Evaluation of Technologies for Mapping Representation in Ontologies

Olga Kovalenko[1], Christophe Debruyne[2], Estefanía Serral[1], and Stefan Biffl[1]

[1] Vienna University of Technology, CDL-Flex,
Favoritenstrasse 9-11/E188, A-1040 Vienna
firstname.lastname@tuwien.ac.at
[2] Vrije Universiteit Brussel STARLab, Pleinlaan 2, B-1050 Brussel
firstname.lastname@vub.ac.be

Abstract. Ontology mapping is needed to explicitly represent the relations between several ontologies, which is an essential task for applications such as semantic integration and data transformation. Currently, there is no standard for representing mappings. Instead, there are a number of technologies that support the representation of mappings between the ontologies. In this paper we introduce a set of mapping categories that were identified based on requirements for the data integration projects of an industry partner. An evaluation of available technologies for mapping representation regarding the support for introduced mapping categories has been performed. The results of the evaluation show that the SPARQL Inference Notation would fit the best in the described use case scenario.

1 Introduction

Ontologies are widely used to obtain semantic integration and semantic interoperability among different systems. For many applications it is necessary to explicitly specify the relations and correspondences between different ontologies. The common technique here is to define a set of mappings that bind a set of entities in one ontology to the entities in another ontology.

To our knowledge there is no standard or conventional technique on how to represent mappings in ontologies. The Web Ontology Language[1] (OWL) provides some support to express mappings between the entities of different ontologies [5]. These mapping are stored as a part of the ontology and are coupled with it [10]. However, the expressivity of OWL is restricted to rather simple mappings, such as one-to-one mapping for ontology concepts and properties. Differences in granularity or understanding of the domain semantics require support for more sophisticated mappings and cannot be expressed in pure OWL [5]. In order to support defining more complex mappings between ontologies various technologies have been developed, e.g., the Datalog language, the Semantic Web Rule Language (SWRL)[2], SPARQL CONSTRUCT queries[3], and the SPARQL

[1] OWL overview: `http://www.w3.org/TR/owl-features/`
[2] SWRL specification: `http://www.daml.org/2003/11/swrl/`
[3] SPARQL specification: `http://www.w3.org/TR/rdf-sparql-query/`

R. Meersman et al. (Eds.): OTM 2013, LNCS 8185, pp. 564–571, 2013.

Inference Notation (SPIN)[4]. The degree of support for defining mappings between ontologies provided by these technologies vary in many aspects, e.g., by the expressivity for mapping representation and on how mappings are stored, which must be taken into account for choosing the appropriate technology for a specific task. It is important to note that the selection of an optimal technique for mapping representation will strongly depend on the application at hand, e.g., the best fitting technology for ontology merging or for semantic integration can be different [11].

In this paper we create a benchmark for comparing the abilities of technologies to express different categories of mappings. We investigate categories of complex mappings between ontologies needed in a typical data integration scenario in multi-disciplinary engineering projects [1]. In these projects the data from the ontologies of the disciplines involved in the project must be transformed into a common ontology to allow the integration of the local discipline ontologies. The contribution of the paper is twofold: a) we categorize mappings that are required in the projects of an industry partner to fulfill effective and efficient data integration and data transformation; and b) we evaluate the available technologies for mapping representation regarding the support provided for introduced mapping categories. We focus on OWL and RDF(S)[5] ontologies, as these are currently prevailing ontology languages.

The rest of the paper is organized as follows: Section 2 describes the related work on ontology mapping; in Section 3 we present the application scenario from our industry partners that is used across the paper to illustrate introduced mapping categories and determine a set of mapping categories that may be needed in the described use case scenario; and in Section 4 we evaluate the existing technologies for mapping representation regarding the support for defined mapping categories and discuss which technology would be a better choice in the presented use case scenario.

2 Related Work

Mapping captures the semantic and structural relationships between the entities of two ontologies. Or in other words, mappings specify the connection between entities in enough details that it will be possible to apply or execute them for a certain task [3].

According to Noy et al three dimensions can be distinguished in ontology mapping research: a) mappings discovery; b) mappings representation (MR) and c) reasoning with mappings [9]. In this paper we focus on the second dimension - representation of mappings.

Several authors have addressed the problem of MR in their works. Noy at al. distinguish three main ways on how mappings can be represented: a) as first-order logic axioms serving as semantic bridges between ontologies; b) using views

[4] SPIN overview: `http://www.w3.org/Submission/spin-overview/`
[5] RDF(S) specification: `http://www.w3.org/TR/rdf-schema/`

that link global ontology to local ontologies; and c) mappings themselves are instances in a mappings ontology [9]. An approach using the Semantic Bridges between entities of two ontologies was presented by Maedche et al. [8]. A language to specify mappings between ontologies and requirements for such language are described by Scharffe and de Bruijn in [10]. An interesting approach was presented by Brockmans et al. [2], which is based on Meta Object Facility (MOF)[6] and the Unified Modeling Language (UML)[7] to support formalism independent graphical modeling of mappings between OWL ontologies.

The general problem is that there are many systems developed by different vendors, which represent mappings in dissimilar formats. A standard language for this purpose would highly facilitate the reusability and interoperability of the results of such systems, however there is no such single standard established yet [10]. Up to date there is also a lack of work investigating the abilities and appropriateness of current technologies for MR in different application scenarios. We believe that this problem has high practical importance as the specific application will influence the choice of optimal MR technique for the task at hand [11]. Focused on the data integration and data transformation applications, this paper provides a state of the art on existing technologies for MR between ontologies, and evaluates their support for a set of essential mapping categories.

3 Use Case Scenario and Mappings Description

As a running example we present a data integration scenario from an industry partner, a hydro power plant systems integrator.

As is shown on the Figure 1 ontologies are applied to integrate data from all disciplines involved in the project engineering. The data model of each discipline is represented by its local ontology. The common data model, which includes only concepts that are relevant to at least two disciplines, is built on the top of local ontologies. For simplicity reasons we limit the set of concepts and attributes shown to the minimal set necessary to illustrate all the identified mapping categories introduced further. On the top of the figure 1, the local ontologies representing data models from three different domains (software engineering (SE), mechanical engineering (ME) and project management (PM)) are shown. The concept *PLC* describes the controllers' behavior of the devices; the concept *PhysicalComponent* describes the components of the plant, and the concept *Connection* captures direct connections between the devices; the concept *Project* describes the important characteristics of the project under development. The figure 1 also shows relations between each local model and the common model represented by mappings. Different mapping categories are identified as M1-M7.

Value Processing (M1-M4). Often a relation between entities in two ontologies is not "exactly-the-same", but can be represented by some function that

[6] OMG's MetaObject Facility specification: `http://www.omg.org/mof/`

[7] Unified Modeling Language specification: `http://www.omg.org/spec/UML/`

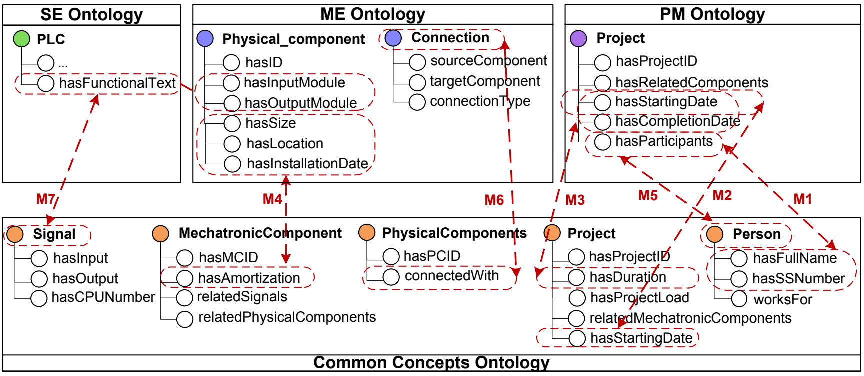

Fig. 1. Mapping categories: M1 - string processing; M2 - data type transformation; M3 - math functions; M4 - user-defined functions; M5 - structural differences → granularity; M6 - structural differences → schematic differences; M7 - conditional mappings. Mappings M1-M2 could be also considered as bidirectional mappings (M8).

takes a value of a property in source ontology as an input and returns a value of a property in the target ontology as an output. Below 4 types of suchlike mappings are described.

String Processing (M1). In the PM ontology an employee is represented by the attribute *hasParticipants* of the *Project* concept, which value is a list of strings in the "name/surname/social security number" format. In the CC ontology the concept *Person* represents employee. Thus, corresponding mapping defines that initial string must be split into 3 parts, which then will be used as values for the *hasFullName* and the *hasSocialSecurityNumber* attributes.

Data Type Transformation (M2). It can happen that in a proprietary data model the data type of a certain concept was not modeled in an optimal way, e.g. dates are represented as strings. In the example, the value of the *hasStartingDate* attribute in the PM ontology may be string in the "DD/MM/YYYY" format. If the data type of corresponding attribute in the CC ontology is *Date* then the data type transformation is needed.

Math Functions (M3). Here the relation between the entities represents some mathematical expression. For an instance, to obtain a value of the *hasDuration* attribute in the CC ontology a substraction of *hasStartingDate* from *hasEndingDate* in PM ontology must be performed.

User-Defined Functions (M4). This category comprises functions that are not supported by the used technology, but must be additionally implemented. In the example, a value of the *hasAmortization* attribute of an mechatronic component will depend on its location and size and also on its installation date.

Granularity (M5). In the example, each employee in the PM ontology is represented by a string value of the hasParticipants attribute, while in the CC ontology the concept *Person* serves the same purpose.

Schematic Differences (M6). This type of mappings can arise when two ontologies on a similar domain were created separately and thus, the same semantics was modeled differently. In the example, a connection between physical devices in the ME ontology is represented by the *Connection* concept with the *sourceComponent* and *targetComponent* attributes, while in the CC ontology the same semantics is expressed with the *connectedWith* attribute of the *PhysicalComponent* concept.

Conditional Mappings (M7). One of the mapping aspects we wish to examine is to what extent technologies allow for the transformation of descriptions of one type of thing into descriptions of another type. In the example, the concept "signal" models a connection between physical devices and their controllers and corresponding PLC code. A value of the *hasFunctionalText* attribute of the *PLC* concept from the SE ontology is a string in "CPU number/input module/output module" format. Corresponding mapping states that for any pair of PLC and physical component, such that their values of *hasFunctionalText* and *hasInputModule* and *hasOutputModule* attributes have certain conformity an instance of *Signal* concept must be created in the CC ontology with appropriate values in its *hasInput*, *hasOutput* and *hasCPUNumber* attributes.

Bidirectional Mappings (M8). Usually mappings are specified in a way that knowledge/data flow can occur in one direction only. However, for some applications it is beneficial to define bidirectional mappings between the entities in order to reduce total amount of mappings and facilitate their maintenance [4].

4 Technology Evaluation for Mapping Categories

Below we describe the technologies that can be applied to create mappings between OWL ontologies before providing an overview on how well these technologies cover the examples in Fig. 1.

The Web Ontology Language (OWL) is an ontology language where one can declare relations between concepts such as equivalence, subsumption, etc. and allows one to infer additional information about instances by reasoning over the properties of classes and relations. OWL itself neither supports the representation of mappings between ontologies nor provides means for translating instances from one ontology to another. One way to represent mappings is to use the **SPARQL CONSTRUCT** query form, which returns an RDF graph based on template instantiated with he results of a query.

A second option is to use rules, which can be declared on top of OWL ontologies. Apache Jena[8] includes a rule-based inference engine called **Jena Rules** for reasoning with RDF and OWL data sources based on a Datalog implementation. Datalog is a declarative logic programming language that is becoming increasingly popular for data integration tasks [6]. **SPARQL Inference Notation** (SPIN) [7] is currently submitted to W3C and provides – amongst others – means to link class definitions with SPARQL queries (ASK and CONSTRUCT)

[8] Apache Jena: `http://jena.apache.org/`

Table 1. Evaluation of Technologies for Different Mapping Categories

Legend: Y = yes, P = partial, N = no; * = depends on provider	String	Data Type	Math	User-defined	Granularity	Schematic	Conditional	Bi-directional	
	M1	M2	M3	M4	M5	M6	M7	M8	Standard
Jena-Rules	P	P	P	*	Y	Y	Y	P	-
SPIN	Y	Y	Y	Y*	Y	Y	Y	P	W3C Sub.
SWRL	Y	Y	Y	*	Y	Y	Y	P	W3C Rec.
SPARQL CONSTRUCT	P	Y	P	*	Y	Y	Y	N	W3C Rec.

to infer triples. The **Semantic Web Rule Language**, (SWRL) is a W3C recommendation for a Semantic Web rules language, combining OWL DL – a decidable fragment of OWL – with those of the Rule Markup Language. Rules are thus expressed in terms of OWL concepts. Note that SPARQL CONSTRUCT is *not* a rule language and "merely" allows one to make a transformation from one graph match to another graph; i.e., a one-step transformation.

Now we provide an overview on how well above mentioned technologies cover the examples in Fig. 1. This comparison is depicted in Table 1. Where necessary, we provide some extra information on certain caveats of the technologies.

String Processing (M1). The built-in string functions for Jena-Rules are limited; there exists for instance no function for obtaining substrings or string replacement. SPARQL supports some additional basic string functions. We therefore consider the string processing coverage of Jena-Rules and SPARQL partial. SWRL and SPIN offer additional string functions. Some frameworks – such as Jena – allows one to create custom functions (cfr. M4).

Data Type Transformation (M2). There are two aspects to be analyzed: xsd data type transformations and transforming values into other values. SPARQL 1.1 – and hence, also SPIN – allows one to cast the value of a variable to another xsd data type. The Jena-Rules and SWRL data type transformations are limited; e.g., Jena-Rules has a built-in primitive for creating a URI out of a set list of strings. As SWRL and Jena-Rules are used to infer additional triples to query, the SPARQL queries built on top of these rules can do the casting. The second type of transformation is covered by M1 and M4.

Math Functions (M3). All considered technologies support arithmetic, albeit Jena-Rules implementation has some limitations as mentioned in the previous Section. SWRL and SPIN even provide some additional functions (e.g., *sin*, *cos*, etc.), which are not defined in the SPARQL specification and their existence can depend on the provider. Again, when necessary, some function can be provided as a user-defined function (cfr. M4).

User-Defined Functions (M4). The implementation of user-defined functions is interesting to look at. Some simple functions can be built via rules and others might need to be implemented or already exist. Defining functions as part of the mapping can only be done in SPIN. For Jena Rules, SWRL and SPARQL, one can include custom functions if the software allows so. Jena, for instance, allows one to create Java classes representing functions that can be imported.

Jena thus covers the inclusion of custom functions, albeit via code. This is why we denoted this possibility in the table with an asterisk. SPIN "inherits" this aspect by depending on Jena.

Granularity (M5) and Schematic Differences (M6) tackled the problem of structural differences. All technologies provide means for transforming information into different representations. The limits actually depend on the specific requirements of a domain. In the presented use case scenario, one could use the social security number to create a URI for that person. Otherwise, blank nodes have to be introduced. The application domain of the knowledge bases could also have business rules that tell us how to construct URIs; which can be accomplished with aforementioned mappings.

Conditional Mappings (M7). Unlike OWL, which only allows relations between classes and properties via class- and property relations, both rules and SPARQL CONSTRUCT allow the transformation of descriptions of one type of concept into descriptions of other types of (related) concepts.

Bidirectional Mappings (M8). Ideally, mappings are bidirectional and maintained as such in one artifact. However, this can only be guaranteed if there is a bijective mapping between models of two ontologies. Take, for example, the mapping from one entity containing with a height and width as an attribute to another entity with a surface. Computing the surface by multiplying the height and width is easy, but the inverse yields more than one possibility. None of the studied technologies allow for bidirectional rules. However, we can examine the problem from a different angle and see to what extent we can "simulate" bidirectionality by storing both mappings in one artifact. When possible, bidirectional mappings can be declared with Jena-Rules, SWRL and SPIN. This is not possible with SPARQL CONSTRUCT queries, as a SPARQL CONSTRUCT queries allows one to populate a graph based on the result set of a query.

For our use case, SPIN and SPARQL CONSTRUCT are suitable candidates to represent mappings. SPARQL has the advantage of being W3C recommendation for quite a while and therefore different software solutions exist. Some of these solutions support the definition of custom functions. The downside is that mappings that depend on other mappings have to be executed in a certain order and are not a part of the knowledge base. Using SPIN, rules are part of the knowledge base. SPIN even has a notion of function that can be reused in different parts of the knowledge base. The downside is that SPIN is still in consideration by W3C. An implementation of SPIN is available from TopBraid[9] that depends on Jena. Given the fact that it is more desirable to have the mappings as part of the knowledge base, SPIN proves to be the more suitable candidate.

5 Conclusion and Future Work

In this paper we examined available technologies that support mapping representation between ontologies regarding the ability to represent complex mappings

[9] `http://www.topquadrant.com/`

that were identified based on requirements for the data integration projects of an industry partner. We evaluated existing technologies for mapping representation regarding the support provided for introduced mapping categories and discussed which technology would fit the best for the described use case scenario. The results show that in the introduced use case scenario the SPIN would be the optimal choice.

Possible directions for future work could include: a) Investigation on how different applications scenarios will influence the choice of optimal mapping representation technology; b) Following the work of Shvaiko et al. [12], focusing on the construction of mappings and considering it as a collaborative (and therefore social) process.

Acknowledgements. This work has been supported by the Christian Doppler Forschungsgesellschaft and by the Erasmus Mundus Action 2 Programme of the European Union.

References

1. Biffl, S., Moser, T., Winkler, D.: Risk assessment in multi-disciplinary (software+) engineering projects. International Journal of Software Engineering and Knowledge Engineering 21(02), 211–236 (2011)
2. Brockmans, S., Haase, P., Stuckenschmidt, H.: Formalism-independent specification of ontology mappings – A metamodeling approach. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 901–908. Springer, Heidelberg (2006)
3. Ehrig, M.: Ontology alignment: bridging the semantic gap, vol. 4. Springer Science+ Business Media (2007)
4. Ghidini, C., Serafini, L., Tessaris, S.: On relating heterogeneous elements from different ontologies. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 234–247. Springer, Heidelberg (2007)
5. Haase, P., Motik, B.: A mapping system for the integration of owl-dl ontologies. In: Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems, pp. 9–16. ACM (2005)
6. Huang, S.S., Green, T.J., Loo, B.T.: Datalog and emerging applications: an interactive tutorial. In: Sellis, T.K., Miller, R.J., Kementsietsidis, A., Velegrakis, Y. (eds.) SIGMOD Conference, pp. 1213–1216. ACM (2011)
7. Knublauch, H., Handler, J.A., Idehen, K.: SPIN - Overview and Motivation. W3C Member Submission, W3C (February 2011), `http://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/`
8. Maedche, A., Motik, B., Silva, N., Volz, R.: MAFRA – A MApping FRAmework for distributed ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 235–250. Springer, Heidelberg (2002)
9. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. ACM Sigmod Record 33(4), 65–70 (2004)
10. Scharffe, F., de Bruijn, J.: A language to specify mappings between ontologies. In: Proc. of the Internet Based Systems IEEE Conference (SITIS 2005). Citeseer (2005)
11. Scharffe, F., Fensel, D.: Correspondence patterns for ontology alignment. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 83–92. Springer, Heidelberg (2008)
12. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1164–1182. Springer, Heidelberg (2008)