

User-friendly ontology editing and visualization tools: the OWLeasyViz approach

Nadia Catenazzi, Lorenzo Sommaruga, Riccardo Mazza
Semantic and Multimedia Systems Lab. (LSMS), Dept. of Innovative Technologies (SUPSI-DTI)
University of Applied Sciences of Southern Switzerland, CH-6928 Manno
{nadia.catenazzi; lorenzo.sommaruga; riccardo.mazza}@supsi.ch

Abstract

This paper aims to propose solutions to the issue of ontology visualization, by presenting intuitive and user-friendly ontology editing and visualization environments mainly oriented to domain experts. It starts with an overview of existing ontology visualization methods; afterwards it describes the Semantic DB system and the OWLeasyViz ontology editor. Semantic DB is a web application framework to create simple complete semantic web applications, integrating an ontology editor, a resource editor, an inference rule editor, a reasoner, and a search engine. OWLeasyViz is an ontology editor that combines a textual and a graphical representation of OWL ontologies. It meets different user needs by providing a simple and intuitive interface to end-users who are not ontologists, and offering more advanced tools to ontology experts. The OWLeasyViz editor is intended to be a module of a semantic web integrated working environment, developed within the context of a Swiss Government funded CTI applied research project in the domain of waste water management.

Keywords-- ontology visualization, ontology editing, semantic web framework.

1. Introduction

The process of ontology engineering is a complex and time-consuming task. The major problem deals with knowledge acquisition and maintenance, achieving consensus among the domain experts and other interested parties. Ontology engineering requires the support of specific tools that help users carry out some of the main activities of the ontology development process, such as conceptualization, implementation, consistency checking, and documentation [5]. There are a number of studies where different ontology development tools are analysed and compared [1,3,5]. One of the most important issues emerged from these works is the need for user-friendly interfaces. From a study of ontology

development tools [4], it emerges that “the overall sentiment expressed by users of the various ontology development environments clearly reflected the need for facilitating the use of such tools by domain experts rather than by ontologists”. In fact, domain experts have knowledge and competencies to create the conceptualization of the domain, i.e. the ontology, and should be therefore the primary target users of these frameworks. On the other hand, the current ontology development tools are commonly designed for the interaction with ontologists, who usually speak the RDF and OWL terminology.

To facilitate the process of ontology editing, a higher abstraction level of the ontology constructs is required. The ontology constructs must be expressed in a more intuitive and powerful way, by means of a visual interactive representation that is a simplification of the world, reducing, where necessary and possible, complexity without losing completeness.

This paper aims to provide a solution to this issue, by proposing intuitive and user-friendly ontology editing and visualization environments mainly oriented to domain experts.

The starting point for this work is an analysis of the existing approaches for ontology visualization; the results are reported in the next section.

2. Analysis of existing ontology visualization methods

Different studies in the field have been conducted. A pioneering book [6] presents the state of the art in the area of semantic web visualization, focusing on several topics such as visualization of semantic data and metadata, topic maps, ontology visualization, SVG/X3D for Semantic Web, etc.

Concerning specifically ontology visualization, Katifori [8] provides a detailed overview of the existing visualization methods and their advantages and disadvantages. These methods are grouped in several categories: indented list, node-link and tree, zoomable, focus + context, space-filling, and 3D Information Landscapes. Among these, the most frequently used for

ontology visualization are the first four categories, briefly considered below.

The *indented list* methods represent the taxonomy of the ontology following the file system explorer-tree view. The Protégé Class Browser [13] is an example (see left part of the window in figure 1). These methods are intuitive and simple to implement, as also confirmed by the results of a user evaluation [9] where different visualization methods were compared. Their main drawback is that they represent a tree hierarchy and not a graph, and do not visualize role relations.

The *node-link and tree* methods represent another approach frequently used for ontology visualization. The ontology is represented as a set of interconnected nodes. They offer a good overview of the hierarchy and connections, but may produce cluttered displays when used to visualize more than hundred of nodes. Examples of this category are OWLViz [12], OntoViz [11] and RDF Gravity [14]. The right part of the window shown in figure 1 displays the ontology hierarchy using the OWLViz Protégé plug-in.

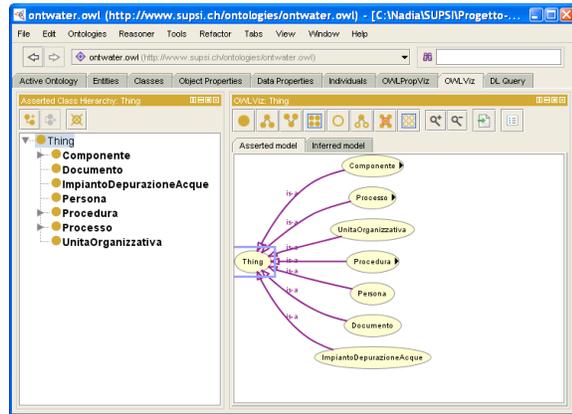


Figure 1: Protégé Class Browser and OWLViz plug-in

The *zoomable* methods present the nodes in the lower levels of the hierarchy nested inside their parents and with smaller size. The user may zoom-in to the child nodes to enlarge them. An example based on this approach is Jambalaya [7] (see figure 2). These methods seem to be successful for browsing to locate specific nodes, but they do not offer an effective overview of the hierarchical structure and may produce disorientation after zooming-in several times. In order to avoid the loss of context problem, some orientation clues could be added.

Finally, the *focus + context* methods present the node on the focus in the centre and the connected nodes around it, usually reduced in size. The TGVizTab Protégé plug-in [16] is an example. This technique is effective to provide global overviews, to focus on specific nodes, and for quick browsing; however, it may produce messy visualizations and make difficult for the user to create a mental model of the ontology as the

graph is continuously redrawn, and node positions re-arranged.

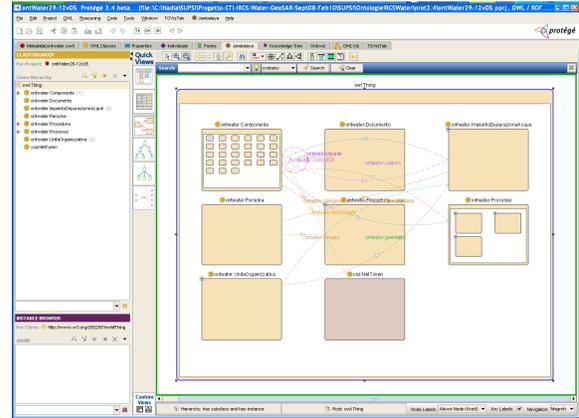


Figure 2: Protégé Jambalaya plug-in

Considering the variety of methods and approaches to visualize ontologies, the question now is which method to choose. A key issue to be taken into account is the specific user task the visualization method is expected to support, e.g. overview, zoom, filtering, editing, etc. Another factor to be considered is the ontology scalability. The choice of the method also depends on the number of nodes to be visualized. When this number is over 1000, 3D visualization methods, by exploiting more space availability, seem to perform better than 2D; alternatively, clustering or filtering techniques may be used. Another issue to be considered is the end-user profile: while an ontology expert can easily understand the ontology “syntax”, the domain expert needs an abstraction of the ontology constructs to be able to use them to create a conceptualization of the domain.

The conclusion is that there is not a method that is always the best. Katifori [8] suggests to give the user the option to choose among several visualization methods.

In spite of the differences among the techniques presented above, some features which should always be available are query mechanisms to identify nodes and relations of interest, and filtering facilities to hide part of the ontology, reducing the information overload. Another desirable feature, currently rarely supported, is the incorporation of reasoning mechanisms, and the visualization of their effects.

3. User-friendly visualization tools for the Semantic Web: the Semantic DB

Some previous works carried at SUPSI in the context of Semantic Web visualization include tools to visualize and navigate RDF resources, such as the Star Resource Navigator [2], and general frameworks for Semantic Web applications, such as the Semantic DB.

We focus here on the description of the latter, being relevant to ontology visualization.

The Semantic DB [10] is a web application framework to create simple complete semantic web applications; it was carried out at LSMS SUPSI DTI in partial fulfillment for a master thesis. The Semantic DB system provides a number of facilities to create and manage semantic worlds, including searching and reasoning mechanisms, and offers an intuitive graphical visualization of ontologies and resources. In particular, the framework integrates an (OWL) ontology editor, a (RDF) resource editor and navigator, an inference rule editor, a reasoner to make inference on the basis of the defined rules, a search engine to find resources, and a path search engine to identify paths among any pair of resources. All these facilities are provided through an intuitive graphical interface. A key feature of this work is that the framework represents a fully integrated working environment where the user can simultaneously operate (define, change, etc.) both on the structure of the world, i.e. the ontology, and on its individuals, i.e. resources.

The ontology editor is shown in figure 3. It consists of a textual editing area (left part of the window enlarged in figure 4), and a graphical editing area (right part).

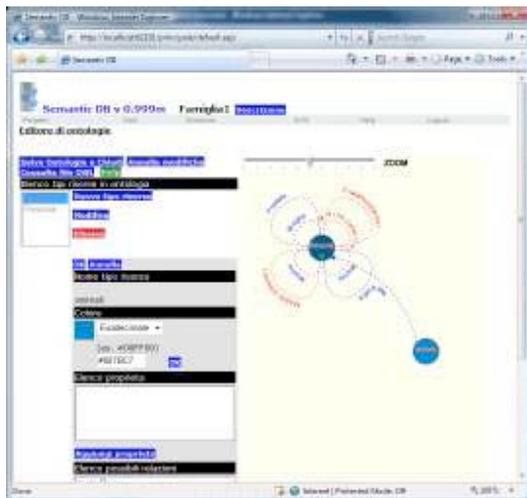


Figure 3: The Semantic DB ontology editor

The visualization method is graph-based: circle nodes represent classes and arcs are relations. Properties are visible only if the user moves the mouse on the top of a node. A small green circle inside a node indicates whether it has properties. The graph shows both explicit and inferred relations by using a different color. Search, zooming and filtering mechanisms are also provided. The graphical interface is mainly based on SVG and Ajax, currently implemented as a rich internet Asp.Net

2.0 application with a core Jena API library and an SQL server DBMS.

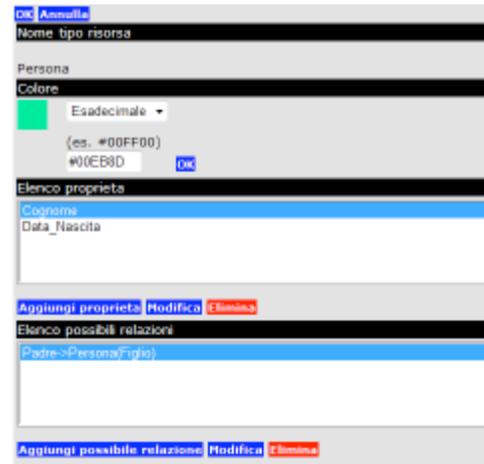


Figure 4: The textual editing area for class editing

The main advantage of this tool is that it allows users with basic semantic web knowledge to create and manage any kind of semantic worlds, and make inferences on them. The system is mainly used as a didactic tool to demonstrate the potential of the semantic web technologies to non-expert people.

At the moment the ontology editor does not provide full support to the OWL specification, and considers a limited number of OWL constructs. However, it is sufficiently expressive to allow a simple ontology and semantic world of resources to be defined and integrated into the framework.

4. The OWLeasyViz approach for a user-friendly ontology editing and visualization tool

On the basis of the analysis of existing methods for ontology visualization (section 2) and taking into account the experiences described in section 3, the OWLeasyViz approach for ontology visualization is proposed.

The main drawback of many ontology visualization tools is that they are too complex to be used by domain experts who are not ontologists. Other visualization mechanisms are intuitive but restricted: for instance OWLViz is simple but limited to a hierarchical visualization, hiding essential elements of the ontology, such as role relations.

The driving philosophy for our approach is meeting different user needs, i.e. providing a simple and intuitive interface to end-users who are not ontologists, and offering more advanced tools to ontology experts. Following the Semantic DB model, the ontology editor is intended to be integrated in a more general working framework, including both resource and ontology editor tools.

The OWLeasyViz approach combines a textual and graphical representation of OWL ontologies (see figure 5).

The textual representation presents class, data properties and object properties in a three-column table. The first column is the class browser, which provides a simple representation of the class hierarchy (*indented list method category*). The second one contains the object properties, and the third one data properties. By selecting a class, all its properties are shown. Those properties inherited from a super-class are visualized in a different color.

and properties. Therefore class and property names are visualized without their namespace. This basic textual visualization strategy has been introduced because, as already highlighted, it is intuitive and effective.

The graphical representation separates the “is-a” inheritance relationship from the role relations, simplifying the visual presentation of the world. Hierarchies are represented as nested sets. Child nodes are visualized inside their parents, with smaller size. Different shapes distinguish nodes that are at the lowest level of the hierarchy (leaves) from the others. Leaf nodes are visualized as rounded rectangles, while parent nodes as elliptical shapes.

When the user clicks on a child node, the node is enlarged and its content is made visible, using the zoomable technique.

Searching and filtering mechanisms are also provided, facilitating ontology access, navigation, and visualization. Searching mechanisms allow full text

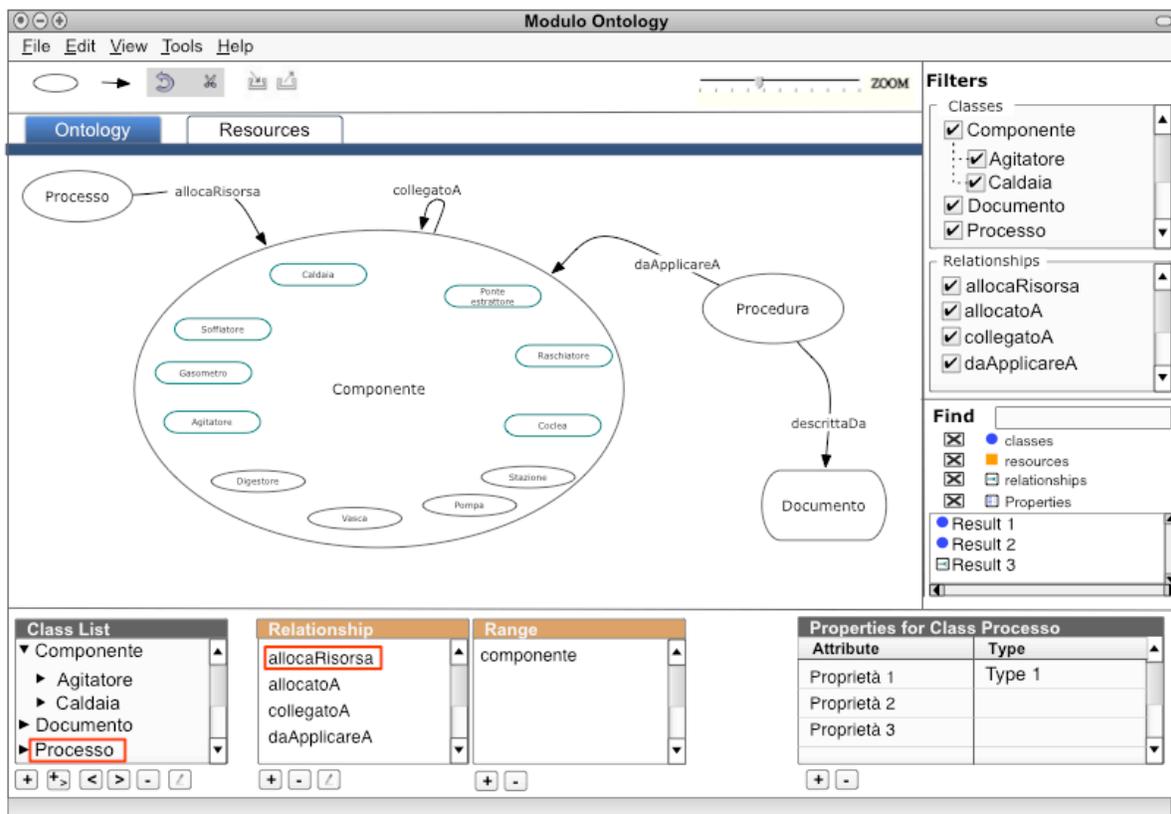


Figure 5: the EasyViz interface

In order to simplify the user interaction, technical terms have been replaced by more intuitive terms; for instance *relationship* is used instead of *ObjectProperty*, and *property* instead of *DatatypeProperty*. Fully qualified identifiers used in the OWL ontology are hidden in order to simplify the naming of OWL classes

search to be carried out and the list of results to be shown. Items selected in the list are highlighted in the graphical visualization. The filtering tool allows the graphical representation to be simplified by showing only the selecting classes, relations, and properties.

A possible drawback of this technique is the lack of an effective visualization of the hierarchical structures;

however, the fact that it is used in combination with the class browser guarantees a clear representation of the hierarchies and easy orientation.

This ontology visualization model exploits the visualization strategies used in *Grokker* (<http://www.grokker.com/>), a general system for the display of knowledge maps, and *Jambalaya*, a Protégé plug-in specifically developed for ontology visualization (<http://www.thechiselgroup.org/jambalaya>). With respect to Grokker, that is limited to visualize hierarchies, the OWLeasyViz approach also shows role relations and allows a graph to be effectively represented. Compared to Jambalaya, the OWLeasyViz approach is simpler and neater.

The representation described so far allows a subset of OWL constructs to be visually presented, including classes, properties, and relations. In order to keep the interface simple, and, at the same time, to extend the OWL constructs the tool is able to manage, the adopted solution is to make additional OWL constructs available through “advanced options”. These constructs include restrictions, relation features, and multiple inheritance. The OWLeasyViz visualization approach is therefore adaptable and meets the needs of different kinds of target user.

A prototype, which implements the basic functionalities illustrated in figure 4, is currently under development within the context of the IRCS Project (CTI project nr. 9402.1 PFES-ES), a Swiss Government funded applied research project in the domain of waste water management.

In the following phase the more advanced options will be considered. This visualization approach is currently used to represent an ontology describing a water plant and the involved resources from physical devices to organization units and documentation. Some of this ontology classes are shown in figure 5.

Conclusions

A user-friendly ontology visualization environment produces benefits both for the ontology developer team, who necessarily involves domain experts, and for any user who has to create resources based on the ontology specifications.

This paper has presented different experiences in the field of ontology visualization.

One of the drawbacks of simple and intuitive systems is that keeping the interface design simple and intuitive means limiting the number of constructs that it is possible to use, reducing the ontology expressivity.

This problem emerged in the Semantic DB system. The OWLeasyViz approach proposes a solution to this issue by keeping the main working area as simple as possible, hiding levels of detail or making abstractions of some ontology constructs, and inserting more complex ontology constructs under “advanced” options.

Our experience demonstrate the importance and effectiveness of integrating in a single working environment all the tools required to work with a semantic world: an ontology editor, a resource editor, searching and filtering tools, and inference mechanisms. This integration allows users to operate simultaneously on ontology, resources, and inference rules, exploiting the potential of semantic web technologies to solve real-life problems.

References

- [1] J. Cardoso. Editing construction for ontology, chapter 4 of *Semantic Web Services: Theory, Tools and Applications*. 2006.
- [2] N. Catenazzi, L. Sommaruga. Practical experiences towards generic resource navigation and visualization, *Proceedings of SWAP 2005, the 2nd Italian Semantic Web Workshop, Trento, Italy, December 14-16, CEUR Workshop Proceedings, ISSN 1613-0073, URL: <http://ceur-ws.org/Vol-166/58.pdf>*. 2005.
- [3] M. Denny. *Ontology Tools Survey*, Revisited URL: <http://www.xml.com/pub/a/2004/07/14/onto.html>. 2004.
- [4] D. Gašević, D. Djurić, and V. Devedžić. Chapter 2: ontologies in *Model Driven Architecture and Ontology Development*, Springer. 2006.
- [5] A. Gómez-Pérez, F. Ortiz-Rodriguez, B. Villazón-Terrazas. *Ontological Engineering*, Springer. 2004.
- [6] V. Geroimenko, C. Chen, (Eds.) *Visualizing the Semantic Web, XML-based Internet and Information Visualization*, Springer. 2006.
- [7] Jambalaya, URL: <http://www.thechiselgroup.org/jambalaya>.
- [8] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Gainnopolou. *Ontology visualization methods - a survey*, *ACM Computing Surveys (CSUR)*, Volume 39 , Issue 4. 2007.
- [9] A. Katifori, E. Torou, C. Halatsis, G. Lepouras, C. Vassilakis. *A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results*, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01648294> . 2006.
- [10] R. Mantegazzini. *Semantic DB*, *Advanced Computer Science Master Thesis*, SUPSI DTI. 2008.
- [11] *OntoViz* URL: <http://protegewiki.stanford.edu/index.php/OntoViz>
- [12] *OWLviz* URL: <http://www.co-ode.org/downloads/owlviz/>
- [13] *Protégé* URL : <http://protege.stanford.edu/>
- [14] *RDF Gravity* URL: <http://semweb.salzburgresearch.at/apps/rdf-gravity/>

[15] K. Serguei, V. Ferdinando, W. Richard and W. Xindong. ON VISUALIZATION OF OWL ONTOLOGIES, Chapter 10 of Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, Christopher J. O. Baker and Kei-Hoi Cheung, Springer US. 2007.

[16] TGVizTab Protégé plug-in URL:
<http://users.ecs.soton.ac.uk/ha/TGVizTab/>