IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Ontology Editing Tools: A Comparative Perspective

D. Ramya[1*], V. Kiran Kumar [2]

[1]Computer Science, Dravidian University, Kuppam, India
[2] Computer Science, Dravidian University, Kuppam, India

*Corresponding Author: ramyadoma@gmail.com

Available online at: www.ijcseonline.org

Accepted: 17/May/2018, Published: 31/May/2018

Abstract: With the revolutionary advent of the Semantic Web and the corresponding technologies, Ontologies have lately gained tremendous patronage and popularity from a large cross-section of users from the realm of knowledge management. This paper seeks to make an insightful comparative analysis of a few select software tools, associated with 'Semantic Web'. As a matter of fact, this study attempts an insightful comparative description of six ontology-editors, viz. Protégé, Swoop, Vitro and Fluent. Also, it describes the key structural aspects of the above editors, including its basic features, apart from the methods of use. In essence, convenience for the user as well as the multiple possibilities of applications is considered as the primary criterion for comparison among these editors. Diverse user groups prefer to use Ontology Building/Management tools while handling several tasks. Even though, each tool defines a specific functionality, many users prefer to use it only for a single purpose for migration within their host ontologies, from one tool to another. Additionally, we evaluated the compatibility among various ontologies by applying many development and management tools. Finally, it detects the many similarities and diversities found among the analyzed ontologies, both within a particular domain (application area) as well as with other domains.

Keywords: Semantic Web, Intelligent Web, Ontology, Ontology building tools

I. Introduction

The semantic web is regarded as an interconnected network of shared information or distributed databases created on a global platform. It primarily seeks to ensure easy processing of data by machines. It endows many mechanisms capable of being applied for classifying information and defines the contextual setting for smart retrieval of information from the web. It is primarily attained by application of knowledge representation languages creating unambiguous domain formulations like ontologies. This formulation includes a set of concepts, their definitions, apart from their shared relationships. In fact, in recent times, we notice emergence of a lot of ideas and tools that have ensured the expansion of ontologies. In such as scenario, ontology offers the most trust worthy support for the many forms of information management tasks that include information retrieval, storage, and info sharing on the web.

The world still needs many software tools to escalate the growth in ontologies, though a range of open source and commercial tools are already in circulation for ensuring the development of Ontology Editors, as a case in point. Such tools have potential application in many stages of the ontology lifecycle, consisting of creation, implementation, and maintenance of ontologies.

II. Review of Literature

Kapoor and Sharma [1] conducted a study on "A Comparative study on ontology Building tools for Semantic web Applications". This study was carried out on existing ontology tools like protégé3.4, Apollo, Isaviz and Swoop. Those are open source and the review in terms of interoperability, openness, easiness to update and maintain, market status and penetration. This study reveals that each tool provides different functionalities, most of the users use only one because they are not able to interchange their ontologies from one tool to other, the compatibility of different ontologies with different development and management tools and also concern the detection of commonalities and differences between the examined ontologies both on the same domain and among different domains.

Altrish [2] examine "Comparision some of ontology editors". In this study he examined the some ontology editors like Apollo,OntoStudio,Protégé,Swoop and TopBraid Composer free edition. In this the structure and basic features are explained. The comparison done based on general description of the tools, software architecture and tool evaluation, interoperability of the tools, knowledge representation, inference services and the usability of the tools. The findings are ontology development is an ad-hoc approach. Among several alternatives user need to find which one is work better for projected task and which one is easily and effectively maintained and expressed.

Burag, Cojocaru and Nichifor [3] conducted a “Survey on Web Ontology editing tools”. In this focused on OilEd, OntoEdit, POWL, Protégé and Swoop. The comparison based on granularity of expressivity, web standards compliance, reasoning support, provided APIs and interoperability. The findings are some tools are intended for simple ontology development, others are complex and the best ontology suited for individual or academic study tools are protégé and swoop. While we need to develop large or complex architectures is OntoStudio.

III. Ontology-Building Method

Ontology represents a model belonging to an actual real time sphere, which ontological concepts typically reflect. Once the preliminary version of ontology is defined, the evaluation and debugging tasks are integrated into the applications by following either problem-solving methods or taking into account the opinion of experts. In fact, in subsequent phases, we are essentially required to make a revision of the early ontology, an iterative process expected to continue all through the complete lifecycle of the ontology. In order to develop ontology, the required building blocks or steps are enumerated below:

- Identifying the domain and its scope.
- Evaluation of reuse.
- Defining the key terms.
- Characterizing classes and class hierarchy.
- Determining class properties as well as constraints.
- Forming illustration of classes

Moreover, for construction of ontology, an ontology-specification language is required to be developed from the select list. As far as ontology languages are concerned, the Web Ontology Language (OWL) has a reputation as an authentic one engaged in expression and transmission of knowledge and information in the semantic web environment. OWL belongs to the class of Resource Description Framework (RDF) as well as the DARPA Agent Markup Language (DAML). In essence, OWL leverages the RDF definition of classes and properties to create certain potent modeling primitives for reiterating expressions. It also ensures a framework for syntactic conversion of the complete ontological references into the developing ontology.

IV. ONTOLOGY EDITING TOOLS

The ontology editors belong to classes of tools permitting users to make visual manipulation, inspection, browsing and coding of ontologies in order to augment ontology development and maintenance tasks. Here, we propose to project the summary of some existing ontology editor tools by briefly describing each of them, apart from highlighting their group, major properties and functions etc.

4.1 Protégé:

Protégé belongs to a class of free, open source java-based platforms offering a collection of tools to a dynamic user community who wish to develop domain models and knowledge-based applications applying ontologies. Protégé executes a prolific collection of knowledge modeling structures and actions for augmenting the conception, construction, and utilization of ontologies while working on various representation formats. It is possible to customize Protégé for providing domain-compatible support that creates knowledge models and data entry. Moreover, protégé may have extendable applications through plug-in architecture. In fact, some services can be expected to be programmed through application of the java API. The protégé specifically augments the process of building the ontologies, the largely frame-based ones as per the specifications laid down by Open Knowledge Base Connectivity protocols (OKBC). It also ensures compatibility with OWL ontologies, wherein a special plug-in helps in creating the representations of the editing ontologies in terms of graphs[2].

The protégé offers special advantages for time tool builders as well as knowledge engineers and domain specialists. The available tools seem to favour knowledge engineers more as they are found wanting in possessing the requisite flexibility for meta-modeling. In fact, Protégé helps construct domain ontologies by customizing data entry forms, apart from defining classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and the nature of relationships.

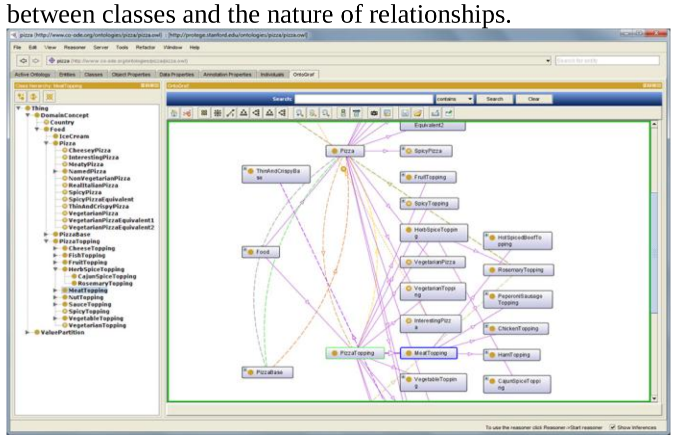

Figure 1: Screenshot of Protégé ontology editor

4.2 *Swoop:*

Swoop belongs to the category of open source, web-based OWL ontology editors and browsers, containing Owl validation. It also provides a syntactic view of many OWL presentations. It offers augmentation in terms of providing reasoning support for a manifold ontology environment in which entities and relationships of multiple ontologies is compared, edited and merged seamlessly. It also ensures a comparison between diverse ontologies by using Description Logic-based definitions, associated properties as well as classes. In Swoop, it is simple and easier to navigate due to the presence of hyperlinked properties. However, it doesn't pursue any specific methodology for ontology construction. The users get the chance to reapply external ontological data through simple linkages with the external entity, or when the total external ontology is imported. Significantly, Swoop applies ontology search algorithms by combining keywords with DL-based constructs to discover connected concepts in already available ontologies, apart from having mutual annotation with the Annotea plug-in[1,3].

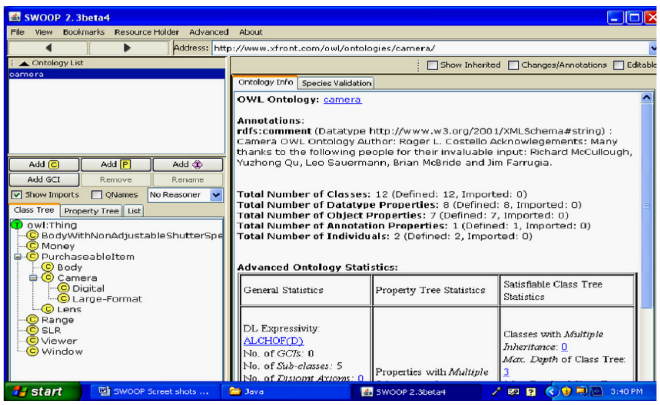

Figure 2: Screenshot of Swoop ontology editor

4.3 *Vitro:*

Vitro refers to a composite ontology editor that also displays features of a semantic web application. Originating from a research and scholarship initiative at Cornel University, today it is globally run as a java web application that operates on a tomcat Servlet Container. Using a Vitro, one can produce or insert ontologies into the OWL format, edit classes as well as associations. It can also develop a public web site for displaying our data, apart from searching our data in alignment with Apache Solar[6,7].

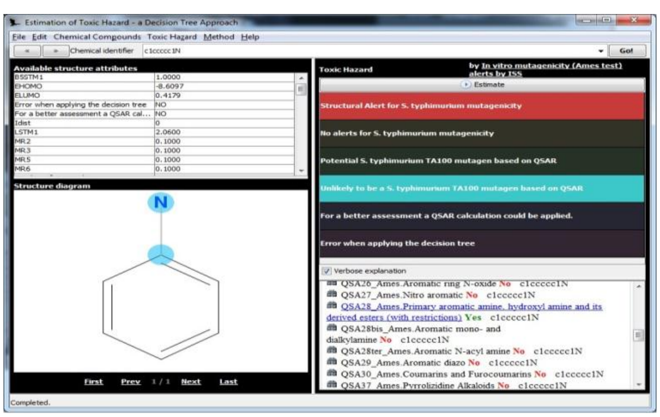

Figure 3: Screenshot of Vitro ontology editor

4.4 *Fluent:*

Fluent editor refers to a tool having applications in editing, modification and querying of complex ontologies expressed in OWL, RDF or SWRL. A Fluent Editor shares a complete compatibility with a majority of Semantic Web W3C standard (OWL, RDF, SPARQL, SKOS,..), while possessing intuitive user interface with the Ontorion Controlled Natural Language (OCNL). Though it is a human-friendly substitute for XML ontology languages like OWL or RDF, yet it offers total compatibility with OWL2, RDF and SWRL. Additionally, the OCNL has potential application as a query language having compatibility with SPARQL[8].

A Fluent Editor is armed with tools that enable the user to operate complex ontologies comprise the following: a reasoner window for generating queries to the ontology; a SPARQL window for executing SPARQL queries; an XML preview window to verify the reflection of written OCNL sentence in the corresponding OWL; a taxonomy tree to view; and finally, an annotation window. In addition, it provides two plug-ins viz., a Protege interoperability plug in (to export/import to/from Protege) and an R plug in using the ROntorionpakages to plot and list the content of the Ontology[9].

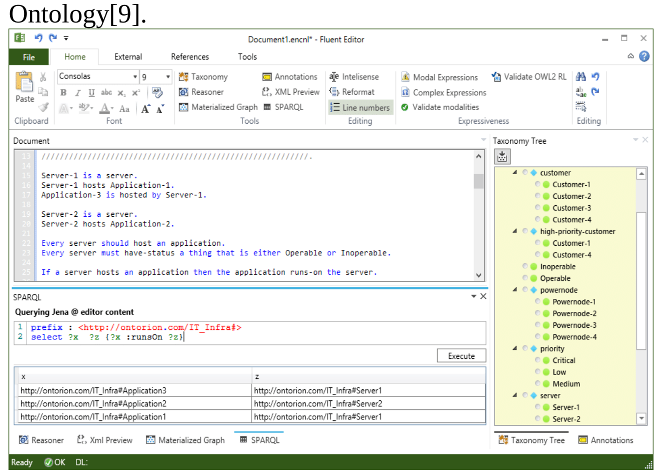

Figure 4: Screenshot of Fluent ontology editor

V. Comparison of Tools

The comments related to this section are derived from its relation to tools presented above. The general descriptions of tools include information about developers, apart from the availability factor.

Table 1: General description of the tools

Feature	Protégé	Swoop	Vitro	Fluent
Developers	SMI(Stanford University)	MND(University of Maryland)	Cornel University	Cognitum
Availability	Open Source	Open Source	Open Source	Open Source

Software Architecture and Tool evaluation includes information about the requisite platforms to use the above tools. The information provided below has three broad parts such as default architecture, extensibility and backup management.

Table 2: Software Architecture and Tool evaluation

Feature	Protégé	Swoop	Vitro	Fluent
Semantic Web Architecture	Standalone and client/Server	Web based and Client/Server	Standalone Java Web Application	Standalone
Extensibility	Plug-ins	Plug-ins	Plug-ins	Plug-ins
Backup Mangement	No	No	No	No
Ontology Management	Files and DBMS	As HTML Models	Files	XML models

Interoperability includes information about the exchange and interplay between the tools and other ontology development tools and languages, translations from certain ontology languages. It is an important feature of the integration of ontologies in various applications.

Table 3: Tools Interoperability

Feature	Protégé	Swoop	Vitro	Fluent
With other Ontology tools	PROMPT,OKBC ,JESS,Fact and Jena	No	Jena, Semantic Web	No
Imports from Languages	XML(S), RDF(S), OWL,HTML,text file, RDF file, Excel, DataMaster	OWL,XM L, RDF and text formats	XML, OWL,RDF(S)	OWL2,SWRL, RDF,SPARQL
Exports to Languages	XML(S),RDF(S), OWL,HTML,Jav a,Eclipse, F-Logic,OWLdoc	RDF(S),OI L,DAML	XML, OWL,RDF(S),SPARQL	OWL2,SWRL, RDF,SPARQL
Merging	Via Anchor point Plug-in	No	Yes	Yes

Inference Services includes built-in or other inference engines, constraint and consistency checking mechanisms, automatic classifications and exception handling among others.

Table 4: Inference Services of tools

Feature	Protégé	Swoop	Vitro	Fluent
Built-in Inference Engine	Yes,PAL	No	Yes	Yes
Other Attached Inference Engine	Racer,FACT,FACT ++, F-Logic and Pallet	Pallet	No	No
Consistant/Consistency Checking	Yes, Via plug-in	Only with reasoner plug-in	Only with reasoner plug-in	Yes
Exception Handling	No	Yes	No	Yes

VI. Conclusion

It is quite clear that ontology development is primarily an ad-hoc approach. Among several available alternatives, a user needs to find out which one would work better for the projected task as well as one that offers easy, effective maintenance and expression. Though the foundation of ontology is logical, yet it is a model of reality which envisages that the concepts of ontology must reflect this reality. In fact, we have suggested a tool-assisted method for building the basis for ontologies that are adopted from domain analysis.

References

[1] Bhaskar Kapoor, Savitha Sharma, " A Comparative Study Ontology Building Tools for Semantic Web Applications", International Journal of Web & Semantic Technology (IJWesT), 1(3), 2010.

[2] Emhimed Alatrish, "Comparison of ontology Editors. Management Information Systems", 8(2), 18-24, 2013.

[3] S.C.Buraga, L.Cojocaru,O.C. Nichifor, "Survey on web Ontology Editing Tools". Retrieved February 25,2018 from Alexandru loan Cuza University,2006.

[4] A.Funk,V.Tablan,K. Bontcheva,H. Cunningham, B.Davis,S. Handschuh, "Clone: Controlled language for ontology editing", Sixth International Semantic Web Conference, (ISWC). Berlin: Springer Berlin Heidelberg. (pp. 142-155).2007.

[5] A. Harith ,O. kieron, S.Nigel, "Common Features of Killer apps : A comparison with Protégé". 8th International Prot Conference, 18-21 July 2005. Madrin ,Spain.2005.

[6] https://sourceforge.net/p/vivo/vitro/home/

[7] vitro.mannlib.cornell.edu/

[8] www.cognitum.eu/Semantics/FluentEditor

[9] https://www.w3.org/2001/sw/wiki/Fluent_Editor

Authors profile

Mrs. D. Ramya pursed Master of Computer Applications from Sri Venkateswara University, Thirupathi in 2009 and Master of Technology in Computer Science Engineering from Jawaharlal Nehru Technological University, Ananthapuramu in 2016. She is currently pursuing Ph.D. and currently working as an academic Consultant in Dravidian University, Kuppam. Her research work focus on Semantic Web Technologies. She is having 7 years of teaching experience.

Dr. V. Kiran Kumar is working as a associate professor in the department of computer science, Dravidian University, Kuppam. He did his Ph.D. from Acharya Nagarjuna University in 2013. He has 18 years of teaching experience. His research areas are semantic web and web technologies. He has published good number of journals in the same areas.

