

# ADAMAP: Automatic Alignment of Relational Data Sources using Mapping Patterns (*Abstract*)

Diego Calvanese<sup>1,2</sup>, Avigdor Gal<sup>3</sup>, Naor Haba<sup>3</sup>, Davide Lanti<sup>1</sup>,  
Marco Montali<sup>1</sup>, Alessandro Mosca<sup>1</sup>, and Roei Shraga<sup>3</sup>

<sup>1</sup> Free-University of Bozen-Bolzano, Bolzano, Italy, *lastname@inf.unibz.it*

<sup>2</sup> Umeå University, Sweden

<sup>3</sup> Technion – Israel Institute of Technology, Haifa, Israel

{avigal@, naor-haba@campus., shraga89@campus.}@technion.ac.il

**Abstract.** This extended abstract presents our work on automatic discovery of RDB-to-OWL 2 QL *mapping patterns* published at CAiSE 2021 [1].

## 1 Introduction and Contribution

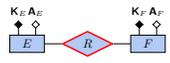
Modern industrial and business processes require intensive use of large-scale data alignment and integration techniques to combine data from multiple heterogeneous data sources into meaningful and valuable information. Such integration is performed on structured and semi-structured data sets from various sources such as SQL and XML schemata, entity-relationship (ER) diagrams, ontology descriptions, process models, and web forms. One of the main challenges of data integration is to create a common semantic understanding from the multiple available data sources. In ontology-based data access (OBDA) and integration [6], this is achieved through two main components: (i) an ontology that captures the relevant concepts and relations of the domain of interest at a high level of abstraction, in turn acting as a vehicle for reaching a semantic consensus; and (ii) a mapping specification that dictates how the data in relational sources can be used to (virtually) populate the classes and properties of the ontology.

A major impediment towards the adoption of OBDA is that data sources typically lack a proper semantic documentation, which makes it extremely difficult and error-prone to obtain both the ontology and the mapping. In this work, we aim at reconstructing such lost domain semantics by inspecting relational data sources, without any additional documentation. To do so, we start from the key observation that while the relational model may be semantically-poor with respect to ontological models, the original semantically-rich design of the application domain leaves recognizable footprints that can be converted into the aforementioned ontology and mapping specifications. Therefore, we propose to use *ontology mapping patterns* [2], which systematically collect recurring ways of linking relational data sources to ontologies via mapping assertions. Based on such patterns, we propose an algorithmic technique called ADAMAP that, given a relational data source, automatically determines how suitable fragments of its schema align with corresponding mapping patterns. Once mapping patterns are suitably instantiated on a

---

Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Table 1: Portion of Schema-driven Patterns from [2]

E-R DIAGRAM	DB SCHEMA	MAPPING PATTERN	ONTOLOGY
<b>Schema Entity (SE)</b>			
	$T_E(\mathbf{K}, \mathbf{A})$	$s: T_E$ $t: C_E(t_E(\mathbf{K})),$ $\{d_A(t_E(\mathbf{K}), A)\}_{A \in \mathbf{K} \cup \mathbf{A}}$	$\{\exists d_A \sqsubseteq C_E\}_{A \in \mathbf{K} \cup \mathbf{A}}$
<b>Schema Relationship (SR)</b>			
	$T_E(\mathbf{K}_E, \mathbf{A}_E)$ $T_F(\mathbf{K}_F, \mathbf{A}_F)$ $T_R(\mathbf{K}_{RE}, \mathbf{K}_{RF})$	$s: T_R$ $t: p_R(t_E(\mathbf{K}_{RE}), t_F(\mathbf{K}_{RF}))$	$\exists p_R \sqsubseteq C_E$ $\exists p_R^- \sqsubseteq C_F$
In case of $(\_, 1)$ cardinality on role $R_E$ (resp., $R_F$ ), the primary key for $T_R$ is restricted to the attributes $\mathbf{K}_{RE}$ (resp., $\mathbf{K}_{RF}$ ).			

given data source, they can be employed for a number of downstream data engineering tasks, *e.g.*, *ontology bootstrapping* [3,4,5,8] and *schema cover* [7].

*Contributions.* The contribution of this work is twofold. On a conceptual level, we offer an approach to semantically enrich a relational model by exploiting the footprints left by the conceptual design on which the relations are based. We then offer an algorithmic solution to align the relations (for which we assume to have a definition of primary keys, foreign keys, and unique constraints) with the mapping patterns introduced in [2].

## 2 Ontology Mapping Patterns

A mapping pattern is a quadruple  $(C, S, M, O)$ , where  $C$  is a conceptual schema,  $S$  is a database schema,  $M$  is a set of mappings, and  $O$  is an ontology. In such mapping pattern, the pair  $(C, S)$  puts into correspondence a conceptual representation to one of its (many) admissible (*i.e.*, formally sound) database schemata. Such variants are due to differences in the applied methodology, efficiency and performance optimizations, and database space consumption. The *database schema ontology* ontology  $O$  [9] is the OWL 2 QL encoding (hence, not lossless) of  $C$ , and the set  $M$  of *database schema mappings* provides the link between  $S$  and  $O$ . Table 1 shows two examples of patterns, namely, **Schema Entity (SE)** and **Schema Relationship (SR)**. SE is a fundamental pattern that considers a single table  $T_E$  with primary key  $\mathbf{K}$  and other attributes  $\mathbf{A}$ . The pattern captures how  $T_E$  is mapped into a corresponding class  $C_E$ . The primary key of  $T_E$  is employed to construct the objects that are instances of  $C_E$ , using a template  $t_E$  specific for that class. Each relevant attribute of  $T_E$  is mapped to a data property of  $C_E$ . *Example.* A projects table whose primary key is the attribute *id*, together with their funding scheme and their reference in the CORDIS portal, is mapped to a *Project* class using the *id* attribute to construct its instances. In addition, every attribute in the table is mapped to a corresponding data property.

SR considers three tables  $T_R$ ,  $T_E$ , and  $T_F$ , in which the primary key of  $T_R$  is partitioned into two parts  $\mathbf{K}_{RE}$  and  $\mathbf{K}_{RF}$  that are foreign keys to  $T_E$  and  $T_F$ , respectively.  $T_R$  has no additional attributes. The pattern captures how  $T_R$  is mapped to an object property  $p_R$ , using the two parts  $\mathbf{K}_{RE}$  and  $\mathbf{K}_{RF}$  of the primary key to construct respectively the subject and the object of each triple in  $p_R$ .

<https://cordis.europa.eu/projects/en>

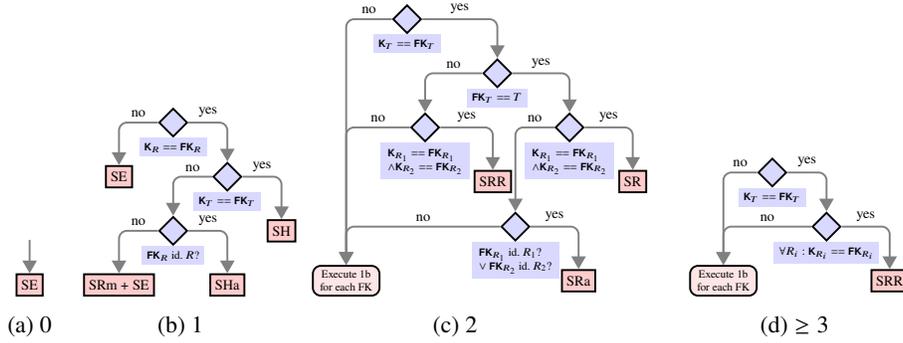


Fig. 1: ADAMAP inference for a table  $T$  by the number of foreign keys it contains.

*Example.* A `projects_erc_panels` table connecting through two foreign keys the projects to their corresponding ERC panels. Such table is mapped to an `:ercPanel` object property, for which the ontology asserts that the domain is the class `:Project` and the range is an additional class `:ERC-Panel`, which corresponds to the `erc_panels` table according to the SE pattern.

### 3 The ADAMAP Algorithm

Figure 1 shows the ADAMAP inference for a table  $T$  by the number of foreign keys it contains. The rectangular red nodes denote the identified mapping patterns (see Table 2 and [2]), and the decision points represent the choices that the algorithm makes to determine such patterns.

*Evaluation.* To assess the feasibility of the approach in practice, we focus on non-trivial and real-world scenarios. We identified two such scenarios, analyzing in total more than one thousand mapping assertions, namely:

- *CORDIS*, which is designed around the domain of competitive research projects, provided by SIRIS Academic S.L. (<https://www.sirisacademic.com/wb/>), a consultancy company specialized in higher education and research; and
- *NPD*, which is built around the domain of oil and gas extraction, and contains data coming from the FactPages portal (<https://factpages.npd.no/en>).

The full validation in [1], confirms that the patterns automatically identified by ADAMAP mostly conform to those manually identified by human experts. Hence, the patterns identified by ADAMAP provide a sound basis that can be further manually improved by experts.

*Acknowledgements.* This research has been partially supported by the EU H2020 project INODE (863410), the Italian PRIN project HOPE, the European Regional Development Fund Investment for Growth and Jobs Programme 2014-2020 through the project IDEE (FESR1133), and the Free University of Bozen-Bolzano through the projects QUADRO, KGID, GeoVKG, and STyLoLa.

Table 2: Patterns Abbreviations

Pattern	Abbreviation
Schema Entity	SE
Schema Relationship	SR
→with Identifier Alignment	SRa
→with Merging	SRm
Schema Reified Relationship	SRR
Schema Hierarchy	SH
→with Identifier Alignment	SHa

## References

1. Calvanese, D., Gal, A., Haba, N., Lanti, D., Montali, M., Mosca, A., Shraga, R.: Adamap: Automatic alignment of data sources using mapping patterns. In: Proc. of the 33rd Int. Conf. on Advanced Information Systems Engineering (CAiSE 2021). Lecture Notes in Computer Science, Springer (2021)
2. Calvanese, D., Gal, A., Lanti, D., Montali, M., Mosca, A., Shraga, R.: Mapping patterns for virtual knowledge graphs. CoRR Technical Report arXiv:2012.01917, arXiv.org e-Print archive (2020), <https://arxiv.org/abs/2012.01917>
3. Jiménez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M.G., Thorstensen, E., Mora, J.: BootOX: Practical mapping of RDBs to OWL 2. In: Proc. of the 14th Int. Semantic Web Conf. (ISWC). Lecture Notes in Computer Science, vol. 9367, pp. 113–132. Springer (2015)
4. de Medeiros, L.F., Priyatna, F., Corcho, O.: MIRROR: Automatic R2RML mapping generation from relational databases. In: Proc. of the 15th Int. Conf. on Web Engineering (ICWE). pp. 326–343. Springer (2015)
5. Pinkel, C., Binnig, C., Kharlamov, E., Haase, P.: IncMap: pay as you go matching of relational schemata to OWL ontologies. In: Proc. WS on Ontology Matching. CEUR Workshop Proceedings, <http://ceur-ws.org/>, vol. 1111 (2013)
6. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. *J. on Data Semantics* **10**, 133–173 (2008)
7. Saha, B., Stanoi, I., Clarkson, K.L.: Schema covering: a step towards enabling reuse in information integration. In: Proc. of the 26th IEEE Int. Conf. on Data Engineering (ICDE). pp. 285–296. IEEE Computer Society (2010)
8. Sequeda, J.F., Miranker, D.P.: Ultrawrap Mapper: A semi-automatic relational database to RDF (RDB2RDF) mapping tool. In: Proc. of the 14th Int. Semantic Web Conf., Posters & Demonstrations Track (ISWC). CEUR Workshop Proceedings, <http://ceur-ws.org/>, vol. 1486 (2015)
9. Spanos, D.E., Stavrou, P., Mitrou, N.: Bringing relational databases into the Semantic Web: A survey. *Semantic Web J.* **3**(2), 169–209 (2012)