SEMANTIC WEB LANGUAGES – STRENGTHS AND WEAKNESS

Sinuhé Arroyo, Rubén Lara, Ying Ding, Michael Stollberg, Dieter Fensel
Universität Innsbruck
Institut für Informatik
Technikerstraße 13
6020 Innsbruck, Austria

{sinuhe.arroyo, ruben-lara, ying.ding, michael.stollberg, dieter.fensel}@uibk.ac.at

ABSTRACT

The current web suffers information overloading: it is increasingly difficult and time consuming to obtain information desired. Ontologies, the key concept behind the Semantic Web, will provide the means to overcome such problem by providing meaning to the available data. An ontology provides a shared and common understanding of a domain and information machine-processable semantics. To make the Semantic Web a reality and lift current Web to its full potential, powerful and expressive languages are required. Such web ontology languages must be able to describe and organize knowledge in the Web in a machine understandable way. However, organizing knowledge requires the facilities of a logical formalism which can deal with temporal, spatial, epistemic, and inferential aspects of knowledge. Implementations of Web ontology languages must provide these inference services, making them much more than just simple data storage and retrieval systems. This paper presents a state of the art for the most relevant Semantic Web Languages: XML, RDF(s), OIL, DAML+OIL, and OWL, together with a detailed comparison based on modeling primitives and language to language characteristics.

KEYWORDS

Ontologies, Semantic Languages, Semantic Web.

1. INTRODUCTION

The current World Wide Web (WWW) is a syntactic web where structure of the content is presented while the content itself is difficult only readable by humans. Although the WWW has resulted in a revolution in information exchange among computer applications it still cannot fulfill the interoperation among various applications without some pre-existing, human-created agreements.

The next generation of the Web aims to alleviate such problem. The Web resources will be much easier and more readily accessible by both humans and computers with the added semantic information in a machine-understandable and machine-processable fashion [Berners-Lee, 1999]. "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [Berners-Lee, et al, 2001].

Ontologies are considered to be a key technology to make the Semantic Web become reality. They play a pivotal role by providing a source of shared and precisely defined terms that can be understood and processed by machines. A typical ontology consists of a hierarchical description of important concepts and their relations in a domain, task or service. The degree of formality employed in capturing these descriptions can vary – ranging from natural language to logical formalisms – but increased formality and regularity clearly facilitates machine understanding. Therefore a powerful ontology language which can help to formalize the web is the most wanted thing in the Semantic Web.

Various requirement of Web ontology languages have been announced. The most desired features are that such a language should be well designed for the intuition of human users without loosing the adequate expressive power; it should be well defined with clear specified syntax and formal semantics; it should be compatible with existing web standards, etc.

In this paper, we intent to have a broad coverage for various existing web ontology languages, starting

The research presented in this paper was funded by European Commission in the context of the Esperonto Services project (http://www.esperonto.net) under contract number IST-2001-34373

from XML(s), and following the line with RDF(s), OIL, DAML+OIL, and OWL.

The survey is organized in four sections: Section 2 gives and description of the capabilities and limitations of language families. Section 3 presents a comparison of modeling primitives, a specific language to language evaluation, the state of the different languages related to the W3C wish list and finally an outline of their strengths and weaknesses. Section 4 sketches the final conclusions

2. SEMANTIC WEB LANGUAGES

The layered tower of Semantic Web Languages shown in Fig. 1 is Semantic is the vehicle dreamed of to bring the Semantic Web to its full potential. The recognition of the importance of ontologies for the Semantic Web has led to the revolution and extension of the current web markup languages surveyed here.

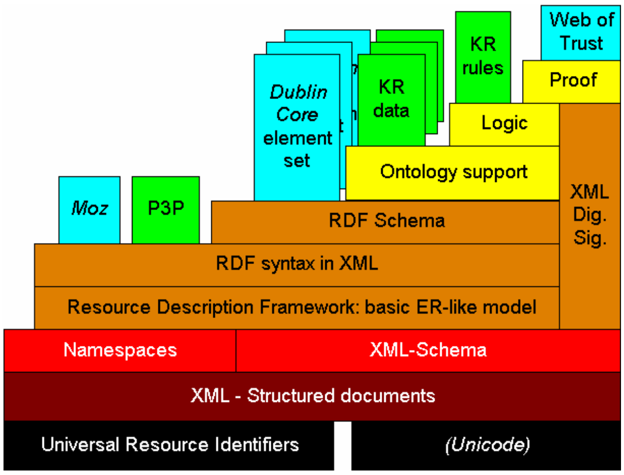

Figure 1. Semantic Web Language cake

The W3C has defined design criteria for Semantic Web Languages, namely: Ontology Sharing and Versioning, Interoperability, Reasoning support, a balance of expressiveness and scalability, Ease of Use and Compatibility with standards, and Internationalization [Heflin, et. al, 2002]. In our comparison we will focus on Interoperability, Reasoning support, Expressiveness and Scalability, and Compatibility with web standards.

2.1 XML

In the beginning of the web, HTML was designed as a easily applicable languages for defining the presentation of information. As the amount of information available on the web increased exponentially, a way for defining the structure of information was desired in order to allow automated processing of web content.

Therefore the eXtendible Markup Language [XML]has been developed. XML is a tag-based language for describing tree structures with a linear syntax. It offers the facilities for users to define their own tags, which are needed for describing the structure of the documents. By this the content of a web document can be processed automatically. As thus XML provides an means for exchanging information over the web it is the basic language for the semantic web.

2.2 RDF

XML provides a language for describing the structure of information but fails to define the semantics in a machine understandable and processable way. The Resource Description Framework [RDF] comes to fill up the hole.

RDF is an XML application (i.e., its syntax is defined in XML) customized for adding meta-information to Web documents. Basically, RDF defines a data model for describing machine processable semantics of data which consists of three object types:

- **Resources**. A resource may be an entire Web page; a part of a Web page; a whole collection of pages; or an object that is not directly accessible via the Web e.g. a printed book. Resources are always named using URIs.
- **Properties**. A property is a specific aspect, characteristic, attribute, or relation used to describe a resource.
- **Statements**. A specific resource together with a named property plus the value of that property for that resource is a RDF statement.

These three individual parts of a statement are called, respectively, the subject, the predicate, and the object. In a nutshell, RDF defines *object-property-value*-triples that represent the semantics of a web resources and introduces a standard syntax for them. As RDF statements are also resources, statements can be recursively applied to statements allowing their nesting. Thus RDF

The RDF modeling primitives are defined in RDF Schema (RDFS). RDFS in particular can be recognized as an ontology language as it defines the semantics for classes and properties, range and domain constraints, and subclass and subproperty relations. However, it is a very limited language and more expressive power is clearly demanded to describe data in sufficient detail. Furthermore, descriptions should be able to support automated reasoning.

2.3 OIL, DAML+OIL and OWL

As RDF(S) is not sufficient as ontology language, efforts have been allocated for developing more expressive ontology languages for the Semantic Web. In this section, we shortly present the most important developments, namely OIL, DAML+OIL and especially OWL.

2.3.1 OIL

OIL [OIL] has been developed in the context of the European IST project On-To-Knowledge [1]. OIL provides modeling primitives used in frame-based and Description Logic oriented ontologies, coming along with a simple and clean semantics. The syntax definition uses RDF(s) and XML(s). in order to maintain backward compatibility.

OIL unifies three important aspects provided by different communities: (1) formal semantics and efficient reasoning support as provided by Description Logics [Horrocks], (2) epistemologically rich modeling primitives as provided by the Frame-based community, and (3) a standard proposal for syntactical exchange notations as provided by the Web community. OIL provides the means for describing structured vocabulary with well-defined semantics which is considered as the main contribution of OIL.

For describing ontologies OIL distinguishes three different layers [Klein M., et. al, 2000]. The *object level* where concrete instances of an ontology are described, the *first meta-level* (called ontology definition in OIL) where the actual ontological definitions are provided. Here we define the terminology that may be instantiated at the object level, and the *second meta-level* (i.e., the meta-meta level, called ontology container in OIL) is concerned with describing ontology features, like author, name, subject, etc. Reasoning support is a feature that OIL adds upon RDF whereby the expressiveness of a ontology language is increased as it allows functionalities like automatic consistency checking on ontology data.

2.3.2 DAML+OIL

DAML+OIL is a semantic markup language for Web resources created as a joint effort of the American and European ontology communities for the Semantic Web [DAML] by merging DAML-ONT (an early result of the DARPA Agent Markup Language DAML program) and OIL .

DAML+OIL exploits existing Web standards (XML and RDF) by adding ontological primitives of object oriented and frame-based systems and formal rigor of expressive description logic. It implements an object-oriented approach, with the structure of the domain being described in terms of classes and properties, and the set of axioms that assert characteristics of these classes and properties. The meaning of DAML+OIL is

---

[1] see www.ontoknowledge.org.

defined by a standard model-theoretic semantics based on interpretations, where an interpretation consists of a domain of discourse and an interpretation function.

DAML+OIL has been accepted as an ontology language with sufficient expressiveness throughout the research community.

2.3.3 OWL

OWL [Bechhofer et al, 2003] is the web ontology language currently under the development by the W3C Web Ontology Working Group. OWL is mainly based on OIL and DAML+OIL, therefore the main features of OWL are very similar to the languages introduced above.

OWL consist of three main components. **Ontologies** are defined as a sequence of axioms and facts, plus inclusion references to other ontologies, which are considered to be included in the ontology. OWL ontologies are web documents referred to by URIs. **Axioms** are used to associate class and property IDs with either partial or complete specifications of their characteristics and to give other logical information about classes and properties, and thirdly **Facts** state information about particular individuals in the form of a class that the individual belongs to plus properties and values. Individuals can either be given an individual ID or be anonymous (blank nodes in RDF terms).

In order to allow usability by various users , OWL provides three increasingly expressive sublanguages:

- **OWL-Lite**. Roughly consists of RDFS plus equality and 0/1-cardinality. Layered and easy-going language for tool builders. Developed to capture many of the commonly used features of DAML+OIL. It attempts to provide more functionality than RDFS, which is important in order to support web applications.
- **OWL DL**. Contains the whole OWL vocabulary, interpreted under a number of simple constraints. Primary among these can be found the type separation. Class identifiers cannot simultaneously be properties or individuals. Similarly, properties cannot be individuals.
- **OWL Full**. Composed of the complete vocabulary but interpreted more broadly than in OWL DL. A class can be treated simultaneously as a collection of individuals (the class extension) and as an individual in its own right (the class intension).

Apart from the RDF style syntax, the OWL specification also includes an abstract syntax which provides a higher level and less cumbersome way of writing ontologies. It has the advantage of allowing a more succinct statement of the semantics. It is interesting to observe that the OWL abstract syntax has reverted to grouping axioms into frame structures. The basic idea of having a semantic web language to represent ontologies is to allow computer programs to inter-operate without pre-existing, outside-of-the-web agreements. If this language also has an effective reasoning mechanism, then computer programs can manipulate this interoperability information themselves.

3. COMPARISON

In this section a general comparison among the different languages is presented. Section 3.1 compares the modeling primitives, in section 3.2 we provide a specific language to language comparison. Section 3.3 summarizes the strengths and weaknesses of the different languages.

3.1 Modeling primitives comparison

The comparison carried on in this part deals with factual knowledge (data models), terminological knowledge (ontologies), and inference knowledge.

Factual knowledge: Data Models.

The data-models underlying the semantic web languages present the following differences:

- XML takes labeled trees as its basic data-model. Thus information can be presented as hierarchical structures.
- RDF's data model consists of three object types (resources, properties and statements). This data model is a syntax-neutral way of representing RDF expressions. It is based on binary relations, enhanced with a reification mechanism to enable relations between relations, and statements

about the statements. RDFS uses this data model for defining the semantics of RDF modeling primitives.
- The data model of OIL, DAML+OIL's and OWL is based on description logic and Frame-based logic. Therefore, these languages have the rich class, property, and axiom to model the world.

Terminological knowledge: Ontologies.
Ontologies can define rich semantics of complex objects and therefore they are well-suited for describing heterogeneous, distributed and semi-structured information sources such as contexts on the Web:
- XML provides mechanisms for defining document structure and content. It allows inheritance for element, attribute, and datatype definitions and the creation of user-defined datatypes.
- RDF(S) can be used directly to describe an ontology with its Objects, Classes, and Properties. The expressiveness of RDF is rather limited as intentional definitions or complex relationships via axioms can be defined. A strong feature of RDF is the reification mechanism.
- OIL, DAML+OIL and OWL aim at complete support for defining ontologies. They provide richer constructors for forming complex class expressions and axioms for enabling reasoning on ontology data. An important feature of these languages is that they are layered in order to fulfill different needs and allow the definition of simple or complex ontologies.

Inference knowledge.
Regarding inference knowledge these are the main differences:
- XSLT (XSL Transformation language) allows expressing transformations of XML structures which can be used to express certain inferential knowledge.
- RDF/RDFS' subclass relation can be used to represent class subsumption.
- OIL, DAML+OIL and OWL allow definition of complex rules or axioms.

Table 1 gives the summary of the comparison on modeling primitives, where the range from - to ++ indicates the expressive power in the corresponding category.

Table 1: Summary of the modeling primitives

	Facts	Terminology	Inference
XML	+	+/-	-
RDF(S)	+	+/-	-
OIL	++	++	++
DAML+OIL	++	++	++
OWL	++	++	++

3.2 Specific language comparison

This section provides a one-to-one comparison of the different Semantic Web Languages introduced above in order to point out specific aspects of the technologies.

XML vs. RDF. RDF is an application of XML to represent meta-data. For example, an RDF statement can be represented in linear XML syntax. However, RDF provides a standard way to represent meta-data in XML. Using plain XML directly for representing meta-data would result in different syntax. RDFS provides a fixed set of modeling primitives for defining an ontology (classes, resources, properties, is-a, element-of relationship, etc. [Borgida, et. Al, 1994]) and a standard way on how to encode them in XML.

XMLS vs. RDFS. RDF Schema defines semantics of the RDF modeling primitives: class and subclass relationships, domain and range restrictions for properties, and subproperties. Although these modeling primitives are limited, RDFS allows basic definitions of ontologies while XML Schema does not. XML Schema, as well as DTDs, prescribe the order and combination of tags in an XML document. In contrast, RDFS provides information about the interpretation of the statements given in a RDF data model.

XML vs. OIL. XML can be used as a serial syntax for OIL, enabling support of web standards. Validation and rendering techniques developed for XML can directly be used for ontologies specified in OIL. Central for an ontology is the “is-a” relationship, and the fact that XML schemas incorporate the notion of inheritance. [Klein M., et. al, 2000] discussed a translation procedure that enables XML documents to capture the semantics of an ontology described in OIL by using type refinement in XML schemas to model the subsumption between concepts in OIL.

XMLS vs. OIL. XML schemas and OIL have as their main goal in common to provide a vocabulary and a structure for exchanging information sources. Thereby OIL provides much richer modeling primitives as it distinguishes classes and slots and class (or slot) definitions can be used to derive the hierarchy (and their corresponding inheritance). On the other hand, XML schemas provide richer modeling primitives concerning the variety of built-in datatypes and the grammar for structuring the content of elements. Models in OIL can be viewed as a high level description that becomes further refined when aiming for a document structure model.

RDF(S) vs. OIL. RDF can be used as a representation format for OIL. To ensure maximal compatibility with existing RDF/RDFs applications and vocabularies, the integration of OIL with the resources defined in RDF-schema is essential:

- The abstract OIL class `OntologyExpression` is a subclass of `rdfs:Resource`. The abstract OIL class `OntologyConstraint` is a subclass of `rdfs:ConstraintResource`.
- OIL slots are realized as instances of `rdf:Property` or as subproperties of `rdf:Property`. The subslot relationship can be expressed via `rdfs:subPropertyOf`. `rdf:Property` is enriched in OIL with a number of properties that specify inverse and transitive roles and cardinality constraints, what is not possible in RDF/RDFS.

OIL uses the existing primitives of RDFS as much as possible to retain an unambiguous mapping between the original OIL specification and its RDFS serialization. Therefore, the RDFS contained in the definition of domain ontologies in OIL can be easily understood or interpreted by any non-OIL-aware RDFS applications, while OIL-aware applications can tale advantage of the added features of formal semantics and reasoning support. In a nutshell, any valid OIL document is also a valid RDFs document when all the elements from the OIL-namespace are ignored. According to the layers distinguished in OIL, the sub-language OIL Core has been defined to exactly coincide with RDFS.

RDFS vs. DAML+OIL. DAML+OIL is tightly integrated with RDFS by using RDFS to express the syntax of DAML+OIL. Therefore, the existing RDFS infrastructure can be easily reused and ontologies defined by DAML+OIL can be partially or fully compatible with those defined by RDFS. DAML+OIL is regarded as a complete ontology specification language On the other hand, DAML+OIL also inheritates the “strange” modeling concepts of RDFS, such as restrictions with multiple properties and classes. DAML+OIL’s relation to RDFS also leads to the consequence of the decidability of the language. Decidability is lost when cardinality constraints can be applied to properties that are transitive, or that have transitive sub-properties. So decidability in DAML+OIL depends on an informal prohibition of cardinality constraints on non-simple properties.

DAML+OIL vs. OWL. As DAML+OIL has been a major reference for the OWL-Specification, the difference between DAML+OIL and OWL is rather trivial. But OWL abstract syntax has reverted to grouping axioms into frame like structure, which makes frame-based tools such as Protégé [Grosso, et. al, 1999] or DL based ones like OilEd [Bechhofer et al, 2001] easy to use. In this sense, OWL is closer to OIL due to its frame-based feature while DAML+OIL is more DL-like. Due to the fact that OWL is still a very young semantic web ontology language, (less than one and a half years old), more work and development will be carried on in this field. Significant work is going on in the Web Ontology Working Group of W3C [WOWG].

3.3 Strengths and weaknesses

This section concludes the analysis of Semantic Web languages. We summarize the strength and weaknesses of the languages presented and point out underlying design principles of Semantic Web Languages.

RDF(S). Its expressive power is quite limited and the reasoning capabilities are not the strongest among the different languages, providing a limited reasoning mechanism only suitable for constraint checking. It counts with partial interoperability facilities where mapping rules can be defined. It has a XML-based syntax. There are many tools and examples that could be either used or followed to learn about the language which makes it very widespread. Regarding internationalization it supports different natural languages and it is compatible with HTML, of which it is considered to be a super set. The community is actively developing and improving this language.

OIL. OIL counts with a much richer expressive power than RDFS for defining ontologies. The reasoning capabilities of OIL provide atomic consistency checking and allows cross linking the inter-ontology relations and check for implied relations. Regarding interoperability, OIL allows partial definition of mapping rules. It incorporates internationalization facilities supporting different natural languages. OIL is easy to use; there is a lot of documentation and examples about it, as well as tools and support for them. As long as compatibility is concerned its design is based on Description Logics, F-Logics and Web standards (RDFs and XML). Core OIL coincides with RDF Schema, except for the reification features of RDFS. OIL is no longer under development.

DAML+OIL. Its reasoning capabilities are useful for ontology sharing. Regarding interoperability, it allows the partial definition of mapping rules. Reasoning in DAML+OIL is specially suited for DL reasoning supporting design maintenance and deployment of ontologies. Its expressive power is much richer than the one of its predecessors; it supports different natural languages; it is quite easy to use, and regarding its compatibility it is important to notice that it supports the full range of XML Schema datatypes since it is based on the existing Web standards XML and RDF. Finally, it counts with partial interoperability facilities where mapping rules can be defined.

OWL. The reasoning functionalities of OWL could be used like in the case of DAML+OIL to provide sharing capabilities. Unlike the languages presented so far, OWL provides built-in versioning functionalities. Its reasoning mechanism is the same as in DAML+OIL and it is based on open world assumption (OWA). It is equipped with a rich expressive power and counts with a layered architecture for scalability. The easy of use is a common feature to all the languages presented so far. It supports different natural languages as the rest of its colleagues and regarding compatibility, should be outlined that OWL is based on OIL and DAML+OIL which makes it compatible with both. OWL is divided in three sub-languages suitable for different proposes (*OWL-Lite, OWL-DL, OWL-Full*). The main drawback of OWL resides in the fact that it is still under development.

As the underlying design criteria of Semantic Web languages, the following aspects have been worked out:

Compatibility. All languages are XML or RDF syntax based. This enables compatibility with web standards. Existing tools supporting XML or RDF can be easily reused for OIL, DAML+OIL and OWL. The layered language tower as aimed at by the W3C thus becomes possible.

Semantics. Adding semantics to the existing information is the main goal for the semantic web languages. But how to add and represent clear, explicit, machine-understandable semantics is not a trivial task. It is a clear go-direction for the design of languages. Different semantics needs different expressive power; therefore the layered structure is essential.

Layered structure. There will not be one single language, which fulfills all the needs of various web users. In a layered design, a simple core can accommodate simple taxonomies and relationships, while additional layers of expressivity, functionality, and complexity can be added for groups requiring more expressive power. Also the scalability and maintenance burden can be distributed to the different layers of the language, therefore can be alleviated accordingly. The languages that present such a clear layered structures are OIL and OWL.

4. CONCLUSIONS AND FUTURE WORK

The easy information access based on the success of the web has made it increasingly difficult to find, present, and maintain the information required by a wide variety of users. In response to this problem, many new research initiatives and commercial enterprises have been set up to enrich available information with machine understandable semantics. The Semantic Web will provide intelligent access to heterogeneous, distributed information, enabling software products to mediate between user needs and the information sources available.

Ontologies are considered to be a key enabling technology for the Semantic Web as they provide a means to formally specify semantics of web resources. But giving real semantics to the semantic web language tower as sketched in Figure 1 [Berners-Lee, 1999] requires much more work. Currently many layering ideas oriented to syntactical and semantic extensions compete with each other. Working out the proper relationship will be much more challenging than just developing one layer for it.

This paper has presented the state of the art on Semantic Web Languages including the most relevant ones (RDF(s), OIL, DAML+OIL and OWL). The first part of the paper presented a short description of each of these languages, outlining their main capabilities and strengths. In the second part, a detailed comparison according to different criteria and perspectives was detailed.

The main goal of the paper has been to present the actual state of the art of the Semantic Web technology regarding the languages it is using as foundational component for its development. It is our hope that this paper is found useful whenever a decision about which language to use has to be made. To achieve this goal, we have provided specific language-to-language comparisons, the strengths and weakness of each language have been presented, and finally the capabilities of each one of them regarding the W3C have been put to test. In coming papers, the evolution experimented by actually evolving languages (RDF(s) and OWL) will be presented, taking under special consideration the criteria sketched by W3C.

The comparison has shown that the development of suitable languages for ontology specification has reached a level of maturity. But in order to lift the Semantic Web to its full potential a lot more work has to be undertaken, but the path has already been delineated and semantic web languages are the basis for this technological reality.

REFERENCES

Bechhofer, S. et al, 2001. DAML+OIL is not enough. *Proceedings of the First Semantic Web Working Symposium (SWWS'01)*, 151–159.

Bechhofer S. et al, 2003. OWL Web Ontology Language Reference. *W3C Candidate Recommendation.* www.w3.org/TR/owl-ref/.

Berners-Lee, T. et al, 2001. The Semantic Web. Scientific America.

Berners-Lee, T. 1999. Weaving the Web. *San Francisco: Harper*.

Borgida, A. and Patel-Schneider, P. F., 1994. A semantics and complete algorithm for subsumption. *In the CLASSIC description logic. Journal. of Artificial Intelligence Research* 1:277–308.

DAML. DARPA Agent Markup Language (DAML). www.daml.org.

XML. Extensible Markup Language (XML). www.w3.org/XML.

Grosso, W. et. al, 1999. Knowledge modeling at the millennium (the design and evolution of protégé-2000). *In Proc. of Knowledge acquisition workshop (KAW-99)*.

Heflin, J.et. al, 2002. Requirements for a Web Ontology Language. W3C Working Draft.

Horrocks I. The FACT System. www.cs.man.ac.uk/~horrocks/FaCT/.

Klein M., et. al, 2000. The relation between ontologies and schema-languages: Translating OIL-specifications in XML-Schema. *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany.

Nejdl, W. et. al, 2000. The RDF Schema Revisited. *In Ebert J. et al. (eds.), Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik, Modellierung*, St. Goar.

OIL. Ontology Inference Layer (OIL). www.ontoknowledge.org/oil/.

RDF. Resource Description Framework (RDF). www.w3.org/RDF/.

WOWG. Web Ontology Working Group. www.w3.org/2001/sw/WebOnt/