A RuleML Syntax for Answer-Set Programming*

Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits

Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
{eiter, ianni, roman, tompits}@kr.tuwien.ac.at

1 Introduction

The need for sharing knowledge on the Web—and in particular rules in a standardized format—is an important issue. RuleML [1] has been the most prominent effort in this direction so far. Nonetheless, it turns out that none of the variants available is suitable for expressing nonmonotonic logic programs as used in the *answer-set programming* (ASP) paradigm, in which the former are assigned with a declarative semantics, known as *answer-set semantics* or *stable-model semantics* [3]. ASP in general is an important declarative problem-solving paradigm, gaining increasing attention in the recent years (see, e.g., [4]).

In this work, we present a new language variant in addition to the current RuleML proposal for expressing an ASP core language. Then we provide an extension to this core to accommodate different ASP dialects, substantially based on the notion of an *oracle atom*. Oracle atoms are rooted in the notion of an *external atom* [2]. A working translator from RuleML to ASP and vice versa is available. This way, RuleML specifications are made executable under the ASP semantics.

The framework we present here is supposed to be a starting point that should encourage both the Semantic Web and the ASP community to discuss and achieve a comprehensive RuleML interchange format for the ASP semantics. It is work in progress which, as we believe, will attract other participants after initial dissemination.

2 Description of the Work

Our approach towards extending RuleML to answer-set programs consists of several layers. First, we define a RuleML schema called *aspbase*, which encapsulates the syntax of traditional ASP. We then present an extension (*asporacle*) to this schema, facilitating the expression of a number of advanced constructs which are provided by current ASP solvers, such as aggregates, built-ins or external atoms, by a general syntactical element.

Figure 1 shows how an ASP specification written in RuleML can be processed. Translators rewrite an answer-set program in the general RuleML syntax into a textual representation suitable for a given reasoner. Each translator might accept a set of specific syntactic features. It enforces a specific meaning to each particular feature by rewriting it into the construct expected by the corresponding reasoner.

* This work was partially supported by the Austrian Science Fund (FWF) under grant P17212-N04, and by the European Commission through the IST Networks of Excellence REWERSE (IST-2003-506779).

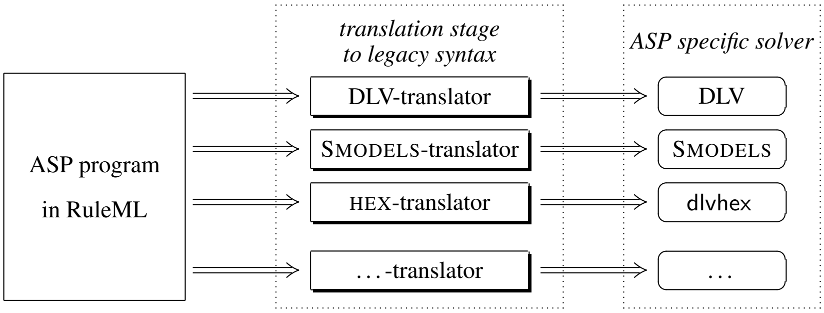

Fig. 1. ASP RuleML architecture

3 An Example

A complex construct like a *cardinality constraint*, featured by SMODELS, such as $1\ \{a, b, not\ c\}\ 2$. can be modeled in our setting in the following way:

```
<Oracle>
  <Rel>cardCons</Rel>
  <Input>
    <Data xsi:type="xs:integer">1</Data>
    <Atom>
      <Rel>a</Rel>
    </Atom>
    <Atom>
      <Rel>b</Rel>
    </Atom>
    <Naf>
      <Atom>
        <Rel>c</Rel>
      </Atom>
    </Naf>
    <Data xsi:type="xs:integer">2</Data>
  </Input>
</Oracle>
```

For further examples of this language variant and the use of oracle atoms, we refer the reader to `http://www.kr.tuwien.ac.at/research/ruleml`.

References

1. H. Boley, S. Tabet, and G. Wagner. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. In *Proc. SWWS 2001*, pages 381–401, 2001.
2. T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits. A Uniform Integration of Higher-Order Reasoning and External Evaluations in Answer Set Programming. In *Proc. IJCAI 2005*, pp. 90–96.
3. M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9:365–385, 1991.
4. S. Woltran (ed.). Answer Set Programming: Model Applications and Proofs-of-Concept. WASP, 2005. `http://www.kr.tuwien.ac.at/projects/WASP/report.html`.