Ontology Transformation as a Procedural Extension of the Description Logic Reasoning

Igor Toujilov and Paul Taylor

Centre for Health Informatics and Multiprofessional Education,
University College London, Highgate Hill, London N19 5LW, UK
i.toujilov@ucl.ac.uk,p.taylor@chime.ucl.ac.uk
http://www.chime.ucl.ac.uk

Abstract. An approach to OWL DL expressivity enrichment, based on automatic transformation of ontologies, is proposed. This direction of research follows and further develops existing approaches of ontology transformation and preprocessing. A presented OWL preprocessor, implemented using the OWL API and SWRL, brings wide range of numeric calculations and custom theory implementations. The preprocessor makes object property cardinality values accessible in SWRL rules, which allow implementing many numeric aggregation functions directly in OWL DL. Theory implementation is discussed on a probability theory example.

Key words: OWL, description logic, ontology preprocessing, SWRL, probability theory

1 Introduction

Description logic (DL) reasoning has some attractive features such as guaranteed decidability, very high level representation of knowledge, and a concise simple notation [1]. However the price paid for these advantages is the very limited expressivity of DL, e.g. the almost complete absence of ordinary numeric calculations. The purpose of the research presented here is to overcome some of these drawbacks while retaining the advantages.

Our work is based on ontology transformation, an approach first introduced in [2], who subsequently developed the Ontology Preprocessing Language (OPPL, currently version 2). OPPL takes a non-procedural approach, which thus inherits some of the limitations of DL.

Our approach pragmatically extends the concept of ontology transformation/preprocessing to a custom procedural preprocessing, implemented in a general programming/scripting language, e.g. Java. The transformation retains the key feature of OWL DL - that reasoning with the ontology is decidable and enables two additional features: the extension of numeric calculations and the implementation of custom theories. Of course, the ontology transformation may be executed in multiple steps where we have a chain of transformations.

2 Numeric Calculations

Some calculation features are already available in SWRL. However the successful application of SWRL calculations in practice almost always requires access to property cardinality values that are not available in SWRL. Our idea is to preprocess an ontology in order to instantiate the property cardinalities as data properties that are always available in SWRL. Unfortunately cardinality variables are not allowed in OPPL2 because they can lead to the ontology becoming undecidable. We have implemented this idea in a simple Java application called an OWL preprocessor, which combines the DL reasoning, operating inside the ontology, with some simple procedural operations at the meta-level.

Once the property cardinality values are available in SWRL, we can implement various aggregation calculations in SWRL, for example, percentage. This test case consists of two OWL modules. The first module is a generic ontology, required for all cardinality-enabled calculations in OWL DL. The second module imports the generic ontology and contains application specific instances. Only the second module needs to be pre-processed.

3 Custom Theory Implementations

Another important application of the ontology transformation is the automatic generation of ontology axioms for theories to support customised approaches to reasoning. Here we provide only a draft of an approach for the generation of probability theories. In this example we also have two OWL modules: a generic ontology for the probability theory, and an example of a custom theory, importing the generic probability theory module.

The classic probability theory defines probabilities of propositions that may be uncertain. In DL the most generic form of proposition is one asserting that an entity is a member of a class. Hence we define a class the probabilistic entity class of conforming to the axioms of probability theory. This is a super-class of all classes dealing with probability. Note that the axioms are not implemented in the upper-level module. They need to be generated automatically by an ontology preprocessor in a custom theory module, and they are expressed as equivalence class and superclass axioms, and as rules.

Acknowledgments. The presented work is a part of the DebugIT project, `http://www.debugit.eu`, funded within the 7th EU Framework Programme.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., and Patel-Schneider, P.F. (eds). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Egana, M., Antezana, E., Stevens, R.: Transforming the Axiomisation of Ontologies: The Ontology Pre-Processor Language. In: Proceedigns of OWLED 2008 DC OWL: Experiences and Directions. (2008)