-to-ontology approaches: the approaches for creating ontology from a database and those for mapping a database to an existing ontology.

In this paper, we are interested with the first category. Indeed, we propose a new tool for automatically creating a new ontology from a relational database schema.

We choose OWL, the latest standard recommended to the semantic web, as the ontology language. Indeed, OWL enables us to represent not only the data itself, but also its interpretation. Furthermore, building the domain ontology in OWL, gives the ability to use DL reasoning on it.

The construction process is based on automatic instantiation of a meta-ontology describing the database structure.

The rest of this paper is structured as follows. First, section two is devoted to the presentation of database to ontology approaches. Second, section three gives the

Fayech I. and Ounalli H..
Automatic Constructing OWL Ontology from Relational Database Schema.
DOI: 10.5220/0004603100740081
In *Proceedings of the 2nd International Workshop on Web Intelligence* (WEBI-2013), pages 74-81
ISBN: 978-989-8565-63-1

specification of the meta-ontology used in our approach. Section four describes the proposed approach. Finally, a discussion of future work and concluding remarks will be presented.

2 Database to Ontology Approaches

A wide variety of works has been proposed in the literature in order to convert a relational database to XML, RDF or OWL ontology. We present, in this section, these different works and we show their limits.

The authors of [2] proposed a set of rules in order to construct RDF ontologies from Relational databases (RDBs). Their approach is semi-automatic because the user must choose the rules to be applied.

Perez and Conrad have proposed a tool called Relational.OWL to transform database schema and database data to OWL ontology [3]. The transformation is automatic but the proposed approach doesn't consider ontological relations between tables and attributes.

DataGenie[1] is a Protégé plug-in for importing a RDB to Protege ontology. The transformation process is simple and direct. Indeed, each table is converted into a class and each attribute is converted into a property. This plug-in has many limitations. Firstly, the generation process is semi-automatic. Indeed, the transformation is automatic but the user intervention is necessary to choose tables to be included in the ontology. Secondly, DataGenie doesn't allow importing only schema and doesn't support the OWL language.

To overcome the drawbacks of DataGenie, another Protege plug-in, called DataMaster, is proposed [4]. This plug-in allows importing the database structure and eventually the database data. With DataMaster, the user specifies tables to be imported and choose to import them as OWL classes or individuals of a class. The data extracted from the RDB is inserted into the generic ontology Relational.OWL. This later is OWL FULL whereas the ontology generated by DataMaster is OWL DL, allowing logical reasoning.

The authors of [5] have proposed a set of rules in order to discover mapping between a RDBS and an ontology contents (classes and properties). Their rules can be used to create an initial ontology which can be modified later. Their approach is semi-automatic and not implemented.

OWLFROMDB is an ontology generator allowing converting a RDB to an OWL ontology [6]. This tool considers only simple database structure such as database where primary and foreign keys are single columns.

In [7] a tool called RDB2On is proposed to construct OWL ontology from a RDB. The authors have proposed a set of rules to convert each table (respectively attribute) into an OWL class (respectively property).

OGSRO [8] is also a tool for transforming a RDB to OWL ontology such as RDB2On [7]. The transformation process is automatic but there is a difference between the generated ontology and the one created manually from the considered

[1] http://protege.cim3.net/cgi-bin/wiki.pl?DataGenie

RDB.

Another approach for transforming a DB to ontology is proposed in [9]. Contrary to those proposed in [6] and [8] where the DB structure is supposed to be simple, this approach considers other relation types between tables (one to one, one to many and many to many relations).

Let us discuss the drawbacks of the previous approaches.

The goal of converting a RDB into ontology is the semantic enrichment of the considered database. However, the available approaches for transition between RDBs and ontologies suffer at least from one of the following limitations:

- The approach is not implemented.
- The approach is semiautomatic.
- The approach is a plug-in of an ontology editor such as Protege and the user must be familiarized with this environment.
- The approach can't be used to transform only the database schema.
- The approach considers simple database structure.
- When the approach transforms both database data and structure, the size of the generated ontology will increase.
- When the approach considers a direct transformation, the generated ontology will be similar to the RDBS. However, there are many differences between ontologies and databases. Indeed, ontology must be represented, and eventually semantically enriched, in different ways according to the need.
- When the approach transforms database attributes into ontology properties, we can't distinguish between attributes and relationships of concepts. Moreover we can't add semantic relations between attributes.
- The approach doesn't support the OWL language. Contrary to RDF or XML, OWL allows uniform representation for relational database schema and data.
- The approach generates the ontology in OWL language but it needs the entire language (OWL FULL) and not OWL DL. This later is characterized by its expressiveness and reasoning power. But, OWL FULL prevents using decidable inference on the generated ontology.

To overcome the drawbacks of the previous approaches, we propose a novel approach to construct domain ontology from a RDBS. The transformation process is automatic and the generated ontology is in OWL DL language.

The proposed approach is described by a generic algorithm in order to be used with any RDB. So, we propose a tool, called DB_DOOWL, based on automatic instantiation of a specific meta-ontology. This later is the conceptual level of the ontology. It describes the structure of any RDBS.

3 The OWL Meta-ontology

In this section, we give the specification of the meta-ontology used to construct the domain ontology. This meta-ontology is developed in OWL DL which present some limitations related to the representation of the lexicalization of a concept. Indeed, in

OWL we use the rdfs: label annotation property to represent the term denoting a concept. So, we can't add a property to a term nor a relation between terms.

This problem was treated in the literature and different models were proposed in order to represent the terminological part of an ontology [10], [11], [12].

These models don't give the same importance to the notions of concept and term. For more details, readers are recommended to refer to [11] where the authors have presented a meta-model to model terms in OWL DL. They propose to distinguish terms from concepts: concepts are subclasses of the DomainThing class and terms are subclasses of the Term class. Their meta-model can't distinguish between properties concerning a term and those concerning its instances and doesn't consider how to associate a term to a semantic relation.

In order to manipulate a term as easily as a concept, we represent both in the form of an owl:Class. Consequently, the meta-model related to the domain ontology is based on two generic concepts: "c_domain_concept" and "concept_term" which represent respectively, the concepts of the considered domain and their lexicalizations.

Each "concept_term" denotes one domain concept ("c_domain_concept"). The later is divided into two sub-concepts in order to distinguish the concepts representing tables ("concept_table") and those describing attributes ("concept_attribute").

Since in OWL DL, a relation can be defined only between two individuals or between an individual and a literal, we choose to model terms as instances of the class "concept_term". Furthermore, concepts describing tables and those describing attributes will occur as instances respectively, of the "concept_table" and the "concept_attribute" classes. Figure 1 shows the specification of the proposed meta-ontology.

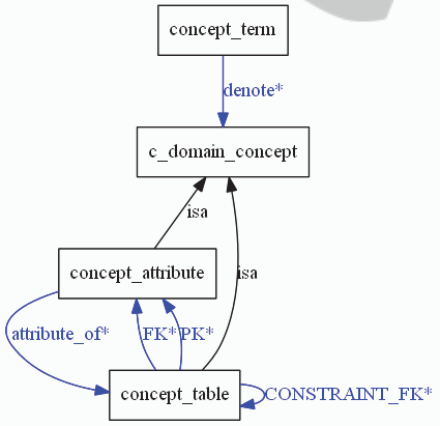

Fig. 1. The specification of the meta-ontology.

We consider the following object properties which link individuals to individuals.

- denote(C, T) means that the individual T of concept_term denotes the individual C of a c_domain_concept.
- attribute_of(CA, CT) means that CA, individual of concept_attribute, is an attribute of CT which is an individual of concept_table.

- PK(CT,CA) means that CA (individual of concept_attribute) is a primary key of CT (individual of concept_table).
- FK(CT,CA) means that CA (individual of concept_attribute) is a Foreign key of CT (individual of concept_table).
- CONSTRAINT_FK(CT1, CT2) means that there is a foreign key in CT1 (individual of concept_ table) that references CT2 (individual of concept_ table).

Moreover, we consider the property DB-Elt() applied to each individual of c_domain_concept in the meta-ontology. DB-Elt(C) means that C describes a database element. This property is useful when we enrich the ontology by adding new properties or concepts. It allows distinguishing concepts describing database contents (tables and attributes) from the others.

4 The proposed Approach

The goal of DB_DOOWL is to automatically construct a domain ontology (DO) from a RDBS. In this section, we describe the ontology building process and we give the proposed algorithm. The overall idea is to explore the meta-ontology describing the database structure. So, the domain ontology will be constructed by automatic instantiation of the proposed meta-ontology. The structure of our tool can be divided into two parts:

1) Extracting Database Contents: After connecting to RDB, we can extract data tables, data columns, primary keys, foreign keys and other metadata information.
2) Meta-ontology Instantiation: This step deals with the automatic instantiation of the meta-ontology in order to cover the RDBS. This later is a finite set of tables and a set of integrity constraints which are here limited to key constraints (primary and foreign keys). Each table has a name and a set of attributes.

Since our objective is to generate a domain ontology describing a RDBS, we distinguish different transformation types:

- Table Transformation: Each table T is transformed into an instance CTAB of concept_table class in DO. Moreover, we create a new instance T of concept_term class and we add the object property denote (CTAB, T) and the data property DB-Elt(CTAB).
- Attribute Transformation: Each attribute A is transformed into an instance CA of concept_attribute class in DO. Furthermore, we create a new instance A of concept_term class and we add the object property denote (CA, A) and the data property DB-Elt(CA).
- Converting a Relation between a Table and an Attribute: Each attribute A of a table T can be a simple attribute, a primary key or a foreign key. We have to convert these relations into the generated ontology DO.
- The relation “A is an attribute of a table T” is expressed in DO by the object property attribute_of (CA, CTAB).
- The relation “A is a primary key of T” is converted by using the object property PK (CTAB, CA).

- The relation “A is a foreign key of T” is converted by using the object property FK (CTAB, CA).
- Converting relations between tables: Since a RDB table can be related to other tables trough foreign key constraints, we have to express these relations in DO. For each foreign key of a table T1 referencing a table T2, we add the object property CONSTRAINT_FK (CT1, CT2), where CTi is an instance of the concept_table class describing the database table Ti (i={1, 2}).

Our proposed approach is described by an algorithm named Construction. This later is elaborated in a generic way in order to be reusable with any RDB.

The inputs of the algorithm are a relational database (RDB) and the OWL meta-ontology. The result is a domain ontology (DO) describing the database schema.

```
Algorithm Construction
Inputs: RDB, MO ; Output: DO
DO is initialized to MO
Create a database connection and
Extract data tables, data columns, primary keys, for-
eign keys and other metadata information.
For each table T in RDB {
Create a new instance CTAB of the concept _table
Create a new instance T of the concept_term
Add a new relation denote() between  CTAB and T
Add the data property DB-Elt(CTAB)
For each attribute A of the table T {
Create a new instance CA of the concept_attribute
Create a new instance A of the concept_term
Add a new relation denote() between  CA and A
Add the data property DB-Elt(CA)
Add a new relation attribute_of() between CA and CTAB
If A is a primary key then
Add a new relation PK between CA and CTAB
If A is a foreign key then
Add a new relation FK between CA and CTAB } }
Add foreign key constraints between tables
END
```

An example that illustrates the proposed approach is given in Figure 2 showing the generated ontology (by using OntoViz, the Protege plug-in) related to the following fragment of a RDBS describing the faculty of sciences:

```
contractuel (matricule, nom, numdep)

PKey(matricule), FKey(numdep) REFERENCES  dept

dept(numdep, nom, code_univ)

PKey(numdep)
```

We denote a primary key by PKey(A) where A is an attribute. Furthermore," FKey(A) references TAB " refers to a foreign key constraint: The attribute A references the primary key of the relation TAB.

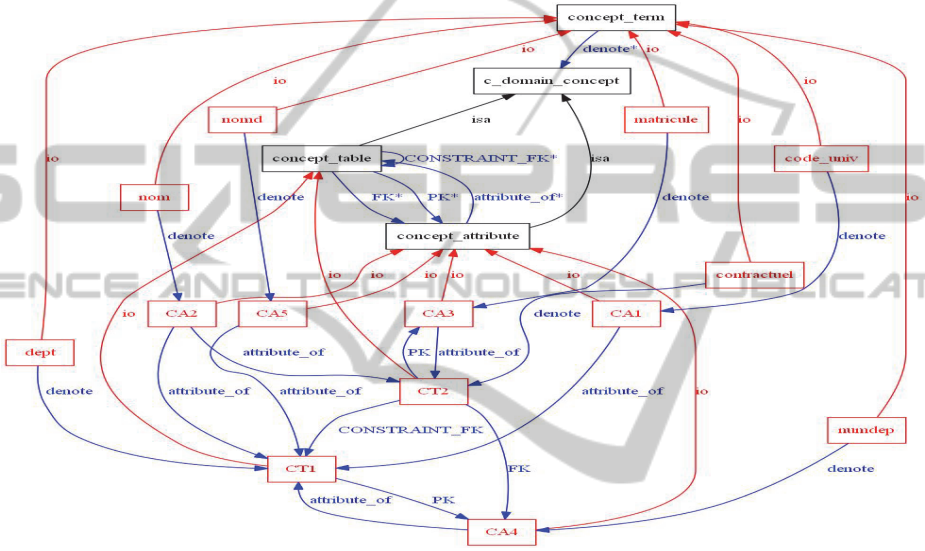

Fig. 2. The generated ontology.

5 Conclusions

The research is nowadays focused on how to enrich traditional databases with additional semantic support in order to facilitate their interrogation. Indeed, the majority of information found in the web is stored in databases that present some limitations related to their semantic poverty. To meet these requirements, we have developed DB-DOOWL, a tool that allows the user to automatically import a RDBS into OWL ontology. It is implemented based on Jena in Java development platform.

The transformation is based on the instantiation of a proposed meta-ontology. This later is a generic ontology used to describe the structure of any RDBS.

In the generated ontology, terms describing table/attribute names in the RDBS are often abridged (e.g. dept). So, it is difficult or even impossible to deduce the meaning of data from those names. Since ontology should capture consensual knowledge accepted by the communities, we intend to enrich the generated ontology by adding

different semantic relation types between concepts.

References

1. Gruber T. R.: A translation approach to portable ontology specification. knowl. Acquis, vol. 5 num2, Academic Press Ltd. p199-220 (1993)
2. Stojanovic, L., Stojanovic, N., Volz, R.: Migrating Data-intensive Web Sites into the Semantic Web. Proc. 17th ACM Symposium on Applied Computing, Madrid (2002)
3. Christian Perez de Laborda and Stefan Conrad. : Relational.OWL - A Data and Schema Representation Format Based on OWL. In Sven Hartmann and Markus Stumptner, editors, Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005) , volume 43 of CRPIT, Newcastle, Australia. Pages 89–96 (2005)
4. Nyulas C. O'connor, M. Tu, S.: DataMaster – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé (2007). (http://protege.stanford.edu/ conference/2007/presentations/10.01_Nyulas.pdf).
5. Hu C., Li H., Zhang X., Zhao C.: Research and Implementation of Domain-Specific Ontology Building from Relational Database. 3rd IEEE Int. Conf. ChinaGrid Annu. pp. 289-293 (2008)
6. He-ping C., Lu H., Bin C.: Research and Implementation of Ontology Automatic Construction Based on Relational Database. Int. Conf. Comput. Sci. Softw. Eng., pp. 1078-1081 (2008)
7. Zhou S., Meng G.: Tool for Translating Relational Databases Schema into Ontology for Semantic Web. 2nd Int. Workshop Educ. Technol. Comput. Sci., pp. 101-108 (2010)
8. Zhang L., Li J.: Automatic Generation of Ontology Based on Database. J. Comput. Inf. Syst., pp. 1148-1154 (2011)
9. Saeed M. Sedighi, Reza Javidan: A novel method for improving the efficiency of automatic construction of ontology from a relational database, International Journal of Physical Sciences Vol. 7(13), pp. 2085 - 2092, (2012)
10. Bontcheva K., Tablan V., Maynard D. Cunningham H.: Evolving GATE to Meet New Challenges in Language Engineering, Natural Language Engineering, 10(3/4), 349–373 (2004)
11. Axel Reymonet, Jérôme Thomas, Nathalie Aussenac-Gilles : Modélisation de Ressources TerminoOntologiques en OWL. Actes d'IC 2007. 169-181 (2007)
12. Anita C., Liang, Boris Lauser, Margherita Sini.: From AGROVOC to the Agricultural Ontology Service / Concept Server - An OWL Model for Creating Ontologies in the Agricultural Domain", OWLED (2006)