Conceptual Schema Transformation in Ontology-based Data Access (Extended Abstract)

Diego Calvanese[1], Tahir Emre Kalayci[2,1], Marco Montali[1], Ario Santoso[3,1], and Wil van der Aalst[4]

[1] KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano (Italy)
[2] Virtual Vehicle Research Center, Graz (Austria)
[3] Department of Computer Science, University of Innsbruck (Austria)
[4] Process and Data Science (PADS), RWTH Aachen University (Germany)

During the last two decades, (structural) conceptual schemas have been increasingly adopted not only to understand and document the relevant aspects of an application domain at a high level of abstraction, but also as live, computational artifacts. In particular, the paradigm of Ontology-Based Data Access (OBDA) exploits conceptual schemas (also called ontologies) as an intermediate layer for accessing and querying data stored inside legacy information systems [10]. In the context of OBDA, the conceptual schema provides end-users with a vocabulary they are familiar with, at the same time masking how data are concretely stored, and enriching those (incomplete) data with domain knowledge. In this light, we call such a conceptual schema *domain schema*.

OBDA has been subject of extensive research, and its advantages have been concretely shown in a variety of application domains (see, e.g., [4,6,7]). Surprisingly, though, no research has been carried out on how to suitably extend the OBDA approach to handle the common situation where a higher-level conceptual schema (which we call *upper schema*) is needed to further abstract the knowledge captured by the domain schema. This happens when there is the need of viewing the domain schema and, in turn, the underlying data, according to a predefined structure, described by a reference model or an interchange format. In addition, different users may need to generate different views on the data, possibly using multiple upper schemas.

We illustrate the need for such a multi-level approach to data access on the common situation where certain types of users adopt reference models as an upper schema to understand the business relationships existing between an organization and its external stakeholders. For example, the managers of an e-commerce company need to reconstruct the state of contractual relationships and mutual commitments with customers, on top of the domain concepts of orders, payments, and deliveries. At the same time, managers employ the commitment-based core reference ontology for services (UFO-S) [8] as an upper schema, to understand and monitor the state of commitments that contractually relate the company and its customers. When managers need to inspect which commitments currently exist, and in which state they are, they cannot directly formulate queries of this form on top of the legacy data, due to a vocabulary mismatch.

A possible solution would be to create a dedicated OBDA specification that directly connects the legacy data to the UFO-S upper schema. However, this is unrealistic from the conceptual modeling point of view, for two reason. First and foremost, linking data directly to concepts and relations in the upper schema requires to first understand the data in terms of domain notions, and only then to establish suitable connections between the domain and the upper level schemas. In addition, an OBDA specification connecting data to the domain schema could be in place independently on these UFO-S related needs. It is well-known that creating an OBDA specification, especially for what concerns the understanding of the legacy data structures and the construction of corresponding mappings, is a labor-intensive and challenging task [3], similarly to alternative approaches to data access and integration. If such a specification is already in place, it would be beneficial to build on it so as to gracefully integrate the upper schema into the picture, instead of creating another OBDA specification from scratch. A second issue is related to the fact that reference models and upper ontologies are typically meant to capture a plethora of concepts and relations spanning over a wide range of application domains, with the purpose of resolving ambiguities and misunderstandings [8,9,5]. In a specific application domain, only a small portion of the whole reference model is needed to capture the commitments of interest.

To attack these issues, we propose to adopt a standard OBDA approach to make sense of the legacy data in terms of the domain schema. Once this OBDA specification is in place, domain experts can forget about the schema of the legacy data, and work directly at the level of the domain schema. In addition, a declarative specification is provided that declares how the domain schema, e.g., of orders, can be transformed into (a portion of) the UFO-S upper schema. Once the mapping and transformation rules are specified, managers can express queries over UFO-S, with the aim to obtain answers that are computed over the legacy data. This 2-level approach favors modularity and separation of concerns, since the mapping and the transformation rules can vary independently from each other. In particular, if the underlying data storage changes, only the mapping to the domain schema needs to be updated, without touching the definition of commitments. If instead the contract is updated, the domain-to-upper schema transformation needs to change accordingly, without touching the OBDA specification.

To tackle such challenging but common scenarios, we propose to suitably extend the OBDA framework so as to take into account multiple conceptual layers at once. We focus on the case where two conceptual layers are present, accounting for the domain and upper schemas, and call the resulting setting *2-level OBDA* (2OBDA for short).

Specifically, our contribution is threefold. *(i)* We introduce the 2OBDA model as an elegant extension of OBDA. The core part of the framework is the *conceptual transformation* of concepts and relations in the domain schema into corresponding concepts and relations in the upper schema. This is specified in a declarative way, similarly to OBDA mappings but in this case accounting for ontology-to-ontology correspondences. *(ii)* We show how a 2OBDA specification can be automatically compiled into a classical OBDA specification that directly connects the legacy data to the upper schema, fully transparently to the end-users. Consequently, these can query the legacy data through the upper schema, by resorting to standard OBDA systems like *ontop* [1]. *(iii)* We realize the approach in a tool-chain that supports end-users in modeling the domain and upper schemas, and in specifying the corresponding transformations as annotations of the domain schema, whose types and features are derived from the concepts in the upper schema. Notably, the tool-chain fully implements the above compilation technique.

The full paper has been published in the proceedings of the 21st International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018) [2].

References

1. 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 J. **8**(3), 471–487 (2017). doi:10.3233/SW-160217
2. Calvanese, D., Kalayci, T.E., Montali, M., Santoso, A., van der Aalst, W.: Conceptual schema transformation in ontology-based data access. In: Proc. of the 21st Int. Conf. on Knowledge Engineering and Knowledge Management (EKAW). Lecture Notes in Computer Science, Springer (2018)
3. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: The onprom tool and methodology. In: Proc. of the 20th Int. Conf. on Business Information Systems (BIS). Lecture Notes in Business Information Processing, vol. 288, pp. 220–236. Springer (2017). doi:10.1007/978-3-319-59336-4 16
4. Daraio, C., Lenzerini, M., Leporelli, C., Naggar, P., Bonaccorsi, A., Bartolucci, A.: The advantages of an ontology-based data management approach: Openness, interoperability and data quality. Scientometrics **108**(1), 441–455 (2016). doi:10.1007/s11192-016-1913-6
5. Guizzardi, G.: On Ontology, ontologies, conceptualizations, modeling languages, and (meta)models. In: Databases and Information Systems IV – Selected Papers from the 7th Int. Baltic Conference (DB&IS 2006). Frontiers in Artificial Intelligence and Applications, vol. 155, pp. 18–39. IOS Press (2006), http://www.booksonline.iospress.nl/Content/View.aspx?piid=5421
6. Kharlamov, E., Hovland, D., Skjæveland, M.G., Bilidas, D., Jiménez-Ruiz, E., Xiao, G., Soylu, A., Lanti, D., Rezk, M., Zheleznyakov, D., Giese, M., Lie, H., Ioannidis, Y.E., Kotidis, Y., Koubarakis, M., Waaler, A.: Ontology based data access in Statoil. J. of Web Semantics **44**, 3–36 (2017). doi:10.1016/j.websem.2017.05.005
7. Mehdi, G., Kharlamov, E., Savkovic, O., Xiao, G., Kalayci, E.G., Brandt, S., Horrocks, I., Roshchin, M., Runkler, T.A.: Semantic rule-based equipment diagnostics. In: Proc. of the 16th Int. Semantic Web Conf. (ISWC). Lecture Notes in Computer Science, vol. 10588, pp. 314–333. Springer (2017). doi:10.1007/978-3-319-68204-4_29
8. Nardi, J.C., de Almeida Falbo, R., Almeida, J.P.A., Guizzardi, G., Pires, L.F., van Sinderen, M.J., Guarino, N., Fonseca, C.M.: A commitment-based reference ontology for services. Information Systems **54**, 263 – 288 (2015). doi:10.1016/j.is.2015.01.012
9. Scherp, A., Saathoff, C., Franz, T., Staab, S.: Designing core ontologies. Applied Ontology **6**(3), 177–221 (2011). doi:10.3233/AO-2011-0096
10. Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., Zakharyaschev, M.: Ontology-based data access: A survey. In: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 5511–5519. Int. Joint Conf. on Artificial Intelligence Organization (2018). doi:10.24963/ijcai.2018/777