Pay-as-you-go Matching of Relational Schemata to OWL Ontologies With *IncMap*⋆

Christoph Pinkel[1] ⋆⋆, Carsten Binnig[2], Evgeny Kharlamov[3], and Peter Haase[1]

[1] fluid Operations AG [2] University of Mannheim [3] University of Oxford

Abstract. Ontology Based Data Access (OBDA) enables access to relational data with a complex structure through ontologies as conceptual domain models. A key component of an OBDA system are mappings between the schematic elements in the ontology and their correspondences in the relational schema. Today, in existing OBDA systems these mappings typically need to be compiled by hand. In this paper we present *IncMap*, a system that supports a semi-automatic approach for matching relational schemata and ontologies. Our approach is based on a novel matching technique that represents the schematic elements of an ontology and a relational schema in a unified way. Finally, *IncMap* can extend user-verified mapping suggestions in a pay-as-you-go fashion.

1 Introduction

Today, enterprise information systems of large companies typically store petabytes of data across multiple relational databases, each with hundreds or thousands of tables (e.g., [1]). Effective understanding of complex schemata is a crucial task for enterprises to support decision making and retain competitiveness on the market. Ontology-based data access (OBDA) [2] is an approach that has recently emerged to provide semantic access to complex structured (relational) data. However, in many existing real-world systems (e.g. [2]) that follow the ODBA principle, the mappings have to be created manually, which constitutes a significant entry barrier for applying OBDA in practice.

To overcome this limitation, we propose a novel semi-automatic schema matching approach and a system called *IncMap*. We focus on finding one-to-one correspondences of ontological and relational schema elements, while we also work on extensions for finding more complex mappings.

The matching approach of *IncMap* is inspired by the Similarity Flooding (SF) algorithm [3] that works well for schemata that follow the same modeling principles. However, we show that applying the SF algorithm naively for matching relational schemata to OWL ontologies results in rather poor suggestion quality due to a conceptual mismatch between ontologies and relational schemata. The contributions of the paper are the following: In Section 2, we propose a novel graph structure called *IncGraph* to represent schema elements from ontologies and relational schemata in a unified way. In Section 3, we present our matching algorithm that supports an incremental pay-as-you-go approach that can

⋆ The research was supported by the EU Commission's FP7 grant Optique (n. 318338).

⋆⋆ E-Mail: christoph.pinkel@fluidops.com

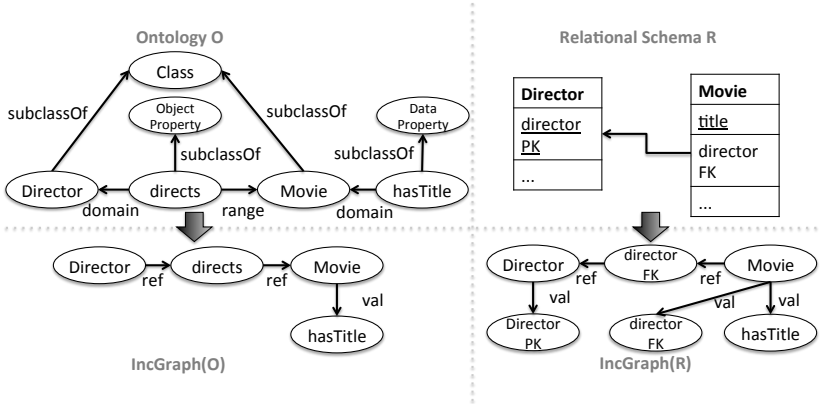

Fig. 1. *IncGraph* Construction Example

leverage existing mappings. Finally, Section 4 presents an experimental evaluation using different (real-world) relational schemata and ontologies. Experiments show that the basic version of *IncMap* reduces the effort for creating a mapping up to 20% compared to applying SF in a naive way. The incremental version of *IncMap* can reduce the total effort by another $50\% - 70\%$.

2 The *IncGraph* Model

The *IncGraph* model used by *IncMap* represents schema elements of an OWL ontology $\mathcal{O}$ and a relational schema $\mathcal{R}$ in a unified way. An *IncGraph* model is defined as directed labeled graph $(V, \texttt{Lbl}_V, E, \texttt{Lbl}_E)$. V represents a set of vertices, E a set of directed edges, $\texttt{Lbl}_V$ a set of labels for vertices and $\texttt{Lbl}_E$ a set of labels for edges. A label $l_V \in \texttt{Lbl}_V$ represents a name of a schema element whereas a label $l_E \in \texttt{Lbl}_E$ is either "ref" representing a so called `ref`-edge or "value" representing a so called `val`-edge. Figure 1 shows a cinematography related ontology $\mathcal{O}$ and relational schema $\mathcal{R}$, as well as the result of constructing graphs *IncGraph*($\mathcal{O}$) and *IncGraph*($\mathcal{R}$) according to the *IncGraph* model. While $\mathcal{O}$ and $\mathcal{R}$ describe the same entities *Directors* and *Movies* and their relationship in a different way, the *IncGraph* $\mathcal{O}$ and $\mathcal{R}$ is designed to represent both in a structurally similar fashion.

However, after constructing the *IncGraph* models, structural differences between *IncGraph*($\mathcal{O}$) and *IncGraph*($\mathcal{R}$) might still exist due to the mismatch between the high level view of the domain in ontologies and the low level view of data in relational databases. *IncMap* therefore adds *annotations* in *IncGraph* to bridge these structural gaps. Annotations are added as inactive `ref`-edges which can be activated during the schema matching process. For instance, additional `ref`-edges are added to *IncGraph* ($\mathcal{R}$) as *shortcuts* for join-paths to better match the *IncGraph* ($\mathcal{O}$). Moreover, another idea is to add *inverse* `ref`-edges to unify the structure resulting from modeling relationships in different directions (e.g., the *directs*-predicate in $\mathcal{O}$ vs. the *directorFK*-relationship in $\mathcal{R}$ in Figure 1. Finally, results from reasoning over an ontology $\mathcal{O}$ can also be integrated into *IncGraph* ($\mathcal{O}$). Analyzing these annotations in detail is a future work.

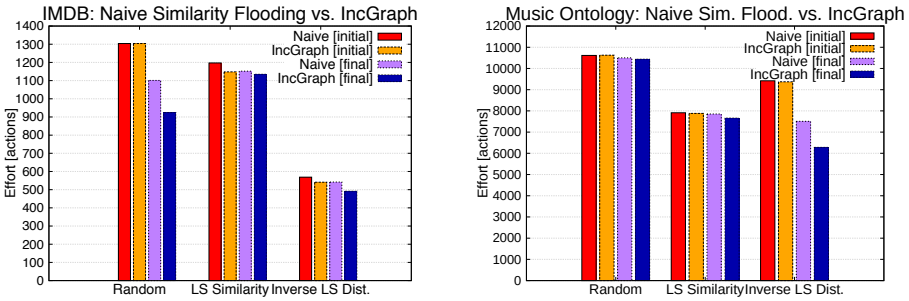

Fig. 2. Naive vs. *IncGraph*

3 The *IncMap* System

IncMap takes the *IncGraph*s produced for a relational schema $\mathcal{R}$ and for an ontology $\mathcal{O}$ as input. In its basic version, *IncMap* applies the original SF algorithm and thus creates initial mapping suggestions for the *IncGraph* of $\mathcal{O}$ and $\mathcal{R}$. Additionally, *IncMap* can activate `ref`-edges (i.e., annotations) before executing the SF algorithm to achieve better results.

One important extension is the incremental version of *IncMap*. In this version the initial suggestions are re-ranked by *IncMap* by including user feedback. The idea of user feedback is that the user confirms those mapping suggestions of the previous iteration, which are required to answer a given user query over $\mathcal{O}$.

We support three methods for incorporating user feedback into the matching process: First, the naive *Initializer* method changes the score of confirmed or rejected mappings to initialize the next run to 1.0 and 0.0, respectively. Second, *Self-Confidence Nodes* work similar but the initialization is repeated during the fix-point computation of the SF algorithm which results in a stronger influence of the user feedback. Finally, *Influence Nodes* include additional nodes in the graph structure to locally influence the score of a confirmed or rejected mappings. Please refer to [4] for a more detailed description of those methods.

IncMap is designed as a framework and provides different knobs to control which extensions and variations to use. A major avenue of future work is to apply optimization algorithms to find the best configurations automatically.

4 Experimental Evaluation

We evaluate *IncMap* using to two real-world scenarios that provided hand crafted mappings as gold standard. As a first scenario, we evaluate a mapping from movie database IMDB to the Movie Ontology (`http://www.movieontology.org`) The second scenario is a mapping from the MusicBrainz database to the Music Ontology (`www.musicontology.com`) We evaluate *IncMap* w.r.t. reducing *work time* (i.e., effort) needed to correct the correspondences suggested by *IncMap* to match the gold standard. The effort is defined as the sum of steps that users need to validate the suggested mappings for each node in the *IncGraph* ($\mathcal{O}$). For validating one mapping the user needs to reject all suggested correspondences in the decreasing order of their final ranking score until reaching the correct mapping whereas each rejection is counted as one step.

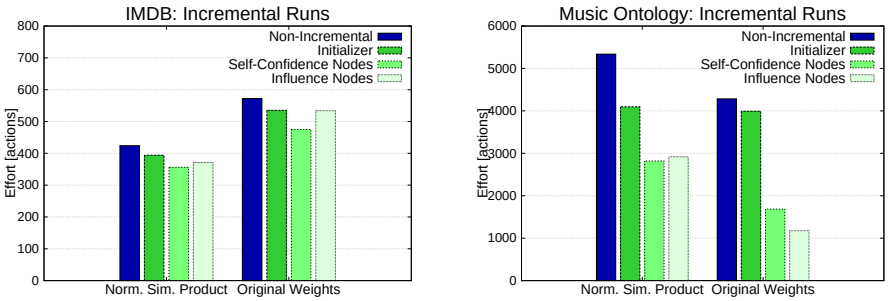

Fig. 3. Incremental Evaluation

Experiment 1 – Naive vs. IncGraph. In our first experiment we compare the work time required to correct the mapping suggestions when the schema and ontology are represented naively as schema graphs, or using *IncGraph*s. Additionally, we vary the lexical matcher using three alternatives: randomly assigned scores (base line), Levenshtein similarity and inverse Levenshtein distance. Figure 2 shows that *IncGraph* works better in all cases than the naive approach.

Experiment 2 – Incremental Mapping Generation. In the second experiment we evaluate the incremental schema matching in *IncMap*. Figure 3 show the resulting work time for the three incremental methods. Most significantly, incremental evaluation reduces the overall effort (work time) by up to $50\% - 70\%$ compared to the naive non-incremantal version. For both scenarios Self-Confidence Nodes and Influence Nodes work much better than the naive Initializer approach.

5 Conclusions and Outlook

We presented *IncMap*, a novel semi-automatic matching approach for matching relational schemata to ontologies. Our approach is based on a novel unified graph model called *IncGraph* for ontologies and relational schemata. Based on the *IncGraph* model, *IncMap* implements a novel semi-automatic matching approach inspired by the Similarity Flooding algorithm to derive mappings using both lexical and structural similarities of ontologies and relational schemata. Our experiments with *IncMap* on real-world relational schemata and ontologies showed that the effort for creating a mapping with *IncMap* is up to 30% less than using the Similarity Flooding algorithm in a naive way. The incremental version of *IncMap* reduces the total effort of mapping creation by another $50\% - 70\%$.

References

1. SAP HANA Help: http://help.sap.com/hana/html/sql_export.html (2013)
2. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The mastro system for ontology-based data access. Semantic Web Journal **2**(1) (2011) 43–53
3. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In: ICDE. (2002)
4. Pinkel, C., Binnig, C., Kharlamov, E., Haase, P.: IncMap: Pay-as-you-go Matching of Relational Schemata to OWL Ontologies. In: OM. (2013)