Learning Highly Structured Semantic Repositories from Relational Databases
The RDBToOnto Tool

Farid Cerbah

Dassault Aviation
DPR/ESA
78, quai Marcel Dassault 92552 Saint-Cloud – France
farid.cerbah@dassault-aviation.fr

Abstract. Relational databases are valuable sources for ontology learning. Methods and tools have been proposed to generate ontologies from such structured input. However, a major persisting limitation is the derivation of ontologies with flat structure that simply mirror the schema of the source databases. In this paper, we show how the RDBToOnto tool can be used to derive accurate ontologies by taking advantage of both the database schema and the data, and more specifically through identification of taxonomies hidden in the data. This extensible tool supports an iterative approach that allows progressive refinement of the learning process through user-defined constraints.

1 Motivation

Ontology learning from relational databases is not a new research issue. Several methods and tools have been developed to deal with such structured input (e.g. [1–3]). However, a major persisting limitation of the existing methods is the derivation of ontologies with flat structure that simply mirror the schema of the source databases. For example, the DataMaster Protégé plugin [3] is a convenient tool that allows to import schema definition and data into Protégé, but the target populated models are simply based on ontologies of the relational model (such as Relational.OWL [4]). Such tools can significantly ease the transitioning task by automatically expressing legacy data into ontology representation formats. However, the results might not fully meet the expectations of users that are primarily attracted by the rich expressive power of semantic web formalisms and that could hardly be satisfied with target knowledge repositories that look like their source relational databases. A natural expectation is to get at the end of the learning process ontologies that better capture the underlying conceptual structure of the stored data.

Ontologies with flat structure is the typical result of learning techniques that exclusively exploit information from the schema without considering the data. One of the main motivations behind the RDBToOnto tool is to implement a process that allows to learn populated ontologies with rich taxonomies by exploiting

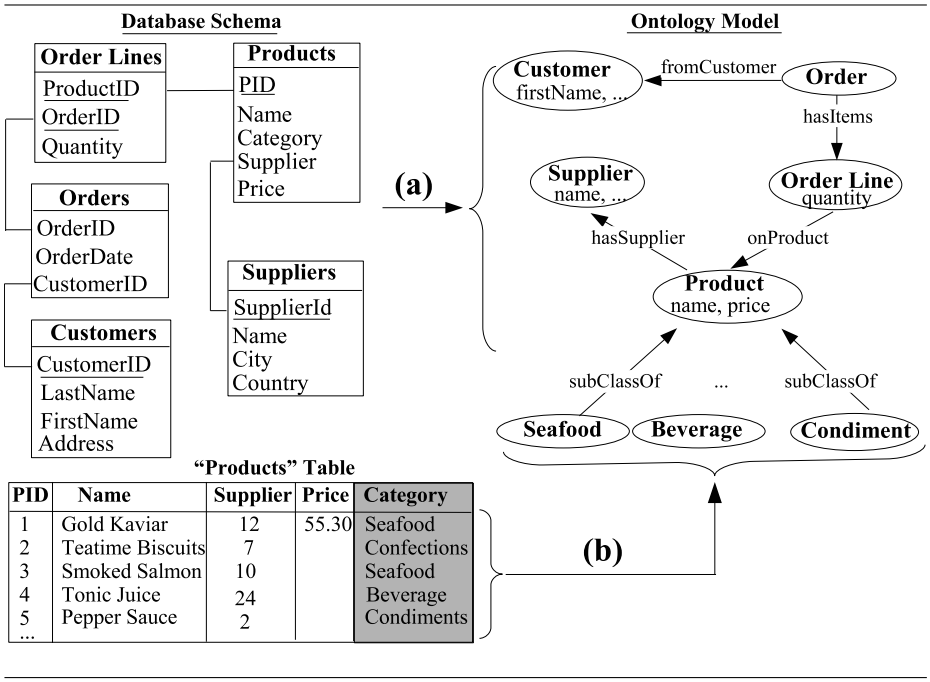

Fig. 1. Ontology model built by exploiting both the schema and the data

both the schema and the data in the identification of the ontology structure. Additionally, a second major objective is to provide support for an iterative approach that allows progressive refinement of the learning process through user-defined constraints.

To give an illustration of how both schema definition and data can be exploited as input, let us start by depicting the typical transitioning process on an academic example. Figure 1 shows the input and the potential output of such of a process when applied on a sample database.

The derivations applied to get the target ontology can be divided in two parts. The first part, named **(a)** in the figure, includes derivations that are motivated by the identification of patterns from the database schema. In this example, each relation (or table) definition from the schema is the source of a class in the ontology. Such simple mappings from relations to classes are often relevant (though some exceptions need to be handled). Datatype properties are derived from some of the relation attributes and binary key-based associations between tables are the most reliable source for linking classes through object properties.

The derivations applied to obtain this upper part of the ontology are well covered by current methods and, if applied on this database example, most of the methods would provide the result of the **(a)** derivations as final output. However, by looking closer at the data, we can notice that the process can go further. In the Products table, additional structuring patterns can be exploited to make

the ontology more accurate. More particularly, the **(b)** part of the derivations shows how the Product class can be refined with subclasses derived from the values of Category column in the source Products table. In the same vein, the Supplier class can be extended with a two-level hierarchy by interpreting the values in both Country and City columns of the corresponding table (resulting in subclasses Sweden Supplier $\longrightarrow$ Stockholm Supplier, Göteborg Supplier, etc).
These are typical examples of subsumption relations that can be discovered by mining the database content. One of the key issues addressed in this work is the identification of relation attributes that may serve as good *categorisation sources*. In our ontology learning approach, it is assumed that these attributes can be revealed by combining identification of lexical cues in attribute names and entropy-based estimation of data redundancy in attribute extensions.

We give in other publications a formal description of our comprehensive learning method which takes advantage of both the database schema and the data. It is the main method implemented in the RDBToOnto tool described below.

2 The RDBToOnto Tool

RDBToOnto[1] is a highly configurable tool that ease the design and implementation of methods for ontology learning from relational databases. It is also a user oriented tool that supports the complete transitioning process from access to the input databases to generation of populated ontologies. The settings of the learning parameters and control of the process are performed through a full-fledged dedicated interface (figure 2).
A basic principle in the design of RDBToOnto is to allow the derivation of an exploitable ontology in a fully automated way. By using the tool with its default configuration, a user can get a populated ontology by simply providing as input the uri of the input database. However, it should also allow the user to iteratively refine the result. This is performed by adding local constraints. Several types of constraints are pre-defined while allowing (experienced) users to define new ones. As briefly discussed in previous section, the main learning method implemented in RDBToOnto includes data-driven mechanisms to automatically mine categorisation patterns in the database content. To further refine the ontology structure, the user can add local constraints to specify categorisation patterns that have been missed by the automated mechanisms (i.e. by selecting relevant categorisation attributes through the interface). Constraints on instance naming are also highly useful when building fine-tuned ontologies. Instead of letting the system assign arbitrary names to instances, it is possible to specify through local constraints attached to source relations how names should be derived from attribute values (e.g., to an Employees source relation, it is possible to attach a constraint specifying that instance names should be formed by combining values of FirstName and LastName attributes).
Another key improvement over existing tools is the inclusion of a database normalisation step in the supported process. It is often assumed that the ontology

[1] http://www.tao-project.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html

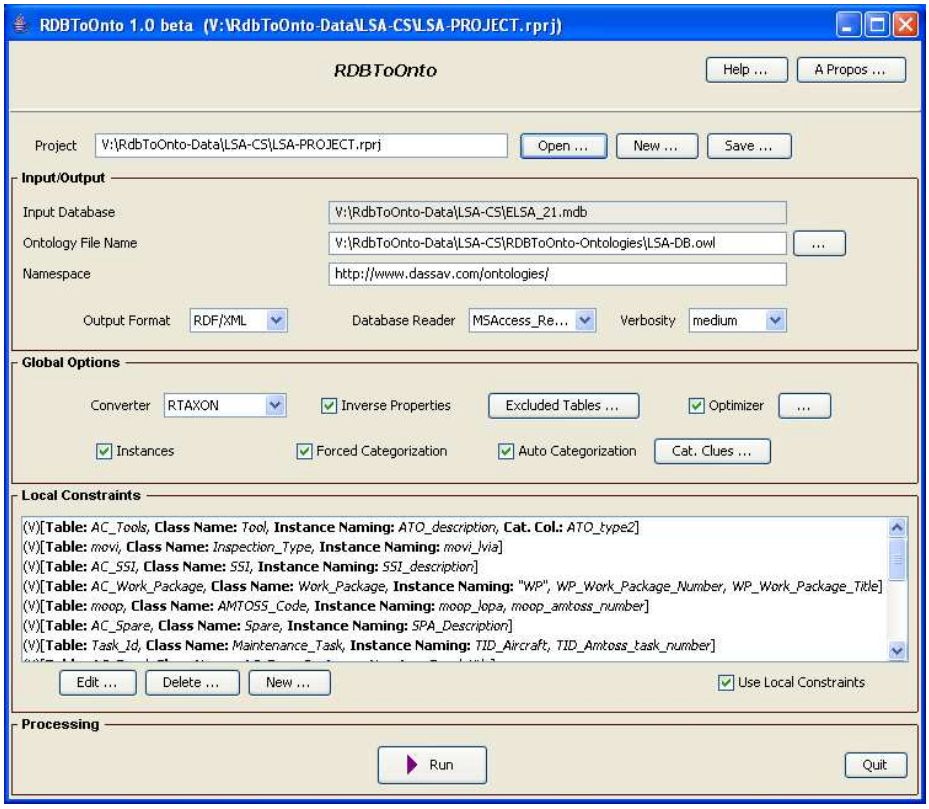

Fig. 2. The user interface of RDBToOnto. This extensible tool is designed to allow the integration of several learning methods (each method is implemented by a "converter"). A method can have its specific types of global options and local constraints.

learning process starts with well-designed databases. While theoretically acceptable, this assumption has some drawbacks in practice as many databases that are relevant for ontology learning suffer from redundancy problems. Without a proper integrated support for database normalisation, users might be tempted to directly take the databases as input even if badly designed. In RDBToOnto, main effect of the normalisation step is to eliminate data duplication in the source tables (through the interpretation of inclusion dependencies defined by the user). The model transformation performed to eliminate redundancy ultimately results in the introduction of inter-class relations (i.e. object properties).

A set of reusable components can be directly exploited to implement new methods. More particularly, database readers for some of the most common database formats are included in the tool and new ones can be integrated. Additionally, the database normalisation task is supported by a reusable component. The user interface can be extended to handle the specific constraints of new methods.

3 Evaluation

RDBToOnto has been evaluated on a set of 50 databases from different domains. One of the representative transitioning experiments performed with this tool has been conducted in the context of the TAO project[2]. In this significant case study, the input is a complex database in the domain of aircraft maintenance that includes technical descriptions of aircraft parts and all logistic resources involved in maintenance operations (spares, tools, manpower, ...). For this project, a thorough specification of the learning process has been performed resulting in 70 constraints (mostly, inclusion dependencies to optimise the model and naming constraints for classes and instances). The process produced an ontology of 600K triples corresponding to 70 classes populated with 50K instances (50 object properties with 40K instances and 120 datatype properties with 350K assigned values). The ten class hierarchies that have been discovered appeared to be relevant and the variety of some prominent concepts (such as tools and spares) are captured in these hierarchies.

4 Conclusion

We described in this paper the functionalities of RDBToOnto, a tool that implements a novel approach to ontology learning from relational databases. The prominent features of the supported approach are:

- A method that takes advantage of both database schema and content, and that can identify reliable categorisation patterns hidden in the data.
- A fully automated learning process that can be influenced through user-defined local constraints of various types.
- A database normalisation step incorporated in the implemented process that can reduce the redundancy of the source databases before ontology learning.
- A framework that eases the implementation of new methods.

References

1. Stojanovic, L., Stojanovic, N., Volz, R.: Migrating data-intensive web sites into the semantic web. In: Proc. of ACM Symp. on Applied Computing, Madrid (2002)
2. Astrova, I.: Reverse engineering of relational databases to ontologies. In: First European Semantic Web Symp. (ESWS 2004), Greece, Stringer-Verlag (2004)
3. Nyulas, C., O'Connor, M., Tu, S.: Datamaster - a plug-in for importing schemas and data from relational databases into protégé. In: 10th Intl. Protégé Conference, Budapest (2007)
4. de Laborda, C.P., Conrad, S.: Relational.OWL: a data and schema representation format based on OWL. In: APCCM '05: Proc. of the 2nd Asia-Pacific conference on Conceptual modelling, Darlinghurst, Australian Computer Society, Inc. (2005)

[2] http://www.tao-project.eu/