An architecture for efficient knowledge-driven information and data access (abstract)

Pablo Rubén Fillottrani[1,2] and C. Maria Keet[3]

[1] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, prf@cs.uns.edu.ar
[2] Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina
[3] Department of Computer Science, University of Cape Town, South Africa
mkeet@cs.uct.ac.za

Advanced information systems require the orchestration of many components, including ontologies or knowledge graphs, and efficient data management, in order to provide a means for better informed decision-making and to keep up with new requirements in organisational needs. A major question in delivering such systems, is which components to design and put together to create a 'knowledge to data' pipeline, as each component and process has trade-offs, such as the computational complexity of the representation and query languages, which reasoning services that operate under an open or closed world, and maintainability. Such a combination of knowledge with data is named with various terms, including ontology-based data access (OBDA) [4], with (virtual) knowledge graphs gaining in popularity in research and industry (e.g., [12, 15]). OBDA has become synonymous with the approach of declaring a mapping layer between knowledge represented in an OWL file and data stored in a relational database whilst using query rewriting in answering conjunctive queries. That mapping layer is known to be costly both computationally [7] and in design and maintenance [10]. Also, database users may want to retain full SQL expressiveness when querying data, and remain within the closed world assumption they are more familiar with cf. the open world assumption in OBDA systems.

In an attempt to avoid these issues, an "Abstract Relational Model" (ARM) with special object identifiers and a strict extension to SQL for path queries (SQLP) has been proposed [3] and experimentally shown to simplify queries [11]. This approach avoids the costly mapping layer through transformations and offers more than full SQL, but the ARM is not an ontology. Put differently this ARM+SQLP falls short of the knowledge layer.

We aim to address this limitation of the ARM+SQLP option. To this end, we introduce a new knowledge-to-data architecture, KnowID: **Know**ledge-driven **I**nformation and **D**ata access. It pulls together both recently proposed components of the knowledge layer and, to complete the pipeline, we add novel transformation rules between EER and the ARM, which enhances and adapts rules from the regular EER-to/from-Relational Model (RM) transformation with important additional expressiveness. KnowID's components and the addition to ARM+SQLP is visualised in Fig. 1. Regarding the four steps in the top block: 1) If the model is not in EER, one can convert it into EER by means of a metamodel or common core (e.g., [5, 9]); 2) If the EER diagram was not formalised

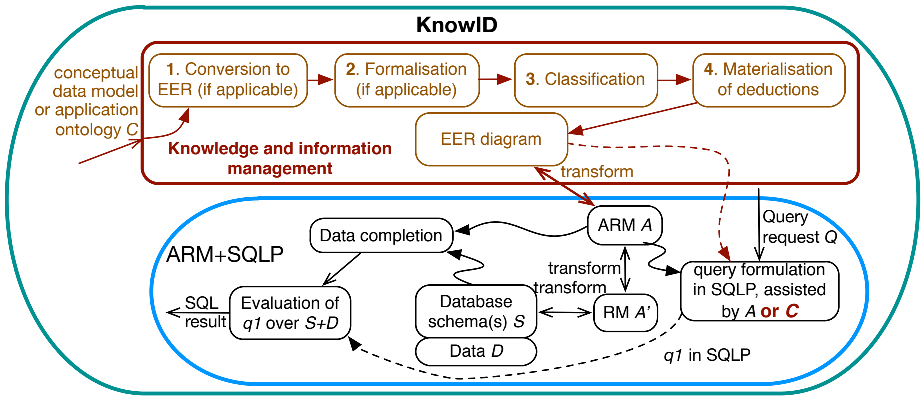

Fig. 1. Extending ARM+SQLP toward KnowID, i.e., adding a knowledge layer to the architecture, into the **Know**ledge-driven **I**nformation and **D**ata access, KnowID, architecture.

yet, one of the logic-based reconstructions may be used (e.g., [2, 14]) that, ideally, supports all that KnowID supports as modelling language features: entity type (weak and strong), n-ary relationship ($n \geq 2$), attribute, basic cardinality constraints (0..n, 0..1, 1, 1..n) and identifier, entity type subsumption, and disjointness and covering constraints; 3) inferences can be computed (e.g., with a DL reasoner) and undesirable deductions dealt with by the modeller as usual; and 4) materialising the deductions amounts to modifying the EER diagram by adding the acceptable deductions to the model, in a similar fashion as used to be possible in the earlier Protégé tool for OWL [6]. The model resulting from completing step 4 is the one that will be closed and transformed into an ARM by means of our proposed set of rules and then used for querying the data.

KnowID's functionality is thus similar to OBDA systems [4] and to 'enhanced' databases such as OntoMinD [1]: one can reason and pose queries at the knowledge layer—i.e., supporting a user in *what* to query, without the labour-intensive discovery of *how and where*—that will be evaluated over an 'intelligent' database that avails of the formally represented knowledge in the conceptual data model or ontology. Architecturally, a distinct practical advantage is that it achieves this through a series of automated transformations that are linear in the model's size, rather than (manual or automated) specifications of non-trivial mappings in a separate mapping layer. Further advantages are closed world assumption commonly used in information systems and full SQL augmented with path queries. The latter has been shown in user experiments to make query formulation faster with at least the same level of accuracy or fewer errors [11, 8], and discovery through paths is seen as essential for data integration [13].

We are currently implementing the EER↔ARM transformation rules as a first step toward concretely realising KnowID as a usable and scalable software system.

References

1. Al-Jadir, L., Parent, C., Spaccapietra, S.: Reasoning with large ontologies stored in relational databases: The OntoMinD approach. DKE **69**, 1158–1180 (2010)
2. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Reasoning over extended ER models. In: Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B. (eds.) Proc. of ER'07. LNCS, vol. 4801, pp. 277–292. Springer (2007), auckland, New Zealand, November 5-9, 2007
3. Borgida, A., Toman, D., Weddell, G.E.: On referring expressions in information systems derived from conceptual modelling. In: Proc. of ER'16. LNCS, vol. 9974, pp. 183–197. Springer (2016)
4. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., Xiao, G.: Ontop: Answering SPARQL queries over relational databases. Semantic Web Journal **8**(3), 471–487 (2017)
5. Fillottrani, P.R., Keet, C.M.: Evidence-based languages for conceptual data modelling profiles. In: Morzy, T., et al. (eds.) Proc. of ADBIS'15. LNCS, vol. 9282, pp. 215–229. Springer (2015), 8-11 Sept, 2015, Poitiers, France
6. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-Computer Studies **58**(1), 89–123 (2003)
7. Gottlob, G., Kikot, S., Kontchakov, R., Podolskii, V.V., Schwentick, T., Zakharyaschev, M.: The price of query rewriting in ontology-based data access. Artif. Intell. **213**, 42–59 (2014)
8. Junkkari, M., Vainio, J., Iltanenan, K., Arvola, P., Kari, H., , Kekäläinen, J.: Path expressions in SQL: A user study on query formulation. Journal of Database Management **22**(3), 22p (2016)
9. Keet, C.M., Fillottrani, P.R.: An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2. DKE **98**, 30–53 (2015)
10. Lubyte, L., Tessaris, S.: Automated extraction of ontologies wrapping relational data sources. In: Proc of DEXA'09. pp. 128–142. Springer (2009)
11. Ma, W., Keet, C.M., Oldford, W., Toman, D., Weddell, G.: The utility of the abstract relational model and attribute paths in sql. In: Faron Zucker, C., Ghidini, C., Napoli, A., Toussaint, Y. (eds.) Proc. of EKAW'18. pp. 195–211. Springer (2018), 12-16 Nov. 2018, Nancy, France
12. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: Lessons and challenges. Queue **17**(2), 20:48–20:75 (Apr 2019)
13. Stonebraker, M., F., I.I.: Data integration: the current status and the way forward. IEEE Data Engineering **41**(2), 3–9 (2018)
14. Toman, D., Weddell, G.E.: On adding inverse features to the description logic $\mathcal{CFD}^{\forall}{}_{\mathrm{nc}}$. In: Proc of PRICAI 2014. pp. 587–599 (2014), Gold Coast, QLD, Australia, December 1-5, 2014.
15. Xiao, G., Ding, L., Cogrel, B., Calvanese, D.: Virtual knowledge graphs: An overview of systems and use cases. Data Intelligence **1**, 201–223 (2019)