Choosing an Ontology Language

Anna V. Zhdanova, Uwe Keller

***Abstract*—**We summarize information that facilitates choosing an ontology language for knowledge intensive applications. This paper is a short version of the ontology language state-of-the-art and evolution analysis carried out for choosing an ontology language in the IST Esperonto project. At first, we analyze changes and evolution that took place in the filed of Semantic Web languages during the last years, in particular, around the ontology languages of the RDF/S and OWL family. Second, we present current trends in development of Semantic Web languages, in particular, rule support extensions for Semantic Web languages and emerging ontology languages such as WSMO languages.

***Keywords*—** OWL, RDF/S, Semantic Web Languages, WSML

I. Introduction

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 [1].

For the Web to scale, tomorrow's programs must be able to share and process data even when these programs have been designed totally independently. The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

One of the biggest problems we nowadays face in the information society is information overload. The Esperonto project[1] aims to overcome this problem by adding meaning to the Web, which can be exploited by software agents to whom people can delegate tasks.

Esperonto's objective is to bridge the gap between the actual World Wide Web and the Semantic Web by providing a service to "upgrade" existing content to Semantic Web content. Ontologies play a key role in this effort, aiming at unifying, bridging and integrating multiple heterogeneous, international and multilingual digital content. Once having Semantic Web content, the project aims at exploiting the content through innovative graphical navigation, and prototyping distributed added value services.

In this paper, we provide up-to-date information about the established existing web ontology languages, such as RDF(s), OIL, DAML+OIL, OWL, and review trends in development of new ontology languages, such as WSMO languages. The provided information is purposed to help in the decision which of the current and emerging Semantic Web languages should be used in the in different kinds of knowledge-based applications, specifically, for Semantic Annotation Service provider developed in the Esperonto project.

The structure of this paper is as follows. In section 2, we introduce current Semantic Web languages and outline the latest advances in evolution of these languages. In section 3, we specify requirements for the ontology language in the Esperonto project, analyze the current Semantic Web languages versus the requirements, describe current rule extensions and support available for the current languages, and specify the ontology language choice for the Esperonto project. Section 4 concludes the paper.

II. Evolution of Ontology Languages

In this section, we overview the most renowned Semantic Web languages, and outline their evolution trends.

A. RDF Schema

RDF is a general-purpose language for representing information in the Web. RDF schema[2] is a formal description of eligible RDF expressions. In particular, a schema defines the properties of the resource (e.g., title, author, subject, size, colour, etc.) and the kind of resources being described (e.g., books, Web pages, people, companies, etc.).

RDF Schema became a W3C Proposed Recommendation on 15 December 2003, and a W3C Recommendation on 10 February 2004. The main latest updates and improvements accomplished within the context of this language are related to grammar updates, ways to link the XML serialization to formal semantics, tutorials on how to use RDF in applications, syntax and test cases.

B. OIL

OIL[3] is built on top of RDF and RDFS, employing their constructs to a large extent, in order to maintain backward compatibility. OIL provides modelling primitives used in frame-based and Description Logic oriented ontologies, coming

Manuscript received January, 2005. This work was supported by the European Commission under the projects DIP, Knowledge Web, SEKT, SWWS, Esperonto; by Science Foundation Ireland under the DERI-Lion project.

A. V. Author is with the Digital Enterprise Research Institute, Innsbruck, Austria and Galway, Ireland (phone: +43-512-507-6467; fax: +42-512-507-9872; e-mail: anna.zhdanova@deri.org).

U. K. Author is with the Digital Enterprise Research Institute, Innsbruck, Austria and Galway, Ireland (e-mail: uwe.keller@deri.org).

[1] http://esperonto.semanticweb.org

[2] http://www.w3.org/TR/rdf-schema/

[3] http://www.ontoknowledge.org/oil/

along with a simple and clean semantics. It has a syntax definition using web standards such as RDF(S) and XML(S).

OIL unifies three important aspects provided by different communities: (1) formal semantics and efficient reasoning support as provided by Description Logic, (2) epistemologically rich modelling primitives as provided by the Frame-based community, and (3) a standard proposal for syntactical exchange notations as provided by the Web community.

OIL is not an evolving language any longer. The natural continuer of the work carried out by the OIL team was DAML+OIL, a joint effort of the American and European ontology communities for the Semantic Web.

C. *DAML+OIL*

DAML+OIL[4] is an ontology language specifically designed for the Semantic Web, created as a joint effort of the American and European ontology communities for the Semantic Web. DAML+OIL exploits existing de-facto Web standards (XML and RDF), adding ontological primitives of object oriented and frame-based systems, and the formal rigor of expressive description logic. As an ontology language, DAML+OIL is designed to describe the *structure* of a domain. DAML+OIL takes 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.

Similar to OIL, DAML+OIL is not an evolving language. The latest DAML+OIL drafts are dated from December 2001.

D. *OWL*

OWL[5] is the web ontology language, developed by the W3C Web Ontology (WebOnt[6]) Working Group. OWL is mainly based on OIL and DAML+OIL, and therefore the main features of OWL are very similar to those of OIL. OWL includes three sub languages called:

OWL-Lite. Roughly consists of RDFS plus equality and 0/1-cardinality. It represents a migration path from other taxonomies. It is intended for classification hierarchies and simple constraints. It should be kept as simple as possible in order to facilitate the tool development.

OWL DL. Contains the language constructs but with hierarchy restrictions. It provides computational completeness and decidability, and has a maximum expressive power within DL Description Logics fragment.

OWL Full. Composed by the complete vocabulary interpreted more broadly than in OWL DL. The language incorporates maximum expressive power and syntactic freedom, and offers no computational guarantees.

Besides the DAML+OIL style RDF syntax, the OWL specification also includes an abstract syntax, which provides a higher level and less cumbersome way of writing ontologies.

OWL became a W3C Proposed Recommendation on 15 December 2003, and a W3C Recommendation on 10 February 2004. At the moment, OWL is at the pruning stage, and frequent and major changes are not expected to take place. The main efforts of the Web Ontology Working Group were switched to the activities within the newly established W3C Semantic Web Best Practices and Deployment Working Group (SWBPD)[7]. The major efforts of the SWBPD group aim at providing hands-on support for developers of Semantic Web applications, which will contribute to dissemination and pruning of OWL (and RDF) languages.

[4] http://www.daml.org/

[5] http://www.w3.org/TR/owl-ref/

[6] http://www.w3.org/2001/sw/WebOnt/

E. *Ontology Languages in the WSMO Project*

The *Web Service Modelling Ontology* (WSMO)[8] project is a major European initiative in Semantic Description of Web Services. It is carried out by the SDK-Cluster[9] in the context of three EU-funded projects, namely SEKT[10], DIP[11] and Knowledge Web[12], and aims at providing the conceptual model for semantically describing various aspects of Web Services relevant for discovery, composition and mediation. The WSMO initiative hosts two sub-projects for modelling language related issues (Web Service Modeling Language, WSML) and for the design and implementation of a reference implementation (Web Service Execution Environment, WSMX).

In the context of the WSML project there are some efforts related to developing ontology languages with specific characteristics that particularly useful in the area of Semantic Web Services, namely OWL-Lite-, OWL-Flight, OWL-DL-, OWL-Full-, WSML-Core. Although, these languages are being developed in a specific context, namely Semantic Web Service Description, they are useful and relevant in their own right. We will briefly discuss these languages below.

1) OWL-Lite-

OWL Lite is the least expressive species of OWL. However, this language already requires reasoning with equality, which significantly increases computational complexity. Cardinality restrictions, in their current form, introduce equality in a non-intuitive way. There is no notion of constraints in OWL Lite. Furthermore, because the expressiveness of the *SHIF* Description Logic language is beyond the capabilities of efficient rule-based engines, and because straightforwardly extending a Description Logic with Horn-like rules leads to undecidability issues [4], one cannot easily extend OWL Lite with a rule language without loosing computational guarantees which so far has been considered as an important feature of Semantic Web languages.

OWL Lite- [3] is a proper subset of OWL Lite that can be translated into Datalog. OWL-Lite- restricts the syntax and semantics of OWL Lite. The OWL-Lite- authors argue that some of the above mentioned limitations can be overcome by using a more restricted form of OWL Lite, which can be translated into a Datalog program (without equality) [3]. This language can then be straightforwardly extended to include

[7] http://www.w3.org/2001/sw/BestPractices/

[8] http://www.wsmo.org

[9] http://www.sdk-cluster.org

[10] http://sekt.semanticweb.org

[11] http://dip.semanticweb.org

[12] http://knowledgeweb.semanticweb.org

database-style integrity constraints, which can be used for both cardinality and value constraints. Furthermore, in Datalog rules can be added directly on top of the ontology.

2) OWL-Flight

One of the important purposes with inventing OWL Lite- has been to come up with a clean conceptual starting point for a powerful and practically useful ontology language with efficient reasoning support.

OWL Lite- already overcomes some of the limitations of OWL Lite, but in some cases the expressivity had to be significantly reduced. For example, we had to leave out cardinality restrictions, because they introduce equality. Furthermore, many limitations of OWL Lite still exist in OWL Lite-, such as the lack of constraints and a sharp distinction between classes and instances. Also, OWL Lite- does not provide support for datatypes, which is commonly considered as essential for real-world applications.

For this reason, de Bruijn [3] extends OWL Lite- with a number of features, such as datatype support, unique name assumption, constraints, classes-as-instances, local-closed world assumption.

3) WSML-Core

In addition to the OWL- family of ontology languages presented above, there is an ontology language called *WSML-Core* currently being developed in the WSML working group which essentially combines two things: OWL Lite- and the conceptual meta-model for ontologies as defined in the WSMO Ontology [5]. Roughly speaking, OWL-Lite- provides the semantic basis for WSML-Core whereas the conceptual model for ontologies of WSMO provides the basic modelling elements of the language.

WSML-Core represents the most basic language that can be used for describing the semantics of web services. It represents the intersection between two prominent knowledge representation paradigms, namely Description Logics and Rule Languages.

III. CHOICE OF AN ONTOLOGY LANGUAGE

A. Requirements of the Esperonto Project

As in any other Semantic Web project, the ontology language chosen in the Esperonto project requires a balance between expressive power and reasoning support.

Other requirements are:

(1) *Ease of use.* Due to actual lack of tools and that the project is an early adopter of the language; it should have a human readable syntax and fast learning curve.

(2) *Compatibility.* There exists the possibility of using some other languages such as RDF(S) and OWL in some parts of the project. In order to minimize the efforts towards the final consolidation into a single language, compatibility with other standards should be supported by the chosen language.

(3) I*nternationalization.* Due to the multilinguality support Esperonto must offer (English, Spanish, Catalan) ontologies will be shared among different speaking agents. Therefore, internationalization must be supported by the language.

(4) *Sharing and versioning.* Ontologies will be shared and will evolve as the project evolves. If the language counts with support for these features, then the amount of work within the project related to these matters will be reduced.

(5) *Simple extendibility to support advanced application.* Ontologies by themselves are static components of Semantic Web, and they are not useful if they can not be processed by applications, software agents and Web Services. One of the most important and widely recognized aspects here is extended expressivity of the languages for advanced applications like Semantic Web Service applications by means of rule support. Rules represent a very flexible mechanism to add domain knowledge to ontologies and derive implicit knowledge from the given facts in an elegant and efficient way. The rule extendibility of the candidate ontology languages is analyzed in section III.C of this paper.

B. Candidate Languages

1) RDF Schema This language lacks of sharing and versioning capabilities. 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. 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 XML, of which it is considered to be a super set. The language is an attractive choice if its expressive power is considered being enough. RDF(S) has been the language selected for the first prototype of the Esperonto system and of the case studies. Furthermore, a major argument for using RDF(S) is that RDF as seen as almost non-alternative languages for representing instances, since this is the language used by RDFS and OWL for expressing instances.

2) OIL OIL is no longer an alive language, so choosing OIL would damage impact and visibility of the project.

3) DAML+OIL Like OIL, DAML+OIL is no longer an alive language, so choosing DAML+OIL would damage impact and visibility of the project.

4) OWL Reasoning functionalities of OWL could be used (like in the case of DAML+OIL) to provide sharing capabilities, basing on the *open world assumption (OWA)*. Unlike the languages presented above, OWL incorporates expressions for versioning. OWL has a rich expressive power and possesses a layered architecture for scalability. OWL supports different natural languages as the other considered languages.

OWL is very well positioned in the community mobilizing lots of efforts to make it become “the” Semantic Web language to use in the future. From an impact and visibility point of view, this would be the language to chose, even though taking up the language requires in the development phase. Numerous initiatives developing OWL tool support exist, such as Jena – a toolkit for managing OWL ontologies [2], OWL reasoning-related activities, e.g., by NetworksInference[13] and Pellet[14] at the Maryland Information and Network Dynamics Lab.

[13] http://ww.networkinference.com

[14] http://w w w .mindswap.org/2003/pellet/index.shtml

5) Ontology Languages in the WSMO Project At present, there is only a precise definition and discussion for OWL-Lite-, a proper subset of OWL Lite with nice computational and extendability properties: There exists a direct translation into the deductive database language Datalog. Thus, any OWL Lite- ontology can be translated into Datalog in order to allow for efficient query answering. Practically, most current ontologies fall inside this fragment. An ontology language for which a translation to Datalog exists has several advantages. Most notably, it can benefit from highly optimized query answering engines and allows for easy implementation of a rule and a query language on top of the ontology. Since OWL Lite- is a subset of OWL-Lite, all tools that can deal with OWL Lite can automatically be used for OWL Lite-. Moreover, by implementing the translation from OWL Lite- to Datalog, one can use any existing Datalog engine for efficient reasoning on ontologies and large sets of instances.

C. Rule Extensions of Candidate Languages

Rules are considered to be a design issue for the Semantic Web (on top of the ontology layer in Tim Berners-Lee's Semantic Web layer cake) and have been a topic of discussion in the W3C Web Ontology Working Group, but have not been included in the Web ontology language OWL. It is expected that there will be a W3C Working Group for developing a W3C rule markup language, possibly having a RuleML[15] working group as a core.

RDF/S rule extensions: RDF extension: TRIPLE

TRIPLE [6, 7] is a rule language for the Semantic Web. TRIPLE is based on Horn logic and borrows many basic features from F-Logic but is especially designed for querying and transforming RDF models. In this respect, it can be seen as a query and inference language for RDF. TRIPLE can be viewed as a successor of SiLRI (Simple Logic-based RDF Interpreter). One of the most important differences to F-Logic and SiLRI is that TRIPLE does not have a fixed semantics for object-oriented features such as classes and inheritance. TRIPLE's module system allows such features to be easily defined for different object-oriented and other data models like UML, Topic Maps, or RDF Schema. TRIPLE can be applied for instance to various e-learning tasks such as querying learning objects, inference for personalization, and ontology mapping.

OWL rule extensions: OWL extension: SWRL

The Semantic Web Rule Language (SWRL)[16] is a proposal for a rule language for the Semantic Web based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. The proposal extends the set of OWL axioms to include Horn-like rules. It thus enables Horn-like rules to be combined with an OWL knowledge base. A high-level abstract syntax is provided that extends the OWL abstract syntax described in the OWL Semantics and Abstract Syntax document. An extension of the OWL model-theoretic semantics is also given to provide a formal meaning for OWL ontologies including rules written in this abstract syntax.

[15] http://www.ruleml.org

[16] http://www.daml.org/rules/proposal/

D. Candidate Solution

In the previous sections, characteristics of the different ontology languages have been presented emphasizing the important features that can influence choice of an ontology language for knowledge intensive applications, and, particularly, in applications and tools of Esperonto project.

Given the nature of the Esperonto project and the position (evolution/characteristics/impact) of the ontology languages, the RDFS and OWL languages were chosen for the implementation works in the project. The practical advantages of a choice in favor of these languages were their broad scope and visibility, expressivity, active community behind and permanent improvement of related ontology management and reasoning tools.

Though in the near future, the ontology languages of the WSMO initiative, in particular the OWL- family of languages, can be of interest to consider, due to their advanced computational properties.

IV. Conclusions

Any Web ontology language enables describing and organizing knowledge on the Web in a machine understandable way. However, existing ontology languages substantially vary in their status and properties. Monitoring and understanding of state-of-the-art, development and evolution processes of ontology languages are crucially important for choosing an ontology language in any knowledge intensive application.

Acknowledgment

The authors thank Sinuhe Arroyo, Jos de Bruijn and other colleagues from Esperonto and DERI for useful inputs and discussions.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American 284(5), 2001, pp. 34-43.

[2] J.J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson, *Jena: Implementing the Semantic Web Recommendations*. Proc. of the Thirteenth International World Wide Web Conference (WWW 2004), 2004, Alternative track papers and posters, pp.74-83.

[3] J. de Bruijn (Ed.): *OWL Flight*. WSML Deliverable D20.1 Available: http://www.wsmo.org/2004/d20/d20.1/v0.2/20040719/.

[4] A.Y. Levy., M.C. Rousset, "Combining horn rules and description logics in CARIN", Artificial Intelligence, 1998, 104:165-209.

[5] D. Roman, H. Lausen (Eds.): *Web Service Modeling Ontology - Standard (WSMO - Standard)*, WSMO deliverable D2 version 1.0 Working Draft 29 July 2004. Available: http://www.wsmo.org/2004/d2/v1.0/20040729/.

[6] M. Sintek, S. Decker, "TRIPLE - An RDF Query, Inference, and Transformation Language", DDLP'2001, Japan, October 2001.

[7] M. Sintek, S. Decker, "TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web". International Semantic Web Conference (ISWC), Sardinia, June 2002.