Framework for Analyzing Ontology Development Tools

Violeta Damjanović, Vladan Devedžić, Dragan Djurić, Dragan Gašević
FON – Faculty of Organizational Sciences, University of Belgrade, Serbia and Montenegro
vdamjanovic@gmail.com, devedzic@fon.fon.bg.ac.yu, dragandj@gmail.com, gasevic@yahoo.com

INTRODUCTION

Nowadays, ontologies and ontological engineering span such diverse fields as qualitative modeling, language engineering, database design, information retrieval and extraction, knowledge management and organization, ontology-enhanced search, possibly the largest one, e-commerce (e.g., Amazon.com, Yahoo Shopping, etc.), and configuration [1]. The engineering part of developing ontologies comprises a complex set of activities that are conducted during conceptualization, design, implementation and deployment of ontologies. Ontological engineering covers a whole range of topics and issues, such as the basics (philosophical and metaphysical issues and knowledge representation formalisms), methodology of ontology development, recent Web technologies such as eXtensible Markup Language (XML) and its relatives [2], business process modeling, common sense knowledge, systematization of domain knowledge, Internet information retrieval, standardization, evaluation, ontology integration with agents and applications, and many more [3]. It also gives us design rationale of a knowledge base, helps us define the essential concepts of the world of interest, allows a more disciplined design of a knowledge base, and enables us to accumulate the knowledge about it. As a consequence, the use of specific software tools that enable ontology conceptualization, representation, construction, and use becomes an important aspect of building ontologies.

CLASSIFICATION OF ONTOLOGY DEVELOPMENT TOOL

There are numerous ontology development tools today. The most of them have resulted from efforts of research groups and university labs, and are currently free. However, these tools can differ a lot in terms of:

- support they provide to the ontology development process,
- the level of sophistication,
- the format(s) used for storing ontologies,
- many format converters that support translating ontologies to/from other formats,
- the way(s) other applications can interoperate with different ontology development tools,
- the tool stability and maturity,
- support for querying information about an ontology, and so on [4].

So far, there were several efforts to develop a comprehensive classification of ontology development tools, as well as to compare and evaluate many different tools. All such efforts start from a well-developed methodology and a great number of evaluation criteria, thus developing a complex evaluation framework first. Then many tools are compared and evaluated against the framework. Till now, the most comprehensive among such approaches has been the one proposed by OntoWeb Consortium [4]. The approach starts from grouping all ontology-based software tools into the following large categories:

- ontology development tools – the tools, environments and suites that can be used for building a new ontology from scratch or reusing existing ontologies;
- ontology merge and integration tools – the tools helping to solve the problem of merging or integrating different ontologies on the same domain;
- ontology evaluation tools – support tools that enable getting insight to the level of quality of ontologies and their related technologies;
- ontology-based annotation tools – the tools enabling the users to insert ontology-based markups in Web pages;
- ontology storage and querying tools - the tools that allow using and querying ontologies easily;
- ontology learning tools - the tools used to (semi) automatically derive ontologies from natural language texts.

For each of the categories above, the authors from the OntoWeb Consortium have developed a specific evaluation framework, based on numerous criteria, and have evaluated selected tools according to that

framework. The evaluation results are presented in the form of several tables, showing the characteristics of each selected tool. For example, the tools from the first of the categories above are evaluated according to the following groups of criteria:

- general description of the tools (such as information about developers, releases and availability);
- software architecture and tool evolution;
- interoperability with other ontology development tools and languages;
- knowledge representation paradigm (knowledge model used);
- inference services attached to the tool;
- tools usability.

A similar study by M. Denny covered ontology editors only [5]. Ontology browsers without an editing focus and other types of ontology building tools were not included. The study was still very useful, because it helped identify a cross-section of ontology editing tools.

Another group of comparative studies is focused on ontology development languages only. A good example coming from an academic environment is the study of languages for the Semantic Web [6]. The languages are compared according to the criteria, like how suitable they are for representing concepts, taxonomies, instances, axioms, relations, and functions.

FRAMEWORK/HIERARCHY OF ONTOLOGY DEVELOPMENT TOOLS

This section suggests a simple framework for analyzing ontology development tools and surveys many current tools in order to evaluate and compare them according to the proposed framework. The ultimate goal is to indicate an existing hierarchy of current ontology development tools and illuminate current trends in tools-related support for ontology development. The framework is graphically shown in Figure 1 and is characterized by:

- a wider focus than that of ontology editors alone, used in [5];
- yet, a more narrow focus than that of covering all ontology-related tools as in [4];
- ontology development languages themselves are included, although much less formally than in [6];
- ontology learning tools are included, since ontology learning is also a way of building ontologies.

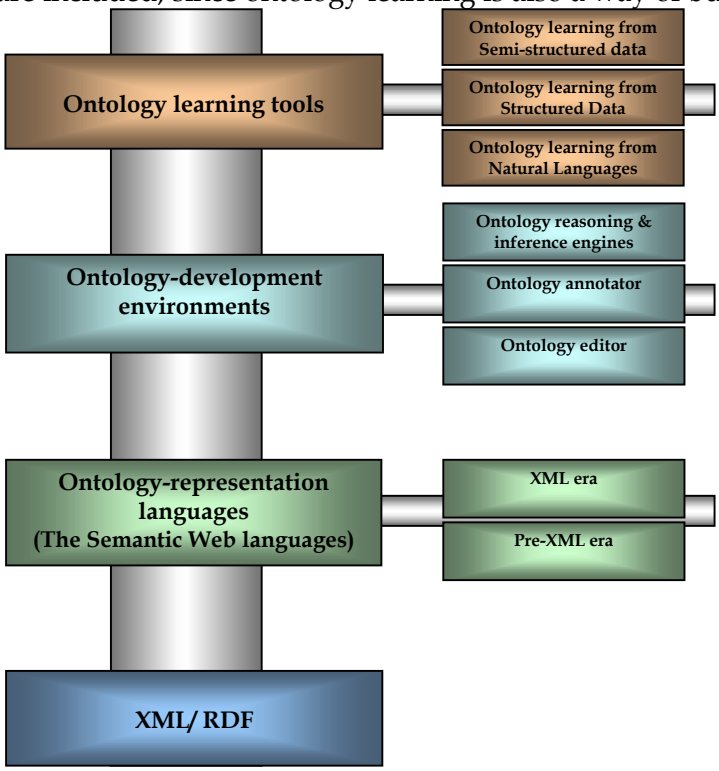

Figure 1. Framework/hierarchy of ontology development tools

First part of this framework is XML/RDF, which consists of XML/XMLS (XML Schema), RDF/RDFS (Resource Definition Framework Schema) and the corresponding development tools. Second, ontology-representation languages are languages of different expressive power, based on different representation

paradigms (regardless of the underlying technology). With the use of the Web standards and also the industry standards, it is possible to separate the two groups of ontology representation languages: some 'classical' representation languages, which coming from pre-XML era and the other one that coming from XML era. Third, ontology development environment presents integrated graphical tools, while ontology learning tools employing machine learning

In this paper, we summarize certain criteria as the basis for evaluations and comparisons of ontology development languages, tools and environments, as well as ontology learning tools, since ontology learning is also a way of building new ontologies, and a way of reaching and extending existing ontologies. We can emphasize the following criteria:

- *Ontology development tool maturity* – measures how ontology development tools may handle development problems, and even reduce the number and intensity of the future problematic situations. This criterion comprises the ability to deal constructively with real environments, the capacity to adapt to change, the capacity to relate and combine with other development tools and environments, and so on.
- *Ontology development tool extensibility* –measures how adaptable ontology development tools may be to the future technological advances. It is crucial for preserving a full development evolution of ontology development tools.
- *Interoperability* – the ability of systems to operate in conjunction with each other, encompassing communication protocols, hardware, software applications, and data compatibility layers (definition of interoperability from www.ichnet.org/glossary.htm).
- *Ontology development tool portability* – the ability to adopt any ontology development tool, technique or method within a new environment without redeveloping it.
- *Ease-of-use* – covers ease-of-learning, intuitiveness, efficiency and functionality. Simultaneously measures how long it takes for one to learn to use a certain product, how intuitive the product is, and how logical it is to use, create or modify a program.
- *Overall capabilities* – this is important when there is more than one criterion that an ontology development tool or environment should satisfy. In this case, performance has to be measured against a composite of individual criteria (e.g., discovery and search, security and access control).

Table 1 shows the summary of analysis of ontology development tools included in this survey, according to the above mentioned criteria. This analysis gives us a survey of currently existing tools and languages for ontology development from the point of their maturity, extensibility, interoperability, portability, as well as other criteria that could tell us about the further roles and capabilities of these tools. Based on mentioned analysis, we come to the following conclusions about current ontology development tools:

- Ontology languages from the pre-XML era have attained a level of maturity. Unlike them, tools and ontology development languages from the XML-era still aren't mature. Many existing tools have not yet reached maturity and also, many tools are still lacking it. Hence, they are continuously evolving. New research areas emerge from deploying the idea of intelligent Web services (a combination of the emerging Semantic Web and Web services technologies), but require new research efforts, new development tools, and new tools for dynamic management of the Web.
- From the criteria for ontology development tool extensibility, we notice that there is a trend of further adaptation of existing ontology development tools to the new Web standards (W3C recommendations), such as RDF (Resource Definition Framework), OWL (Web Ontology Language). Also, we can stress the importance of the newly proposed ISO standard, known as CL (Common Logic) that will be compatible with all the accepted W3C standards [7]. But, this trend is not equally represented in all of these tools.
- Certain problems are related to the ontology development tools interoperability. Usually, different research groups develop different tools and, as a consequence, ontology development environments and tools are not interoperable. These tools have different knowledge models, use the different technology, and it is often difficult to integrate them. More recent ontology development tools allow for exporting and importing ontologies in XML and other markup languages as a mean of exchanging ontologies between the tools. This can improve the level of interoperability between the tools.
- Like the ontology development tools extensibility criteria, the portability criteria pertain to the ability of a tool to adapt easily to a new environment as well. A good example is Protégé-2000 ontology editor (see http://protege.stanford.edu), which has a component framework for easily integrating other components

via plug-ins. Thus, Protégé-2000 brings on a great potential to expand and also, adapt itself to the new development environment. But, this is not a case with all these tools.

– The 'ease-of-use' criteria is very important since it implies a necessity to use intuitive screen designs for anyone who will work in the area of ontology development, maintenance, deployment, merging, and update. However, current ontology development tools require their users to be trained in knowledge representation and abstraction.
– The use of ontology development tools in the sense of discovery and search criteria is important in the Web environment to find some potentially interesting new knowledge. Moreover, this criterion is related to the ability of validating, evolving, and maintaining this knowledge.

The results of evaluation and comparison of some existing ontology development tools is shown in Table 1, since Table 2 shows a summary and explanation of symbols used in Table 1.

Table 2. The analysis of ontology development tools based on proposed framework

	Maturity	Extensibility	Interoperability	Portability	Ease- of - use	Discovery / Search
Pre-XML era languages						
F-Logic		E	E	G	F	E
Ontolingua		E	E	G	G	G
KIF		E	E	P	F	P
LOOM		P	F	F	F	G
XML era languages						
SHOE		M	E	G	G	P
CKML		G	E	G	G	P
OIL		G	E	E	G	G
DAML+OIL		G	G	G	G	G
RDF, RDFS		E	E	G	E	G
OWL		E	E	G	G	E
CL		E	E	E	-	G
Ontology editors						
Protégé-2000		E	E	G	E	E
OilEd		G	G	G	E	E
OntoEdit		E	G	G	E	E
WebOnto		G	M	M	M	G
OntoSaurus		M	M	M	G	G
Chimaera		M	M	G	M	G
Jena		G	E	G	E	E
Ontology annotators						
Annotea		G	G	G	G	-
Annozila		G	G	G	G	-
Amaya		E	G	E	E	-
MnM		E	E	E	G	-
OntoMat		E	G	G	G	-

SMORE		G	E	G	G	-
Ontology reasoning/ inference						
FaCT		M	G	G	G	E
RACER		G	G	G	G	E
F-OWL		E	E	E	G	E
DAMLJessKB		E	E	E	G	E
OWLJessKB		G	G	G	G	E
Algernon		E	E	G	G	E
Ontology learning tools						
ASIUM		NA	G	G	G	G
CORPORUM OntoBuilder		NA	M	G	G	E
OntoLearn		NA	F	NA	NA	G
Text-To-Onto		M	M	G	G	G
OntoBuilder		M	G	M	G	G

Table 2. Summary of symbols and theirs meanings

	Early phase	E	Excellent	P	Poor
	Development phase 1	G	Good	NA	Not Available
	Development phase 2	M	Moderate	ND	Not Defined
	Full implementation	F	Fair		

CONCLUSION

This paper provides a comprehensive survey of both earlier and current ontology development tools. It starts from some of the existing surveys of ontology development tools, but its approach is different – it is based on the suggested simple framework/hierarchy of currently available ontology development tools.

References

McGuinness, D. L.: Ontologies Come of Age. In: Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (eds.), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. Boston: MIT Press. 2002, pp. 171-194.

Klein, M.: Tutorial: The Semantic Web - XML, RDF, and Relatives. *In IEEE Intelligent Systems.* Vol. 16, No. 2, 2001, pp. 26-28.

Devedžić, V.: Understanding Ontological Engineering. *Communications of the ACM.* Vol. 45, No. 4, 2002, pp. 136-144.

OntoWeb Consortium: A Survey of Ontology Tools. IST Project IST-2000-29243, *OntoWeb: Ontology-based Information Exchange for Knowledge Management and Electronic Commerce*, Deliverable 1.3, 2002.

Denny, M.: *Ontology Building: A Survey of Editing Tools*. (2002) Available from: http://www.xml.com/pub/a/2002/11/06/ontologies.html [Accessed October 2003].

Gómez-Pérez, A., Corcho, O.: Ontology Languages for the Semantic Web. *In IEEE Intelligent Systems.* Vol. 17, No. 1, 2002, pp. 54-60.

1. Sowa, J.: *ISO Common Logic Standard Proposed*. [Online]. (2002) Available from: http://xml.coverpages.org/CommonLogicStandardAnnounce.html