IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-7, Special Issue, 4, Feb 2019 E-ISSN: 2347-2693

A Review on Ontology Languages

R. Revathy

Department of Computer Applications, Idhaya College for Women, Kumbakonam, Tamil nadu, India

Corresponding Author: revasakthi06@gmail.com

Available online at: www.ijcseonline.org

Abstract— Ontology languages play key roles for representing and process info regarding the important world for the rising linguistics net. Efforts are created to develop varied metaphysics languages. Every metaphysics language provides totally different communicatory power and conjointly process quality for reasoning. Metaphysics question languages were developed to question the data outlined by these metaphysics languages and reasoning systems. We have a tendency to conduct a study to match their communicatory power, efficiency, and scalability and best performing arts scenario.

***Keywords*—** Ontology, RDF, OWL.

I. INTRODUCTION

The linguistics internet is taken into account by some because the next generation of the globe Wide internet. the event of appropriate languages for representing metaphysics and economical reasoners are 2 keys tasks to be resolved for the linguistics internet. Currently, many metaphysics languages and metaphysics question languages are developed. For these languages and querying systems, there doesn't exist a radical comparison and analysis among them. This paper evaluates those thought metaphysics languages and metaphysics question languages and provides a read concerning the benefits and drawbacks of every of them. After that, it provides our evaluations on what question languages and what question systems to settle on underneath a particular state ofaffairs.

In this paper, we'll specialise in 3 metaphysics languages: RDF(S) [1], OWL [2], and SWRL [3]. The Resource Description Framework (RDF) could be a framework developed by the W3C (World Wide net Consortium) for representing info within the World Wide net. RDF inherits XML syntax and exploits URI to spot resources. RDF Schema (RDFS) is employed to specify the vocabularies in RDF. RDF(S) provides a foundation for different advanced languages for similar functions. the net metaphysics Language (OWL) could be a linguistics language for metaphysics illustration. It extends RDF syntax and springs from DAML+OIL [4] and plenty of different influencers. raptor provides 3 more and more communicatory sublanguages: raptor nonfat , raptor metric capacity unit (Description Logic) and raptor Full [5]. The linguistics net Rule Language (SWRL) couldbearulelanguagecombiningraptorandRuleML[3]. It's still underneath development by the Joint language Committee2. SWRL extends raptor metric capacity unit with binary/unary 1st order Horn like rules. This rule extension makes SWRL additional powerful and versatile than raptor metric capacity unit. As a trade-off, the machine quality of SWRL increased to semi-decidable [6].

II. REQUIREMENTSFOR ONTOLOGY LANGUAGES

Ontology languages allow users to write explicit, formal conceptualizations of domains models. The main requirements are:
1. A well-defined syntax
2. A well-defined semantics
3. Efficient reasoningsupport
4. Sufficient expressive power
5. Convenience ofexpression

The importance of well-defined syntax is obvious, and illustrious from the realm of programming languages; it's a necessary condition for machine-processing of data. All the languages we've conferred up to now have well-defined syntax. DAML+OIL and OWL hinge upon RDF and RDFS and have a similar quite syntax. after all it's questionable whether or not the XML-based RDF syntax is extremely easy, there are alternatives higher appropriate for humans (for example, see the OIL syntax). but this disadvantage isn't terribly significant, as a result of ultimately users are going to be developing their ontologies victimisation authoring tools, or a lot of typically metaphysics development tools, rather than writing them directly in DAML+OIL or OWL.

III. ONTOLOGYLANGUAGES

This section shortly discusses numerous metaphysics languages for the linguistics internet (technically, these languages area unit in several layers of the linguistics internet cake [10] however area unit all used for representing ontology). The focus is on the variations and correspondences among completely different languages. Since all of those languages are influenced by RDF and RDFS, We 1st in brief describe RDF and RDFS. DAML+OIL is extremely the same as hooter and is additionally very in brief delineate. During this section, we'll focus to a small degree additional on the upper finish of the Semantic net cake, hooter and SWRL.

A. RDF and RDFS

The goal of the linguistics net is to form data on the net each accessible and graspable by not solely humans, however additionally computers. RDF(S) was so developed to represent data within the net. In RDF(S) the resources within the net ar known by net identifiers (Uniform Resource symbol or URI) [11]. To form it machine processible, RDF inherits XML- based syntax. When years of development, RDF(S) currently has formal syntax, formal linguistics and XML Schema data types. It's a W3C suggested data illustration normal for the linguistics net. The RDF abstract syntax includes a graph pattern, wherever the statements ar pictured as N-triples [12] (format: Subject – Predicate – Object or node-arc-node link, thence the term 'graph') [13]. RDF will specific resource properties and their values. RDFS extends RDF by providing the power to outline RDF vocabularies like categories, properties, types, ranges, domains, etc. However, RDF and RDFS solely have terribly restricted communicative powers [11][14]:

- RDF(S) cannot express equality andinequality;
- RDF(S) cannot define enumeration of propertyvalues;
- RDF(S) cannot apply cardinality and existenceconstraints;
- RDF(S) cannot describe unique, symmetric, transitive, inverse relationships amongproperties;
- RDF(S) cannot describe union, intersection andcomplement;
- The domain and range in RDF(S) can only be specified globally [14]. As a result, several more sophisticated languages have been developed to meet theserequirements.
-

B. DAML+OIL

DAML+OIL [4] could be a combination of DAML (DARPA Agent Markup Language) and OIL (Ontology abstract thought Layer) [15]. Its associate degree RDF/XML syntax supported the frame paradigm [16] associate degreed describes the structure of a site (schema) in an object-oriented vogue. DAML+OIL consists of a group of axioms declarative the relationships between categories and properties. DAML+OIL uses an outline Logic vogue model theory to formalize the which means of the language [16]. This can be a awfully vital feature to scale back arguments and confusions, therefore giving the language the power to exactly represent the meanings of data. This ability is crucial for automatic reasoning that is that the goal of the linguistics net. The new options DAML+OIL supports area unit the following:

_ Constraints (restrictions) on properties (existential/universal and cardinality),

_ Boolean combinations of classes and restrictions, e.g., union, complement and intersection,

_ Equivalence and disjointness,

_ Necessary and sufficient conditions, and

_ Constraints on properties.

However, DAML+OIL tries to be compatible with RDF syntax, however this raises some serious syntactical and linguistics issues [16]. Another drawback is that DAML+OIL datatypes aren't compatible with RDF, since RDF didn't give datatype definition ability once DAML+OIL was being developed.

C. OWL

OWL [5] was developed on top of RDF and borrowed from DAML+OIL. Like RDF, OWL is the standard recommended by W3C for Semantic Web. OWL is powerful in expression, but complex for computation. To compromise between expressive power and acceptable computational complexity, OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. Among them, OWL Lite is a subset of OWL DL, and OWL DL is a subset of OWL Full. OWL Full contains all the OWL language constructs and provides the free, unconstrained use of RDF constructs [2]. In OWL Full, *owl: Class* is equivalent to *rdfs: Class*.

OWL Full also permits classes to be individuals. A class can even be a property of itself. In OWL Full, *owl: Things* and *rdfs: Resource* are equivalent too. This means that object properties and datatype properties are not disjoint. The advantage of this jointness is that it provides high expressive power. Unfortunately, the drawback is that it is computationally undecidable [16]. As the result, it is very difficult to build a reasoning tool for OWL Full. Although theoretically, OWL Full can be processed via some FOL engine, it cannot guarantee quick and complete answers. As a sublanguage of OWL Full, OWL DL introduces several restrictions on the usage of OWL constructs. These restrictions are carefully chosen to make sure that OWL DL is decidable. OWL DL does not support all of RDF(S) [2]. In OWL DL, classes, datatypes, individuals, and properties are all pairwisely disjoint. Datatype properties and object properties are also disjoint. As a result, inverse, transitive and symmetric relationships can not be applied to datatype properties. Cardinality constraints are also forbidden on transitive properties. These restrictions guarantee that OWL DL is computationally decidable. It is equivalent to DL *SHOIN* (**D**) [16] whose worst case is non-deterministic exponential time(NEXPTIME).

Reasoning for OWL DL can be supported via DL or FOL reasoners without losing accuracy. It is the best choice for users who require accurate results with maximal expressive power. OWL Lite can be considered as a simplified version of OWL DL. It supports simple classification hierarchies and simple qualification restriction. Constructs such as *one of*, *unionOf*, *complementOf*, *hasValue*, *disjointWith*and *DataRange*are not allowed in OWL Lite. Furthermore, some constructs also restrain the use of certain resources. The cardinality restrictions in OWL Lite can only have value of 0 or 1. The computational complexity of OWL Lite is equivalent to that of DL *SHIF* (**D**), which is exponential time (EXPTIME) in the worst case [16]. The purpose of OWL Lite is to provide a minimal useful subset of OWL with an efficient complete reasoner. Building a DL reasoner for OWL Lite is relatively straightforward. Several currentDL reasoning systems perform very well on OWL Lite repositories.

D. The OWLLanguage

1) Syntax:

OWL builds on RDF and RDF Schema, and uses RDF's XML syntax. Since this is often the first syntax for raptor, we are going to use it here, however it'll presently become clear that RDF/XML doesn't give a really decipherable syntax. owing to this, different grammar forms for raptor have conjointly been defined:

_ AN XML-based syntax that doesn't follow the RDF conventions. This makes this syntax already signi_cantly easier to browse by humans.

_ AN abstract syntax that is employed within the language speci_cation document. This syntax is way a lot of compact and decipherable then either the XML syntax or the RDF/XML syntax nine a graphical syntax supported the conventions of the UML language (Uni- versalModelling Language). Since UML is wide used, this may be a straightforward manner for folks to induce acquainted with raptor.

2) Header:

OWL documents are usually called OWL ontologies, and are RDF documents. So the root element of a OWL ontology is an rdf:RDF element which also speci_es a number of namespaces. Forexample:

```
<rdf:RDF
xmlns:owl ="http://www.w3.org/2002/07/owl#"
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:xsd ="http://www.w3.org/2001/XLMSchema#">
```

An OWL ontology may start with a collection of assertions for house-

keeping purposes. These assertions are grouped under an owl:Ontology element which contains comments, version control and inclusion of other ontologies. For example:

```
<owl:Ontologyrdf:about="">
<rdfs:comment>An example OWL ontology</rdfs:comment>
<owl:priorVersionrdf:resource="http://www.mydomain.org/uni-ns-old"/>
<owl:importsrdf:resource="http://www.mydomain.org/persons"/>
<rdfs:label>University Ontology</rdfs:label>
</owl:Ontology>
```

The only of those assertions that has any consequences for the logical which means of the metaphysics is owl: imports: this lists alternative ontologies whose content is assumed to be a part of this document. Ontology Notice that whereas namespaces ar used for elucidation functions, foreign ontologies give de_nitions that may be used. typically there'll be Associate in Nursing import part for every used namespace, however it's doable to import extra ontologies, for instance ontologies that give de_nitions while not introducing any new names. additionally note that owl:imports could be a transitive property: if metaphysics A imports metaphysics B, and metaphysics B imports metaphysics C, then metaphysics A additionally imports metaphysics C. category components categories arde_ned employing a owl:Class element7. for instance, we will de_ne a category associateProfessor as follows:

```
<owl:Classrdf:ID="associateProfessor">
<rdfs:subClassOfrdf:resource="#academicStaffMember"/>
</owl:Class>
```

We can additionally say that this category is disjoint from the faculty member and assistantProfessor categories victimisationowl:disjointWith parts. These parts will be enclosed within the de_nition higher than, or will be adscititious by relating the id victimisationrdf:about. This mechanism is inheritable from RDF.

```
<owl:Classrdf:about="associateProfessor">
<owl:disjointWithrdf:resource="#professor"/>
<owl:disjointWithrdf:resource="#assistantProfessor"/>
</owl:Class>
```

Equivalence of classes can be de_ned using aowl:equivalentClasselement:

```
<owl:Classrdf:ID="faculty">
<owl:equivalentClassrdf:resource="#academicStaffMember"/>
</owl:Class>
```


Finally, there are two prede_ned classes, owl:Thing and owl:Nothing. The former is the most general class which contains everything (everything is a thing), the latter is the empty class. Thus every class is a subclass of owl:Thing and a superclass of owl:Nothing.

3) Propertyelements:

In OWL there are two kinds of properties:

_ Object properties which relate objects to other objects. Examples are isTaughtBy, supervises etc.

_ Datatype properties which relate objects to datatype values. Examples are phone, title, age etc. OWL does not have any prede_ned data types, nor does it provide special de_nition facilities. Instead it allows one to use XML Schema data types, thus making use of the layered architecture the Semantic Web Here is an example of a datatype property.

```
<owl:DatatypePropertyrdf:ID="age">
<rdfs:rangerdf:resource="http://www.w3.org/200
1/XLMSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
```

User-defined data types will usually be collected in an XML schema, and then used in an OWL ontology.

Here is an example of an object property:

```
<owl:ObjectPropertyrdf:ID="isTaughtBy">
<owl:domainrdf:resource="#course"/>
<owl:rangerdf:resource="#academicStaffMember"/>
<rdfs:subPropertyOfrdf:resource="#involves"/>
</owl:ObjectProperty>
```

E. SWRL

The Semantic Web Rule Language (SWRL) [3] is under development as a combination of the OWL DL with the Unary/Binary DatalogRuleML sublanguages of the Rule Markup Language [3]. SWRL can be considered as DL + function-free FO Horn rules . SWRL adds rules to OWL DL. The reason is that these rules provide more expressive power to Description Logic. For example, you can use FOL to define (represented in the human readable syntax of SWRL) the concept *Aunt.* Parent(?x, ?y) ^ Sister(?y, ?z) ⇒ Aunt (?x, ?z) Here 'Parent(?x, ?y) ^ Sister(?y, ?z)' is called the antecedent (body) and 'Aunt (?x, ?z)' is called the consequent (head) of the rule. Whenever the antecedent holds, the consequent holds. OWL can not define relationships like this. Applying rules one can also extend OWL with compositioncapability.

Although SWRL is comparatively new, FOL has already been completely studied. Also, the mix with FOL permits SWRL to simply communicate with ancient computer database systems. This feature is incredibly engaging since most data within the planet remains hold on in relative databases. The disadvantage of SWRL is its procedure complexness. though it's still underneath construction, developers have in agreement that it'll support most of the options of Description Logic and partial Horn Logic Programs. it's absolute to be undecidable [3]. However, since FOL has been studied for over a century, it'll still like the precedent model theory and FOL engines. an honest answer for this can be to borrow the concept of bird of prey – developing multiple sublanguages of SWRL with decreasing complexities, from full to easy versions and decreasing the complexities from being undecidable to being decidable. This task has not formally started nevertheless, however has already been thought-about.

IV. CONCLUSION

We studied several languages for the Semantic Web. RDFS is a framework that provides limited expressiveness for representing metadata. The advantage of RDFS is that it is very simple and can be very efficient for reasoning. OWL is a successful ontology language which is recommended by W3C. It provides standardized syntax and is downward compatible with RDFS. That is anything represented in RDFS can be translated into OWL. To overcome the complexity problem, OWL introduces a very attractive idea to design a family of sublanguages with different levels of expressive power and computational complexity. Users can flexibly select a proper sublanguage for a specific representation task. Another advantage of this strategy is that users can easily upgrade existing ontology into higher level of sublanguages. For example, users can start creating ontology in OWL Lite since it is much easier to understand and manipulate. When users become more familiar with OWL Lite and the ontology requires more powerful features, users can smoothly upgrade ontology from OWL Lite to OWLDL. For those who want even more expressive power, the process of upgrading from OWL DL to OWL FULL is very similar. However, users should consider the increasing complexity before upgrading. Especially for OWL Full, since it is undecidable, it is difficult to find an efficient reasoner that can generate correct answers.

REFERENCES

[1] McBride, P.H.a.B., *RDF Semantics.* W3C,2004.

[2] Mike Dean, G.S., Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider and Lynn Andrea Stein, *OWL Web Ontology Language Reference*, ed. W3C. 2004.

[3] Ian Horrocks, P.F.P.-S., Harold Boley,SaidTabet,BenjaminGrosof and Mike Dean, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* available at http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/,2004.

[4] Horrocks, I., *DAML+OIL: a Description Logic for the Semantic Web.* the IEEE Computer Society Technical Committee on Data Engineering, 2002. 25(1): pp.4-9.

[5] Michael K. Smith, C.W.a.D.L.M., *OWL Web Ontology Language Guide.* W3C,2004.

[6] Becker, M.Y. *Godel's Completeness Theorem*.in *Computer Laboratory University of Cambridge*.28.2.2003.

[7] Seaborne, A., *Jena Tutorial A Programmer's Introduction to*

RDQL. HP Labs,2002.

[8] Richard Fikes, P.H., and Ian Horrocks, *OWL-QL – A Language for Deductive Query Answering on the Semantic Web*.KSL 03-14, 2003.

[9] Voronkov, A.R.a.A., *Vampire 1.1*. IJCAR, 2001. LNAI 2083: pp.376-380.

[10] Palmer, S.B., *The Semantic Web: the introduction.* W3C, 2001.39

[11] Frank Manola, E.M.a.B.M., *RDF Primer*. W3C,2004.

[12] Brian McBride, J.G.a.D.B., *RDF Test Cases.* W3C, 2004.

[13]]D. McGuinness and F van Harmelen (eds) OWL Web Ontology Language Overview http://www.w3.org/TR/2003/WD-owl-features-20030331/

[14] M. Dean, G. Schreiber (eds), F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, . Patel-Schneider, L. Stein, OWL Web Ontology Language Reference http://www.w3.org/TR/2003/WD-owl-ref-20030331/

[15] M. Smith, C.Welty, D. McGuinness, OWL Web Ontology Language Guide http://www.w3.org/TR/2003/WD-owl-guide-20030331/P. Patel-Schneider, P. Hayes, I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax http://www.w3.org/TR/2003/WD-owl-semantics-20030331/