

SweetProlog: A System to Integrate Ontologies and Rules

Loredana Laera, Valentina Tamma and Trevor Bench-Capon

University of Liverpool
Liverpool L69 7ZF, UK
{lori, valli, tbc@csc.liv.ac.uk}

Abstract

In this poster, we present our prototype system to represent semantic derivation rules to support reasoning on the Semantic Web. The prototype, called SweetProlog, is able to reason over the instances of an ontology within rule definitions using a Prolog engine and enables the integration of ontologies and rules.

1 Introduction

Rules represent an important issue for the Semantic Web and constitute the next language level over the ontology languages on the Semantic Web, as envisioned by Tim-Berners Lee [Berners-Lee *et al.*, 2001]. Rules give the ability to draw inferences, to express constraints and to describe some aspects of applications, not expressible in ontology languages. In this context, several rule languages and standards have been proposed, such as RuleML [Boley *et al.*, 2001] and SWRL [Horrocks *et al.*, 2004], a RuleML-oriented extension of OWL. However, the SWRL proposal provides a syntactical and semantical interoperability between ontologies and rules but not *inferential*, due to its undecidability. The crucial point is to provide some reasoning support and to provide *semantic*, *syntactic* and *inferential* interoperability between ontologies and rules. In other words, we need a way to enable the rules to use the vocabulary of an ontology and to reason in a semantically consistent way to draw inferences.

Here we propose our prototype for reasoning, both with ontologies and rules, that uses a Logic Programming Language, such as Prolog, to offer efficient automatic reasoning. The system, called SweetProlog, shows how derivation rules may be used to enhance the content of a ontology and allow the dynamic inclusion of derived facts that are not captured by the ontological taxonomy alone.

2 Features of SweetProlog

2.1 A “meta-ontology” for RuleML

In order to define the rules we have used a OWL ontology of Courteous Logic Programs (CLP) RuleML rule syntax [Grosz *et al.*, 2002], a sub-language of the Rule Markup Language. Intuitively, the resulting language, called OWL-RuleML, can be viewed as a “meta-ontology” of the RuleML

syntax. OWLRuleML allows to specify derivation rules and preserves the semantics of all constraints in RuleML. An interesting aspect of OWLRuleML is the possibility to represent non-monotonicity via Courteous Logic Programs. CLPs [Grosz, 1997] extend the expressivity of Ordinary Logic Programs (OLP), and they are tractably compilable to OLP by a Courteous Compiler. CLPs provide a method to resolve conflicts between rules using partially prioritized information to guarantee a consistent and unique set of conclusions (answer-set).

2.2 Rules on top of ontologies

In order to integrate rules with OWL ontologies, we follow the approach proposed [Grosz, 2003] to build *rules on top of ontologies*. Principally, the OWLRuleML rules can be linked to descriptions in OWL. The names of predicates in the OWL-RuleML rules are URI's that link to classes and properties in an OWL ontology. The OWL ontology that is referenced forms a background theory for the rulebase. Formally, the Knowledge Representation contained within the intersection of LP rules on top of DL ontologies corresponds to Description Logics Programs (DLP) [Grosz *et al.*, 2003]. DLP is a fragment of OWL corresponding to Horn clauses, i.e. to the logic programming fragment of OWL. DLP provide a significant degree of expressiveness, substantially greater than the RDFS fragment of Description Logics.

2.3 Translation of OWL in Prolog

Choosing the DLP fragment imposes some restrictions on OWL DL in order to guarantee that all axioms stated are interpretable with Prolog. In general, the sub-set of OWL that can be evaluated with Prolog includes the whole of the OWL fragment of RDF Schema, the equality of classes, properties and individuals of OWL, all property characteristics, cardinality constraints with minimum cardinality 0 and maximum cardinality, range restrictions on properties, conjunction of classes and construction of classes by enumeration. The result is necessarily less expressive than description logics. However, as shown in [Volz, 2004] very few available ontologies use constructs outside the DLP language fragment.

3 SweetProlog Architecture

SweetProlog enables reasoning over OWL ontologies by rules via a translation of OWL subsets into simple Prolog



Figure 1: Rules on top of Ontologies

predicates, which a SWI-Prolog engine can understand and process. SweetProlog is implemented in Java and makes use of different components (see Figure 2): JPL¹, a Java interface to Prolog, for providing a *bridge* between Java and SWI-Prolog; a set of inference rules to translate OWL into Prolog; IBM Courteous Compiler² to resolve conflicts between rules using partially prioritized information and Jena³ to parse OWL ontologies. SweetProlog extracts RDF triples

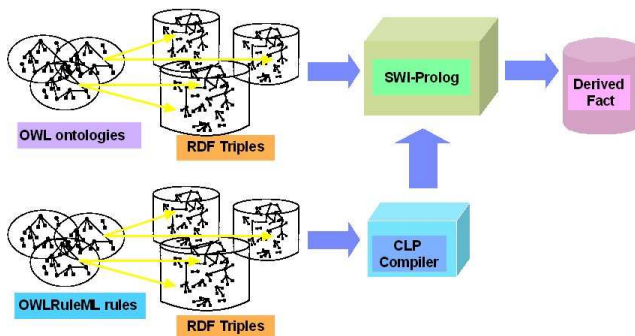


Figure 2: Architecture overview

by parsing OWL ontologies and OWLRuleML rules. The RDF triples that represent OWL concepts and instances are translated into Prolog predicates via a set of inference rules to translate OWL into Prolog. The RDF triples that represent the rules are translated into Courteous Logic Program rules. Such rules are transformed by a Courteous Compiler into Prolog rules. Finally, the translated predicates are then fed into the working memory of a SWI-Prolog engine to infer new

knowledge. SweetProlog also includes an editor that permits the interactive editing of OWLRuleML rules.

4 Conclusions

In this poster we have presented a prototype application that attempts to provide semantic, syntactic and inferential inter-operation between OWL ontologies and RuleML rules. The main strength of the approach is that it provides not only reasoning support for OWLRuleML rules, via a OWL *meta-ontology* for RuleML syntax, but also reasoning support for OWL ontologies combined with OWLRuleML rules. This approach develops Grosf and Horrocks's idea of specifying RuleML rules on top of OWL ontologies, suggesting a mapping of a Description Logic subset (OWL) into Logic Programs (Prolog). The intersection of DL with LP, called Description Logic Programs, covers RDF Schema and a significant fragment of OWL.

5 Acknowledgement

The research presented in this paper is funded by PIPS (FP6-IST No 507019)

References

- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*. 284(5):3443, May, 2001.
- [Boley *et al.*, 2001] H. Boley, S. Tabet, and G. Wagner. Design rationale of RuleML: A markup language for semantic web rules. In *International Semantic Web Working Symposium (SWWS)*, 2001.
- [Grosf, 1997] B. Grosf. Courteous logic programs: Prioritized conflict handling for rules. Technical report, *IBM Research Report RC 20836*, Dec. 30 1997, revised from May 8 1997.
- [Grosf, 2003] B. Grosf. SweetDeal: Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions. In *Proceedings of the twelfth international conference on World Wide Web*, Budapest, Hungary, 2003.
- [Grosf *et al.*, 2002] B. Grosf, M. D. Gandhe and T. W. Finin. SweetJess: Translating DAMLRuleML to JESS. In *Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, Sardinia, Italy, 2002.
- [Grosf *et al.*, 2003] Benjamin N. Grosf, Ian Horrocks, Raphael Volz, and Stefan Decker. Description Logic Programs: Combining Logic Programs with Description Logic. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003)*, pages 48–57. ACM, 2003.
- [Horrocks *et al.*, 2004] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosf and M. Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission, 21 May 2004.
- [Volz, 2004] R. Volz. Web Ontology Reasoning with Logic Databases. *PhD thesis*, AIFB, University of Karlsruhe, 2004.

¹<http://www.swi-prolog.org/packages/jpl>

²<http://alphaworks.ibm.com/tech/commonrules>

³<http://jena.sourceforge.net>