ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(24), DOI: 10.17485/ijst/2015/v8i34/54249, September 2015* ISSN (Online) : 0974-5645

Ontology Development: A Comparing Study on Tools, Languages and Formalisms

Thabet Slimani*

Department of Computer Science, College of Computer Science and Information technology Taif University, Thabet.slimani@gmail.com

Abstract

This paper reviews and compares some Ontology Development Tools, Formalisms and Languages from those reported in the Literature, with a special attention accorded to the interoperability between them. Additionally, this paper presents the Structure and Basic Features of Tools, Formalisms and languages. The main criterion for comparison of these tools and languages was the user interest and their application in different kind of real world tasks. The primary goal of this study is to introduce several tools and languages to ensure more understanding from their use. Consequently,we can solve the problems of current tools and languages and ensure the easy development of a new generation of tools and languages.

Keywords: Ontology Building Tools, Ontology Development, Ontology Formalisms, Ontology Languages, Ontology Tools, Ontologies

1. Introduction

Ontologies describing concepts and relations in a specific domain are necessary for knowledge representation and knowledge exchange[1]. Waterson and Preece[2] define ontology as "The Specification of Shared Knowledge". Ontologies (semantic data), facilitate the usability of e-technology and realize its full power. There are several Languages,Tools and Formalisms that accompany the term ontology. Among those languages,we enumerate XML (Extensible Markup Language),RDF (Resource Description Framework), RDFS (RDF Schema), DAML+OIL and OWL that are used in several contexts (compatibility with other concepts, expressiveness, etc.). Moreover,there exist several ontology formalisms with a definite advantages and drawbacks, which support particular features. Ontologies should be created for a specific function in the most appropriate formalism satisfying needs of a fine target community. Many tools have been developed for.

In the last few years,several tools for implementing metadata of ontologies (building ontologies) using these languages have been developed. WebOnto[3], OntoEdit[4], Protégé[5], OilED[6],Ontolingua[7], WebODE[8], OntoSaurus[9], TODE[10], Hozo[11], Swoop[12], TopBraid[13], OWLGrEd[14], Graffo[15] etc. A study combing some of them can be found in[16]. Another purposes have been concerned by Ontology Tools and Services Building,for example ontology merging (PROMPT[17],Chimaera[18], Ontomorph[19] etc.), Ontology access (OKBC[20]), etc. Moreover,some applications based-Ontologies have been built, like PlanetOnto[21], Ontobroker[22], MKBEEM[23], (KA)[24], etc. All the developed tools participate as a great contributor to the promotion of the ontology community and the solid foundations of an emergent research, the semantic web.

Ontology development (building) needs to clarify several points related to the Formalisms, Tools and Languages to be used. We suppose that ontology is developed from scratch, it should be required:

- To specify the tool(s) that give/s support to the ontology development process.
- To identify the ontology storage type (in databases or files) and if the tool has an inference engine.
- To define if the used tool have translators for different ontology languages or formats.
- To indicate which language (s) should be used to im-

* *Author for correspondence*

plement this ontology and the expressiveness that characterize an ontology language.
- To analyze if the ontology development tool support the chosen language and if the chosen language is appropriate for information exchange between different applications.
- To verify if the language compatible with other languages can be used for knowledge and information representation on the Web.
- To check if the application requires integrating other ontologies that was already implemented in other languages.

The rest of this paper is organized as follows: In the next section, we describe the concept of Ontology, its Definitions, its Applications and its Modularizations. Then languages are presented with a comparing table as a guideline that helps to understand the characteristic of each language. In the following section, Ontology Formalisms are described. After that, Ontology Tools are explained to help researchers in this field to obtain answers to the previous questions. And finally, this paper is wrapped up with a conclusion.

2. Ontologies

Ontologies, which are one of the basic concepts of the Semantic Web, have appeared in the early 90s in the Knowledge engineering field, as part of the steps of knowledge acquisition for Knowledge-Based Systems(CBS). Following the experts that separated knowledge base into "Declarative" and "Procedural" systems (Based Inference Engine), CBS proposed to specify, in one hand, knowledge of the domain being modeled and on the other hand, the reasoning knowledge describing the heuristics rules enabling the use of these domain knowledge. The idea of this modular separation was to build better and faster CBC by reusing generic components as possible, whether at the level of reasoning or knowledge of the domain.

Knowledge system building requires creating a particular domain. The domain being modeled has an abstraction under format of model, specifically, useful as it abstract from irrelevant details.Therefore, an Abstract Model enables to focus on the aspects of the domain of interest. The model of a domain building involves specifying the entities in the domain in a distinguishable manner and the necessary relations relating these entities. Additionally, it requires specifying the types of entities and the relations types existing between entities.

Each domain requires to be split into concepts which constitute a conceptualization of the domain under consideration. The conceptualization defines the types of entities and the existing relations between them. It is important to specify that the created conceptualization is not, obviously, a defined process.

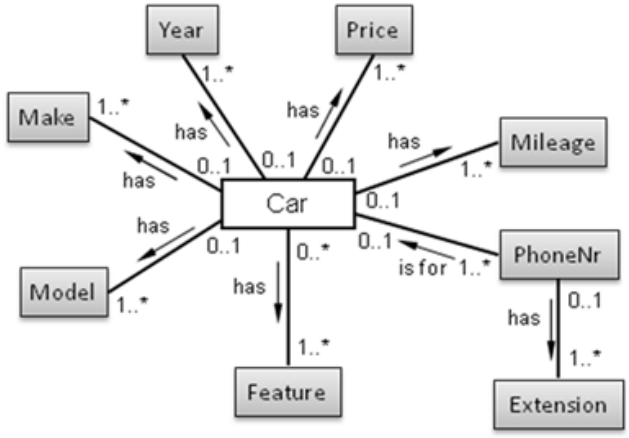

Figure 1. Example of Car Ontology.

It is possible to not conceptualize some entities in the world, but other entities could be specified more or less abstractly. In brief, a conceptualization making is a process that comes with a considerable amount of autonomy. Figure 1 includes an example of Car Ontology. Figure 2 includes another example of Person Ontology.

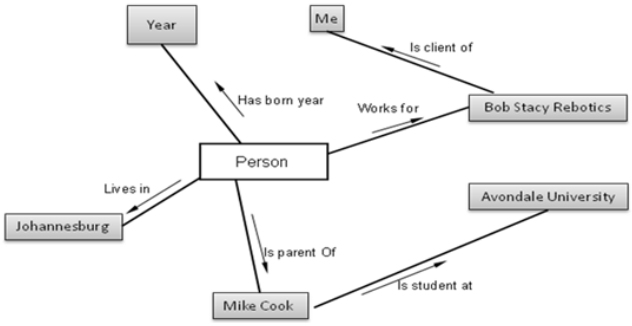

Figure 2. Example of Person Ontology.

2.1. Definitions of Ontology

The word ontology is employed in the field of AI research, as it is useful to make the conceptualizations of a domain explicit which enables their comparison and analyzes. Gruber[25] ,defines an ontology as an explicit specification of a conceptualization. An ontology is a description of knowledge-level[26] ,where the representational formalism is independent[27]. Additionally, ontology is a

representation of the entities type, their relations, and their constraints[27]. It consists of a set of Classes, Relations, Instances, Functions and Axioms ordered hierarchically. Formally, ontology is a description of data that remains constant over various data/knowledge bases in a certain domain[28]. For example,ontology could specify that all knowledge bases of the university domain include shared or understood classes (Department, Staff, Students, or Course) by users which are familiar with universities all over the world. For more interpretation of the word ontology,Guarino and Giaretta[28], have discussed several interpretations of the word 'Ontology'.

2.2. Application of Ontologies

There are immense computer-based applications which take advantage of Ontologies, enumerated in the following section, based either on great performance or simplicity of use:

- **Information Retrieval:** Ontologies are used in information retrieval as intelligent search tool using inference mechanism as an alternative of keyword matching. It facilitates information retrieval without the use of the complicated Boolean logic. Additionally, ontology helps to improve recall based on the synonym relations of query expansion and to improve precision based on Word Sense Disambiguation (identifying the meaning of a word in a given context).
- **Texts:** Ontologies can be exploited in text processing applications and it can be extracted from textual sources (recognized as Text-Based Ontology learning). For example, the technique used to identify useful words (likely to represent concepts or relations) for the learning method is called ontology learning which is a Pattern-Based Technique. For instance,a special syntactic pattern has been developed to identify ontological relations like Hyponymy29 or Mereonymy30.
- **Digital Libraries:** Ontologies can exploit the sense for Automatic Indexing and Annotation of documents. Additionally, it can be used for dynamic catalogues building from machine readable metadata.
- **Knowledge Management:**The use of Ontologies in knowledge management systems supports the existing systems/databases integration and the ability to generate implicit information. Moreover, ontology can be used as a knowledge management tools enabling semantic oriented access and to guide the knowledge discovery.
- **NLP (Natural Language Processing):** The use of ontology enables queries with natural language and allows a better machine translation. The grouping between NLP,IR and Ontologies seem to outline an attractive alternative to use context in texts processing and to create research and development opportunities.As an example MUMIS system 31 relies on NLP techniques.

2.3. Ontology Modularization

The differentiation between Ontologies based on their generality level can be presented in the classification presented by Guarino in[32] and showed in Figure 3. Ontologies can be classified according to the conceptualization subject (content). Very general things such as Time, Space, Insubstantial or Concrete Objects and so on can be covered by the Top-Level Ontologies, independent to the domain usage.The construction of either domain or task Ontologies can be done based on these Top-Level Ontologies. The first category includes Ontologies dedicated to cover a given domain (medical or university, for example) independently to the task that uses the ontology. The second category includes Ontologies specified for a generic mission (content annotation or situation recognition, for example) irrelevant of usage domain. In conclusion, the development of application Ontologies helps particular tasks to be solved within particular domains, and consequently often requires both domain and task Ontologies for reusability.

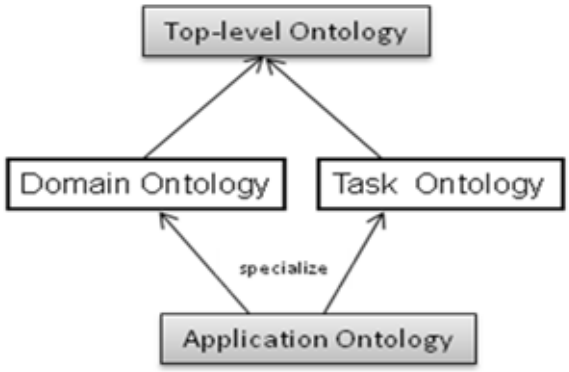

Figure 3. Guarino's Ontology Classification[32].

Ontology may be classified as follows, based on the scope of the ontology (see also figure 3):

- Upper/Top-level Ontology: it describes general knowledge (i.e. what time is and what space is).

- Domain Ontology: it describes the domain (medical domain, personal computer domain or electrical engineering domain).
- Task Ontology: it is suitable for a specific task (assembling parts together).
- Application Ontology: it is developed for a specific application (assembling personal computers).

Modularization can be used at each level. For instance, upper ontology could includes modules for Real Numbers, Time and Space (parts of Upper Ontology, generally are called generic Ontologies).Upper levels Ontologies could be imported by Ontologies at lower levels and adding them specific knowledge. Domain and Task Ontologies may be independent and are combined to be used for application ontology.

3. Ontology Languages

In the last few years, a variety of ontology languages have been developed, and they become ontology languages adopted in the context of the Semantic Web.The following section enumerates the most recognized languages whether it is created especially by Wide Web Consortium (W3C) working groups or not :

- **KIF:** short for Knowledge Interchange Format[33], is a language based on first order logic created in 1992 as an interchange format for diverse knowledge related systems.
- **Loom:** Loom[34] is a knowledge representation language implemented by researchers in the AI research group at the University of Southern California's Information Sciences Institute.Loom is not designed for implementing Ontologies,but for general KBs. It is developed based on DLs and production rules, and offers automatic classifications of concepts.
- **OCML:** short for Options Configuration Modeling Language.The development of OCML was created in 1993[35] at the Open University KMI as a kind of "Operational Ontolingua". Indeed, most of the definitions that can express in OCML are analogous to the corresponding definitions in Ontolingua.But some added components are defined operational definitions for functions and deductive production rules and OCML was constructed for developing executable Ontologies and models in problem solving methods.
- **FLogic:** short for Frame Logic, FLogic[36] merge frames and first order logic, to allow Concepts,Concept Taxonomies,Functions,Binary Relations, Instances,Axioms and Deductive rules representation. Ontobroker[22] can be used underlying FLogic based inference engine to check constraint and deduce new information.
- **SHOE:** SHOE[37] was built in 1996 as a simple html ontology extension allowing web page authors the annotation of their web pages with machine-readable knowledge. SHOE makes the possibility for agents to gather meaningful information about Web pages and Documents, which improves search mechanisms, and knowledge gathering. Moreover, SHOE combines Markup Languages,Knowledge Representation,Datalog and Ontologies features aiming to address the unique problems of semantics on the Web.
- **OML:** Short for Ontology Markup Language,OML[38] was initially developed at the University of Washington, and partially based on SHOE. In reality,it was initially considered an XML serialization of SHOE. Additionally,OML forms a subset of CKML (Conceptual Markup Language) that allows rich knowledge representation capabilities.
- **XML:** it is a W3C recommendation stands for EXtensible Markup Language[39], was built in 1996 much like HTML and designed to describe data and not to display data. As an effect, XML has been used to modify SHOE syntax and subsequently, additional ontology languages were built on the XML syntax.
- **XOL:** short for Ontology Exchange Language[40], was developed by the AI center of SRI international, in 1999. It is designed by the US bioinformatics community and based on XML language. Any tool is allocated for the development of Ontologies using XOL. Although, based on syntax of XML, we can use an XML editor to author XOL files.
- **RDF:** stands for Resource Description Framework[41], was developed by the W3C to describe Web resources and allows the specification of the semantics of data based on XML in a Homogeneous,Interoperable Manner. It also provides mechanisms to clearly represent Services, Processes and Business Models, while allowing recognition of information not clear.
- **RDFS:** stands for RDF Schema[42] and was built by the W3C as an extension to RDF with Frame-Based Primitives. RDF(S) is obtained by the combination of both RDF and RDF Schema. RDF(S) just allows the representation of Concepts,Taxonomies of Concepts and Binary Relations for that reason it is not very expressive. Three additional languages have been developed as extensions to RDF(S) and described in the following section (OIL, DAML OIL and OWL).
- **OIL:** stands for Ontology Interchange Language[43], and developed in the OntoKnowledge project(www.

ontoknowledge.org/OIL),allowing semantic interoperability between Web resources. Its syntax and semantics combines the existing proposals OKBC, XOL, and RDF(S). OIL was built on top of RDF(S) and includes the following four layers: Core OIL grouping the OIL primitives;Standard OIL including the complete OIL model that uses additional primitives than the ones defined in RDF(S); Instance OIL adding roles and instances of concepts to the previous model and Heavy OIL as a layer for future extensions of OIL. OILEd, Protégé2000, and WebODE tools can be adapted to author OIL Ontologies.

- **DAML+OIL:** Stands for DARPA Agent Markup Language+OIL. DAML+OIL[44], has been developed by a joint committee from the US and the European Union (IST) in the context of DAML, a DARPA project for allowing semantic interoperability in XML. Therefore, the same objectives as OIL are shared by DAML+OIL, it is built on RDF(S). DAML+OIL language was based on OIL as indicated by its name. The OIL and DAML+OIL languages allow Concepts, Taxonomies, Functions, Binary Relations and Instances representation. The tools that can author DAML+OIL Ontologies are OILEd, OntoEdit, Protégé2000 and WebODE.
- **OWL:** stands for Web Ontology Language, created in 2001 by a working group formed by W3C. The formed group has defined a list of main use cases for the Semantic Web, taking into account the DAML+OIL features as the main input for developing OWL and proposing the first specification of this language45. Currently OWL may be distinguished between OWL1 and OWL2, OWL1 includes three classes:OWL Full, OWL DL and OWL Lite, detailed later in this paper.
- **OWL2:** OWL2 adds new functionalities with respect to OWL1. Specifically, OWL2 maintains OWL Full and OWL DL by adding Richer Datatypes, Qualified Cardinality Restrictions, Reflexive, Asymmetric and Disjoint Properties, in addition to the definition of different profiles: OWL 2DL, OWL 2EL, OWL2 QL and OWL 2RL.
- **CycL:** CycL was developed by Cycorp and it is a formal language whose syntax is a derivative from first-order predicate calculus and that extends first-order logic based on the second order concepts. CycL is adopted to express common sense knowledge and to represent the knowledge stored in the Knowledge Base Cyc. The CycL vocabulary includes terms: Semantic Constants, Integer, Strings, Non-Atomic Terms, Variables, etc. A knowledge base can be formed by a set of sentences[46]. In brief, CycL uses predicate logic extended by typing, reification and microtheories that define a context for the truth of formulas.

We have built some conclusions of the languages presented above by comparing them with respect to the similar evaluation framework[47]. Table 1 presents a comprehensive comparison that details how each language manages to meet attributes Criteria, Facets Criteria, Taxonomies Criteria, Functions Criteria or general issues criteria. A √ symbol denotes that the corresponding language satisfies that particular requirement and an X symbol denotes that the corresponding language not satisfies or satisfies partially that particular requirement.

4. Ontology Formalisms

It is recommended to choose the formalism of ontology modeling language that will allow the expression of all the needed distinctions. To be used for automatic processing in computers, Ontologies need to be specified formally. There are several formalisms that are used for expressing Ontologies. In this section we summarize some of them. These formalisms provide means for representing particular ontology:

- **Frame-Based Formalism:** similar to OOA, frame is a data structure used to represent a concept in a domain. Schaerf [48] define a frame as a concept (or a class) representation formed by an identifier and a set of data elements named slots,where each slot corresponds to an attribute contained in the members of the class. The attributes values are either concrete domain elements (e.g., Integers, Float, Strings) or other frames identifiers. Frame based Ontologies contain classes and the information concerning the property specified at a generic class level inherited by subclasses and instances. For instance, if a class course has as properties degree and exam date,a specific course instance could have degree:75 and exam date: 10/11/2014. Any subclass in the is-a hierarchy of course, for example class course (or home course), has all course properties and relations and it can include additional properties or relations if necessary. This organizing knowledge structure is analogous to the recognized object oriented modeling techniques. Chaudhri et al[49],proposes an example of frame based Ontologies providing a uniform model, where the knowledge could be shared based on the ordinary classes conceptualization,slots,individuals, facets and inheritance. As examples of ontology construction tools

Table 1. A Comparison between Ontology Languages (updated from [47])

	LOOM	OCML	FLogic	SHOE	OML	XOL	RDF(s)	OIL	DAML + OIL	OWL	OWL2
General issues											
Concept partitions	√	X	X	X	√	X	X	√	√	X	√
documentation	√	√	X	√	√	√	√	√	√	√	√
Attributes											
Instance attributes	√	√	√	√	√	√	√	√	√	√	√
Class attributes	√	√	√	X	√	√	X	√	√	X	√
Local scope	√	√	√	√	√	√	√	√	√	√	√
Global scope	√	X	X	X	√	√	√	√	√	√	√
Facets											
Default slot value	√	√	√	X	X	√	X	X	X		
Type constraints	√	√	√	√	√	√	√	√	√	√	√
Cardinality constraints	√	√	X	X	X	X	X	√	√	√	√
Slot documentation	√	√	X	√	√	√	√	√	√	√	√
Taxonomies											
Subclass of	√	√	√	√	√	√	√	√	√	√	√
Disjoint Decomposition	√	√	X	X	X	X	X	√	√	X	√
Relations											
n-ary relations	√	√	X	√	√	X	X	X	X	X	√
Operational definition	√	√	√	X	X	X	X	X	X	√	√
Axioms											
1st order logic	√	√	√	X	√	X	X	X	X	√	√
2nd order logic	X	X	X	X	X	X	X	X	X	√	√
Named axioms	X	√	X	X	X	X	X	X	X	X	X
Embedded axioms	√	√	X	X	√	X	X	X	X	√	√
Instances											
Concept instances	√	√	√	√	√	√	√	√	√	√	√
Facts	X	X	X	X	√	X	X	X	X	√	√
Other											
Rules	√	√	X	√	√	X	X	X	X	√	√

supporting the frame based Ontologies construction (between other models), Protégé is very recognized. Additionally, the most known UML language offers all the needed elements (classes, 'is-a" Hierarchy, Cardinality, Property Definition,Specification and Value Restrictions) to construct frame based models having relation with software engineering. However, with representation capabilities, UML is more reduced for frame construction than the presented with complex frame based models.

- **Logic-Based Formalism:** Description Logics (DL) has replaced Frame-based languages. It is very expressive knowledge representation formalism. FOL formalism (first-order logic based) is ideal for Ontologies representations that are difficult from the computational point of view. DL is more expressive for the representation of the popular bio-medical Ontologies. SNOMED and GALEN are two import examples of bio-medical ontology represent using DL formalism.
- **OBO Formalism:** short for Open Biomedical Ontology. OBO formalisms require the ontology to be a graph with labeled nodes. OBO formalism is a collaborative experiment including biological researchers and ontology developers proposing a new paradigm for the development of the biomedical Ontologies. The researcher adopted a set of best practice principles to ensure a high quality and formal rigor in Ontologies and assuring their interoperability from the start. In the bio-ontology field, in-house tools have been developed, such as DAG-Edit and OBO-Edit.

- **Semantic Networks:** Semantic network is a graph,where vertices symbolize concepts and where edges symbolize relations between concepts.At the ontology level, semantic network expresses vocabulary that is useful especially for human, but that can be functional for machine processing.The concepts relations adopted in semantic networks are as follows:
- Synonym: a concept A expresses the same thing as another concept B.
- Antonym: concept A expresses the opposite of concept B
- Meronym, holonym: includes relations like part-of and has-part relation between concepts.
- Hyponym, hypernym: is the inclusion of semantic range between concepts in both directions.
- **Conceptual Graph Formalism:** Conceptual graph (CG) is a logical formalism including Classes, Relation, Quantifier and Individuals. CG formalism is based on semantic networks, but it takes its semantics from first order predicate logic. A conceptual graph expresses meaning with logically precise, readable by humans, and computationally tractable form. A conceptual graph is useful as an intermediate language for translating computer-oriented formalisms to and from natural languages using a direct mapping to language. It offers a graphic representation, which is useful as a readable and formal design and specification language. CG has been implemented in several projects for information retrieval, expert systems, database design and natural language processing.

Table **2.** gives a classification of a number of ontology languages based on structure and syntax

5. Ontology Tools

Several software tools related to Ontologies have been proposed by researchers in Semantic web.Especially, there exist significant attention accorded to Semantic web editors (responsible to the creation and manipulation of Ontologies). The following paragraphs contain some of these tools:

- **DUET[1]:** The DAML UML Enhanced Tool[50], DUET is a software tool that enables DAML Ontologies importing into IBM Rational Rose and AgroUML and the exportation of UML models into DAML. This tool is actually a plug-in for AgroUML. DUET uses a simple UML profile containing stereotypes for Ontologies modeling (based on UML package) and properties (based on a UML class).
- **UBOT[2]:** short for UML Based Ontology Tool-set[51]. UBOT is a project intended to build ontology engineering and natural language processing-based text annotation tools for DAML. UBOT use UML as a front-end for DAML Ontologies visualizing and editing. The approach is used to extend UML by defining a UML profile for DAML that maps UML stereotypes to DAML-specific elements[50].
- **OntoEdit[3]:** OntoEdit[4] is an ontology editor integrating various aspects of ontology engineering. OntoEdit is quite exceptional in its category since it is based on a modern method for ontology development and

Table 2. Ontology Languages Classification Based on Structure or Syntax

	Classification by structure			*Classification by syntaxes*	
	Frame based	*Description logic-based*	*1st order logic-based*	*Traditional syntax ontology languages*	*Markup ontology languages*
KIF			√	√	
LOOM				√	
OCML				√	
F-Logic	√			√	
SHOE					√
OML				√	
RDF(s)					√
OIL					√
DAML+OIL					√
OWL		√			√
CycL			√	√	

because it makes comprehensive use of inferencing.

- **Protégé[4]:** Protégé[5] is an ontology editor created at Stanford University and is very popular in the field of Semantic Web and the level of computer science research. Protégé is free, developed in Java and its source code is released under a free license (the Mozilla Public License). Protégé can read and save ontologies in most ontologies formats: RDF, RDFS, OWL, etc. It has several competitors such as Hozo[11], OntoEdit and Swoop. It is recognized for its ability to work on large Ontologies.

[1] http://codip.grci.com/Tools/Tools.html

[2] http://ubot.lockheedmartin.com/

[3] http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/

[4] http://protege.stanford.edu/

- **OILEd[5]:** OIL Editor (OilEd)[6] is a simple ontology editor that supports OIL-based Ontologies construction. The basic design has been deeply influenced by similar tools such as Protégé[5] and OntoEdit, but OilEd extended these approaches in several manners, especially using an extension of expressive power and a reasoner. OilEd supports the construction of OIL-based Ontologies as an ontology editor.
- **Ontolingua[6]:** The Ontolingua[7] is an ontology tool created the Knowledge System Laboratory at Stanford University. Ontolingua is devoted for Ontologies development using a form-based Web interface. The ontology editor of Ontolingua is a tool supporting distributed, browsing,collaborative editing and Ontologies creation.Using Ontolingua, it is possible to export or import the following formats: KIF, DAML+OIL, OKBC, Prolog, LOOM, Ontolingua and CLIPS (C Language Integrated Production System). Additionally, it is also possible to only import Classic Ocelot and Protégé format, but not their export.
- **OntoSaurus[7]:** OntoSaurus[9] , is a web browser for LOOM providing a graphical hyperlinked interface to Loom Knowledge bases. It provides automatic consistency checking, expressive knowledge representation and deductive support via its deductive engine.
- **WebODE[8]:** WebODE[8] ,can be defined as described in the Ontological Engineering Group webpage, "WebODE was built as a Scalable, Extensible, Integrated workbench that covers and gave support to most of the activities involved in the ontology development process (conceptualization, reasoning, exchange, etc.) and supplied a comprehensive set of ontology related services that permit interoperation with other information systems". WebODE exports to WebODE's XML, RDF(S), Prolog, OIL, Java/Jess, DAML+OIL, X-CARIN, UML and OWL, and imports from WebODE's XML, RDF(S), UML, X-CARIN and OWL.
- **WebOnto[9]:** WebOnto[3] is a tool which provides a web-based visualization, browsing and editing support to develop and maintain Ontologies and knowledge models specified in OCML. An ontology can be viewed as a model of the conceptual structure of some domain and WebOnto provides the capability to represent this graphically.
- **Hozo[10]:** Hozo is an environment for ontology development created at Osaka University. Hozo is a free tool representing a role model based on frames, where role denotes a generic role concept class. The main idea of class definition in Hozo indicates that all concepts are defined independently of the possible wholes they belong to, and each class as a whole is defined by specifying the roles whose parts play[11]. Hozo features includes: 1) role concept supporting, 2) ontology visualization in well considered format, and 3) management of dependencies between Ontologies for distributed development.
- **Swoop[11]:** short for Semantic Web Ontology Editor, SWOOP[12] (swoop, 2004) is a tool for creating, editing, and debugging OWL Ontologies. It was produced by the MIND lab at University of Maryland, College Park, but is now an open source project with contributors from all over the world.
- **Dogma Studio Workbench:** Dogma Studio Workbench is the not free recent toolsuite supporting DOGMA ontology engineering approach. It contains a wokbench (server and Eclipse, in addition to several plugins). This tool support ORM format and the mapping of ORM into OWL-DL.
- **TopBraid Composer[12]:** The Free Edition (FE) of TopBraid Composer [13] is a professional tool for ontologies development. It uses the Eclipse platform and the Jena API. TopBraid Composer is a complete editor for RDF(S) and OWL models, additionally it is a platform for other RDF-based components and services. Any OWL2 file in formats such as RDF/XML or Turtle can be loaded and saved by TopBraid Composer (FE).
- **OWLGrEd:** short for OWL Graphical Editor[14] is a

[5]http://img.cs.man.ac.uk/oil.

[6]http://ontolingua.stanford.edu/

[7]http://www.isi.edu/isd/ontosaurus.html

[8]http://delicias.dia.fi.upm.es/webODE/

[9]http://webonto.open.ac.uk/

[10]www.hozo.jp

[11]http://code.google.com/p/swoop/

[12]TopBraid Composer, http://www.topquadrant.com/products/TB_Composer.html.

Table 3. Ontology Development Tools Comparison

	Release Date	Base Language	Imort/Exports from/to Languages	Exception Handling	Ontology storage	Availability	Ontology library
DUET	17/03/2002	UML	DAML+OIL	Yes	No	Free	No
UBOT	09/2002	UML	**Inp:**UML , XML, **Exp:**DAML, Slang	Yes	No	Free	Yes
OntoEdit	04/03/2004	F-Logic	RDFS, F-Logic, DAML+OIL; RDB, schemas	No	Files	Free	No
Protégé	22/06/2004	OKBC+ CLOS based meta-model	RDF, RDFS, DAML+OIL; XML, OWL, Clips; UML	No	Files &DBMs (JDBC)	Free	Yes
OiLED	31/10/2003	DAML+OIL	RDF URI's; limited XML Schema, export : HTML.	No	Files	Free	Yes
Ontolingua	11/2001	Ontolingua	**Imp/Exp:** KIF, OKBC,Loom,Prolog, Ontolingua, CLIPS **import only:** Ocelot, classic, Protégé	No	Files	Free	Yes
OntoSaurus	03/2002	Files	LOOM, IDL, ONTO, KIF C++	No	Files	Open source	No
WebODE	03/2002	HTML forms and Java applets	**Imp/exp:** XML, RDF(S), XCARIN, OWL **Exp:** OIL DAML + OIL FLogic, Prolog Jess, Java	No	DBMS (JDBC)	Free	No
WebOnto	05/2001	OCML	**Imp/exp:** OCML **Exp:** Ontolingua GXL, RDF(S), OIL	No	Files	Free	Yes
Swoop	08/2007	OWL	**Imp:** OWL, XML, RDF and text **exp:** RDF(S), OIL and DAML	No	HTML models	Free	No
Dogma Studio Workbench	Not reported		OWL, RDF, DAML+OIL			Not free: OnDemand	No
TopBraid Composer	2011/06/04	RDFS/OWL	**Imp:** RDFa, OWL, RDF(s) ,XHTML, Microdata, and RDFa, Data sources, SPIN, **Exp:** Merge /Convert RDF Graphs, RDF(S), OWL	Yes	Files	License	Yes
OWLGrEd	10/ 2011	OWL	OWL, OWL2, UML, RDF/XML	No	Files	Free	Yes
Anzo	2011/09/01	XLS file	XML, RDF, XLS file	No	Files	Pay Licensed Closed	Yes
Graffoo	28/10/2013	OWL	OWL2, Turtle, RDF/XML, Manchester Syntax, or OWL/XML	No	Files	Open source	No

free UML style graphical editor for OWL Ontologies. It has further features for the exploration and development of graphical ontology. OWLGrEd provides a complete graphical notation for OWL2, based on UML class diagrams and take into account the interoperability with Protégé.

- **Anzo[13]:** Anzo is a tool released in 2011 and used with Excel to generate an initial Ontology. Anzo includes a Straightforward Ontology Editor, two rules Engines, Work Flow Capabilities, Pubsub, and Integration with XML, DRFS, OWL, Relational Databases, and Web Services.
- **Graffoo[14]:** Graffoo stands for Graphical Framework for OWL Ontologies[15], is a superb new open source tool developed by Silvio Peroni that can be used to present the classes, properties and restrictions within OWL ontologies, or sub-sections of them, as clear and easy-to-understand diagrams. Several Graffoo diagrams have been developed to explain SPAR ontologies, or portions of them, and are to be found in the appropriate ontology directories.

Below we present some conclusions comparing the tools presented above,whose results are also summarized in Table 3. The comparison of these tools could be completed in different manners. In this paper, the comparison is done based on some properties selected from the work in[47]. Comparison could be done by specifying different properties of editors. We restrict the comparison only to the following properties:

- **General description of the tools:** which includes information about tools release date (used to specify the first date when a tool was released) and availability (free, open source, pay license, on demand, etc.) .
- **Software architecture and tool evolution:** The selected property is limited to the ontology storage property (databases, text files, etc.).
- **Interaction with other ontology development tools and languages:** includes information about the interoperability of the tool (import to languages and export from languages and basic language used).
- **Inference services attached to the tool:** Only the property Exception Handling is selected which tells if the tool is able to manage the exceptions in taxonomies.
- **Usability:** Only the property Ontology library is selected to show whether it provides libraries of Ontologies.
- **Fluent Editor:** is an ontology tool useful for editing, manipulating and querying complex ontologies that supports OWL, RDF or SWRL. Additionally, Fluent Editor[52] is fully compatible with most of the Semantic Web W3C standard (OWL, RDF, SPARQL, SKOS,...).
- **CSNePS:** is an ontology visualization tools used to assist in creating and maintaining textual term definitions[53.] The authors in[53] discusses two features of the CSNePS GUI designed to make term definition easier: the ability to easily explore ontological terms and their properties within a graph visualization, and to easily pick out undefined terms in an ontology in an optimal order for writing definitions.

5. Conclusion

Nowadays, the maturity of ontology technology is not completely sufficient in spite of the existence of an amalgam of tools and languages. For example, Ontology markup languages are continuously evolving, but not really mature making it difficult to have advanced technology for managing them. However, the main problem addressed in this area is the need of an integrated environment for (excepting some environments, like protégé, OntoEdit, etc.)ontology development. However, the work required in this field should give an attention to a common workbench for ontology development to make easy ontology development, evaluation and management. This workbench should integrate the technology components that are currently available, instead of its creation from scratch.

6. References

1. M Obitko. Ontologies-Description and Applications. Gerstner Laboratory for Intelligent Decision Making and Control Series of Research Reports. 2001. Report No. GL 126/01, 35 p. http://cyber.felk.cvut.cz/gerstner/reports/GL126.pdf.
2. A Waterson. A Preece. (1999)-Verifying Ontological Commitment in Knowledge-Based Systems. Knowledge Based System. 1999; 12(1-2) 45-54.
3. J Domingue, TADZEBAO, WEBONTO. Discussing, Browsing, and Editing Ontologies on the Web. KAW'98. Proceedings of the Eleventh International Workshop on Knowledge Acquisition, Modeling, and Management; Banff: Canada. 1998 Apr. p. 18–23.
4. Y Sure, J Angele, and S Staab. OntoEdit: Multifaceted inferencing for ontology engineering. Journal on Data Semantics. 2003 Nov; 1(1):128–52.

[14]http://www.cambridgesemantics.com/products/anzo-express

[14]http://www.essepuntato.it/graffoo/

5. EG William, E Henrik, RW Fergerson, HG John, WT Samson and AM Mark. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). KAW'99. Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management. 1999. p. 16-21.
6. Bechhofer S, Horrocks I, Goble CA, Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web. Advances in Artificial Intelligence. 2001. p.396–408.
7. Farquhar A, Fikes R, Rice J. The Ontolingua server. A Tool for Collaborative Ontology Construction. International Journal of Human-Computer Studies. 1997 Jun; 46(6):707–27.
8. Corcho O, F-L M, G-PA, Vicente O. WEBODE: An Integrated Work Bench for Ontology Representation, Reasoning, and Exchange. EKAW2002. Proceedings of the Thirteenth International Conference on Knowledge Engineering and Knowledge Management; Sigüenza: Spain. 2002 Octo. p. 1–4.
9. Swartout B, Ramesh P, Knight K, Russ T. Toward Distributed Use of Large-Scale Ontologies. Proceedings of the Spring Symposium on Ontological Engineering of AAAI; Stanford University; California. 1997 Mar. p. 24–6.
10. Noman I, Mohammed SS, Zubair AS. TODE. A Dot Net Based Tool for Ontology Development and Editing. ICCET 2010. Proceedings of the 2nd International Conference on Computer Engineering and Technology; Chengdu: China. 2010 Apr 16–18. p. 229-3.
11. Mizoguchi R, Sunagawa E, Kozaki K, Kitamura Y. A Model of Roles within an Ontology Development Tool. Hozo. Journal of Applied Ontology. 1997; 2(2):159-79.
12. Swoop. Semantic Web Ontology Overview and Perusal. 2004. Available from: http://www.mindswap.org/2004/SWOOP/
13. W3C. TopBraid. 2001. Retrieved from W3C: http://www.w3.org/2001/sw/wiki/TopBraid
14. Barzdins J, Barzdins G, Cerans K, Liepins R, Sprogis A. OWLGrEd: a UML Style Graphical Notation and Editor for OWL 2. OWLED 2010 proceedings of 7th International Workshop OWL; Experience and Directions. 2010. p. 614.
15. Falco R, Gangemi A, Peroni S, Shotton D, Vitali F. Modelling OWL ontologies with graffoo. Proceedings of the 11th European Semantic Web Conference Satellite Events; Crete: Greece. 320–25.
16. Duineveld AJ, Studer R, Weiden MR, Kenepa B, Benjamins VR. A Comparative Study of Ontological Engineering Tools. International Journal of Human Computer Studies. 2000; 52 (6):1111–33.
17. Fridman N, Musen M. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignement. AAAI-2000. In The proceedings of the Seventeenth National Conference on Artificial Intelligence; Austin: Texas. 2000 July 30-August 1.
18. McGuinness D, Fikes R, Rice J, Wilder S. The Chimaera Ontology Environment. AAAI-2000. Proceedings of the Seventeenth National Conference on Artificial Intelligence; Austin: Texas. 2000 July 30-August 1.
19. Chalupsky H. OntoMorph: A Translation System for Symboloic Knowledge.KR-2000. Proceedings of the Seventh International Conference: Principles of Knowledge Representation and Reasoning; San Francisco: CA. 2000. p. 471-482.
20. Chaudhri VK, Farquhar A , Fikes R , Karp PD, Rice JP. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. AAAI-98. Proceedings of the Fifteenth National Conference on Artificial Intelligence; Madison: Wisconsin. 1998July. p. 26-30.
21. Domingue J. Tadzebao and Webonto: Discussing, Browsing and Editing Ontologies on the Web. KWW98. Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management; Alberta: Canada. 1998Ap. p.l 18-23.
22. Fensel D, Angele J, Decker S, Erdmann M, Schnurr H, Staab S, Studer R, Witt A. Ont2broker. Semantic-Based Access to Information Sources at the WWW. WebNet99. Proceedings of WebNet World Conference on the WWW and Internet., Honolulu: USA. 1999 Oct. p. 366-71.
23. MKBEEM. Avaialble from: http://www.lt-world.org/kb/players-and-teams/projects/obj_61579.
24. Benjamins VR, Fensel D, Decker S, Perez AG. $(KA)^2$: Building Ontologies for the Internet : A Mid-Term Report. Int. J. Human-Computer Studies. 1999; 51:687-712.
25. Gruber T. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies. 1995;43:907–928.
26. Newell A. The Knowledge Level. Artificial Intelligence. 1982; 18 (1).
27. Van Heijst G, Schreiber AT, Wielinga BJ. Using Explicit Ontologies in KBS Development. International Journal of Human-Computer Studies. 1997; 46(2-3):183 – 292.
28. Guarino N, Giaretta P. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. J. I. Mars (ed.), Towards Very Large Knowledge Bases; Amsterdam: Netherlands. IOS Press; 1995. p. 25-32.
29. Guarino N, Welty C . Evaluating Ontological Decisions with OntoClean. Communication of the ACM . 2002;(2):61-5.
30. Berland M , Charniak E. 37th Annual Meeting of the ACL. 1999. p. 57-64.
31. Kuper J, Saggion J, Cunningham H, Declerck T, de Jong T, Reidsma D, Wilks Y. Wittenburg P. Intelligent Multimedia Indexing and Retrieval through Multi-source Information Extraction and Merging. IJCAI '2003. Proceedings of the International Joint Conference of Artificial Intelligence; Acapulco: Mexico. 2003.
32. Guarino N. Semantic Matching: Formal Ontological Distinctions for Information Organization,Extraction, and Integration. In M. T. Pazienza (ed.) Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. Springer; Verlag.1997. p. 139-170.
33. Genesereth M, Fikes R. Knowledge Interchange Format, Technical Report Logic-92-1. Computer Science Department, Stanford University.1992.
34. MacGregor R. Inside the LOOM classifier, SIGART bulletin. 1991;2(3):70–6.
35. Shadbolt N, Motta E, Rouge A. Constructing Knowl-

edge-Based Systems. IEEE Software; 1993;10(6):34-39.
36. Kifer M, Lausen G, Wu J. Logical Foundations of Ob¬-ject-Oriented and Frame-Based Languages. Journal of the ACM. 1995; 42(4):741–843.
37. Luke S, Heflin J. SHOE 1.01 Proposed Specification. Parallel Understanding Systems Group. 2000 Feb. Retrieved from: ww.cs.umd.edu/projects/plus/SHOE/ spec1.01.htm.
38. Kent RE. Conceptual Knowledge Markup Language. The Central Core. KAW'99. Proceeding of the Twelfth Work¬shop on Knowledge Acquisition, Modeling and Manage-ment; Banff, Alberta: Canada.1999 Oct.
39. Bray T, Paoli J, S-M CM, Maler E. Extensi¬ble Markup Lan-guage (XML) 1.0, second ed., W3C Recom¬mendation. 2000. Available from: <http://www.w3.org/TR/REC-xml>
40. Karp R, Chaudhri V, Thomere J. XOL: An XML-Based On¬tology Exchange Language (version 0.4). Techni¬cal Report Artificial Intelligence Center. SRI International; 1999 Aug.
41. Lassila O, Swick R. Resource Description Framework Mod-el and Syntax Specification. W3C Recommen¬dation. 1999. Available from: http://www.w3.org/TR/REC-rdf-syntax/
42. Brickley D, Guha RV. RDF Vocabulary Description Lan¬guage 1.0: RDF Schema. W3C Working Draft. 2002. Avail¬able from: <http://www.w3.org/TR/PR-rdf-schema>
43. Horrocks I, Fensel D, Harmelen F, Decker S, Erd¬mann M, Klein M. OIL in a Nutshell. ECAI'00 Proceed¬ing of Workshop on Application of Ontologies and PSMs; Berlin: German. 2000.
44. Horrocks F, Van H. Reference Description of the DAMLOIL (March 2001) Ontology Markup Language. 2001. Available from: <http://www.daml.org/2001/03/ reference.html>.
45. Dean M, Connolly D, Van Harmelen F, Hendler J, Hor¬rocks I, McGuinness DL, Patel-Schneider PF, Stein LA. OWL Web Ontolog Ontology Language 1.0 Reference, W3C Working Draft. 2002. Available from: <http://www.w3.org/TR/owl-ref/>.
46. Cycorp. Inc. 2014 .Available from: http://www.cyc.com/ [Accessed 11.08.2014].
47. Corcho O, F-L M, G-P A. Methodo¬logies, Tools and Lan-guages for Building Ontologies .Where Is Their Meeting Point?. Data and Knowledge Engineering. 2003 46(1):41-64.
48. Schaerf A. Query answering in Concept-Based Knowl-edge Representation Systems: Algorithms, Complexity and Se¬mantic Issues. [PhD thesis]. Dipartimento di Informat-ica e Sistemistica. Universita di Roma "La Sapienza". 1994.
49. Chaudhri VK , Farquhar A, Fikes R, Karp PD, Rice JP. Open Knowledge Base Connectivity 2.0 Knowledge Systems Lab¬oratory. Stanford: CA. 1998.
50. Baclawski K, Kokar M, Kogut P, Hart L, Smith J, Holmes W, Letkowski J, Aronson M. Extending UML to Support On¬tology Engineering for the Semantic Web. UML2001. Pro¬ceeding of the 4th International Conference on UML; Toronto: Canada. 2001Oct.
51. Paul Kogu. DAML-UML Based Ontology Toolset (UBOT). Lockheed Martin Integrated System and Solutions. 2005.
52. 2014.Available from: https://www.w3.org/2001/sw/wiki/ Fluent_Editor.
53. Daniel R. Schlegel, Peter L, Elkin. Ontology Visualization Tools to Assist in Creating and Maintaining Textual Term Definitions. Third International Workshop on Definitions in Ontologies (IWOOD 2015); Lisbon: Portugal. 2015.