DB2OWL: A tool for automatic database-to-ontology mapping

Conference Paper · January 2007

Source: DBLP

3 authors:

Nadine Cullot
University of Burgundy

Raji Ghawi
Technical University of Munich

Kokou Yetongnon
University of Burgundy

DB2OWL: A Tool for Automatic Database-to-Ontology Mapping

Nadine Cullot, Raji Ghawi, and Kokou Yétongnon
Laboratoire LE2I, Université de Bourgogne, Dijon, FRANCE
{Nadine.Cullot, Raji.Ghawi, Kokou.Yetongnon}@u-bourgogne.fr

Abstract. We present a tool, called DB2OWL, to automatically generate ontologies from database schemas. The mapping process starts by detecting particular cases for conceptual elements in the database and accordingly converts database components to the corresponding ontology components. We have implemented a prototype of DB2OWL tool to create OWL ontology from relational database.

Keywords: Interoperability, Ontology, Data Integration and Mapping.

1 Introduction

In order to achieve an efficient interoperability between heterogeneous information systems, many solutions have been proposed. Particularly, ontologies play an important role in resolving semantic heterogeneity by providing a shared comprehension of a given domain of interest. An ontology formally defines different concepts of a domain and relationships between these concepts. In interoperability approaches a local ontology is used for each information. The advantage of wrapping each information source to a local ontology is to allow the development of source ontology independently of other sources or ontologies. Hence, the integration task can be simplified and the addition and removal of sources can be easily supported.

Information sources may contain different types of data structures: data may be structured as databases, semi-structured as XML documents, and/or non-structured as web pages or other type of documents. However, all of these sources must be mapped to a local ontology which will express the semantic of information sources. In this paper, we focus only on the mappings between databases and the local ontology.

We have developed a tool called DB2OWL 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. The created ontology is expressed in OWL-DL language[1] which is based on Description Logics. The mapping process starts by detecting some particular cases for tables in the database schema. According to these cases, each database component (table, column, constraint) is then converted to a corresponding ontology component (class, property, relation). The set of correspondences between database components and ontology components is conserved as the mapping result to be used later.

[1] http://www.w3.org/TR/owl-features/.

2 Database to Ontology Mappings: DB2OWL Tool

2.1 Different table cases

The mapping process used in our approach depends on particular database table cases that are taken in account during the ontology creation. These cases are illustrated using examples from database schema shown in figure 1.

Case 1. When a table T is used only to relate two other tables T1, T2 in a many-to-many relationship, it can be divided into two disjoint subsets of columns A_1, A_2, each participating in a referential constraint with T_1 and T_2 respectively. Therefore all T columns are foreign keys and they are primaries as well because their combination uniquely defines the rows of T. For example, the table PRESENCE is in case 1 because it relates STUDENT and SESSION tables in a many-to-many relationship.

Case 2. This case occurs when a table T is related to another table T_1 by a referential integrity constraint whose local attributes are also primary keys. In this case all the primary keys of T are foreign keys because they participate in a referential integrity constraint. For example, the table STUDENT is in case 2, because it is related to PERSON table by a foreign key which is primary key at the same time.

Case 3. This case is the default case, it occurs when none of previous cases occur.

When these different cases are detected in the database, the mapping process can use them to appropriately map database components to suitable ontology components as follows.

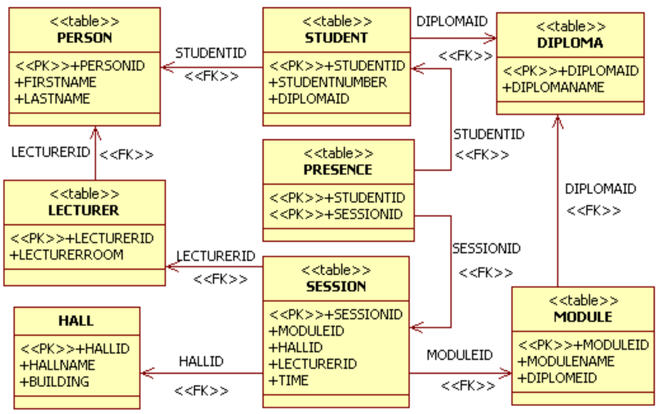

Fig. 1. The schema of schooling database

2.2 Mapping process

The mapping process is done progressively as follows. It starts by mapping the tables to concepts and then mapping the columns to properties. Thus, the table cases mentioned above are used twice: one time for table-to-class mapping and the other time for column-to-property mapping. The mapping process consists therefore of the following steps:

1. The database tables that are in case 3 are mapped to OWL classes.
2. The tables in case 2 are mapped to subclasses of those classes corresponding to their related tables, i.e. if T is in case 2 and related to T_1 by a foreign key which is primary key at the same time, then T is mapped to a subclass of the class corresponding to T_1 .
3. Each table in case 1 is not mapped to class, but the many-to-many relationship that it represents is expressed by object properties. Two object properties are added, one for each class whose corresponding table was related to the current table. In other words, when a table T is in case 1 and relates between T_1 and T_2, and if c_1, c_2 are the two classes corresponding to T_1, T_2 respectively, so we assign to c_1 an object property op_1 whose range is c_2 , and assign to c_2 an object property op_2 whose range is c_1. Each of these two properties op_1, op_2 are inverse to the other.
4. For tables that are in case 3, we map their referential constraints to object properties whose ranges are classes corresponding to their related tables; i.e. if a table T is in case 3 and has a referential constraint with T_1, and if c, c_1 are the classes corresponding to T, T_1 respectively, then we assign to c an object property op whose range is c_1, and we assign to c_1 an object property op' whose range is c. To preserve the original direction of the referential constraint from T to T_1, we set the object property op as functional. So it will have at most one value for the same instance. This characteristic is obvious because it comes from the uniqueness of key.
5. For tables that are in case 2 and have other referential constraints than the one used to create the subclass, we map them to object properties as in the previous step.
6. Finally, for all tables we map their columns that are not foreign keys to datatype properties. The range of a datatype property is the XML schema data type[2] equivalent to the data type of its original column.

2.3 Mapping Generation

During the mapping process, a R2O [2] document is automatically generated to record the relationships between generated ontology components and the original database components. It includes (1) a full description of the database schema, (2) a set of concept map definitions consisting of the name of concepts with their identifying column(s), and (3) a set of relation and attribute map definitions. This document can be used to translate ontological queries into SQL queries and retrieve corresponding instances.

[2] http://www.w3.org/TR/xmlschem-2/.

3 Conclusion and Future Work

Currently there are many approaches and tools to deal with database to ontology mapping [1][2][3][4][5][6][7][8]. We have presented DB2OWL our tool to map relational databases to OWL ontologies. This tool consider particular table cases and take them into account while the mapping process. We have implemented a prototype of this tool in Java and using Jena[3]. This prototype deals currently with Oracle and MySQL databases because they provide specific views about the database metadata. Extension of the presented tool are underway to deal with other DBMS that provide such views. In addition, DB2OWL will be developed further to map several databases to one ontology, and to map databases from other models such as object-relational model.

References

1. An Y., Borgida A., and Mylopoulos J. (2005): Inferring Complex Semantic Mappings Between Relational Tables and Ontologies from Simple Correspondences. In CoopIS, DOA, and ODBASE, OTM Confederated International Conferences, Cyprus, Part II, volume 3761 of LNCS, pages 1152 - 1169. Springer, 2005.
2. Barrasa J., Corcho O., Gómez-Pérez A. (2004): R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language. A. Second Workshop on Semantic Web and Databases (SWDB2004), Toronto, Canada. 2004.
3. Bizer C. (2003): D2R MAP – A Database to RDF Mapping Language, The twelfth international World Wide Web Conference, WWW2003, Budapest, Hungary.
4. de Laborda C. P. and Conrad S. (2005): 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. ACS.
5. Konstantinou N., Spanos D., Chalas M., Solidakis E., and Mitrou N. (2006): VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Contents. International Workshop on Web Information Systems Modeling (WISM2006) Luxembourg.
6. Petrini J. and Risch T. (2004): Processing Queries over RDF views of Wrapped Relational Databases. In 1st International Workshop on Wrapper Techniques for Legacy Systems, WRAP 2004, Delft, Holland, 2004.
7. Volz R., Stojanovic L., Stojanovic N. (2002): Migrating data-intensive Web Sites into the Semantic Web. ACM Symposium on Applied Computing (SAC 2002). Madrid, Spain, March 2002.
8. Volz R., Handschuch S., Staab S., Studer R. (2004): OntoLiFT Demonstrator.

[3] http://jena.sourceforge.net/.