



WSML Deliverable  
D34 v0.1  
THE WEB COMPLIANCE OF  
WSML

WSML Final Draft – April 28, 2007

**Authors:**

Jos de Bruijn  
Dieter Fensel  
Holger Lausen

**Editors:**

**Reviewers:**

**Final version:**

<http://www.wsmo.org/TR/d34/v0.1/20070328/>

**Latest version:**

<http://www.wsmo.org/TR/d34/v0.1/>



# Abstract

This document describes the usage of Web standards by the Web Service Modeling Language (WSML) family. We show how WSML is aligned with the current and future (semantic) Web language standards.



# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>The Semantic Web Language Layer Cake and Its Relation with WSML</b>	<b>7</b>
<b>3</b>	<b>Plain Web Recommendations</b>	<b>11</b>
3.1	Unicode . . . . .	11
3.2	URI . . . . .	11
3.3	Namespaces . . . . .	11
3.4	XML . . . . .	12
<b>4</b>	<b>Semantic Web Language Recommendations</b>	<b>13</b>
4.1	RDF and WSML . . . . .	13
4.2	RDFS and WSML Core . . . . .	13
4.3	OWL DLP and WSML Core . . . . .	14
4.4	OWL DL and WSML DL . . . . .	14
4.5	RIF and WSML Rule and Flight . . . . .	15
4.6	Logic Framework and WSML Full . . . . .	16
<b>5</b>	<b>Web Service Recommendation SAWSDL</b>	<b>17</b>
<b>6</b>	<b>Conclusions</b>	<b>18</b>



# 1 Introduction

The Web Service Modeling Language WSML<sup>1</sup> [8] provides a formal syntax and semantics for the Web Service Modeling Ontology WSMO<sup>2</sup> [9], a framework for modeling semantic description of Web services. WSML is based on different logical formalisms, namely, Description Logics [3], First-Order Logic [10] and Logic Programming [12], which are useful for the modeling of semantic Web services. WSML consists of a number of variants based on these different logical formalisms, namely WSML-Core, WSML-DL, WSML-Flight, WSML-Rule and WSML-Full.

**WSML-Core** corresponds to an intersection [11] of Description Logic and Horn Logic (without function symbols and without equality), extended with datatype support in order to be useful in practical applications. WSML-Core is fully compliant with a subset of OWL.

WSML-Core is extended, both in the direction of Description Logics and in the direction of Logic Programming, to WSML-DL and WSML-Rule, respectively.

**WSML-DL** extends WSML-Core to an expressive Description Logic, namely, *SHIQ*, thereby covering that part of OWL which is efficiently implementable.

**WSML-Flight** extends WSML-Core in the direction of Logic Programming. WSML-Flight has a rich set of modeling primitives for modeling different aspects of attributes, such as value constraints and integrity constraints. Furthermore, WSML-Flight incorporates a fully-fledged rule language, while still allowing efficient decidable reasoning. To be more precise, WSML-Flight allows writing down any Datalog rule, extended with inequality and (locally) stratified negation.

**WSML-Rule** extends WSML-Flight to a fully-fledged Logic Programming language, including function symbols. WSML-Rule no longer restricts the use of variables in logical expressions.

The final WSML variant unifies the Description Logic and Logic Programming paradigms:

**WSML-Full** unifies all WSML variants under a common First-Order umbrella with non-monotonic extensions which allow capturing non-monotonic negation of WSML-Rule.

All WSML variants are described in terms of a normative human-readable surface syntax. Besides the surface syntax, we provide XML and RDF syntaxes for exchange of WSML descriptions over to (semantic) Web. Furthermore, we provide a mapping between WSML ontologies and OWL for basic

---

<sup>1</sup><http://www.wsmo.org/wsml/>

<sup>2</sup><http://www.wsmo.org/>



inter-operation with OWL ontologies through a common semantic subset of OWL and WSML.

Key for the acceptance and usability of WSML is the proper alignment with existing (and upcoming) Web and Web service standards. WSML aims to use and embed them for the purpose of semantically annotating Web services; however, it should neither replace nor compete with them. Achieving this has been a major objective for WSML; in this paper we explain how this objective has been met in the design of WSML.

In particular, the following standards and W3C<sup>3</sup> recommendations are of particular importance for grounding WSML in existing initiatives:

**UNICODE.** The Unicode Standard<sup>4</sup> is a character coding system designed to support the worldwide interchange, processing, and display of written text, considering the diverse languages and technical disciplines of the modern world.

**URIs.** Uniform Resource Identifiers<sup>5</sup> (URIs, aka URLs) are short strings that identify resources in the Web: documents, images, downloadable files, services, electronic mailboxes, and other resources. They make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way.

**Namespaces.** Namespaces were first introduced in XML [5] for the purpose of qualifying names which originate from different XML documents. One motivation for namespaces is modularity: if a markup vocabulary exists which is well-understood and for which there is useful software available, it is better to re-use this markup rather than re-invent it. However this poses problems of recognition and collision. Software modules need to be able to recognize the elements and attributes which they are designed to process. These considerations require that document constructs should have names constructed so as to avoid clashes between names from different markup vocabularies. XML namespaces accomplish this by separating names into two parts: a namespace and a local name.

**XML and XML schema.** The Extensible Markup Language (XML)<sup>6</sup> is a text format derived from SGML (ISO 8879). XML Schema<sup>7</sup> provides a means for defining the structure, content and semantics of XML documents.

**RDF and RDF schema.** The Resource Description Framework (RDF)<sup>8</sup> is a general-purpose language for representing information on the Web. RDF schema defines a lightweight ontology language for RDF.

**OWL.** OWL<sup>9</sup> is an expressive Web ontology language which uses both URIs and RDF. Its further design is heavily based on Description Logics.

---

<sup>3</sup><http://www.w3.org/>

<sup>4</sup><http://www.unicode.org/unicode/standard/standard.html>

<sup>5</sup><http://www.w3.org/Addressing/>

<sup>6</sup><http://www.w3.org/XML/>

<sup>7</sup><http://www.w3.org/XML/Schema>

<sup>8</sup><http://www.w3.org/RDF/>

<sup>9</sup><http://www.w3.org/2004/OWL/>



**RIF.** The Rule Interchange Format (RIF) Working Group<sup>10</sup> is chartered to produce a core rule language plus extensions which together allow rules to be translated between rule languages and thus transferred between rule systems.

**SAWSDL.** The objective of the Semantic Annotations for WSDL (SAWSDL) Working Group<sup>11</sup> is to develop a mechanism which enables the semantic annotation of Web service descriptions. This mechanism will use the WSDL 2.0<sup>12</sup> extension mechanisms to create a simple and generic support mechanism for adding semantics to Web service descriptions.

In Section 2, we show the relation between the framework of WSML language variants and the semantic Web languages stack of W3C. In Section 3, we describe the usage of the Web language standards UNICODE, URI, and XML in WSML. Section 4 describes the usage in WSML of semantic Web recommendations such as RDF, RDFS, the DLP fragment of OWL, OWL, and the upcoming recommendation RIF, as well as an expected future logic framework. Section 5 adds the grounding in Web service standards through the usage of SAWSDL. Conclusions are provided in Section 6.

---

<sup>10</sup><http://www.w3.org/2005/rules/wg/charter>

<sup>11</sup><http://www.w3.org/2002/ws/sawSDL/>

<sup>12</sup><http://www.w3.org/TR/wsdl20/>



## 2 The Semantic Web Language Layer Cake and Its Relation with WSML

The recently updated W3C layer cake of Web and semantic Web languages is shown in Figure 2.1<sup>1</sup>. It is based on the principle that new and enriched Web recommendation should reuse existing specifications as much as possible, to maximize interchange of information.

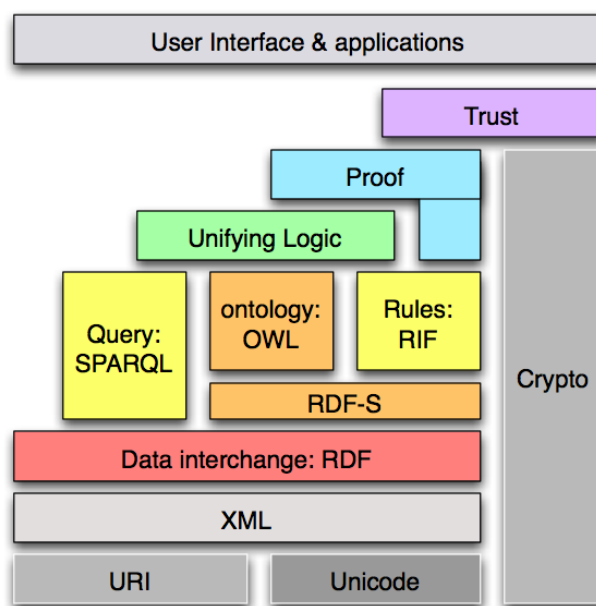


Figure 2.1: Semantic Web Language Layering

WSML intends to maximally reuse these W3C recommendations and their underlying principles, deploying them for the description of Web services. The overall design of WSML is depicted in figure 2.2. The variants differ in the logical expressiveness they offer and in the underlying language paradigm. By offering these variants, we allow users to make the trade-off between the provided expressivity and the implied complexity on a per-application basis. As can be seen from the figure, the basic language WSML-Core is extended in two directions, namely, Description Logics (WSML-DL) and Logic Programming (WSML-Flight, WSML-Rule). WSML-Core describes a very efficient language, which is, however, limited in expressive power to the language elements which are expressible both in Description Logics and Rule languages, in order not to destroy the computational advantages of both paradigms. WSML-DL defines an expressive extension of WSML-Core, staying inside the reasoning paradigm of Description Logics. WSML-Flight and WSML-Rule (the latter introducing

<sup>1</sup>The figure was extracted from [4].

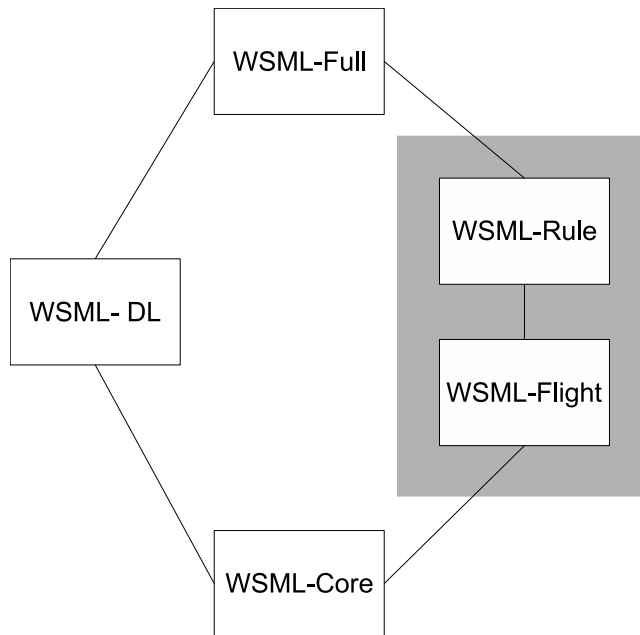


Figure 2.2: The WSML Diamond

function symbols into the language) define extensions of WSML-Core towards more expressive power, however, staying inside the reasoning paradigm of Logic Programming. WSML-Rule and WSML-DL are both extended to a full First-Order Logic with non-monotonic extensions (WSML-Full), which unifies both paradigms. Here, full interoperability between both paradigms is achieved; however, inferencing is highly impractical and undecidable, even for very limited reasoning tasks. Therefore, WSML (and also the Web language layer cake) offers two alternative paths for integrating description logics and rules: an intersection (WSML-Core) which is highly scalable (and has very limited expressiveness); and a union which is highly expressive (and does not scale).

The key question now is: how are both language layerings (the W3C layer cake and the WSML diamond)) related to each other? Spoken in a nutshell, the answer to this question consists of three parts:

1. WSML uses the underlying Web language stack, i.e., it is layered on what is presented in Figure 2.3.
2. WSML can be used by upcoming Web solutions for querying, proof, trust, signature and encryption (see Figure 2.4).
3. WSML mirrors the four core languages of the semantic Web and enriches them with syntactical constructs geared towards describing semantic Web service. Hereby neither the rules part nor the logic framework of the semantic Web is already developed by W3C. Therefore, the WSML working group had to make informed guesses and predictions.

In conclusion, it is hard to imagine a tighter coupling of semantic Web representation formalisms with existing semantic Web standards or the ones

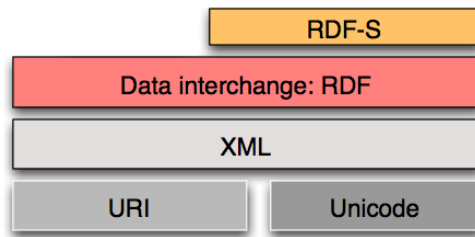


Figure 2.3: The Base Layers of the Web WSML is built upon

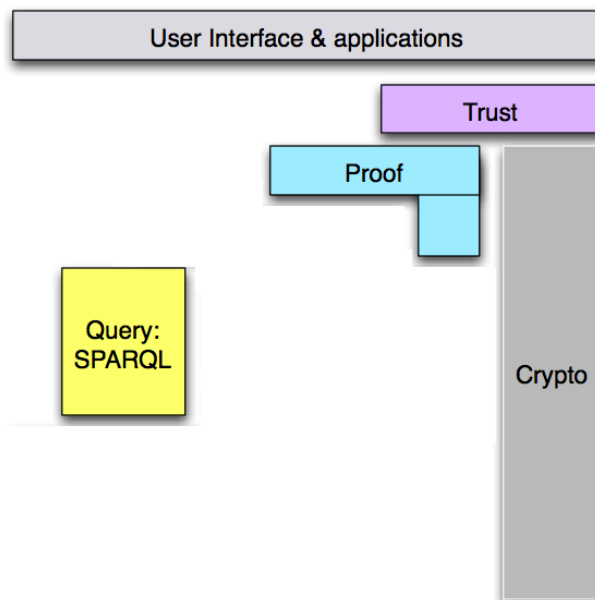


Figure 2.4: The Web Elements beyond WSML Which Can Make Usage of WSML

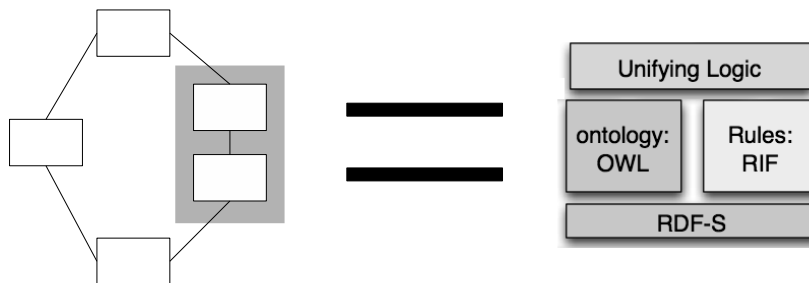


Figure 2.5: The Semantic Web Elements Corresponding to WSML



that are under development. Other approaches such as OWL-S<sup>2</sup>, which only consider OWL as a representation language, miss all the other highly relevant semantic Web language standards that may be need in a semantic Web service context.

More detailed insights on the precise technical couplings are provided in the following sections.

---

<sup>2</sup><http://www.daml.org/services/owl-s/1.0/>



## 3 Plain Web Recommendations

Key for the acceptance and usability of WSML is the proper alignment with existing Web and Web service standards. In this section we describe the relation of WSML with “plain” or “syntactic” standards that are widely used in the context of the Web.

### 3.1 Unicode

The Unicode Standard defines a universal character set and is used, in the surface syntax, by all variants of WSML.

### 3.2 URI

Uniform Resource Identifiers are short strings that identify resources. A resource can be anything and must not necessarily be accessible over the Web. A URI contains a schema and hierarchical path. Examples of common schemas are `http` for hyper text documents and `ftp` for file transfer. The hierarchical path can contain an authority, path, query and fragment, such as in the URI: “`foo://example.com:8042/over/there?name=ferret#nose`”. URIs may contain all kinds of information from all kinds of protocols or formats that use characters beyond the set of ASCII characters. The URI syntax defined in RFC 2396 only allows a subset of the ASCII character set, consisting of about 60 characters. Arbitrary bytes can be encoded using URI characters: a byte is encoded using `%` followed by two hexadecimal digits representing the byte (`%HH`-escaping). Since URIs are not only interpreted by computers but also humans, such an encoding is a limitation for usability. Internationalized Resource Identifiers (IRIs) are the successors of URIs which allow the use of arbitrary Unicode characters to be used in the identifier, thereby catering to international languages beyond English. An IRI is a sequence of characters from the Universal Character Set. There is a mapping from IRIs to URIs, which means that IRIs can be used instead of URIs where appropriate to identify resources.

WSML uses IRIs to identify resources and is thus fully compliant with the latest Web standards for the identification of resources.

### 3.3 Namespaces

In XML, each qualified name consists of a tuple `<namespace, localname>`. RDF adopts the mechanism of namespaces from XML with the difference that qualified names are not treated as tuples, but rather as abbreviations for full URIs.



WSML adopts the namespace mechanism of RDF.

## 3.4 XML

The Extensible Markup Language (XML) is a simple, flexible semi-structured text format. Its primary purpose is to facilitate the sharing of data across different information systems, particularly systems connected via the Internet. XML provides a text-based means to represent information using a tree-based structure. At its base level, all information manifests as text, interspersed with markup that indicates a separation of information into a hierarchy of character data, container-like elements, and attributes of those elements. Due to its widespread adoption, XML facilitates the exchange of data, since parsers and other related software are widely available. However, there is also some criticism. Despite the fact that XML was designed to be reasonably human-legible, it is sometimes considered to be too verbose and redundant.

WSML has defined a normative human-readable surface syntax to specify ontologies, mediators, Web services, and goals. The surface syntax provides a concise notation which aims to facilitate the comprehension by the human modeler and exchange specification without tool support. Acknowledging the importance of XML as exchange syntax within the Web WSML has defined an XML syntax, called WSML/XML. Using the XML syntax one can express everything that can be expressed using the human readable syntax and vice versa.

From the early days of WSML, tools have supported the serialization and parsing of WSML/XML. The reference implementation of the WSMO API<sup>1</sup> is available under LGPL license and provides convenient methods for converting between the syntaxes as required.

---

<sup>1</sup><http://wsmo4j.sourceforge.net/>



## 4 Semantic Web Language Recommendations

Several language recommendations for representation and exchange of knowledge on the semantic Web have been produced by W3C. At the core this is RDF, which defines a graph-based model for representing information, and exchanging this using an XML syntax. RDF Schema (RDFS) defines constructs that allow the specification of lightweight ontologies; RDFS allows to define classes, properties as well as class and property hierarchies. The Web Ontology Language (OWL) and the Rule Interchange Format (RIF) define additional layers of expressivity on top of RDF(S).

### 4.1 RDF and WSML

Similar to XML, RDF is designed to allow information exchange between computers. However, in contrast to XML, which has a tree-based structure and whose atomic exchange unit are documents, RDF is has a graph-based structure, and its basic building block are statements (triples).

Recognizing the importance of RDF as exchange syntax in the context of the semantic Web, WSML has specified an RDF syntax for WSML. The RDF syntax is a fully fledged syntax which allows expressing all constructs of WSML.

The round-tripping between WSML and its RDF syntax has been implemented as part of The Ontology Representation and Data Integration (ORDI) framework<sup>1</sup>. ORDI was developed by Ontotext Semantic Technology Lab; it uses the Sesame<sup>2</sup> RDF Store to persistently store WSML descriptions. furthermore, it allows in-memory as well as file-based conversion between the WSML surface and RDF syntaxes.

### 4.2 RDFS and WSML Core

A direct translation approach to RDF and back is based on the assumption that WSML must be represented completely and accurately in RDF. In this section we relax this assumption. The relation described here is an attempt to transform from WSML to RDF and vice versa only those parts that are expressible in RDFS, trying to capture the intention of a WSML ontology, not all its properties, and viewing RDFS as a restricted but viable ontological language with concepts, relationships, and domain and range restrictions.

This mapping is specified in [7]. It has been implemented as part of the reference implementation of the WSMO-API. Additionally the API allows importing

<sup>1</sup><http://www.ontotext.com/ordi/>

<sup>2</sup><http://www.openrdf.org/>



arbitrary RDF(S) documents; trying to convert them as close to the intention they have been originally written. For example, the import mechanism allows to decide whether RDF properties are converted to relations or attributes. While attributes are usually closer to the intention of the original modeler, they result in difficult to comprehend ontologies in the case extensive property hierarchies where used or many property domains where left unspecified in the original ontology.

### 4.3 OWL DLP and WSML Core

Description Logics Programs (DLP) [11] correspond to a Horn subset of the expressive description logic *SHIQ*, which means that standard Logic Programming and deductive database technology can be used for efficient query answering when using this fragment. Additionally, it can be used as the basis for a rules extension. Since OWL DL is essentially an expressive Description Logic language, DLP can also be seen as a subset of OWL DL; this subset is usually called OWL DLP<sup>3</sup>, which includes the expressiveness of the RDF Schema box in Figure 2.1.

There is a one-to-one mapping between OWL DLP and WSML-Core ontologies. Every OWL DLP ontology can be equivalently translated to a WSML-Core ontology, and vice versa. This means that OWL DLP ontologies can be imported in WSML and used for Web service description. Likewise, WSML-Core ontologies can be imported in the OWL world as OWL DLP ontologies.

Additionally, WSML provides an extension mechanism of OWL DLP with rules, in the form of WSML-Rule: an OWL DLP ontology can be translated to a WSML-Core ontology, which can then be extended with rules using WSML-Flight or WSML-Rule.

### 4.4 OWL DL and WSML DL

There is a close relationship between WSML and OWL DL. In fact, nearly all of OWL DL can be represented as ontologies in WSML-DL. This is because the design principles behind the two language variants are similar. The formal language underlying OWL DL is the Description Logic *SHOIN*, while the language underlining WSML-DL is the Description Logic *SHIQ*. The only essential difference is that WSML-DL does not include nominals (i.e. individuals in class descriptions).

Besides using WSML as an extension mechanism of (fragments of) OWL ontologies, it can also be used for the description of Web services and goals based on OWL ontologies. Viewed from the other way around, WSML Web service descriptions and goal descriptions can simply use OWL ontologies, which are more abundantly available on the semantic Web than WSML ontologies, since OWL is a standard.

Finally, one could view OWL as a more Description Logic-centric syntax for

<sup>3</sup>See also: <http://logic.aifb.uni-karlsruhe.de/wiki/DLP>



WSML-DL ontologies. Axioms in WSML-DL ontologies need to be written in a first-order style, making all the variables explicit. Besides being more verbose than a Description Logic style, the first-order style of DL modeling has a lot of restrictions in the use of the first-order syntax, in order to comply with the target Description Logic, making the syntax generally hard to use. Using OWL, which has a syntax tailored specifically to the Description Logic which underlies it, one could circumvent these limitations in the WSML syntax.

The reference implementation of the WSMO-API includes support for importing OWL-DL documents. The implementation has been donated from the ORDİ project late 2005. In case the imported ontology includes features that are not supported in WSML DL, the ontology will be imported except those definitions in order to be compliant with WSML DL.

In order to export WSML DL to OWL one can use the WSML2Reasoner<sup>4</sup> Framework which includes an implementation of an appropriate serializer.

The current version of WSML (0.21) does not include a mechanism to refer directly to OWL ontologies from goal and Web service descriptions; this mechanism is implicit, because it requires importing OWL (DLP/DL) ontologies into WSML (Core/DL). Future versions of WSML (starting 0.3; expected in the autumn of 2007) will include the possibility to directly use OWL ontologies in Web service and goal descriptions. This mechanism is different from, for example, the mechanism used in OWL-S, where the Web service description itself is part of the ontology. In WSML we decided not to express the meta-ontology (i.e. the ontology describing all the concepts used in ontology and Web service description) in OWL, because this would restrict the language used for Web service description to Description Logics, or extension thereof. Additionally, from Software Engineering we know that it is good practice to separate the language for representing the meta-model level (i.e. the concepts used for ontology and which are description) from the language for representing the model level (i.e. ontologies and Web service descriptions). Instead, WSML relies on WSMO [9], which uses OMG's MetaObject Facility (MOF)<sup>5</sup> for the description of the conceptual model underlying ontology and Web service description.

## 4.5 RIF and WSML Rule and Flight

With the formation of the Rule Interchange Format (RIF)<sup>6</sup> Working Group in December 2005, the W3C has started a standardization activity around rules for the semantic Web. The WSML working group did not have time to wait for this effort (at the time of writing, the RIF working group has yet to release its first draft of the technical specification), and created a rules language for the semantic Web, and semantic Web services, in the form of WSML Rule, and its decidable subset WSML Flight.

In order to share the experiences in creating a rules language for the semantic Web, the WSML working group has submitted the Web Rule Language (WRL) [1], which is based on the ontology components of WSML Core, Flight,

<sup>4</sup><http://tools.deri.org/wsml2reasoner/>

<sup>5</sup><http://www.omg.org/mof/>

<sup>6</sup><http://www.w3.org/2005/rules/>



and Rule, to W3C. WRL is considered to be an input for the RIF working group. It is currently not possible to tell whether, and to what extent, there will be compatibility between RIF and WSML Rule. We do, however, expect some level of compatibility, since the goals of the languages are similar.

## 4.6 Logic Framework and WSML Full

One of the least developed boxes in the semantic Web language layer cake (Figure 2.1) is the logic framework. The position in the layer cake would suggest that it is a formalism which encompasses both the OWL and Rules components. In WSML, WSML Full provides such a unifying formalism for the Rules and DL variants. Because of the close connection between OWL DL and WSML DL and the potential connection between RIF and WSML Rule, the formalism underlying WSML Full [6] could be seen as such a logical framework for the semantic Web.



## 5 Web Service Recommendation SAWSDL

The Semantic Annotations for WSDL and XML Schema (SAWSDL) specification is being developed by a W3C Working Group. It defines mechanisms with which semantic annotations can be added to WSDL components. SAWSDL does not specify a language for representing the semantic models such as ontologies. Instead, it provides mechanisms by which concepts from the semantic models that are defined either within or outside the WSDL document can be referenced from within WSDL components as annotations. These semantics when expressed in formal languages can help disambiguate the description of Web services during automatic discovery and composition of Web services.

Specifically, the SAWSDL specifications defines two kinds of connections to semantic models. The *model reference* is a pointer to a concept within a semantic model, while the *schema mapping* constructs points to a resource that defines a mapping between the XML messages used and the structures used in a semantic annotation.

SAWSDL complements WSML in the sense that it provides a way to ground services described in WSML to Web Services. This grounding can be done from a WSML document (such as in a WSML choreography) but also from the other direction, i.e. from a WSDL document to a WSML description. SAWSDL allows exactly this other way of grounding semantic descriptions.



## 6 Conclusions

When designing the WSML family of languages we had one major principle in mind: maximal compliance with established W3C and OASIS recommendations and standards. WSML is not about competing with existing standards, but rather complementarily extend them with the means required to describe Web services. Therefore, we carefully layered WSML on the relevant Web and Web service standards such as URIs, XML, WSDL, etc. Regarding the semantic Web language recommendations of W3C we were in the situation that the language stack has only been partially developed.

- Therefore, we developed a variant called WSML-Core that basically takes RDFS and extends it with further language constructs that improve the expressiveness without hampering the computational complexity of the language (something which was not done in OWL Lite).
- Second, we developed a language variant called WSML-DL which corresponds to OWL DL (however, we excluded nominals due to their computational complexity).
- Third, we developed WSML-Flight and WSML-Rule as informed guesses, anticipating the upcoming W3C recommendation on rules languages called RIF.
- Finally, with WSML-Full we anticipated a logical framework which is expressive enough to fully cover DL and rule type of representation. Again, we were forced to move faster on this than W3C is developing.

Finally, we had to enrich these languages with means to express goals and Web services, i.e. establishing epistemological concepts such as pre- and post-conditions that are not present in generic semantic Web ontology languages such as OWL.

OWL-S [2], another language for describing Web services, commits to one semantic Web language, namely the Description Logic language OWL DL. This actually limits the overall integration of the semantic Web service description means into the layered language stack of W3C. Furthermore, choosing to use OWL DL as a language for expressing both the conceptual model (ontology) of the Web service description, and large parts of the Web service descriptions themselves<sup>1</sup>, limits the way Web services can be modeled.

Not to mention, OWL was not intended for designing other languages, i.e., OWL-S requires non-standard and ad hoc combination of OWL with other, non-standard, languages which hamper interoperability.

---

<sup>1</sup>The parts of OWL-S Web service descriptions which are expressed using OWL DL are the inputs, outputs and the overall classification of the service.



## Acknowledgments

The work is funded by the European Commission under the projects ASG, EASAIER, enIRaF, Knowledge Web, Musing, Salero, Seemp, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the DERILion Grant No.SFI/02/CE1/I13; by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects Grisino, RW<sup>2</sup>, SemBiz, SeNSE and TSC.

We would like to thank the members of the WSMO and WSML working groups for their advice and input to this document, and especially Michael Brodie for requesting it.



# Bibliography

- [1] Jürgen Angele, Harold Boley, Jos de Bruijn, Dieter Fensel, Pascal Hitzler, Michael Kifer, Reto Krummenacher, Holger Lausen, Axel Polleres, and Rudi Studer. Web rule language (WRL). W3C Member Submission 09 September 2005, 2005.
- [2] Anupriya Ankolekar, Mark Burstein, Grit Denker, Daniel Elenius, Jerry Hobbs, Lalana Kagal, Ora Lassila, Drew McDermott, Deborah McGuinness, Sheila McIlraith, Massimo Paolucci, Bijan Parsia, Terry Payne, Marta Sabou, Craig Schlenoff, Evren Sirin, Monika Solanki, Naveen Srinivasan, Katia Sycara, and Randy Washington. *OWL-S 1.1 Release*. 2004. <http://www.daml.org/services/owl-s/1.1/>.
- [3] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [4] Tim Berners-Lee. Semantic web stack. <http://www.w3.org/DesignIssues/diagrams/sweb-stack/2006a.png>, 2006. Presented in the keynote talk “Artificial Intelligence and the Semantic Web” at the *21st National Conference on Artificial Intelligence*, Boston, USA.
- [5] Tim Bray, Dave Hollander, Andrew Layman, and Richard Tobin, editors. *Namespaces in XML 1.1*. Number W3C Recommendation 4 February 2004. W3C, 2004. <http://www.w3.org/TR/xml-names11/>.
- [6] Jos de Bruijn and Stijn Heymans. A semantic framework for language layering in WSML. In *Proceedings of the First International Conference on Web Reasoning and Rule Systems (RR2007)*, Innsbruck, Austria, June 7–8 2007. Springer.
- [7] Jos de Bruijn, Jacek Kopecký, and Reto Krummenacher. RDF representation of WSML. Deliverable 32v0.1, WSML Working Group, 2006.
- [8] Jos de Bruijn, Holger Lausen, Reto Krummenacher, Axel Polleres, Livia Predoiu, Michael Kifer, and Dieter Fensel. The web service modeling language WSML. WSML Final Draft D16.1v0.21, WSML, 2005.
- [9] Dieter Fensel, Holger Lausen, Axel Polleres, Jos de Bruijn, Michael Stollberg, Dumitru Roman, and John Domingue. *Enabling Semantic Web Services – The Web Service Modeling Ontology*. Springer, 2006.
- [10] M. Fitting. *First Order Logic and Automated Theorem Proving (second edition)*. Springer Verlag, 1996.
- [11] Benjamin N. Groszof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *Proc. Intl. Conf. on the World Wide Web (WWW-2003)*, Budapest, Hungary, 2003.



- [12] John W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer-Verlag, 1987.