A Brief Survey of Enabling Components for Ontology Development Framework

Gurinder P. S. Gosal[1], Neeraj Sharma[1]

[1]*Department of Computer Science, Punjabi University, Patiala*

ABSTRACT

The ontologies have become a key tool of data integration and knowledge representation in different domains of interest. A large number of applications based on ontologies have been successfully built in last two decades by utilizing domain knowledge contained inside some well conceptualized, designed and developed ontologies. The field of ontologies have seen a huge progress from some initial random efforts of integrating domain knowledge to the development of thousands of ontologies in multiple domains. The upsurge in the development and wider acceptance of ontologies has further led to the development of a large number of enabling components, such as ontology languages, editors, reasoners etc., in ontology development environment. In the present paper we survey briefly some of these enabling components used frequently in the development of ontologies.

1 INTRODUCTION

The term ontology originated from a branch of philosophy called metaphysics which defines it as a systematic way of existence and computer science borrowed this definition from it (1). The origin of ontology in the field of computer science can be traced back to 1990's, when there were efforts for devising the new ways of building knowledge based systems by using the knowledge extracted from the reusable components. The DARPA Knowledge Sharing Effort (2) was an important initial attempt in this direction.

World Wide Consortium (W3C) (*http://www.w3.org*) has defined ontology in the following words: *"An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used to classify the terms used in a particular application, characterize possible relationships, and define possible constraints on using those relationships"* (3). One of the simplest and frequently used definitions of Ontology is given by Thomas Gruber in his extensively cited paper "*Toward Principles for the Design of Ontologies Used for Knowledge Sharing*". Gruber defines ontology in simple terms as an explicit specification of a conceptualization (4). Ontology could be seen as a specification of a common representational vocabulary, in terms of classes, relations, axioms, functions, and other related components, to support the sharing and reuse of formally represented knowledge for a shared domain of discourse (1,5).

The reasons for building ontologies in a particular domain are to make a common understanding of the structure sharable among different stakeholders in the domain, to enable reuse of the domain knowledge, to explicitly describe the domain, delineate the domain knowledge from operational commitment, and to explore the domain knowledge (6). Ontologies enable the integration, mining, and reasoning over diverse data sets by conceptually representing knowledge, which makes them distinct from the relational databases (7). This is often seen that the data sources of interest to a community in a specific domain are often large, dissimilar in structure, format and content. These data sources are frequently distributed across many resources, separately controlled and rapidly changing.

The data integration from different sources is not an easy task as there can be disparity in the definitions of schema of these sources as well as in the naming conventions (8). Ontologies are frequently used to deal with the heterogeneity of database schemas of different information sources by providing a shareable, consistent and formal description of the semantics (9). The ontologies could be seen as an answer to the challenges in seamless integration of data from heterogeneous sources using different schema and naming methods.

The initiative that started by some inquisitive experts with the development of a few small ontologies using rudimentary tools and targeting restricted domains of interest has now progressed to a stage where a large number of comprehensive ontologies are being built by the established groups of researchers. These efforts of describing the different complex domains in terms of ontological knowledge are being largely supported these days by the primary funding agencies in different countries. A number of collaborative efforts have been started to streamline the process of ontology development, curation, knowledge sharing and dissemination. The upsurge in the building and the wider acceptance of ontologies has further led to the development of a large number of enabling components for their development. In the next sections of the paper, we look at the ontologies languages, ontological tools such as editors and reasoners that enable the framework of development for ontologies.

2 ONTOLOGY LANGUAGES

Like other languages in computer science, ontologies for their creation also need some sort of formal language consisting of a set of specific symbols together with a sets of specific rules. Ontology languages, mostly declarative languages having their base in first-order logic or on description logic, enable the coding representation of domain knowledge and reasoning rules for describing different domains. These ontology languages have been classified on the basis of different criteria such as syntax based, structure or logic based. The traditional syntax and markup ontology languages are examples of syntax based category whereas description logic and first-order logic based ontology languages are examples of structure or logic based category of classification (10). KIF (Knowledge Interchange Format) (11), DOGMA (Developing Ontology-Grounded Methods and Applications) (12) and OCML (Operational Conceptual Modelling Language) (13) are some key ontology languages developed as traditional syntax languages. DAML+OIL (14), Resource Description Framework (RDF)(15,16), RDF Schema (RDFS) (17) and Web Ontology Language (OWL) (18) are well known languages placed under the category of markup ontology languages.

World Wide Consortium W3C (19) in 2004 recommended Web Ontology Language (OWL) as a key ontology language that has been consolidated further upon the RDF and RDFS languages and since then, OWL has made its mark as a standard ontology language for the Semantic Web. Here we look at RDF and RDFS first to understand them in the context of development of OWL and then at OWL in terms of its different flavors and progression.

2.1 Resource Description Framework (RDF) and RDF Schema(RDFS)

Resource Description Framework (RDF) has been developed by World Wide Web (W3C) Consortium as a simple metadata data model for describing and creating relationships among resources. In 1999, the consortium published a W3C Recommendation specifying the RDF's data model and in 2004 they came up with a set of related specifications as new version of RDF data model. In RDF the information is represented in a minimally restrictive way and RDF's simplification offers greater sharing. An object is defined as a resource in RDF and Uniform Resource Identifier (URI), a formatted string used for identifying abstract or physical resources, is used to uniquely identify it. RDF has been defined with a goal of having a simpler data model, an extensible URI-based vocabulary, using an XML-based syntax and supporting the use of XML schema data types and having formal semantics that provide a reliable base for reasoning about the meaning of the RDF expressions (20).

The basic structure of any expression in RDF has been kept very simple and is made up of a collection of triples. Each RDF triple is made up of:

- A subject,
- A predicate, also called property and
- An object

An RDF graph consists of set of such triples. The set of nodes of an RDF graph represents the set of subjects and objects of triples in the graph. **Figure 1** shows a triple consisting of subject, a gene *AAK1* represented as a URI, having a relationship (predicate) *associatedWith* with an object representing a type of cancer having value *haematopoietic_neoplasm.*

The complete RDF triple is:

http://punjabiuniversity.ac.in/procdio/1.0/AAK1><http://punjabiuniversity.ac.in/procdio/1.0/associatedWith>

<http://punjabiuniversity.ac.in/procdio/1.0/Cancer-haematopoietic_neoplasm>

Figure 1: A RDF statement depicting a triple graphically

The forward direction in the realization of Semantic Web vision has been the introduction of the Resource Description Framework Schema (RDF-S). RDFS, an extensible knowledge representation language, allows resources to be defined in terms of classes, properties and values. RDFS permits the defining of the restrictions and extra relationships that is not allowed in the case of RDF. RDFS enables the organization of classes in a hierarchical fashion and the properties in RDFS also could be inherited from other properties. Further the properties in RDFS could be applied with restrictions such as domain and range constraints.

2.2 The Web Ontology Language (OWL)

The Web Ontology Language (OWL) has been developed by revising the Web ontology languages DAML and OIL to enable richer knowledge representation. OWL derives many concepts from RDFS to make it richer as a Semantic Web language. OWL provides greater expressive power as compared to earlier recommendations of XML, RDF, and RDFS and supports developing ontologies that are explicit representations of terms and their interrelationships.

There are three sublanguage groupings of OWL (shown in **Figure 2**) that are differentiated from each other based on the amount of expressive power they provide.

2.2.1 OWL-Lite

This sublanguage of OWL supports the community of users whose requirements are limited only to representation of a simple class hierarchy and implementation of simple constraints. Therefore, this genre of OWL sublanguage has the least expressive power and it is easier to provide tools for supporting OWL-Lite because of the lesser complexity involved in its realization.

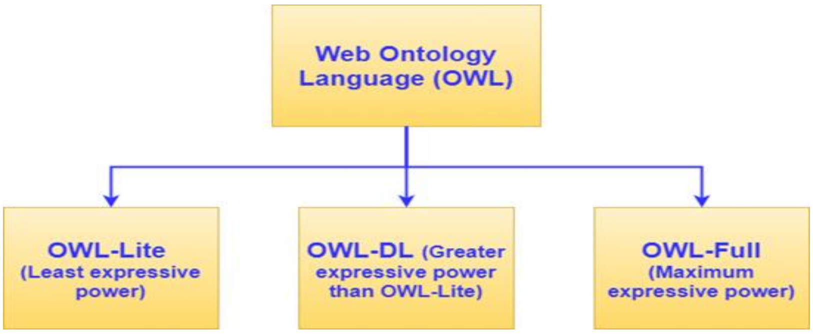

Figure 2: Sublanguages of Web Ontology Language (OWL)

2.2.2 OWL-DL

This sublanguage of OWL makes use of description logic and that is how it gets its name of OWL-DL. It has greater expressive power than OWL-Lite and caters to the need of user community which requires a highly expressive language that is at the same time computationally complete and decidable. The computational completeness means that all computations can be completed in finite time and decidable means that all conclusions are computable.

2.2.3 OWL-Full

The maximum expressive out of the all OWL sublanguages is provided by OWL-Full and it gives the maximum syntactic freedom of RDF. The computational completeness is not guaranteed by this sublanguage. The automated reasoning on OWL-Full ontologies is not possible as to provide different conceptualizations defined in OWL-Full in terms of reasoning tools is not practically feasible.

The choice of use of a specific sublanguage of OWL is clearly driven by the needs of the user. The users might prefer OWL-DL to OWL-Lite in case they need greater expressive power whereas user requiring more meta-modeling capabilities might opt for OWL-Full.

3 ONTOLOGY DEVELOPMENT ENVIRONMENT TOOLS

There are a number of tools designed on different enabling technologies that have been extensively used by ontology developers in the development environment of ontologies. Ontology editors and reasoners are examples of these enabling tools for creation of robust and logically sound ontologies.

3.1 Ontology editors

To provide the users with minimum required functionality in the ontology development and maintenance task, and help them to create, code, browse, examine and manipulate the ontologies visually, a number of ontology editors have been made available in the public domain (21). Many editors out of these have made a respectable place and have been in widespread use amongst the Semantic Web communities as shown in **Table 1**.

Ontology editor **Protégé** (*http://protege.stanford.edu/*) is a free, open-source software system that has become a very popular tool for constructing knowledge-based applications with ontologies that provides a suite of tools to the user community. There are many knowledge-modeling compositions and procedures implemented by Protégé which facilitate in creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications (22). The Protégé-OWL editor provides users support to load and save OWL and RDF ontologies, edit and visualize classes, properties, define logical class characteristics as OWL expressions and execute reasoners such as description logic classifiers (23).

Table 1: Some commonly used open source ontology editors

Ontology Editor	Provider Institution	Availability
Protégé	Stanford Center for Biomedical Informatics Research..	https://protege.stanford.edu/
NeOn Toolkit	The NeOn Foundation	http://neon-toolkit.org/wiki/Download.html
SWOOP	Formerly: The University of Maryland Presently: Clark &Parsia, IBM Watson Research and University of Manchester.	https://github.com/ronwalf/swoop
Neologism	DERI	http://neologism.deri.ie/
Vitro	Anonymous	https://github.com/vivo-project/Vitro
Knoodl	Revelytix, Inc.	http://knoodl.com/ui/home.html
OWLGrEd Fluent Editor	IMCS UL	http://owlgred.lumii.lv/
Semantic Turkey	University of Rome Tor Vergata	http://semanticturkey.uniroma2.it/
VocBench	University of Rome Tor Vergata	http://vocbench.uniroma2.it/

The **NeOn toolkit** is an open source multi-platform ontology engineering environment that is based on the Eclipse platform and provides an extensive set of plug-ins for different ontology engineering activities (24). **Swoop** is a hypermedia inspired Web Ontology Browser and Editor tailored specifically for OWL ontologies and has been based on the UI paradigm (25). Knoodl is a system by Revelytix containing tools for creating, managing, analyzing, and visualizing RDF/OWL descriptions and which has been hosted in the Amazon EC2 cloud for free use (26).

Vitro is another web-based ontology editor for developing and loading OWL ontologies, building a public website to display data and searching data using Apache Solr that has been firstly developed at Cornell University (27). **Neologism** is asimple Web-based editor and publishing system for RDF Schema vocabularies coded in PHP and developed on the Drupal platform that has hosted many popular vocabularies (28).OWLGrEd Fluent Editor, Semantic Turkey and VocBench are some other popular ontology editors that enable the development, browsing and dissemination of ontologies.

3.2 Ontology reasoners

A reasoner, also called reasoning engine or rules engine or semantic reasoner, is the tool that takes a given set of asserted facts or axioms as an input and infers the logical consequences from it. For performing the reasoning many reasoners make use of first-order predicate logic. One of the basic usage of ontology reasoners is classification. Take a very simple example, in an onrtology O, X is an instance of Class1 as well of Class2 and these classes Class1 and Class2 also made disjoint classes. When the reasoner is applied on ontology O, reasoner will mark the ontology as inconsistent. The reasoners further help to check the model satisfiability and class subsumption of ontologies.Some of the open source ontology reasoners are Cwm, OpenRules, FaCT++ Reasoner, Pellet and HermiT as shown in **Table 2**(29).

Table 2: Some commonly used open source ontology reasoners

Ontology Reasoner	Provider Institution	Availability
Cwm (pronounced coom)	World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University)	http://www.w3.org/2000/10/swap/doc/cwm.html
OpenRules	OpenRules, Inc.	http://openrules.com/index.htm
FaCT++	Manchester University, UK	http://owl.cs.manchester.ac.uk/tools/fact/
Pellet	Clark &Parsia, LLC	http://pellet.owldl.com/
Hermit	Oxford University	http://www.hermit-reasoner.com/

Cwm (pronounced coom), a part of a Semantic Web Application Platform (SWAP), is a forward chaining reasoner used for querying, checking, transforming and filtering information (30)**. OpenRules** is an open source business rules and decision management system that includes sequential rule engine and an inferential rule engine (31). **FaCT++** is an open-source tableaux-based reasoner for expressive Description Logics (DL) that is used as one of the default reasoners in the Protege 4 OWL editor (32). FaCT++ has been initially developed together with Ian Horrocks within the WonderWeb project. **Pellet** is a OWL-DL reasoner with extensive support for reasoning with individuals (including nominal support and conjunctive query), user-defined datatypes, and debugging support for ontologies (33). **HermiT**is an OWL reasoner based on a novel “hypertableau” calculus that addresses performance problems due to nondeterminism and model size—the primary sources of complexity in state-of-the-art OWL reasoners(34).

4 CONCLUSION

Ontologies have become a very significant tool of data integration that help in representing the knowledge conceptually and thus benefiting communities in a variety of ways. The developers of the ontologies have used very rudimentary mechanisms for the development in the initial years of their evolution. However, with the greater interest in the development of ontologies in the recent years has given way to a large number of robust enabling components for the development of ontologies. In this paper the popular ontology languages and development environment tools, such as editors and reasoners, have been surveyed briefly to understand the enabling framework for ontology development.

REFERENCES

1. Gruber TR. A translation approach to portable ontology specifications. Knowledge Acquisition. 1993 Jun 1;5(2):199–220.
2. Gruber H-G. Does organisational culture affect the sharing of knowledge? the case of department in a high-technology company [Internet]. Available from: http://dx.doi.org/10.22215/etd/2000-04498
3. W3C: Ontologies in Semantic Web Context [Internet]. W3C: Ontologies in Semantic Web Context. [cited 2017 May 25]. Available from: http://www.w3.org/2001/sw/SW-FAQ#whont
4. Gruber TR. Toward principles for the design of ontologies used for knowledge sharing? Int J Hum Comput Stud. 1995 Nov 1;43(5):907–28.
5. Kramer F, Beißbarth T. Working with Ontologies. Methods Mol Biol. 2017;1525:123–35.
6. Noy NF, McGuinness DL, Others. Ontology development 101: A guide to creating your first ontology [Internet]. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, CA; 2001. Available from: http://www.corais.org/sites/default/files/ontology_development_101_aguide_to_creating_your_first_ontology.pdf
7. Spyns P, Meersman R, Jarrar M. Data Modelling Versus Ontology Engineering. SIGMOD Rec. 2002 Dec;31(4):12–7.
8. Antezana E, Kuiper M, Mironov V. Biological knowledge management: the emerging role of the Semantic Web technologies. Brief Bioinform. 2009 Jul;10(4):392–407.
9. Noy NF, Klein M. Ontology Evolution: Not the Same as Schema Evolution. KnowlInf Syst. 2004 Jul 1;6(4):428–40.
10. Corcho O, Gómez-Pérez A. A Roadmap to Ontology Specification Languages. In: Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Springer Berlin Heidelberg; 2000. p. 80–96.
11. Genesereth MR, Fikes RE, Others. Knowledge interchange format-version 3.0: reference manual. 1992; Available from: https://www.upv.es/sma/teoria/sma/kqml_kif/kif.pdf
12. Jarrar M, Meersman R. Ontology engineering--the DOGMA approach. In: Advances in Web Semantics I. Springer; 2008. p. 7–34.
13. Motta E. An overview of the OCML modelling language. In: the 8th Workshop on Methods and Languages [Internet]. Citeseer; 1998. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.9652&rep=rep1&type=pdf
14. Horrocks I, Others. DAML+OIL: A Description Logic for the Semantic Web. IEEE Data Eng Bull. 2002;25(1):4–9.
15. Lassila O, Swick R. Resource Description Framework (RDF) Model and Syntax Specification [Internet]. 1999. Available from: http://www.w3.org/TR/REC-rdf-syntax/
16. Vertan C, Merkmale RF. Resource description framework (rdf). 2004; Available from: https://nats-www.informatik.uni-hamburg.de/pub/SemanticWeb04/EinfuehrungsFolien/rdf.pdf
17. BRICKLEY, D. RDF Vocabulary Description Language 1.0 : RDF Schema. http://www.w3.org/TR/rdf-schema/ [Internet]. 2004 [cited 2018 Oct 14]; Available from: https://ci.nii.ac.jp/naid/10025277112/
18. Antoniou G, van Harmelen F. Web Ontology Language: OWL. In: Staab S, Studer R, editors. Handbook on Ontologies. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 67–92.
19. Herman I. About the World Wide Web Consortium (W3C). Documentação eletrô [Internet]. 2003; Available from: https://www.w3.org/
20. Resource Description Framework (RDF): Concepts and Abstract Syntax [Internet]. [cited 2018 Mar 17]. Available from: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
21. Alatrish ES. Comparison some of ontology. Journal of Management Information Systems. 2013;8(2):018–24.
22. Jena Ontology API [Internet]. [cited 2018 Mar 17]. Available from: http://jena.sourceforge.net/ontology/index.html
23. Stanford Center for Biomedical Informatics Research. protégé [Internet]. [cited 2018 Oct 21]. Available

from: http://protege.stanford.edu/

24. NeOn Wiki [Internet]. [cited 2018 Oct 21]. Available from: http://neon-toolkit.org/wiki/Main_Page.html
25. Kalyanpur A, Parsia B, Sirin E, Grau BC, Hendler J. Swoop: A Web Ontology Editing Browser. Web Semantics: Science, Services and Agents on the World Wide Web. 2006 Jun 1;4(2):144–53.
26. Knoodl [Internet]. [cited 2018 Aug 10]. Available from: http://knoodl.com/
27. VITRO - Integrated Ontology Editor and Semantic Web Application [Internet]. [cited 2018 Aug 10]. Available from: https://sourceforge.net/p/vivo/vitro/home/
28. Neologism: A simple web-based RDF Schema vocabulary editor and publishing system [Internet]. [cited 2018 Aug 10]. Available from: http://semanticweb.org/wiki/Neologism.html
29. Abburu S. A survey on ontology reasoners and comparison. Int J Comput Appl Technol [Internet]. 2012;57(17). Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.252.5532&rep=rep1&type=pdf
30. Cwm (pronounced coom) [Internet]. [cited 2018 Aug 10]. Available from: https://www.w3.org/2000/10/swap/doc/cwm.html#dev
31. Feldman J. Representing and Solving Rule-Based Decision Models with Constraint Solvers. In: Rule-Based Modeling and Computing on the Semantic Web. Springer Berlin Heidelberg; 2011. p. 208–21.
32. Tsarkov D, Horrocks I. FaCT++ Description Logic Reasoner: System Description. In: Automated Reasoning. Springer Berlin Heidelberg; 2006. p. 292–7.
33. Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y. Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web. 2007 Jun 1;5(2):51–3.
34. Shearer R, Motik B, Horrocks I. HermiT: A Highly-Efficient OWL Reasoner. In: OWLED. cs.ox.ac.uk; 2008. p. 91.