



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	The WSML Rule Languages for the Semantic Web
Author(s)	Polleres, Axel; Fensel, Dieter
Publication Date	2005
Publication Information	Jos de Bruijn, Holger Lausen, Axel Polleres, Dieter Fensel "The WSML Rule Languages for the Semantic Web", Proceedings of the W3C Workshop on Rule Languages for Interoperability, 2005.
Item record	http://hdl.handle.net/10379/630

Downloaded 2024-04-26T02:47:48Z

Some rights reserved. For more information, please see the item record link above.



THE WSML RULE LANGUAGES FOR THE SEMANTIC WEB

Position Paper for the W3C rules workshop, Washington DC, USA, April 2005

Authors:

Jos de Bruijn
Holger Lausen
Axel Polleres
Dieter Fensel

Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
University of Innsbruck, Austria
Email: {jos.debruijn, holger.lausen, axel.polleres}@deri.org

ABSTRACT

The Web Service Modeling Language WSML provides a framework for the modeling of ontologies and semantic Web services based on the conceptual model of the Web Service Modeling Ontology. In this paper we describe the two rule-based WSML-variants and outline our position with respect to a rule language for the Semantic Web. The first rule-based WSML variant, WSML-Flight, semantically corresponds to the Datalog fragment of F-Logic, extended with inequality in the body and locally stratified negation under the Perfect model semantics. The second, WSML-Rule, is an extension of WSML-Flight to the logic programming subset of F-Logic which allows the use of function symbols and unsafe rules (i.e., there may be variables in rule heads which do not occur in the body).

INTRODUCTION

The Web Service Modeling Language (WSML) [de Bruijn et al., 2005] provides an overall framework for different logical languages. The main language paradigms supported in WSML are Description Logics and Logic Programming. The language WSML-Core marks the intersection of both paradigms, based on DLP [Grosz et al., 2003]. WSML-DL marks the Description Logic extension; WSML-Flight and WSML-Rule mark the rule extensions in the spirit of Logic Programming. In this paper, we discuss WSML-Flight and WSML-Rule as rule languages for the Semantic Web.

WSML-Flight and WSML-Rule are rule languages based on the Datalog and Logic Programming fragments of F-Logic, respectively, extended with inequality and default negation in rule bodies under the Perfect Model semantics. [Przymusiński, 1989]. There exist efficient implementations for these paradigms. This allows us to use existing tools and reasoners to reason with WSML-Flight and WSML-Rule.

In the remainder of this paper, we briefly introduce WSML-Flight and WSML-Rule. We describe the relation between WSML and other initiatives for rule languages for the Web. Finally, we summarize our positions with respect to rule languages for the Semantic Web.

WSML

WSML distinguishes between a conceptual and a logical expression syntax. The conceptual syntax is used for the modeling of web services, goals, mediators and ontologies. The logical expression syntax is used for the specification of axioms and constraints in an ontology and inside the pre- and post-conditions of goals and web services.

WSML-Flight and WSML-Rule both provide a conceptual syntax, but we will restrict the discussion to the logical expression syntax in this paper, since rules are defined as logical expressions in WSML. Basically, the logical expression syntax has its foundations in F-Logic [Kifer et al., 1995], but uses slightly different language keywords.

Every value in a WSML document is either a data value of a built-in datatype or an IRI (Internationalized Resource Identifier) [Duerst & Suignard, 2005]. Datatype wrappers are used to structure data values in WSML. Datatype support in WSML is based on XML Schema, but we allow more powerful constructs to reflect the structure of data values. For example, the date "March 15th, 2005" is represented as: `_date(2005,3,15)`. IRIs, being a special and often used datatype in WSML, use a shorter notation: IRIs are delimited with `'_'` and `''`. IRIs can also be abbreviated to sQNames (i.e., serialized QNames) using namespace prefixes. A document may have a default namespace, in which case the prefix may be omitted. Assuming the prefix 'ex' stands for 'http://example.org/', the sQName `ex#person` is an abbreviation for `_'http://example.org/person'`. Variables may be any alphanumeric string preceded by a question mark, e.g., `?x`, `?y`, `?myVariable`.

A WSML rule has the form:

```
head :- body.
```

WSML allows database-style constraints of the form:

```
!- body.
```

An atomic formula in WSML may be one of the following (with A_1, \dots, A_n denote data values, variables or identifiers, or, in the case of WSML-Rule, constructed terms, and p an identifier): (1) $p(A_1, \dots, A_n)$; (2) $A_1[A_2 \text{ op } A_3]$ with $op \in \{ \text{hasValue, ofType, impliesType} \}$; (3) $A_1 \text{ isa } A_2$ with $\text{isa} \in \{ \text{subConceptOf, memberOf} \}$.

The body of the rule may be any atomic formula or a combination of atomic formulae using the connectives **and**, **or** and **naf**. The head of the rule may be any atomic formula or a combination of atomic formulae using the connectives **and**, **implies** and **impliedBy**. The keywords **and** and **or** stand for conjunction and disjunction, respectively; **implies** and **impliedBy** denote the two directions (right and left, respectively) of classical implication; **naf** (negation-as-failure) stands for default negation.

By allowing classical implication in the head of rules, we enable layering WSML-Flight on top of WSML-Core. WSML-Core only has classical implication. Every WSML-Core logical expression corresponds to a WSML-Flight rule with an empty body.

Examples A common example for rules is a rule which models that the brother of your parent is your uncle. The statements in brackets are attribute value specifications. `a[b hasValue c]` intuitively means that object a has an attribute b with value c .

Example 1

```
?x[uncle hasValue ?z] :- ?x[father hasValue ?y] and
    ?y[brother hasValue ?z].
```

The following rule illustrates the use of signature molecules in rule bodies. The symbol **naf** is used for default negation (negation-as-failure); a statement of the form $a \text{ subConceptOf } b$ means that concept a is a subconcept of concept b .

Example 2

```
?x[directSuperConcept hasValue ?y] :- ?x subConceptOf ?y and
    naf ?x[indirectSuperConcept hasValue ?y] and ?x != ?y.
?x[indirectSuperConcept hasValue ?z] :- ?x subConceptOf ?y and
    ?y subConceptOf ?z and ?x != ?y and ?y != ?z.
```

This rule enumerates all direct and all indirect superconcepts for each concept. Actually, the attribute 'indirectSuperConcept' is used here to enable the definition of direct superconcepts. The rule also shows an application of default negation which ensures that for every superconcept for which it is not known to be an indirect superconcept, is a direct superconcept. The second rule ensures that every superconcept which is an indirect superconcept is known to be an indirect superconcept. The example shows an application of meta-modeling where a concept has an attribute value.

Syntaxes for WSML Besides the mentioned normative human-readable syntax, WSML has an XML syntax for exchange between machines and an RDF syntax for exchange over the Semantic Web.

XML

WSML has an XML syntax associated with it. The XML syntax resembles the human-readable syntax, both in keywords and in structure. The XML version of the Example 1, in abbreviated form, looks like:

```
<impliedByLP xmlns="http://www.wsmo.org/wsml/wsml-syntax#">
  <molecule>
    ...
  </molecule>
  <and>
    ...
  </and>
</impliedByLP>
```

RDF

Statements with a complex structure can not be easily captured within the graph data model of RDF, because RDF essentially only allows representation of binary relations. Rules are in general very complex structures and do not allow for straightforward encoding in RDF. Furthermore, problems might occur when formulas should be interpreted according to their context (e.g. web service preconditions; cf. [1]), because all triples in RDF are global and do not have a context. We have not directly defined an RDF syntax for the WSML logical expression syntax. Instead, when using the RDF encoding for WSML, we use literals of the type `rdf:XMLLiteral` for the encoding of rules:

```
"<impliedByLP>...</impliedByLP>"^^rdf:XMLLiteral
```

RELATION WITH OTHER EFFORTS

SWSL SWSL is another recent proposal for a language for describing semantic Web services. SWSL has two parts, a rules language and a process ontology. The SWSL-Rules sublanguage is closely related to WSML-Flight and WSML-Rule. Both languages are largely based on F-logic and they mostly share the logical expression syntax. However, the two groups have pursued complimentary goals. WSML is focused on the end user and developed a "conceptual syntax" for top-level descriptions of services, which we believe, might make the specifications easier to read. WSML also paid special attention to the issue of OWL compatibility. To this end, WSML defines WSML-Core as a subset of both Description Logics and Logic Programming, which serves as a common ground for ontology interoperability. In contrast, SWSL's focus was on extending the functionality of their rule-based language. In particular, SWSL-Rules supports meta-reasoning with its HiLog and reification extensions. It also supports prioritized defaults and classical negation by incorporating Courteous Logic Programming.

RuleML RuleML is an effort to develop a rule language for the Web. The syntax for RuleML is based on XML, although there exists an RDF syntax. RuleML provides different modules with different features for the rule language. For example, there is the Datalog module and also the naf (negation-as-failure) extension module which enables incorporating negation-as-failure in the document. Recently, the FOL (First-Order Logic) version of RuleML has been released. This version is an alternate syntax for First-Order Logic, based on rules. The XML syntax for WSML is partly inspired by RuleML. We do not rule out further alignment between the RuleML and WSML initiatives in the future.

SWRL SWRL [[Horrocks et al., 2004](#)] is a recent proposal for a rule language for the Semantic Web, layered on top of the DL species of OWL. While a subset of SWRL falls inside Logic Programming, a SWRL knowledge base easily goes beyond this fragment, because of the possibility to include arbitrary OWL DL, which includes disjunction in the head and existentially quantified variables. Thus, SWRL does not allow for efficient implementation in current rule engines. Another major difference between SWRL and WSML is that SWRL is based on First-Order logic and is thus monotonic. WSML-Rule allows for nonmonotonic negation. SWRL itself poses a number of syntactic restrictions on the language. It is not possible to use predicates with an arity higher than 2 and function symbols are disallowed. These restrictions were not motivated by computational properties of the language; the satisfiability problem in SWRL is undecidable.

SUMMARY OF POSITION

Here follows our actual position with respect to a rule language for the Semantic Web:

Allow exchange with existing rules implementations

The Logic Programming and Deductive Database communities have developed many tools which are currently in use. A rule language for the Semantic Web should allow reuse of existing systems and existing techniques which have been developed in this area. Many systems implement default negation and therefore, we believe it is an essential component of any rule language.

Relation with a Description Logic language can be done through a common subset

Instead of directly extending a Description Logic, as SWRL [[Horrocks et al., 2004](#)] does, or defining an interface between the two paradigms, as in [[Eiter et al., 2004](#)], we propose to use a common subset, depicted by DLP [[Grosz et al., 2003](#)]. In this way, existing tools can still be used, and no special-purpose tools need to be built to allow interaction between the DL ontology and the rule base.

Different layers of expressiveness

It is useful to distinguish different layers of expressiveness in the rule language to suit different application domains. Some domains might require decidable reasoning; this can be ensured through a restriction to Datalog. Other applications might require greater expressiveness, in which case full Logic Programming (i.e. with function symbols and unsafe rules) could be used.

Human-readable syntax for understandability

In order to improve understandability of the language and in order to improve uptake of the language, the normative syntax for the language should be understandable for humans. Alternative syntaxes, such as XML and RDF, could be based on the human-readable syntax, in the same way the OWL DL RDF syntax is based on the OWL DL abstract syntax [[Patel-Schneider et al., 2004](#)].

Frame-based syntax

Frame Logic [Kifer et al., 1995] allows the use of frames in logical statements. This allows the user to work directly on the level of concepts, attributes, instances and attribute values. Furthermore, variables are allowed in place of concept and attribute identifiers, which enables meta-modeling and reasoning over the signature.

REFERENCES

- [de Bruijn et al., 2005] J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, D. Fensel. *The Web Service Modeling Language WSML*. WSML Deliverable D16.l/v0.2, 2005. <http://www.wsmo.org/TR/d16/d16.l/v0.2/>
- [Duerst & Suignard, 2005] M. Duerst and M. Suignard. *Internationalized Resource Identifiers (IRIs)*. IETF RFC3987. <http://www.ietf.org/rfc/rfc3987.txt>
- [Eiter et al., 2004] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *Proc. of the International Conference of Knowledge Representation and Reasoning (KR04)*, 2004.
- [Franconi and Tessaris, 2004] E. Franconi and S. Tessaris. Rules and queries with ontologies: a unified logical framework. In *Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'04)*, St. Malo, France, 2004.
- [Grosf et al., 2003] B. N. Grosf, I. Horrocks, R. Volz, and S. 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. Available from <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.
- [Kifer et al., 1995] M. Kifer, G. Lausen, and J. Wu: Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741-843, July 1995.
- [Patel-Schneider et al., 2004] P. F. Patel-Schneider, P. Hayes, and I. Horrocks: *OWL web ontology language semantics and abstract syntax*. Recommendation 10 February 2004, W3C, 2004. Available from <http://www.w3.org/TR/owl-semantics/>.
- [Przymusinski, 1989] T. C. Przymusinski. On the declarative and procedural semantics of logic programs. *Journal of Automated Reasoning*, 5(2):167-205, 1989.
- [Roman et al., 2004] D. Roman, H. Lausen, and U. Keller (eds.): *Web Service Modeling Ontology - Standard (WSMO - Standard)*, WSMO deliverable D2 version 1.1. available from <http://www.wsmo.org/TR/d2/v1.1/>.

FOOTNOTES

[1] See also the thread "Representation of SWRL expressions in OWL-S" on the public-sws-ig mailing list: <http://lists.w3.org/Archives/Public/public-sws-ig/2005Feb/0011.html>



\$Date: 2005/04/14 14:21:42 \$