Emhimed Alatrish

Comparison Some of Ontology Editors

Article Info:

Management Information Systems,
Vol. 8 (2013), No. 2,
pp. 018-024

Received 03 April 2012
Accepted 24 April 2013

UDC 004.4

Summary

In this paper some software tools related to Semantic web are considered and compared. In fact, five ontology-editors are described and compared. They are: Apollo, OntoStudio, Protégé, Swoop and TopBraid Composer Free Edition. The structure and basic features of these editors are described, as well as the way of their using. The main criterion for comparison of these editors was the convenience for user and possibility to apply in different kind of applications.

Keywords

Semantic Web, Ontologies, Ontology building tools, Apollo, OntoStudio, Protégé, Swoop, TopBraid Composer Free Edition

1. Introduction

There are a lot of software tools related to Semantic web. Special significant have Semantic web editors – software for the creation and manipulation of ontologies. Many ontology editors could be found on Internet. Some of them (like: Apollo, OntoStudio, Protégé, Swoop and TopBraid Composer Free Edition) are used from huge number of people. (For example, in (Harith, Kieron, & Nigel, 2005), the Semantic web editor Protégé is classified as killer application where killer applications are defined as highly transformative technologies that create new markets and widespread patterns of behavior). Because of that it is interesting to compare some of these editors. Comparison could be done by using different criterions: generality, expressiveness, complexity, documentation, scalability etc. In our consideration the main criterion is easiness to use and spreading of editors. The most often used editors probably are: Apollo (see The Open University, n.d.), Protégé (see Stanford University School of Medicine, n.d.; Escórcio & Cardoso, 2007; Kapoor & Sharma, 2010; Tudorache, 2009), OntoStudio (see Ontoprise, n.d. a, b), TopBraid Composer Free Edition (see TopQuadrant, n.d; W3C, 2001; Knublauch, 2009) and Swoop (see Swoop, 2004; Kalyanpur, Parsia, Sirin, Grau, & Hendler, 2006; Kapoor & Sharma, 2010) and it is the main reason that we decide to compare these editors.

2. Related Works

For the users of Semantic web it is important to have a good review of available software tools and relations between them. Moreover, the comparison of these tools could be done in different ways. There are a lot of papers where the comparison of ontology editors is made (see: Buraga, Cojocaru, & Nichifor, 2006; Kapoor & Sharma, 2010; Funk, Tablan, Bontcheva, Cunningham, Davis, & Handschuh, 2007; Norta, Carlson, & Yangarber, 2008; Florida, Mulholland, Tyler, Hoffman, & Genest, 2004; Stojanovic & Motik, 2002). Different criterions are used for comparisons and different aspects of ontology tools are considered. For example, in Florida, Mulholland, Tyler, Hoffman, & Genest, 2004 more then ten features for the twenty semantic web tools are described. In (Karim & Tjoa, 2006) a simple test of modeling and visualization is used to compare editors. An interesting view about requirements for ontology – tools evolution is presented in Norta, Carlson, & Yangarber, 2008. The accent is on functional requirements and authors separated eight the most important. Moreover, additional eight non-functional requirements are specified too.

We have used and experimented with five ontology editors (Apollo, OntoStudio, Protégé, Swoop and TopBraid Composer Free Edition) and we decide to take into consideration the advantages and shortages of these tools. These tools are used for building a new ontology either from scratch or by reusing existing ontologies, which usually supports editing browsing, documentation, export and import from difference formats, views and libraries. They may have attached interference engines, include support for some programming language etc. (see Buraga, Cojocaru, & Nichifor, 2006).

3. Ontology Editor's Development Tools

We will try to provide a broad overview of some available editors and environments that can be used for the building of ontologies. We will provide a brief description each of tools and presenting the group that has developed it. Comparison could be done by considering different properties of editors. We will take into account the following characteristics (Kashyap, Bussler, & Matthew, 2008):

- **General description of the tools** includes information about developers and availability.
- **Software architecture and tool evolution** includes information about the tool architecture (standalone, client/server, n-tier application). It also explains how the tool can be extended with other functionalities/modules; furthermore, it describes how ontologies are stored (databases, text files, etc.) and if there any backup management system.
- **Interoperability with other ontology development tools and languages** includes information about the interoperability of the tool. Tool's interoperability with other ontology tools can be recognized by functionalities like (merging, annotation, storage, inference, etc.), in addition to translations to and from ontology languages.
- **Knowledge representation** is related to presenting of knowledge model of the tool. It also includes the possibility of providing any language for building axioms and whether tool gives support to methodology.
- **Inference services attached to the tool** tells if the tool has a built-in inference engine or it can use other attached inference engine. It also shows if the tool performs constraint/consistency checking. It also provides the possibility of classifying concepts automatically in concept taxonomy and capabilities to manage the exceptions in taxonomies.
- **Usability** shows the existence of the graphical editors for the creation of concept taxonomies and relations, the ability to prune these graphs and the possibility to perform zooms of parts of it. It also says if the tool allows some kind of collaborative working and whether it provides libraries of ontologies.

The following candidates have been selected for comparison:

- Apollo;
- OntoStudio;
- Protégé;
- Swoop and
- TopBraid Composer Free Edition.

All these tools are widespread in the ontology design and development sector and are accepted by relatively large semantic web communities. These tools also provide the minimum necessary functionality supporting the ontology development process.

The ontology editors are tools that allow users to visually manipulate, inspect, browse, code ontologies, support the ontology development and maintenance task. In this section, we will provide a broad overview of some of the available ontology editor tools with a brief description of each tool, presenting the group that has developed it, its main features and functionalities, its URL, etc.

3.1. Apollo

Apollo (The Open University, n.d.) is a user-friendly knowledge modeling application. Apollo allows a user to model ontology with basic primitives, such as classes, instances, functions, relations and so on. The internal model is a frame system based on the OKBC protocol. The knowledge base of Apollo consists of a hierarchical organization of ontologies. Ontologies can be inherited from other ontologies and can be used as if they were their own ontologies. Each ontology is the default ontology, which includes all primitive classes. Each class can create a number of instances, and an instance inherits all slots of the class. Each slot consists of a set of facets.

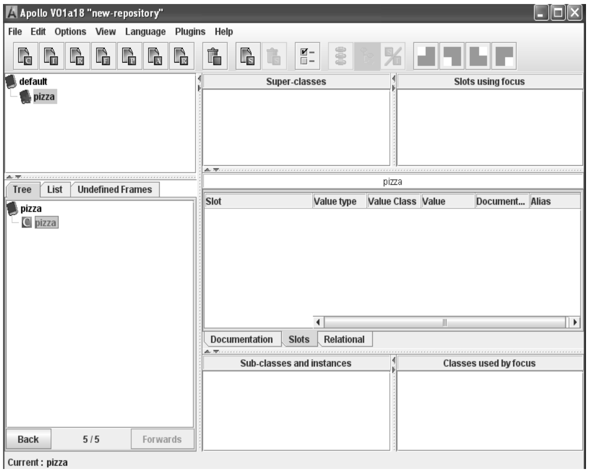
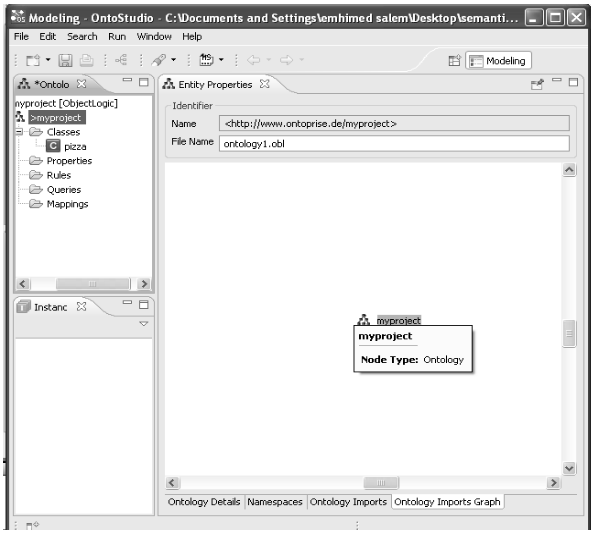


Figure 1 Apollo screenshot

Apollo does not support graph view, web, information extraction and multi-user capabilities or collaborative processing but it features strong type consistency checking, storing the ontologies (files only) and import/export format (I/O plug-in architecture - export plug-ins to CLOS and OCML). Apollo is implemented in Java and it is available for a download from http://apollo.open.ac.uk/index.html.

3.2. OntoStudio

OntoStudio Ontoprise is based on IBM Eclipse framework. It can be downloaded for three months free evaluation period. It is an Ontology Engineering Environment supporting the development and maintenance of ontologies by using graphical means. It is based on client/server architecture, where ontologies are managed in a central server and various clients can access and modify these ontologies. It supports multilingual development, and the knowledge model is related to frame-based languages. It supports collaborative development of ontologies. OntoStudio is built on top of a powerful internal ontology model. The tool allows the user to edit a hierarchy of concepts or classes. OntoStudio (Ontoprise, n.d., a, b) is based on an open plug-in structure. The internal representation data model can be exported to DAML+OIL, F-Logic, RDF(S), and OXML. Additionally, ontologies can be exported to relational databases via JDBC.



Name <http://www.ontoprise.de/myproject>

File Name ontology1.obl

myproject

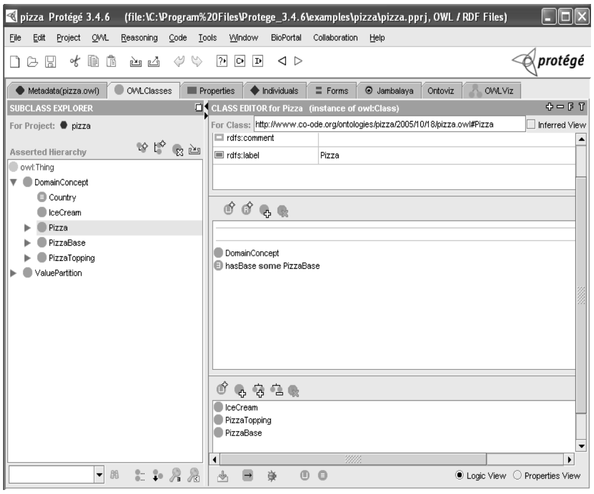
Node Type: Ontology

Figure 2 Screenshot of OntoStudio editor

OntoStudio can import external data representation in DAML+OIL, Excel, F-logic, RDF(S), database schemas (Oracle,MS-SQL, DB2,MySQL), and OXML. OntoSudio can also import and export OWL files. OntoStudio provides an API for accessing ontologies in an object-oriented fashion. The default API implementation stores ontologies in main-memory, but an additional API exists for persistent storage. The inference ngine that OntoStudio uses is OntoBroker (OntoBroker is the result of several years of research and it is now a commercial product). Using this engine, OntoStudio exploits the strength of F-Logic and it can represent expressive rules. OntoStudio supports collaborative ontologies by using the OntoBroker Enhancement Collaborative server.

3.3. Protégé Ontology Editor

Protégé (Stanford University School of Medicine, n.d.) is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-base applications with ontologies. It implements a rich set of knowledge-modeling structures and action that support the creation, visualization and manipulation of ontologies in various representation formats. It can be customized to provide domain-friendly support for creating knowledge models and entering data. Also, it can be extended by a plug-in architecture and Java-based application programming interface (API) for building knowledge-base tools and applications. Protégé allows the definition of classes, class hierarchy's variables, variable-value restrictions, and the relationships between classes and the properties of these relationships.



For Project: ● pizza

For Class: http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl#Pizza

rdfs:comment	
rdfs:label	Pizza

DomainConcept

hasBase some PizzaBase

● Logic View ○ Properties View

Figure 3 Screenshot of Protégé

Protégé is freely available for download. Stanford has a tutorial that covers the basics of using Protégé with the OWL plug-in. Protégé-OWL provides a reasoning API that can access an external DIG-compliant reasoner, enabling the inferences about classes and individuals in an ontology. Protégé (Kapoor & Sharma, 2010) includes an interface for SWRL (Semantic Web Rule Language), which sits on top of OWL to do math, temporal reasoning, and adds Prolog-type reasoning rules.

The significant advantage of Protégé is its scalability and extensibility. Protégé (Escórcio & Cardoso, 2007) allows to build and to process large ontologies in an efficient manner. Through its extensibility Protégé might be adopted and customized to suit users' requirements and needs. The most popular type of plug-ins is tab plug-ins; currently available tabs provide capabilities for advanced visualization, ontology merging, version management, inference, and so on. The OntoViz and Jambalaya tabs, for example, present different graphical views of a knowledge base, with the Jambalaya tab allowing interactive navigation, zooming in particular elements in the structure, and different layouts of nodes in a graph to highlight connections between clusters of data. Protégé (Tudorache, 2009) supports collaborative ontology editing as well as annotation of both ontology components and ontology changes.

3.4. Swoop

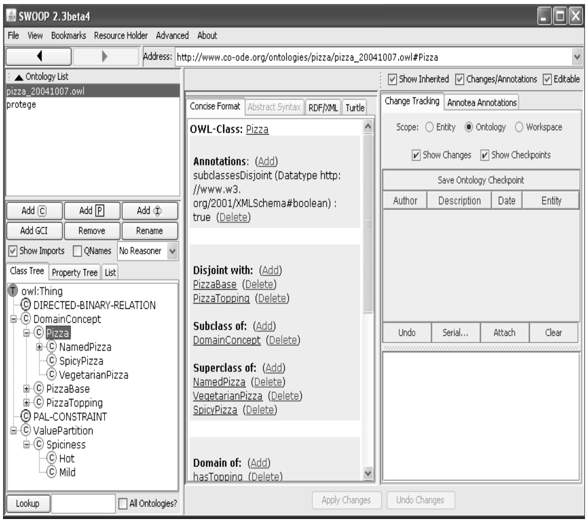
Swoop (Swoop, 2004) is an open-source, Web-based OWL ontology editor and browser. Swoop (Kapoor & Sharma, 2010) contains OWL validation and offers various OWL presentation syntax views (Abstract Syntax, N3 etc). It has reasoning (RDFS-like and Pallet) support (OWL Inference Engine), and provides a Multiple Ontology environment, by which entities and relationships across various ontologies can be compared, edited and merged seamlessly. Different ontologies can be compared against their Description Logic-based definitions, associated properties and instances. Navigation could be simple and easy due to the hyperlinked capabilities in the interface of Swoop. Swoop does not follow a methodology for ontology construction. The users can reuse external ontological data either by simply linking to the external entity, or by importing the entire external ontology. It is not possible to do partial imports of OWL, but it is possible to search concepts across multiple ontologies. Swoop uses ontology search algorithms that combine keywords with DL-based constructs to find related concepts in existing ontologies. This search is made along all the ontologies stored in the Swoop knowledge base. Swoop (Kalyanpur, Parsia, Sirin, Grau, & Hendler, 2006) has collaborative annotation with the Annotea plug-in.

3.5. TopBraid Composer Free Edition

TopBraid Composer (TopQuadrant, n.d) comes in three editions: Free Edition (FE) is an introductory version with only a core set of features. Standard Edition (SE) includes all features of FE plus graphical viewers, import facilities, advanced refactoring support and much more. Maestro Edition (ME) includes all features of SE plus support for TopBraid Live, EVN and Ensemble as well as SPARQLMotion and many other power user features.

TopBraid Composer (FE) a component of TopBraid Suite is a professional development tool for semantic models (ontologies). It is based on the Eclipse platform and the Jena API. It is a complete editor for RDF(S) and OWL models, as well as a platform for other RDF-based components and services. TopBraid Composer (FE) can (Knublauch, 2009) loads and save any OWL2 file in formats such as RDF/XML or Turtle.

TopBraid Composer (FE) (W3C, 2001) supports various reasoning and consistency checking mechanisms. Consistency checking and debugging is supported by built-in OWL inference engine, SPARQL query engine and Rules engine. OWL description logic is supported via a range of built-in OWL DL engines such as OWLIM, Jena and Pellet.



Address: http://www.co-ode.org/ontologies/pizza/pizza_20041007.owl#Pizza

pizza_20041007.owl
protege

☑ Show Imports ☐ QNames No Reasoner

☐ All Ontologies?

☑ Show Inherited ☑ Changes/Annotations ☑ Editable

Scope: ○ Entity ● Ontology ○ Workspace

☑ Show Changes ☑ Show Checkpoints

OWL-Class: Pizza

Annotations: (Add)
subclassesDisjoint (Datatype http://www.w3.org/2001/XMLSchema#boolean) : true (Delete)

Disjoint with: (Add)
PizzaBase (Delete)
PizzaTopping (Delete)

Subclass of: (Add)
DomainConcept (Delete)

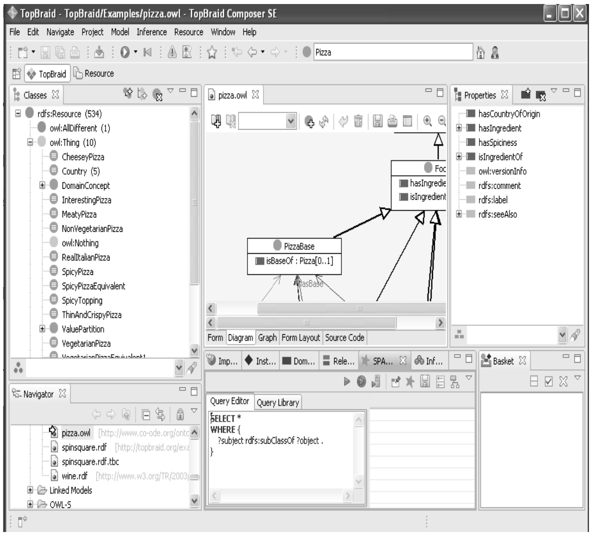
Superclass of: (Add)
NamedPizza (Delete)
VegetarianPizza (Delete)
SpicyPizza (Delete)

Domain of: (Add)
hasTopping (Delete)

Figure 4 Screenshot of Swoop

TopBraid composer (FE) (TopQuadrant, n.d) also supports the SPARQL inference Notation (SPIN). SPIN can also be used to define integrity constraints that can be used to highlight invalid data at edit time. TopBraid Composer (FE) also provides inference explanation facilities for Pellet and SPIN. It can be used to connect RDF resources or ontologies with geospatial ontologies. It can be used to query for resources within a specific region. It can parse such documents to extract RDFa metadata from HTML pages; the metadata can then be treated like any other RDF source, and users can perform DL reasoning or SPARQL queries on it.

TopBraid Composer may use in a single user mode working with ontologies stored as files or in a database. TopBraid Composer (FE) (TopQuadrant, n.d) is a modeling tool for the creation and maintenance of ontologies. It provides: standard-based, syntax directed development of RDF(S) and OWL ontologies, SPARQL queries and Semantic Web rules import/export from a variety of data formats including RDF(S), XML, Excel, etc. Usability, extensibility and robustness are provided by underlying technologies – Eclipse and Jena. TopBraid Composer (FE) can download and evaluate the full version for a 30 days evaluation period.



```
SELECT *
WHERE {
  ?subject rdfs:subClassOf ?object .
}
```

Figure 5 TopBraid Composer(FE) screenshot

4. Comparison of Tools

The comments concerning this section are based on tools that have been described above. The tools are specified in alphabetical order: Apollo, OntoStudio, Protégé, Swoop and TopBraid Composer (FE).

General description of the tools (Table 1) includes information about developers and availability. We can see form (Table 1) that: Apollo, Protégé and Swoop are open source and OntoStudio and TopBraid Composer (FE)are under the software license.

Table 1 General description of the tools

Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
Developers	KMI (Open University)	Ontoprise	SMI (Stanford University)	MND (University of Maryland)	TopQuadrant
Availability	Open source	Software license	Open source	Open source	Software license

Software architecture and tool evolution (Table 2) includes information about the necessary platforms to use the tool. In fact, the following information is provided: default architecture (standalone, client/server, n-tier application), extensibility, storage of the ontologies (databases, ASCII files, etc.) and backup management. All of these tools are moving towards Java platforms, Swoop is web based and Protégé, OntoStudio and Swoop have client/server architecture. Protégé, OntoStudio and TopBraid Composer (FE) use databases for storing ontologies.

Table 2 Software architecture and tool evolution

Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
Semantic web architecture	Standalone	Eclipse client/server	Standalone and Client-server	Web-based and Client-server	Standalone Eclipse plug-in
Extensibility	Plug-ins	Plug-ins	Plug-ins	Yes Via plug-ins	Plug-ins
Backup management	No	No	No	No	Yes
Ontology storage	Files	DBMS	File and DBMS (JDBC)	As HTML Models	DBMS

Interoperability (Table 3) includes information about the tools interoperability with other ontology development tools and languages, translations to and from some ontology languages. It is another important feature in the integration of ontologies in applications.

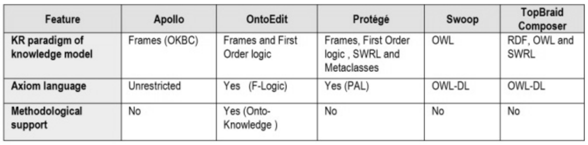
Table 3 Interoperability

Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
With other ontology tools	No	OntoAnnotate, OntoBroker, OntoMat, Semantic and Miner	PROMPT, OKBC, JESS, FaCT and Jena	No	Sesame , Jena and AllegroGraph
Imports from languages	Apollo Meta language	XML(S), OWL, Excel, RDF(S),UML2.0,database schemas (Oracle,MS-SQL, DB2,MySQL), Outlook E-mails	XML(S), RDF(S), OWL, HTML, (RDF,UML, XML)backend, text file, RDF file,Excel, BioPortal and DataMaster	OWL, XML, RDF and text formats	RDFa, WOL, XML(S),RDF(s) ,XHTML, UML, GRDDL, RDB with D2RQ, Microdata and RDFa Web Data Sites, SPIN, Spreadsheets, Oracle database, text file, RDF file, News Feed, Email and Excel
Exports to languages	OCML and CLOS	OWL, RDF(S), RIF, SPARQL, F-Logic and Excel	XML(S), RDF(S),OWL, HTML, Java, Clips, F-Logic SWRL-IQ, Instance Selection, MetaAnalysis, OWLDoc, Queries and (RDF,UML, XML)backend	RDF (S), OIL and DAML	HTML,UML, XSD, Excel , RDB, Oracle database, RDF File, XML File and Text File

Most of these tools support import and export to and from many languages in a variety of formats. TopBraid Composer(FE) supports the import to RDFa, WOL, RDF(s), XHTML, Microdata and RDFa Data sources, SPIN, News Feed, Email and Excel. Swoop supports RDF (S), OIL and DAML. Apollo supports Apollo metalanguage, etc. Protégé supports the import of text files, database tables and RDF files. OntoStudio supports database schemas (Oracle, MS-SQL, DB2, MySQL), Outlook E-mails, etc. Protégé, OntoStudio and TopBraid Composer (FE) are support the export of Merge / Convert RDF Graphs, RDF(S) and WOL. Also Swoop supports RDF (S), OIL and DAML and Apollo supports Apollo metalanguage. Most of them support OWL, RDF(S) and XMl(S). However, there is no comparative study on the quality of each of these translators. Moreover, there are no experimental results about the possibility of exchanging ontologies between different tools and knowledge on the loose in the translation processes.

From the **knowledge representation** point of view (Table 4), there is the family of tools which allows representation of knowledge following a hybrid approach based on frames and first order logic. Additionally, Protégé provides flexible modeling components like metaclasses. OntoStudio gives support to the Onto Knowledge methodology.

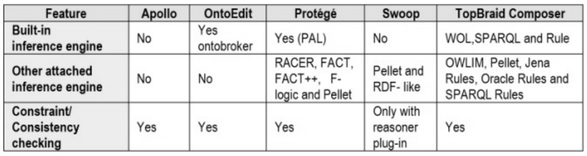
Table 4 Knowledge representation and methodological support



Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
KR paradigm of knowledge model	Frames (OKBC)	Frames and First Order logic	Frames, First Order logic , SWRL and Metaclasses	OWL	RDF, OWL and SWRL
Axiom language	Unrestricted	Yes (F-Logic)	Yes (PAL)	OWL-DL	OWL-DL
Methodological support	No	Yes (Onto-Knowledge)	No	No	No

Inference services are presented in Table 5. This includes: built-in and other inference engines, constraint and consistency checking mechanisms, automatic classifications and exception handling, among others. For built-in inference engine Protégé uses PAL, OntoStudio uses OntoBroker and TopBraid Composer (FE) uses WOL, SPARQL and Rule. Protégé, Swoop and TopBraid Composer (FE) have external attached inference engines. TopBraid Composer (FE) uses the Exception Handling.

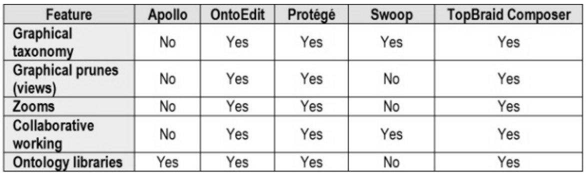
Table 5 Inference services are attached to it



Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
Built-in inference engine	No	Yes ontobroker	Yes (PAL)	No	WOL,SPARQL and Rule
Other attached inference engine	No	No	RACER, FACT, FACT++, F-logic and Pellet	Pellet and RDF- like	OWLIM, Pellet, Jena Rules, Oracle Rules and SPARQL Rules
Constraint/ Consistency checking	Yes	Yes	Yes	Only with reasoner plug-in	Yes

Usability is related to Graphical editors, collaborative working and the provision of reusable ontology libraries. For the most users, Protégé provides friendly possessing and easy to use graphical interface. Additionally, in Protégé and OntoStudio the layout of the interface and visualization of the ontology can be customised. Protégé and OntoStudio allow graphical taxonomy viewing, pruning and zooming. Help systems are essential to users and should be readily available and easy to use. The Apollo, Protégé, OntoStudio and TopBraid Composer (FE) help system is made up of a help on icons, tutorial, users guide. Swoop do not provide a help function in the user interface. Collaboration is essential in the process of building very large and extensive ontologies and Protégé, OntoStudio and TopBraid Composer (FE) allow collaborative construction of ontologies. Swoop allows users to write and share annotation on any ontological entity.

Table 6 Usability of tools



Feature	Apollo	OntoEdit	Protégé	Swoop	TopBraid Composer
Graphical taxonomy	No	Yes	Yes	Yes	Yes
Graphical prunes (views)	No	Yes	Yes	No	Yes
Zooms	No	Yes	Yes	No	Yes
Collaborative working	No	Yes	Yes	Yes	Yes
Ontology libraries	Yes	Yes	Yes	No	Yes

5. Conclusion

To sum up, the Apollo, Protégé 3.4 and Swoop tools are open source ontology and for the tools: OntoStudio and TopBraid composer (FE) software license is requested. Also, there are some tools application ontology demand learning / knowledge of a particular language Swoop. The tools: Protégé, TopBraid Composer (FE) and OntoStudio use databases for storing ontologies. The same applies to backup management functionality, which is just provided by TopBraid Composer (FE). Protégé and OntoStudio are more graphical ontology tools. The Swoop is Web-based application. OntoStudio gives support to the Onto Knowledge methodology. TopBraid Composer (FE) uses the Exception Handling. Some of the tools only support the joint edition of the functions of browsing. Protégé, TopBraid Composer (FE), Swoop and OntoStudio editors provide documentation ontology, ontology import / export to different formats, graphical view of ontologies, ontology libraries and attached inference engines and Apollo supports Apollo metalanguage.

It is quite clear that Ontology development is mainly an ad-hoc approach. Among several viable alternatives, a user needs to find which one would work better for the projected task and which one easily and effectively can be maintained and expressed. The foundation of ontology is logic, but in same time it is a model of reality and the concepts in the ontology must reflect this reality. We have described a tool-assisted method for building the basis for ontologies adopted from domain analysis.

References

Buraga, S. C., Cojocaru, L., & Nichifor, O. C. (2006). *Survey on Web Ontology Editing Tools. Science,.* Retrieved January 11, 2012 from Alexandru Ioan Cuza University: http://profs.info.uaic.ro/~busaco/publications/articles/web-ontology-tool-survey.pdf

Escórcio, A., & Cardoso, J. (2007). Editing Tools for Ontology Creation. In J. Cardoso, *Semantic Web Services: Theory, Tools and Applications (Eds.)* (pp. 1-27). Hershey: Information Science Reference.

Florida, W., Mulholland, P., Tyler, N., Hoffman, R., & Genest, D. (2004). *Ontology editor survey results.* Retrieved January 15, 2012 from O'Reilly xml.com: http://www.xml.com/2004/07/14/examples/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf

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

Harith, A., Kieron, O., & Nigel, S. (2005). Common features of killer apps: A comparison with Protégé. *8th International Prot Conference, 18-21 July 2005.* Madrid, Spain.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., & Hendler, J. (2006). Swoop: A Web Ontology Editing Browser. *Web Semantics: Science, Services and Agents on the World Wide Web , 4* (2), 144-153.

Kapoor, B., & Sharma, S. (2010). A Comparative Study Ontology Building Tools for Semantic Web Applications. *International Journal of Web & Semantic Technology , 1* (3), 1-13.

Karim, S., & Tjoa, A. M. (2006). Towards the Use of Ontologies for Improving User Interaction for People with Special Needs. *Computers Helping People with Special Needs. 4061*, pp. 77-84. Berlin: Springer-Heidelberg.

Kashyap, V., Bussler, C., & Matthew, M. (2008). *The Semantic Web: Semantics for Data and Services on the Web (Data-Centric Systems and Applications).* Berlin: Springer-Verlag.

Knublauch, H. (2009, October 27). *OWL 2 Support in TopBraid Composer.* From Composing the Semantic Web : http://composing-the-semantic-web.blogspot.com/2009/10/owl-2-support-in-topbraid-composer.html

Norta, A., Carlson, L., & Yangarber, R. (2008). *Utility Survey of Ontology Tools.* Retrieved January 15, 2012 from University of Helsinki: http://www.cs.helsinki.fi/u/anorta/publications/OntologyToolEvaluationStatements.pdf

Ontoprise. (n.d.). *Comprehensive ontology building environment developed by Ontoprise GmbH.* Retrieved January 11, 2012 from Ontoprise: http://www.ontoprise.de/en/products/ontostudio/

Ontoprise. (n.d.). *OntoStudio 3 Professional tool for knowledge architects.* Retrieved January 11, 2012 from Ontoprise: http://www.ontoprise.de/fileadmin/user_upload/Flyer_EN/Flyer_OntoStudio_en.pdf

Stanford University School of Medicine. (n.d.). *Protégé ontology editor developed by Stanford Medical Informatics, Stanford University School of Medicine.* Retrieved January 11, 2012 from Stanford University School of Medicine: http://protege.stanford.edu/

Stojanovic, L., & Motik, B. (2002). *Ontology Evolution within Ontology Editors.* Retrieved January 15, 2012 from Knowledge Management Group: http://km.aifb.kit.edu/ws/eon2002/EON2002_Stojanovic.pdf

Swoop. (2004). *Semantic Web Ontology Overview and Perusal.* Retrieved January 11, 2012 from MINDSWAP: http://www.mindswap.org/2004/SWOOP/

The Open University. (n.d.). *Ontology building editor developed by Knowledge Media Institute, The Open University.* Retrieved January 11, 2012 from The Open University: http://apollo.open.ac.uk/

TopQuadrant. (n.d.). *Top Braid Composer.* Retrieved January 11, 2012 from TopQuadrant: http://www.topquadrant.com/products/TB_Composer.html

Tudorache, T. (2009, February 13). *Collaborative protégé.* Retrieved January 12, 2012 from Protégé wiki! : http://protegewiki.stanford.edu/wiki/Collaborative_Protege

W3C. (2001). *TopBraid.* Retrieved January 11, 2012 from W3C: http://www.w3.org/2001/sw/wiki/TopBraid

Emhimed Alatrish
University of Belgrad
Faculty of Mathematics,
Studentski trg 16
11000 Beograd
Serbia
Email: emhimed_Alatrish@hotmail.com