Data & Knowledge Engineering 98 (2015) 8–29

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Ontology-based mappings☆

Giansalvatore Mecca [a], Guillem Rull [b], Donatello Santoro [a], Ernest Teniente [c]

[a] *Università della Basilicata, Potenza, Italy*
[b] *Universitat de Barcelona, Barcelona, Spain*
[c] *Universitat Politècnica de Catalunya, Barcelona, Spain*

A R T I C L E I N F O

Available online 13 July 2015

Keywords:
Mapping
Ontology
View
Tuple generating dependency
Equality generating dependency
Disjunctive embedded dependency

A B S T R A C T

Data translation consists of the task of moving data from a source database to a target database. This task is usually performed by developing mappings, i.e. executable transformations from the source to the target schema. However, a richer description of the target database semantics may be available in the form of an ontology. This is typically defined as a set of views over the base tables that provides a unified conceptual view of the underlying data. We investigate how the mapping process changes when such a rich conceptualization of the target database is available. We develop a translation algorithm that automatically rewrites a mapping from the source schema to the target ontology into an equivalent mapping from the source to the target databases. Then, we show how to handle this problem when an ontology is available also for the source. Differently from previous approaches, the language we use in view definitions has the full power of non-recursive Datalog with negation. In the paper, we study the implications of adopting such an expressive language. Experiments are conducted to illustrate the trade-off between expressibility of the view language and efficiency of the chase engine used to perform the data exchange.

© 2015 Published by Elsevier B.V.

1. Introduction

Integrating data coming from disparate sources is a crucial task in many applications. An essential requirement of any data integration task is that of manipulating *mappings* between sources. Mappings are executable transformations that define how an instance of a source repository can be translated into an instance of a target repository. Traditionally, mappings are developed to exchange data between two relational database schemas [1]. A rich body of research has been devoted to the study of this subject. This includes the development of algorithms to simplify the specification of the mapping [2], the formalization of the semantics of the translation process [3], and various notions of quality of the results [4–6].

This paper investigates how the mapping process changes in the presence of richer *ontology schemas* of the two data sources. Studying this variant of the problem is important for several reasons.

(i) First, the emergence of the Semantic Web has increased the number of data sources on top of which ontology-like descriptions are developed.
(ii) Second, ontologies play a key role in information integration since they are used to give clients a global conceptual view of the underlying data, which in turn may come from external, independent, heterogeneous, multiple information systems [7]. On the contrary, the global unified view given by the ontology is constructed independently from the representation adopted for the data stored at the sources.
(iii) Finally, many of the base transactional repositories used in complex organizations by the various processes and applications often undergo modifications during the years, and may lose their original design. The new schema can often be seen as a set

☆ This work has been partially supported by the Ministerio de Ciencia y Tecnología under projects TIN2011-24747 and TIN2014-52938-C2-2-R.

http://dx.doi.org/10.1016/j.datak.2015.07.003
0169-023X/© 2015 Published by Elsevier B.V.

of views over the original one. It is important to be able to run the existing mappings against a view over the new schema that does not change, thus keeping these modifications of the sources transparent to the users.

It is therefore important to study how the mapping process changes in this setting.

1.1. Contributions

In this paper, we assume that an ontology is provided for the target and, possibly, for the source data repository. The relationship between the domain concepts in this ontology schema and the data sources is given by a set of views that define the ontology constructs in terms of the logical database tables using a relational language of conjunctive queries, comparisons and negations.

We develop a number of techniques to solve this kind of *ontology-based mapping problem*. More specifically:

- we develop rewriting algorithms to automatically translate mappings over the ontology schema into mappings over the underlying databases; we first discuss the case in which an ontology schema is available for the target database only; then we extend the algorithm to the case in which an ontology schema is available both for the source and the target;
- the algorithm that rewrites a source-to-ontology mapping into a classical and executable source-to-target mapping is based on the idea of unfolding views in mapping conclusions; in our setting this unfolding is far from being straightforward; in the paper, we show that the problem is made significantly more complex by the expressibility of the view-definition language, and more precisely, by the presence of negated atoms in the body of view definitions;
- we study the implications of adopting such an expressive language; to handle negation in view definitions we adopt a very expressive mapping language, namely, that of *disjunctive embedded dependencies (deds)* [8]. Deds are mapping dependencies that may contain disjunctions in their heads, and are therefore more expressive than standard embedded dependencies (tgds and egds);
- this increased expressive power makes the data-exchange step significantly more complex. As a consequence, we investigate restrictions to the view-definition language that may be handled using standard embedded dependencies, for which efficient execution strategies exist. In the paper, we identify a restricted view language that still allows for a limited form of negation, but represents a good compromise between expressibility and complexity; we prove that under this language, our rewriting algorithm always returns standard embedded dependencies;
- the classical approach to executing a source-to-target exchange consists of running the given mappings using a *chase* engine [3]. We build on the Llunatic chase engine [9,10], and extend it to execute not only standard tgds and egds, but also deds. We discuss the main technical challenges related to the implementation of deds. Then, using the prototype, we conduct several experiments on large databases and mapping scenarios to show the trade-offs between expressibility of the view language, and efficiency of the chase. To the best of our knowledge, this is the first practical effort to implement execution strategies for deds, and may pave the way for further studies on the subject.

This paper represents a significant step forward towards the goal of incorporating richer ontology schemas into the data translation process. Given the evolution of the Semantic Web, and the increased adoption of ontologies, this represents an important problem that may lead to further research directions.

This paper extends our prior research [11], where we first studied the problem of rewriting ontology-based mappings. We make several important advancements, as follows:

(i) First, previous papers only discussed rewritings based on standard embedded dependencies for a rather limited form on negation. In this paper, we extend our algorithms to handle arbitrary non-recursive Datalog with negation using deds, thus considerably extending the reach of our rewriting algorithm.
(ii) At the same time, we make the sufficient conditions under which the rewriting only contains embedded dependencies more precise, and extend the limited case discussed in previous papers.
(iii) In addition, we present the first chase technique for deds, and a comprehensive experimental evaluation based on scenarios with and without deds. As we mentioned above, this is the first practical study of the scalability of the chase of high-complexity dependencies, an important problem in data exchange.
(iv) Finally, we provide full proofs of all theorems (in Appendix A).

1.2. Outline

The paper is organized as follows. Our motivating example is given in Section 2. Section 3 recalls some basic notions and definitions. Section 4 introduces the ontology-based mapping problem. Section 5 defines disjunctive embedded dependencies which are required by the rewriting when the views that define the mapping are beyond conjunctive queries. Section 6 provides the definition of a correct rewriting. The rewriting algorithm and formal results are in Section 7. Section 8 identifies a view-definition language that is more expressive than plain conjunctive queries but such that it computes correct rewritings only in terms of embedded dependencies. The chase engine is described in Section 9. Experiments are in Section 10. We discuss related work in Section 11.

2. Motivating example

Assume we have the two relational schemas below and we need to translate data from the source to the target.