Study on Ontology Markup Languages for Building Domain Ontology

Myat Su Wai, May Aye Chan Aung
University of Computer Studies, Mandalay
missmyatsuwai@gmail.com , mayayechanaung@gmail.com

Abstract

Ontologies are set to play a vital role in the Semantic Web, e-Commerce, Bio-informatics, Artificial Intelligence, Natural Language Processing, and many other areas by providing a source of shared and precisely defined terms. Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. There are several types of ontology and ontology markup languages. Domain ontologies are reusable vocabularies of the concepts within the domain and their relationships. Domain ontology may also be used to define the domain. When data is marked up using ontologies, software agents can better understand the semantics and therefore more intelligently locate and integrate data for a wide variety of tasks. Many research areas have been increased about ontology, ontology mapping and ontology markup languages. In this paper, we study on several types of ontology markup languages for building domain ontology with example domain.

***Keyword**: Ontology Markup Language, Domain Ontology, Knowledge Representation*

1. Introduction

Today's world is a world of information. Many tasks in the area of information technology require the representation of world knowledge. The history of artificial intelligence shows that knowledge is critical for intelligent systems. In many cases, better knowledge can be more important for solving a task than better algori-thms. Ontology is increasingly seen as a key factor for enabling interoperability across heterogeneous systems and semantic web applications. Ontology differs from a database schema in that it provides more formal expression than a database schema. A schema is thus more restricted, it is generally used for a specific database (not reusable), it does explicit semantic for the data, whereas ontology do.

Often, they can be seen at different level of abstraction where ontology is at higher level. Ontology is an explicit specification of the conceptualization. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. In Computer Science, Ontology is a data model that represents a set of concepts within a domain and the relationships between these concepts. It is used to reason about the objects within that domain. Ontologies are used in Artificial Intelligence, Semantic Web, Software Engineering, Biomedical Informatics and Information Architecture about the world or some part of it. Ontology allows users to organize information into taxonomies of concepts, each with their attributes, and then describe relationship between concepts. When data is marked up using ontologies, not only human but also software agents can better understand the semantics and therefore more intelligently locate and integrate data for a wide variety of tasks. Moreover, there are many types of ontology markup language languages. In this paper, Simple HTML Ontology Extension (SHOE), XML based Ontology Exchange Language (XOL), Resource Description Framework (RDF) and RDF Schema, Ontology Interface Layer (OIL), DARPA Agent Markup Language (DAML-OIL) and Web Ontology Language (OWL) will be studied [12] [14].

2. Domain Ontology

A model of a domain is often specified in two parts, terminological and extensional. The first part of the model identifies the concepts in the domain and relations between them. The second part populates the model with facts about specific individuals in the domain. Domain ontologies are reusable vocabularies of the concepts within a domain and their relationships, of the activities taking place in that domain, and the theories and representative ontologies in the domain of e-commerce, medicine, engineering, enterprise, chemistry, and knowledge management are presented [1] [12].

For example, ontology is considered for Myanmar domain. States and divisions are designed as concepts (classes). Myanmar is also concept (class). Myanmar class has two sub-classes; State and

Division. They are related each other with *has* relation. State class and Division class are not the same. State has seven states; Kachin, Kayah, Kayin, Chin, Mon, Rakhine, and Shan. And Division has seven divisions, Yangon, Mandalay, Sagaing, Bago, Magway, Tanintheryi, and Ayeyarwady. These states and divisions subclasses of State and Division. They also have *has* relation.

Each state and division has five attributes (instances or individuals); Geography, Economy, Popularity, Capital, District city. They are also classes. In other word, Kachin class has class Geography, class Economy, class Popularity, and class Capital, class District city. They are subclass and instances of Kachin class. Kachin are also subClass of State and typeOf State. These are shown in annotation of this class. And Kachin and these attributes are related with *has* relation. They are defined in *Object property annotation*. Geography has three subclasses- Location, Climate, and Foresty. Economy has also three subclasses- Agriculture, Transport, and Resources and power. Popularity has two subclasses - Ethnic and Population and Ethnic has three subclasses; Religion, Spoken language and Settlement pattern. Kachin has capital "Myitkyina". It is defined in *Data property annotation*. All these subclasses are also instances (individuals) of all superclasses respectively.

All subclasses and superclasses are related with *has* relation. In the same way, all other states and divisions have these subclasses (also called attributes or instances) and these relations respectively.

These features are different in SHOE, and XOL. SHOE and XOL based ontology are written in <HTML> tag and <module> tag respectively. These languages are not RDF based languages, so they do not include RDF resources and features.

3. Ontology Markup Languages

Several types of Ontology markup languages have been developed in the areas of Semantic Web, and other applications. Choosing suitable markup languages are important for building ontology. Not all the existing languages have the same expressiveness. Knowledge Representation (KR) ontology implementation languages are diverse: frames, description logic, first and second order logic, semantic networks, etc. This fact makes even more important the correct selection of the language in which the ontology is to be implemented.

All language descriptions will be divided into two main dimensions, which are strongly related to each other: knowledge representation and reasoning mechanisms [[14]. We will study on the main features that each dimension presents. We will study of ontology markup languages for building domain ontology in the following sections.

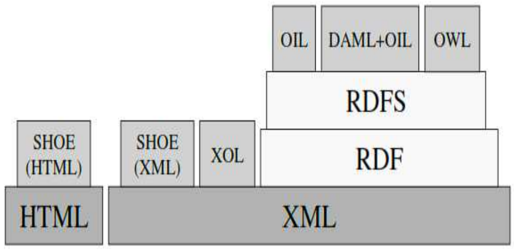

Figure 1 the Stack of Ontology Markup Language

3.1. SHOE

SHOE stands for Simple HTML Ontology Extension. It was developed at the University of Maryland in 1996. SHOE was created as an extension of HTML with the aim of incorporating machine-readable semantic knowledge in Web documents.

SHOE combines features of markup languages, knowledge representation, Datalog, and ontologies in an attempt to address the unique problems of semantics on the Web. It provides specific tags for representing ontologies. As these tags are not defined in HTML, the information inside them is not shown in standard Web browsers. There is also a slight variant of the SHOE syntax for XML compatibility [1] [4].

SHOE ontologies can import terms from other SHOE ontologies with the <USE-ONTOLOGY> tag. Inside this tag we must explicit the name and version of the ontology to be imported, the prefix used to refer to concepts of the imported ontology, and the URL where the ontology is available.

The SHOE base ontology contains data-types like STRING, NUMBER and Date, and top-level concepts and domain ontology about units. The SHOE base ontology should not be mistaken for Knowledge Representation (KR) ontology because it does not provide a formal definition of SHOE KR primitives [9]. SHOE based ontology for Myanmar domain are as follow;

```
<HTML>
…
<BODY>
<ONTOLOGY ID="myanmar-ontology" VERSION="1.1"
        BACKWARD-COMPATIBLE-WITH="1.0">
<USE-ONTOLOGY ID="domain-ontology"
VERSION="1.0" PREFIX="d"
        URL= "http://ontlib.org/dom_v1.0.html">
<DEF-CATEGORY NAME="State " ISA= "Myanmar">
```

```
<DEF-RELATION NAME="hasstates">
<DEF-ARG POS=1 TYPE="Kachin">
</DEF-RELATION>
<DEF-RENAME FROM="m.State" TO="State">
</ONTOLOGY>
</BODY>
</HTML>
```

3.2. XOL

XOL stands for XML-base Ontology exchange Language. It was designed, in 1999. XOL ontologies are written in XML. XOL does not provide Knowledge Representation ontology. This knowledge model allows representing classes, class taxonomies, slots, facets and individuals. [9].The header of XOL ontology includes only information about ontology's name and version. In XOL concepts are called classes. They are represented using the class tag. XOL Ontology header and class for Myanmar domain are shown in below.

```
<module>
   <name>Myanmar Domain Ontology</name>
   <version>3.0</version>
   <documentation>Sample ontlassology for
      Myanmar domain</documentation>
   <package>user</package>
   …
</module>
<class>
   <name>State</name>
   <documentation>Seven states of
      Myanmar</documentation>
   </class>
```

Unlike most of the languages already studied, XOL does not permit importing definition from other ontologies.

3.3. RDF and RDFS

RDF stands for Resource Description Framework. It was developed by the World Wide Web Consortium (W3C) to create metadata for describing Web resources. RDF 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. [5] [6].

RDF(S) can be used directly to describe ontology with its Objects, Classes, and Properties. RDFS offers a fix set of modelling primitives such as *rdfs:Class*, *rdf:Property* or the *rdfs:subClassOf* relationship to define RDF vocabularies for some specific application. In RDFS it is possible to define classes of classes, classes of properties, classes of literals that are strings, integers, booleans and so forth and classes of statements. Using RDFS properties, which are *rdf:type*, *rdfs:subClassOf* and *rdfs:subPropertyOf*, it is possible to define instanceOf relationship between resources and classes, subsumption relationship between classes and subsumption relationship between properties, respectively. Using *rdfs:domain* and *rdfs:range* properties it is possible to restrict the resources that can be subjects or objects of the property.

The expressive power of RDF is rather limited as intentional definitions or complex relationships via axioms can be defined. RDF/RDFS' subclass relation can be used to represent class subsumption. In Myanmar domain RDF class is shown in the <rdf:Class> tag;

```
<rdfs: Class rdf:ID=”State”>
  <rdfs:comment>Seven states of Myanmar
  </rdfs:comment>
</rdfs: Class>
```

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 either be used or followed to learn about the language which makes it very widespread.

3.4. OIL

OIL stands for Ontology Interchange Language and Ontology Inference Layer. It was developed in the context of the European IST project On-To-Knowledge. Like the other languages previously presented, for example, SHOE and RDF(S), OIL was built to express the semantics of Web resources [7].

OIL is Web based Knowledge Representation (KR) language that combines; (a) XML syntax; (b) modeling primitives from the frame-based KR paradigm, and (c) the formal semantics and reasoning support of the description logics (DL) approaches. Thus, OIL can be defined as a frame-based language that uses DL to give clear semantics and also to permit efficient implementations of reasons for the language [1]. In OIL, functions are defined in slots. The function must be binary OIL slots have only two arguments; one input and one output. The definition of this slot would be as follow;

```
slot-def has State
  domain Myanmar
  range Number
  properties functional
```

The OIL KR ontology can be distinguished four layers: Core OIL, Standard OIL, Instance OIL and

Heavy 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, and 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. Core OIL coincides with RDF Schema, except for the reification features of RDFS.

3.5. DAML-OIL

DAML-OIL was developed by a joint committee from the USA and the European Union (mainly OIL developers) in the context of the DARP project DAML (DARPA Agent Markup Language). The main purpose of this language is to allow semantic markup of Web resources [1].

DAML-OIL ontologies are written in XML (no plain text syntax, as in the case of OIL). And they can also be written with the triple notation for RDF. DAML-OIL ontologies are based on RDF(S). Therefore, ontology in DAML-OIL must start with the declaration of the RDF root node. In this root will include the namespaces for the RDF, RDFS and DAML-OIL KR ontologies. When DAML-OIL was created RDF(S), it did not support XML Schema datatypes. [3] [19].

For Myanmar domain, class and restriction of DAML-OIL are written in <daml:Class rdf:ID> tag as shown in below.

```
<daml:Class rdf:ID=”State”>
  <rdfs:comment>Seven states of Myanmar
  </rdf:comment>
  <daml:intersectionOf rdf:parseType=”daml:
     Collection”>
    <daml: Class rdf about=”#State”/>
    <daml:Restriction>
      <daml:toClass rdf resource= “&xsd:St”/>ing
      <daml:cardinality>1/<daml:cardinality>
    </daml:Restriction>
</daml Class>
```

Its reasoning capabilities are useful for ontology sharing. Regarding interoperability, it allows the partial definition of mapping rules. Reasoning in DAML+OIL is especially 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.

3.6. OWL

OWL stands for Web Ontology Language. It is the result of the work of the W3C Web Ontology Working Group. This language derives from and supersedes DAML-OIL. [1].

Like DML-OIL, OWL is built upon RDF(S). Therefore, some RDF(S) primitives are reused by OWL, and OWL ontologies are written eitherin XML or with the triples notation for RDF [10].

As OWL is derived from DAML-OIL, it also shares many features with that language. The main differences between OWL and DAML- OIL are the following:

1. OWL does not include qualified number restrictions.
2. OWL permits defining symmetric properties, which were not considered in DAML - OIL.
3. OWL does not rename the RDF(S) primitives reused by the language, as happened in DAML - OIL.
4. In OWL many DAML - OIL primitives have been renamed.
5. OWL does not include the primitive <daml:disjointWith>.

OWL class and RDF resource are as follow;

```
<owll:Class rdf:ID=”State”>
  <rdfs:comment>Seven states of Myanmar
       </rdf:comment>
   <rdfs:subClassOf >
    <owl:Restriction>
<owl:onProperty rdf:resource=”#geography”/>
      <owl:allValueFromrdf:resource=”&xsd;String”/>
 <owl:cardinality
Rdf:dataType=”&xsd;nonNegativeInteger>
     1/<owl:cardinality>
   </owl:Restriction>
</owl Class>
```

An ontology in OWL starts with the declaration of the RDF root node. In this node must include the namespaces for the RDF, RDFS and OWL KR ontologies. If XML Schema datatypes are used, it may be helpful to include a namespace for XML Schema [1] [15]. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL helps to define semantic specifications for applications that exploit KR and reasoning. 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 [17]. Figure 2 shows Myanmar domain ontology that is built by using OWL with Protégé 4.1.

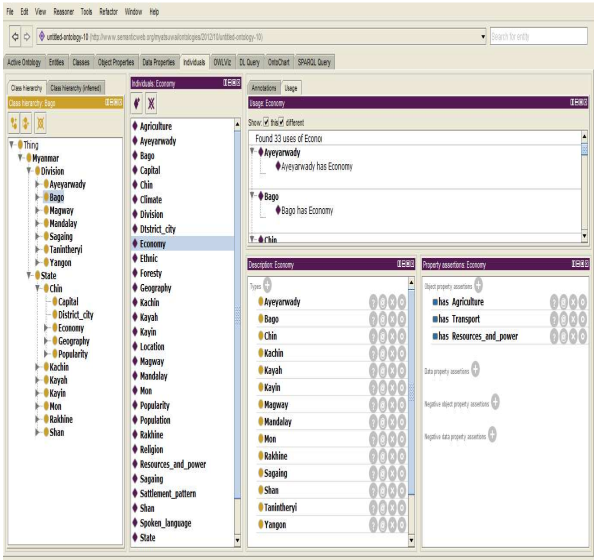

Figure 2 OWL Ontology for Myanmar Domain

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 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 OWL makes the domain ontology more flexible and allowing extensibility and reuse.

4. Comparative Results

Ontology Markup Languages are compared according to their supported features such as instance attribute, class attribute, type constraint, etc. This comparison helps to understand better similarities and differences between these languages and the capabilities of each of them.

Figure 3 shows the comparative results of these languages. Cells in the table are filled using '+' to indicate that it is a supported feature in the language, '-' to indicate for not supported feature and 'W' for not supported feature that can be supported with some workarounds.

	SHOE	XOL	RDF(S)	OIL	DAML-OIL	OWL
Instance attribute	+	+	+	+	+	+
Class attribute	-	+	W	-	W	W
Type constraint	+	+	+	+	+	+
Cardinality constraint	-	+	-	+	+	+
Subclass-Of	+	+	+	+	+	+
Disjoint-Decompositions	-	-	-	+	+	+
Exhaustive-Decompositions	-	-	-	W	W	W
Partition	-	-	-	+	+	W
Integrity constraints	-	-	-	-	-	-
Binary relations	+	+	+	+	+	+
n-ary relations	+	W	W	W	W	W
Relation hierarchies	-	-	+	+	+	+
Binary functions	-	W	-	+	+	+
n-ary functions	-	-	-	-	-	-
Facts	+	+	+	+	+	+
Instances	+	+	+	+	+	+
Procedures	-	-	-	-	-	-
Rules	-	-	-	-	-	-
Inferences	-	-	-	+	+	+

Figure 3 Comparison of Supported Features of Ontology Markup Languages

5. Conclusions

In this paper, I have been studied six types of ontology markup languages for building domain ontology. They have different types of features (primitives), supporting tools and expressive power. SHOE was built on extension of HTML. Later, it was built on XML. XOL does not permit importing definition from other domain ontology. OIL, DAML-OIL and OWL are based on RDF and RDF Schema and their features are derived from RDF(S). RDF(S) is widely used as a representation format in many tools and languages for building domain ontology. The OWL Web Ontology Language facilitates greater machine interpretability of Web content than that supported by XML, The data model of OIL, DAML-OIL's and OWL is based on description logic and Frame-based logic. They provide richer constructors for forming complex class expressions, properties and axioms for enabling reasoning data when domain ontology is constructed.

OWL helps to define semantic specifications for domain ontology that exploits KR and reasoning.

Today, OWL ontology is the most popular and widely used ontology representation language for building because of its precise semantic, better expressive power and reasoning mechanisms. There are many open source supporting tools such as Protégé and OntoEdit with OWL for building domain ontology. These tools provide supports for precise semantic and reasoning. Therefore OWL is more convenient for building domain ontology rather than many other types of ontology markup languages.

References

[1] A Pérez, M. F. Lopez and O. Corcho. Ontological Engineering: with examples from the areas of knowledge management, e-commerce and the semantic web. London. Springer-Verlag, 2004.

[2] Bhaskar Kapoor, A Comparative Study Ontology Building Tools for Semantic Web Applications, Department of Information Technology, MAIT, New Delhi INDIA

[3] Deborah McGuinness Gates , DAML-ONT: An Ontology Language for the Semantic Web Building 2A Stanford University, Stanford, CA 94305, USA, 650-723-9770

[4] Farheen Siddiqui, Web Ontology Language Design and Related Tools: a survey, Hamdard University, India.

[5] Frank Manola and Eric Miller, RDF primer. W3C Working Draft, 23 January 2003

[6] Graham Klyne, Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and abstract syntax, W3C Working Draft, 2003.

[7] Grigoris Antoniou, Enrico Franconi, and Frank van Harmelen, Introduction to Semantic Web Ontology Languages, Faculty of Computer Science, Free University of Bozen–Bolzano, Italy, Department of Computer Science, Vrije Universiteti Amsterdam, Netherlands

[8] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. In S. Staab and R. Studer (Eds), Handbook on Ontologies in Information Systems, Springer 2003

[9] Heflin J, Hendler J, Luke S, Qin Z, 1999, SHOE: A Knowledge Representation Language for Internet Applications, Institute for Advanced Computer Studies, University of Maryland at College Park.

[10] Ian Horrocks, Peter F. Patel Schneider, and Frank van Harmelen; From sHIQ and RDF to OWL: The Making of a Web Ontology Language.

[11] I. Horrocks, "DAML-OIL: a reasonable web ontology language", Lecture Notes in Computer Science (LNCS), Vol. 2287, Springer-Verlag, Berlin, 2002, pp. 2–13.

[12] Janez Brank, Marko Grobelnik, Dunja Mladenic ; A Survey of Ontology Evaluation Techniques, Department of Knowledge Technologies, Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia Tel: +386 1 4773778; fax: +386 1 4251038.

[13] Klyne G, Carroll J, 2004, Resource Description Framework (RDF): Concepts and Abstract Syntax, World Wide Web Consortium.

[14] Myat Su Wai, Information Retrieval System for Myanmar Domain Ontology, Master of Computer Science, AICT Conference, 2013.

[15] M.K. Smith, Chris Welty and D.L. McGuinness. OWL Web Ontology Language Guide

[16] N.F. Noy and D.L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, March 2001.

[17] P.F. Patel-Schneider, P. Hayes and Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax.

[18] R. Karp, V. Chaudhri, J. Thomere, XOL: An XML-Based Ontology Exchange Language, technical report, 1999.

[19] Wroe, Robert Stevens, Carole Goble, Angus Roberts, Mark Greenwood; A Suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data, Chris Department of Computer Science and School of Biological Sciences, University of Manchester, Oxford Road, Manchester, M13 9PL wroec, Robert. Stevens, Carole, Angus, greenwood @cs.man.ac.uk.