Semantic mapping to synchronize data and knowledge bases at the instance level

Olivier Curé
Université de Marne-la-Vallée, Laboratoire ISIS
5, boulevard Descartes
Marne-la-Vallée 77454 France
ocure@univ-mlv.fr

Raphaël Squelbut
Université de Marne-la-Vallée, Laboratoire ISIS
5, boulevard Descartes
Marne-la-Vallée 77454 France
raphael.squelbut@univ-mlv.fr

ABSTRACT

In recent years there has been a growing interest in making existing database (DB) content available to emerging Semantic Web applications. In this paper, a mapping approach between relational DBs and Description Logics (DL) based ontologies has been adopted. Based on the consideration that the DL Abox is a view of the relational DB, we are able to retrieve data en masse from the DBs for integration in the knowledge base (KB). This approach is motivated by the possibility to integrate multiple DBs which may not all be accessible at application run-time. The main contribution of this paper is to tackle the maintenance at the instance level. Thus data inserted, modified or updated in a DB model will be accordingly integrated in the KB model, without a complete processing of the mapping file. A key aspect of this maintenance is the synchronization's efficiency, meaning that integration in the KB of some DB operations may be deferred due to the respect of integrity constraints. This paper focuses on these synchronization issues and their implementation. All synchronization tasks are completely automatized, i.e. human intervention is not required.

1. INTRODUCTION

Several researches are interested in providing ontological engineering tasks such as creation of ontologies and instanciation of the KBs. Most of these solutions are based on the design of expressive and computationally efficient mapping technologies between structured and semi-structured DBs and ontologies. These research solutions usually involve a reverse engineering processing, a task corresponding to the analysis of a "legacy" system in order to identify the system's components and their inter-relationships.

Our system DBOM (DataBase Ontology Mapping) proposes a mapping-based solution for the creation and population of a KB from multiple DBs. But our main contribution is to exploit the mapping file to maintain as synchronized as possible DBs to KB. This synchronization is only considered at the instance level and not at the schema level, meaning that modifications of the DB and ontology schema can not be synchronized between the two models. The maintenance at the instance level are supported by the mapping file and has two flavors (i) a modification of a set of tuples on the DB's side may be reflected on the KB's side, (ii) a modification of a set of concepts and properties instances on the KB's side may be reflected on the DB's side. This paper focuses on the first aspect of the maintenance.

2. RELATED WORK

The primary goal of this paper is to present the maintenance solution of the DBOM system. To our knowledge, there are no researches investigating such an approach. Among related solutions, we distinguish the following categories : (i) creation of a KB (Tbox and Abox) from an existing DB [6, 2, 3], (ii) creation of a DB schema from an existing KB [7], (iii) creation of a mapping between an existing ontology and DB schemata [1, 5], in order to enable information integration. In this approach, an ontology schema corresponding to the DB schema has been manually designed and a mapping is required to enable interoperability. In a nutshell, DBOM belongs to the semi-automatic, like [5, 6], category with loose coupling (data is retrieved en masse from the DBs), like [3, 2] and the target is formalized in OWL DL, corresponding to $\mathcal{SHOIN}(D)$, like [1]. The semi-automatic characteristic is motivated by the fact that DBOM aims, but is not limited, to develop light ontologies supporting inferences in domain-specific applications. By light ontologies we mean KBs that only contain data involved in reasoning activities. These characteristics make DBOM similar to D2R MAP but with the ability to integrate multiple data sources [4]. However another important difference between these two solutions is in the terminological axiomatization possibilities of DBOM which enable the creation of ontologies as expressive as OWL DL. Finally, DBOM proposes additional services one of which, maintenance solutions, is emphasized in the rest of this paper.

3. DBOM FRAMEWORK

3.1 Overview

DBOM is based on the use of a declarative mapping, serialized in XML, which is a set of explicit correspondences between components of the DB and KB models. The processing of the mapping file enables to create a TBox and

instantiate the ABox, considering it as a view of the relational DB. Our contribution to this issue lies in the possibility to richly axiomatize the terminology; thus permitting the creation of expressive ontologies. But the most interesting contribution is the solution proposed to maintain the synchronization between the DB tuples and the Abox. This synchronization is based on automatically created, at mapping processing time, SQL triggers which are fired whenever a "write" query (meaning insert, delete or update SQL queries) is processed on a DB relation used in the mapping file. These triggers are calling Java methods developped within our framework which are responsible for the update of the Abox.

We have developed a Protégé plug-in version of DBOM. This plug-in aims to simplify the creation of mapping files using a graphical user interface and all the features provided by Protégé, principally from the OWL plug-in. We now refer to "members" of the mapping as the set of concepts and object properties. We make the distinction between concrete and abstract members. The comprehension of concrete and abstract members is relatively straightforward as it is equivalent to the assumption made in Object-Oriented Programming. Thus instances (individuals) can be created for a concrete concept and a concrete object property can relate two existing individuals. Abstract members can not be instantiated and they aim to design a hierarchy of members where final (leafs in a tree representation) members should be concrete. The DBOM Protégé plug-in is efficiently integrated in the Protégé framework to enable the design of KBs. In the nutshell, the DBOM plug-in aims to compose concrete members and their SQL queries, via interactions with the mapped DBs presented as a tree. All over ontological tasks can be performed via the OWL tabs, i.e. axiomatizations.

In the following example, we highlight a possible mapping of a relational schema to a TBox.
Relational schema :
person (idPerson, name, idGender)
gender (idGender, name)
The mapping file defines the following TBox (*Person* is an abstract concept).
$Man \sqsubseteq Person$
$Woman \sqsubseteq Person \sqcap \neg Man$
The queries of the concrete concepts (*Man* and *Woman*) are presented in the form of conjunctive queries.
$Man \equiv \{ (X,Y) \mid person(X,Y,Z) \wedge gender(Z,U) \wedge U='male'\}$
$Woman \equiv \{ (X,Y) \mid person(X,Y,Z) \wedge gender(Z,U) \wedge U='female'\}$

This plug-in solution enables to load an existing OWL KB and add new members via the integration with DBs.

3.2 Synchronization issues

A central aspect of the instantiation is the "membership determination solution" which aims to find the appropriate concrete member to create, modify or delete given the firing of a DB trigger. This algorithm enables to detect that either one of the instances of the *Man* or *Woman* concrete concepts of example 2 can be updated from the firing of a trigger on the person relation.
The main issue of the synchronization lies in the respect of integrity constraints (ICs) defined in DB sources. In this paper, we are concerned with the most relevant ICs encountered in relational DBs : key, foreign key and functional dependencies. These ICs force the system to postpone some of the instantiations in the ABox. Thus, a complete synchronization is ensured from the processing of required SQL queries in the DBs. The management of delays has to be taken care of by DBOM's synchronization policy. Four different stages can be encountered : (i) no action, meaning that the "write" queries has no effect on KB instances, (ii) simple action, meaning that a unique object is treated in the KB, (iii) multiple action, meaning that several objects can be treated due to the feedback effect of synchronisation (one action can cause many postponed actions), (iv) postponed action, meaning that no action can be processed but the system is left in a state where future triggers may fire multiple actions. DBOM also supports SQL referential actions for update and delete rules, i.e. cascade, set null, set default and no action. We now consider referential actions as specialized triggers for automatically maintaining referential integrity in DBs.

4. CONCLUSION AND FUTURE WORKS

The DBOM system is implemented using the Java language and Hewlett-Packard's Semantic Web Jena framework and Protégé. Tests have been conducted with PostgreSQL and our DBOM plug-in. The addition of terminological axioms in the DL Tbox enables to highlight inconsistencies on the KB that could not be detected on the DB instance. We are currently working on the detection of the inconsistencies as well as their explanations to end-users. This issue also broadens our approach to DB repair and the view-update problem.

5. REFERENCES

[1] An Y., Borgida, A., Mylopoulos, J. : *Inferring Complex Semantic Mappings Between Relational Tables and Ontologies from Simple Correspondences*. Proceedings of OTM Conferences (2005) 1152-1169.

[2] Bizer, C. : *D2R MAP - A database to RDF Mapping Language*. The Twelfth International World Wide Web Conference (2003).

[3] Borgida, A. : Loading Data into Description Reasoners. Proceedings of ACM SIGMOD International Conference on Management of Data (1993) 315–333

[4] Curé, O., Squelbut, R. : *Data integration targeting a drug related knowledge base* To appear in the proceedings of EDBT 2006 Information Integration and Health Applications Workshop.

[5] Handschuh, S., Staab, S., Volz, R. : On Deep Annotation. 12th International World Wide Web Conference (2003). 431–438.

[6] Stojanovic, L., Stojanovic, N., Volz, R. : *Migrating data-intensive web sites into the semantic web*. Proceedings of the ACM Symposium on Applied Computing SAC (2002) 1100–1107

[7] Moreno Vergara, N., Navas Delgado, I., Francisco Aldana Montes, J. : *Putting the Semantic Web to Work with Database Technology*. IEEE Data Engineering Bulletin Volume 26, issue 4 (2003) 49–54