On the Reduction of Dublin Core Metadata Application Profiles to Description Logics and OWL

Dimitrios A. Koutsomitropoulos

High Performance Information Systems Lab, Computer Engineering and Informatics Dpt.,
University of Patras

kotsomit@hpclab.ceid.upatras.gr

Abstract. This paper gives an account of how traditional metadata application profiles are related to Web ontologies and Description Logics. It is shown that metadata profiles can be reduced to Description Logics; oddly enough though, OWL 2, with its current expressivity, is proven to be inadequate for this purpose. First, we give a brief overview of the recent proposal for representing metadata profiles using the notion of Description Set Profiles and XML Schema. We point out the necessity of an additional representation scheme, using also ontological languages. Following, we introduce the reduction of Description Set Profiles to Description Logics and identify the required expressivity characteristics that are essential for this reduction. Finally, we discuss what expense do these characteristics put on to the complexity of reasoning.

1   Aims and Scope

The need for a consistent framework for developing application profiles has also been recognized from within the Dublin Core Metadata Initiative (DCMI). After the DC 2007 conference, the so-called "Singapore Framework" for developing application profiles based on Dublin Core (DC) was introduced in [7]. Building upon notions of the DCAM [10], a major part of the Singapore Framework is the development of a constraint language, just like an XML schema, that would formally specify what kind of (possibly originating from different standards) properties the particular application profile involves and what kind of values are appropriate for these properties, possibly constrained by specific syntax and vocabulary encoding schemes. The purpose of such a "Description Set Profile" (DSP) [6] is to identify

metadata records that are matching (or conforming) to this DSP. This is turn means that the DSP language may be primarily used for expressing *structural* and *syntactic* constraints that underline the application profile, leaving out of scope *semantic* interoperability. Besides, as mentioned in the Singapore Framework:

"It is important to realize that the semantics of the terms used in application profiles is carried by their definitions, which are independent of any application profile. [...]. As semantic interoperability is provided by a correct use of terms defined in one or more vocabularies, application profiles are about providing high-level syntactic or structural interoperability in addition to the semantic interoperability".

It has been shown elsewhere [2] that the above argument is not valid in the case of *semantic profiles*; In addition, the notion of *semantic profiling* further refines the semantics of terms and drives towards semantic interoperability. Moreover, and despite the implementation of the DSP language in an XML Schema, we believe that its expression in RDF would be more appropriate, having in mind the recent implementations of DC in RDF(S) [8] as well as the DC ontology in OWL [4].

## 2	The Reduction Procedure

First, notice that structural constraints, such as values permitted and typing of resources, have their counterpart in RDFS domain and range restrictions. Also the notion of "allowed properties" can be accommodated as in the following:

The basic structural element of a DSP is a *description template.* A DSP may include any number of such templates. A description template corresponds to the description of resources of a specific type (e.g. items, persons,…) and defines restrictions on the set of properties that are relevant to the specific resource type (i.e. it is in their domain). Restrictions on a property are posed by a *statement template* and thus a *description template* can have any number of statement templates.

A description template can therefore be seen as a single property on its own; a property that is partitioned by the set of allowed properties (i.e. an *n-ary* property). An approach to define such properties can be found in [9]. We can

define a class *Description_ID* for each description template. Then, for each allowed property $P_1, \ldots, P_n$:

$$Description\_ID \sqsubseteq \exists P_1.range_1 \sqcap \ldots \sqcap \exists P_n.range_n \qquad (1)$$

This expresses the constraint that every instance of *Description_ID* has at least one relation, through $P_1, \ldots, P_n$ with an instance from the appropriate range. To express the constraint that $P_1, \ldots, P_n$ can relate *Description_ID* instances *only* to the appropriate ranges, we can use universal quantification:

$$Description\_ID \sqsubseteq \forall P_1.range_1 \sqcap \ldots \sqcap \forall P_n.range_n \qquad (2)$$

Restrictions on the number of allowed fillers for each property can be modelled in the same manner, using qualified number restrictions, thus replacing the (more general) existential restrictions.

To see that the above expressions are also sufficient (not only necessary), suppose that x $\in$ *Description_ID* and $P_k(x, y)$, where $1 \leq k \leq n$ and y does not belong in the allowed range. Then, due to (2) the ontology becomes inconsistent, since y must be an instance of the appropriate P_k range. Also, due to (1) *x must* have (although undeclared yet) n-1 other relations, through allowed properties.

To express the fact that $P_1, \ldots, P_n$ are the *only* properties allowed requires some more elaboration. In fact, what we need is the ability to express a role-disjunction axiom. OWL 2 provides for disjoint roles, but not for role disjunctions in general. Let U be the *universal role*, i.e. the parent of all roles. It holds that:

$$\exists P_1.range_1 \sqcup \ldots \sqcup \exists P_n.range_n \sqsubseteq \exists U.\top$$

We want to express that $\exists P_m.range_m$, where m other than $1 \ldots n$, is not allowed in *Description_ID*. In the presence of a role-disjunction axiom and role complement the set of non-allowed properties can be expressed as:

$$U \sqcap \neg(P_1 \sqcup \ldots \sqcup P_n)$$

Description_ID should not include instances that are related to others through non-allowed properties. That is:

$$Description\_ID \sqcap \exists(U \sqcap \neg(P_1 \sqcup \ldots \sqcup P_n)).\top \equiv \emptyset$$

3   Expressivity Characteristics and Complexity

We therefore come to the conclusion that, in order to express DSPs it is necessary to have the three logical operators (union, intersection, complement) available for role expressions or, at least, role names only. Of the above operators, no one is available in OWL 2. Additionally, we also need qualified number restrictions, which are not included in OWL DL, while the universal role can be safely eliminated [2].

Lutz και Sattler [5] show that the Description Logic *ALC* (⊔, ⊓, ¬) that is, the basic logic *ALC* augmented with logical operators on roles, is in NEXP. Moreover, in [12], it is shown that the Description Logic *ALCIQ* (⊔, ⊓, ¬) that is, the logic mentioned previously augmented with qualified number restrictions and role inverses, which apparently is adequate for expressing DSPs, but does not correspond to any ontology language, is NEXP-complete.

Under the condition that role expressions, transformed to *Disjunctive Normal Form* (DNF, i.e. union of intersections), should have at least one non-negated role in each disjunct (aka *safe Boolean combinations*), this logic becomes PSPACE-complete. For example, the expression $\neg(P_1 \sqcup \dots \sqcup P_n)$ is written in DNF as $\neg P_1 \sqcap \dots \sqcap \neg P_n$, which consists of only one disjunct that includes no non-negative parts, unless the universal role is used.

In addition, Schmidt and Tishkovsky [11] show that the Description Logic *ALBO* that is, the basic logic *ALC* augmented with role union, role complement, inverse roles and nominals, is NEXP-complete (role intersection is not mentioned but included, because $P \sqcap R \equiv \neg (\neg P \sqcup \neg R)$, where P, R role names) and implement a corresponding tableau algorithm. They also mention that applying complement on role chains leads to undecidability, a fact that also holds when applying intersection [1].

Consequently, we argue that the RDF(S) and OWL semantics are not adequate for expressing the structural and syntactic limitations of DC application profiles. In addition, we see that, although DSPs intend to tackle with syntactic constraints only, it turns out that these have considerable semantic implications.

4   Conclusions

Counter-intuitively, the expression of classic metadata application profiles, as they are recently attempted to be standardized around DC with the Singapore Framework, cannot be achieved with the current expressivity of OWL. We have shown exactly what expressivity characteristics are missing from the last specification of this language and are necessary for the specification of such profiles. Nevertheless, we have proven that the syntactic constraints posed by a typical application profile can be reduced to semantic restrictions inside an ontology. The hardness -even undecidability- of reasoning under the addition of these characteristics should not be seen as weakness though: First, XML Schema can easily be used in order to enforce and verify such syntactic restrictions; on the other hand, their counterpart in Description Logics is not necessary to simultaneously include all the rest of the OWL 2 expressivity characteristics, but only a few ones.

References

1. Calvanese, D., & Giacomo, G. d.. Expressive Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, & P. F. Patel-Schneider (Eds.), *The Description Logic Handbook* (2nd ed.). Cambridge (2007)
2. Horrocks, I., Kutz, O., & Sattler, U. The Even More Irresistible SROIQ. *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)* (pp. 57-67). AAAI Press (2006)
3. Koutsomitropoulos, D. A., Paloukis, G. E., & Papatheodorou, T. S. From Metadata Application Profiles to Semantic Profiling: Ontology Refinement and Profiling to Strengthen Inference-based Queries on the Semantic Web. *Int. J. on Metadata, Semantics and Ontologies , 2* (4), pp. 268-280 (2007)
4. Koutsomitropoulos, D. A., Solomou, G. D., & Papatheodorou, T. S. Semantic Interoperability of Dublin Core Metadata in Digital Repositories. *5th Int. Conference on Innovations in Information Technology.* IEEE (2008)
5. Lutz, C., & Sattler, U. Mary likes all Cats. Proc. of the 2000 Int. Workshop in Description Logics (DL 2000), Vol 33 in CEUR-WS, pp. 213-226 (2000).
6. Nilsson, M. Description Set Profiles: A constraint language for Dublin Core Application Profiles. DCMI Working Draft. Retrieved from http://dublincore.org/documents/dc-dsp/ (2008)
7. Nilsson, M., Baker, T., & Johnston, P. The Singapore Framework for Dublin Core Application Profiles. Retrieved from http://www.dublincore.org/documents/ singapore-framework/ (2008).
8. Nilsson, M., Powell, A., Johnston, P., & Naeve, A. Expressing Dublin Core metadata using the Resource Description Framework (RDF). DCMI Recommendation. Ανάκτηση από http://dublincore.org/documents/dc-rdf/ (2008).

9. Noy, N., & Rector, A. (Eds.). Defining N-ary Relations on the Semantic Web. Semantic Web Best Practices and Deployment Working Group Note. Retrieved from http://www.w3.org/TR/swbp-n-aryRelations/ (2006)
10. Powell, A., Nilsson, M., Naeve, A., Johnston, P., & Baker, T. DCMI Abstract Model. DCMI Recommendation. Retrieved from http://dublincore.org/documents/abstract-model/ (2007).
11. Schmidt, R. A., & Tishkovsky, D. Using Tableau to Decide Expressive Description Logics with Role Negation. *Proc. of the 6th Int. Semantic Web Conference (ISWC 2007)* (pp. 438-451). Springer (2007)
12. Tobies, S. Complexity results and practical algorithms for logics in Knowledge Representation. PhD Thesis, LuFG Theoretical Computer Science,RWTH-Aachen (2001)