Towards a Visual Notation for OWL: A Short Summary of VOWL

Steffen Lohmann[1], Florian Haag[1], and Stefan Negru[2]

[1]Institute for Visualization and Interactive Systems (VIS),
University of Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany
steffen.lohmann@vis.uni-stuttgart.de
florian.haag@vis.uni-stuttgart.de
[2]Faculty of Computer Science, Alexandru Ioan Cuza University
Strada General Henri Mathias Berthelot 16, 700483 Iasi, Romania
stefan.negru@info.uaic.ro

Abstract. The Web Ontology Language (OWL) is not inherently visual in contrast to many related modeling languages. However, the visual representation of individual OWL elements and complete OWL ontologies can be very useful in many cases. We have therefore developed the Visual Notation for OWL (VOWL) that defines graphical representations for most of the OWL language constructs. In contrast to related work, we aim for a complete and well-specified notation that is easy to understand and implement. At OWLED 2015, we would like to present the current state of development and discuss our design principles and considerations with the community. We would like to gather feedback on how to further improve the visual notation and collect requirements for its future development.

1 Introduction

The Web Ontology Language (OWL) has become the 'lingua franca' for ontologies. Nearly all modern ontologies are modeled in OWL, and more and more OWL ontologies are developed every week. However, OWL is not a visual language. In contrast to related modeling languages like UML or ER diagrams, which are inherently visual, the OWL specifications do not include any recommendations on how to graphically represent the different OWL language constructs.

Yet, a visualization of individual and combined OWL elements as well as complete ontologies can be very useful in many situations. Among others, it can help in the development, exploration, verification, and sensemaking of ontologies. It can also be useful for teaching OWL, in order to illustrate the language constructs and possible combinations. Finally, visualizations of OWL are known to ease the communication between domain experts and ontology experts.

Although many visualizations for OWL ontologies have been developed in the past, only few define an explicit notation. Most visualizations use basic node-link diagrams or other visualization techniques to depict the concepts and relationships modeled in OWL [23,31], but neither provide any further description of the notation nor a clear visual mapping for the individual OWL elements.

We have developed the Visual Notation for OWL (VOWL) in an effort to close this gap and complement OWL with a well-specified visual notation that defines graphical representations for most language constructs. Although we focused the development on an easy-to-understand notation for casual ontology users, VOWL can also be of use to ontology experts. A precise description of the visual notation can be found in the VOWL specification [35], while details on its implementation and evaluation are given in related publications [30,31]. Recently, we started work on version 3 of VOWL with the goals to incorporate the complete set of OWL language constructs and to further increase the scalability of the visual notation.

In this paper, we summarize the main design principles and considerations related to the development of VOWL, and give an outlook on future directions. We would like to discuss the current state of development at OWLED 2015 and collect feedback and requirements on how to further improve the visual notation.

2 Related Work

Many attempts to visualize OWL ontologies have been presented in the last decade. Surveys can be found in [7,23,28], among others. Most of the approaches visualize OWL ontologies as graphs, which reflects well the way concepts and relationships are organized in OWL [31]. The graphs are typically rendered in a force-directed, hierarchical, or radial layout, often resulting in appealing visualizations. There are also 3D graph visualizations for OWL ontologies [3,16] as well as approaches that visualize the ontologies as hyperbolic trees [8,12].

However, few visualizations show all constructs modeled in OWL, but most approaches focus on certain aspects of ontologies [31]. While some visualize only the class hierarchy of OWL ontologies [19,29,32], others consider different types of properties [10,11,40], but do not include property characteristics and other information required to fully understand the information modeled in OWL. Only a small number of works provide comprehensive visualizations for OWL ontologies. Unfortunately, the different ontology elements are partly hard to distinguish in these visualizations. For instance, the tools TGViz [1] and NavigOWL [22] use plain node-link diagrams where all nodes and links look the same except for their color.

SOVA [2] and GrOWL [27] define more elaborate notations using different symbols, colors, and node shapes. However, the resulting visualizations are still rather hard to read due to many crossing edges and only minor variations in the visual elements. In addition, both notations use abbreviations and mathematical symbols that make them less intuitive for casual ontology users [31].

There is also a bunch of research on reusing UML class diagrams for the visualization of OWL ontologies [4,6,25]. Precise mappings between OWL and UML class diagrams are, for instance, specified in the Ontology Definition Metamodel (ODM) [36]. However, UML has originally not been designed for the representation of OWL, which results in some conceptual limitations and incompatibilities [18,25,26]. It can therefore be irritating to illustrate OWL language constructs with UML, especially in teaching and training contexts. Moreover, people not familiar with UML have difficulties interpreting the diagrams correctly, as we found in a comparative user study [33].

A UML-related type of diagram for the visualization of OWL ontologies is used in OntoViz [38]. It groups classes and datatypes in boxes that are linked by different properties. Another proposal has been made with VisioOWL [13], which has been implemented as a template for the diagram editor Microsoft Visio. However, both types of diagrams share the limitation of UML class diagrams that the resulting visualizations are rather difficult to read for casual ontology users. Futhermore, the visual distinction of the various OWL elements is again limited, as similar shapes and colors are used for different OWL elements.

A related attempt has been made with Concept Diagrams [21] that consider the logic of OWL in particular. Concept Diagrams aim for a formal representation of ontologies that precisely expresses the OWL semantics. However, they do not propose intuitive OWL visualizations that are also understandable to casual users.

This is different in Graffoo [9], which aims at an easy-to-understand notation for OWL and is therefore closely related to the idea of VOWL. It comes with a comprehensive specification [37] and has been implemented as a GraphML extension for the diagram editor yEd. However, Graffoo is rather related to the idea of UML-based modeling by being based on visual elements commonly known from diagram editors. This makes it less scalable than VOWL when it comes to the visualization of OWL ontologies with many classes, properties, and instances. In addition, and similar to the previous attempts, Graffoo defines a rather general notation that uses similar visual elements for different OWL language constructs. Despite these limitations, Graffoo currently seems to be the most promising alternative to UML, Concept Diagrams, and VOWL when a visual notation for OWL is needed.

3 Summary of VOWL

In general, we designed VOWL as a visual notation that is easily understandable to lay users and that supports the communication between domain experts and ontology experts. We did not attempt to create a direct visual mapping of the OWL *syntax*, but rather wanted to create a visual mapping of the *semantics* of the individual OWL elements.

3.1 General Design Principles

For the design of VOWL, we aimed for meeting several design principles that were inspired by the dialog principles as defined in ISO 9241-110 [14]:

- VOWL was designed to be **self-descriptive** by featuring textual descriptions where necessary. At best, no additional legend is required to understand the notation.
- Established patterns were respected by using commonly used shapes and colors (such as arrow heads or gray elements to indicate deprecation), borrowing some visual aspects from notations such as UML. This helps to keep VOWL **conformant with user expectations**.
- We put a focus on the comprehensive specification of the VOWL notation. This complete and unambiguous specification, combined with the self-descriptiveness and conformance with user expectations, makes VOWL **suitable for learning**.

– VOWL is **suitable for its task** of visually depiction OWL concepts by defining a graphical representation for most OWL language constructs.
– VOWL has been defined in a modular way by making certain labels and colors optional or customizable. Furthermore, we took care to make it work in different interaction contexts, by considering both touch and mouse input for example. The VOWL visualizations scale well with different screen sizes, and can be printed on paper in monochrome, without losing any important information. All of these properties make VOWL **suitable for individualization**.

This way, all of the dialog principles defined in ISO 9241-110 are met except for **controllability** and **error tolerance**, which do not apply to the visual notation itself but rather to its implementation in corresponding tools. The design principles were empirically validated in user studies with different user groups [31,33] and benchmarks, both focusing the notation itself and its interactive implementations (cf. Section 4). Besides these primary design principles, we were pursuing some secondary objectives, such as fostering aesthetically pleasing visualizations and a comparatively easy implementation.

3.2 Concrete Design Decisions

The first design decision for VOWL was the basic representation of the OWL structure. The network of classes that can be related by inheritance relationships and properties lends itself to being displayed as a graph. In particular, a node-link-based representation was chosen, as this graph visualization well supports the tasks that could be relevant in the context of ontology visualization, such as finding an indirect connection between two classes or spotting highly connected classes [24].

While indented trees might be more suitable for showing mere hierarchies without multiple inheritance or additional connections by properties, Fu et al. found that node-link visualizations are perceived as “more controllable and intuitive without visual redundancy, especially for ontologies with multiple inheritance” [15]. They are considered particularly “suitable for overviews” and “held [the] attention” better than trees in the user study Fu et al. conducted [15]. However, it needs to be kept in mind that neither indented trees nor node-link diagrams scale particularly well for very large ontologies.

With a node-link visualization, each graph element can be represented exactly once. We decided against this option, by merging some groups of elements in VOWL that conceptually represent units, such as sets of equivalent classes. Likewise, other elements, such as datatypes and *owl:Thing*, are represented several times in the VOWL visualizations. This has the advantage that abstract elements connected with many other ontology elements are prevented from taking central positions in the node-link visualization. At the same time, it allows for shorter edges and fewer edge crossings, both of which enhance the readability of the visualization. As we found in an evaluation on VOWL, users are able to correctly interpret this multiplication [31].

The aggregation and multiplication of certain elements in the visualization reflects why VOWL is not a direct mapping of the OWL syntax, but of the concepts found in OWL ontologies. For example, *disjoint union* constructs are disassembled into their

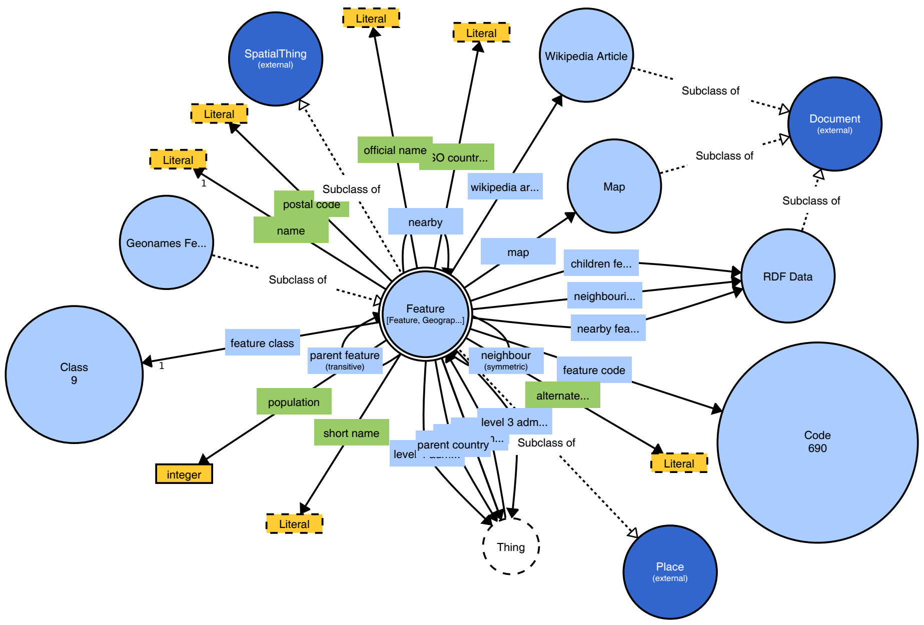

Fig. 1. A part of the Geonames ontology visualized with VOWL.

atomic parts in VOWL, i.e., a union class and pairwise disjoint restrictions between the participants of the union.

Two basic shapes, circles and rectangles, were chosen for nodes in VOWL. Class nodes often feature high degrees of connectivity, which is supported by the circle shape that allows higher numbers of inbound arrows to properly align around the circle without overlapping arrowheads. At the same time, datatype nodes that are usually merely connected to one edge have a rectangular shape. Property labels are shown in a rectangle, as well, but in contrast to the aforementioned shapes, these rectangles have no frame. Like this, the described elements can be clearly distinguished even in monochrome renditions.

The aforementioned arrowheads are found in inheritance relationships, in properties, and in type relationships. VOWL inheritance edges are reminiscent of UML inheritance and implementation edges, which helps users with the respective prior knowledge [31]. Properties defined in the OWL ontology are shown as edges with a label and an arrowhead pointing from the domain to the range. Rather than defining the property as an independent node with two edges to indicate the domain and range, we opted for this graphically simpler visualization as it was preferred by users [33]. Finally, type relationships look similar to properties, but have a specific color and label and only appear in the case that a class is an individual of another class.

In general, the size of class nodes can be scaled VOWL according to the number of individuals in the respective class. It must be pointed out that the exact area covered by circles is difficult to determine for users. Also, a linear mapping between the number of individuals and the radius may not be helpful, anyway, if the numbers of individuals differ a lot between classes. However, the scaling of class nodes is mainly meant to provide an approximate sense of which classes in an ontology contain slightly or significantly more or less individuals than others at a glance. The exact number of individuals can additionally be displayed inside the class node as text. In this number, individuals are counted for each class they are a member of.

Various parts of the visualization were designed to be slightly redundant. For instance,subclass relations are unique in appearance, but also carry a textual label. Likewise, deprecated and imported elements have a unique color, but also have a descriptive text pointing out their special status. This was done to both improve self-descriptiveness and to minimize the amount of information lost when some features of the visualization are not available. For instance, deprecated classes are still recognizable with impaired color vision or when printed as a monochrome depiction due to the descriptive text. Thus, while colors in VOWL help to make elements immediately distinguishable, their absence does not imply the loss of crucial information.

Figure 1 shows a part of version 3.1 of the Geonames ontology [39] visualized with VOWL. The visualization has been generated with the web application described below (cf. Section 4).

3.3 Limitations

Although the graphical representations of the elements described above have many advantages, a few shortcomings still pose open questions in the further development of VOWL. For instance, several reasons beside mere aesthetics support the decision for circular classes, but as labels are often wider than tall, much empty space in class nodes remains. Unless the empty space is used for extra information, the class name can be wrapped, which graphics toolkits are often not able to accomplish within the non-rectangular bounds of a circle. Likewise, IRIs are currently not displayed anywhere in the visualization, as VOWL is directed towards lay users. Those users often do not want to see IRIs, and it is trivial to display them as additional information on the selected element next to the visualization.

More generally, VOWL focuses on the visualization of the TBox of small to medium-size ontologies but does not sufficiently support the visualization of very large ontologies and detailed ABox information for the time being. First attempts to handle larger ontologies by gradually hiding nodes with low degrees of connectivity have been tentatively integrated in one of our implementations (see below). Future analyses of user needs and suggestions will be required to determine how to integrate more ABox information into VOWL. In particular, this issue must be tackled with respect to ontologies where certain key individuals are as generally used and important as the classes and properties defined in the ontology.

4 Implementations of VOWL

VOWL has been implemented in two different tools that demonstrate its applicability: a plugin for the ontology editor Protégé and a responsive web application. Both tools are released under the MIT license and are publicly available at `http://vowl.visualdataweb.org`. The OWL ontologies are rendered in a force-directed graph layout according to the VOWL specification. Interaction techniques allow to explore the ontologies and customize their visualizations. While the Protégé plugin does not include all visual elements defined in VOWL, the web application provides a complete implementation of the current version of VOWL (which is VOWL 2 [35]).

The web application allows users to upload custom ontologies and to interactively explore and adapt the generated VOWL visualizations [30]. It is complemented by a Java-based converter that transforms the OWL ontologies into the required JSON format. The converter parses the ontology representation using the OWL API [20] and outputs a JSON file that is read by the web application. The schema of the JSON file has been designed with regard to VOWL, i.e., its structure differs from common OWL serializations in order to enable an efficient generation of the graph visualization and to ease access at runtime. The web application is easy to use and understand and therefore also appropriate for casual ontology users, as we could confirm in a user study [31].

Ongoing activities related to the implementation of VOWL concern its integration into WebProtégé[1] as well as the development of a Visio template for VOWL [5]. We have also developed a visual query language based on VOWL that addresses the peculiarities of querying Linked Data with SPARQL [17]. Some VOWL elements had to be adapted for this purpose to indicate the variability of the IRIs or values they represent, and to provide for the interactive options that users require to specify their query. For instance, visual elements can act as placeholders in the query language that are not fully specified on a TBox level and for which restrictions can be added by the user. More details on the query language and a prototypical implementation of it are also available at `http://vowl.visualdataweb.org`.

5 Conclusions

In this summary paper, we outlined the main design principles and considerations related to the development of a visual notation for OWL. The interested reader is referred to our papers that detail the development of VOWL [34,31].

The presented VOWL notation apparently provides only one possible way to visualize OWL ontologies using node-link diagrams. As OWL is not an inherently visual language, other types of visualizations would also be possible and could be more appropriate in certain cases. For instance, if users are mainly interested in the class hierarchy contained in an OWL ontology, they might prefer a visualization that uses an intended tree or treemap to depict the ontology.

We believe that VOWL is already a comparatively mature proposal to further discuss the visual representation of OWL. However, although version 2 of VOWL already considers a large portion of the OWL language constructs, it is not yet complete, in particular with regard to OWL 2. Our ultimate goal is to turn VOWL into a visual notation that can represent OWL ontologies as completely as possible. Therefore, we recently started work on version 3 of VOWL, with the goal to incorporate additional OWL language constructs.

[1] see `https://github.com/VisualDataWeb/webprotege`

In addition, we plan to further improve the visual notation, in particular, with regard to the visualization of large OWL ontologies. The current notation does not scale well for large ontologies, calling for a more compact notation that copes for those cases. Alternatives would be to filter certain elements of the notation or to display only parts of the OWL ontology or abstractions of it. While the WebVOWL implementation already provides some functionality to filter the OWL visualization, this is not yet systematically incorporated into the VOWL specification.

The visual representation of very large ontologies is one of the issues we would like to discuss at OWLED 2015. Other issues concern the stability of the visualization and possible alternatives to a force-directed graph layout. For instance, we are currently elaborating whether a hierarchical or radial graph layout could also be used together with VOWL. Moreover, we are thinking about ways to integrate further information about individuals defined in OWL ontologies.

At the OWLED workshop, we would like to discuss these and other issues and challenges related to the development of a visual notation for OWL. We would like to gather feedback on how to further improve VOWL in order to make it even more useful to the community of ontology users.

Acknowledgements

The authors would like to thank David Bold, Vincent Link, and Eduard Marbach for their contributions to the implementation of VOWL and their valuable feedback on the notation.

References

1. H. Alani. TGVizTab: An ontology visualisation extension for Protégé. In *2nd Workshop on Visualizing Information in Knowledge Engineering (VIKE '04)*, 2003.
2. T. Boinski, A. Jaworska, R. Kleczkowski, and P. Kunowski. Ontology visualization. In *2nd International Conference on Information Technology (ICIT '10)*, pages 17–20. IEEE, 2010.
3. A. Bosca, D. Bonino, and P. Pellegrino. OntoSphere: more than a 3D ontology visualization tool. In *2nd Italian Semantic Web Workshop (SWAP '05)*, volume 166 of *CEUR-WS*, 2005.
4. S. Brockmans, R. Volz, A. Eberhart, and P. Löffler. Visual modeling of OWL DL ontologies using UML. In *3rd International Semantic Web Conference (ISWC '04)*, volume 3298 of *LNCS*, pages 198–213. Springer, 2004.
5. T. Chungoora. Visio template for VOWL. `http://ontoweave.com/articles/visio-template-for-vowl/`, 2015.
6. D. Djurić, D. Gašević, V. Devedžić, and V. Damjanović. A UML profile for OWL ontologies. In *European Conference on Model Driven Architecture: Foundations and Applications (MDAFA '03)*, volume 3599 of *LNCS*, pages 204–219. Springer, 2005.
7. M. Dudáš, O. Zamazal, and V. Svátek. Roadmapping and navigating in the ontology visualization landscape. In *19th International Conference on Knowledge Engineering and Knowledge Management (EKAW '14)*, volume 8876 of *LNAI*, pages 137–152. Springer, 2014.
8. P. Eklund, N. Roberts, and S. Green. OntoRama: Browsing RDF ontologies using a hyperbolic-style browser. In *1st International Symposium on Cyber Worlds (CW '02)*, pages 405–411. IEEE, 2002.

9. R. Falco, A. Gangemi, S. Peroni, D. Shotton, and F. Vitali. Modelling OWL ontologies with graffoo. In *ESWC 2014 Satellite Events*, volume 8798 of *LNCS*, pages 320–325. Springer, 2014.
10. S. Falconer. OntoGraf. `http://protegewiki.stanford.edu/wiki/OntoGraf`, 2010.
11. S. Falconer, C. Callendar, and M.-A. Storey. A visualization service for the semantic web. In *17th International Conference on Knowledge Engineering and Knowledge Management (EKAW '10)*, volume 6317 of *LNCS*, pages 554–564. Springer, 2010.
12. D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: The very high idea. In *11th International Florida Artificial Intelligence Research Society Conference (FLAIRS '98)*, pages 131–135. AAAI Press, 1998.
13. J. Flynn. VisioOWL. `http://www.semwebcentral.org/projects/visioowl/`, 2012.
14. I. O. for Standardization. *ISO 9241-110: Ergonomics of Human-system Interaction-Pt. 110: Dialogue Principles*. ISO, 2006.
15. B. Fu, N. F. Noy, and M.-A. Storey. Indented tree or graph? a usability study of ontology visualization techniques in the context of class mapping evaluation. In *12th International Semantic Web Conference (ISWC '13), Part I*, volume 8218 of *LNCS*, pages 117–134. Springer, 2008.
16. S. S. Guo and C. W. Chan. A tool for ontology visualizaiton in 3D graphics: Onto3DViz. In *23rd Canadian Conference on Electrical and Computer Engineering (CCECE '10)*, pages 1–4. IEEE, 2010.
17. F. Haag, S. Lohmann, S. Siek, and T. Ertl. Visual querying of linked data with QueryVOWL. In *3rd Workshop on Human-Semantic Web Interaction*, CEUR-WS, to appear.
18. L. Hart, P. Emery, B. Colomb, K. Raymond, S. Taraporewalla, D. Chang, Y. Ye, E. Kendall, and M. Dutra. OWL full and UML 2.0 compared. Technical report, OMG, 2004.
19. M. Horridge. OWLViz. `http://protegewiki.stanford.edu/wiki/OWLViz`, 2010.
20. M. Horridge and S. Bechhofer. The OWL API: A java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.
21. J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: A case study. In *10th International Semantic Web Conference (ISWC '11), Part I*, volume 7031 of *LNCS*, pages 257–272. Springer, 2011.
22. A. Hussain, K. Latif, A. Rextin, A. Hayat, and M. Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355–367, 2014.
23. A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods – a survey. *ACM Computer Surveys*, 39(4), 2007.
24. R. Keller, C. M. Eckert, and P. J. Clarkson. Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models? *Information Visualization*, 5(1):62–76, 2006.
25. E. Kendall, R. Bell, R. Burkhart, M. Dutra, and E. Wallace. Towards a graphical notation for OWL 2. In *6th International Workshop on OWL: Experiences and Directions (OWLED '09)*, volume 529 of *CEUR-WS*, 2009.
26. K. Kiko and C. Atkinson. A detailed comparison of UML and OWL. Technical Report TR-2008-004, University of Mannheim, 2005.
27. S. Krivov, R. Williams, and F. Villa. GrOWL: A tool for visualization and editing of OWL ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):54–57, 2007.

28. M. Lanzenberger, J. Sampson, and M. Rester. Visualization in ontology tools. In *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '09)*, pages 705–711. IEEE, 2009.
29. T. Liebig and O. Noppens. OntoTrack: A semantic approach for ontology authoring. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):116–131, 2005.
30. S. Lohmann, V. Link, E. Marbach, and S. Negru. WebVOWL: Web-based visualization of ontologies. In *EKAW 2014 Satellite Events*, volume 8982 of *LNAI*, pages 154–158. Springer, 2015.
31. S. Lohmann, S. Negru, F. Haag, and T. Ertl. VOWL 2: User-oriented visualization of ontologies. In *19th International Conference on Knowledge Engineering and Knowledge Management (EKAW '14)*, volume 8876 of *LNAI*, pages 266–281. Springer, 2014.
32. E. Motta, P. Mulholland, S. Peroni, M. d'Aquin, J. M. Gomez-Perez, V. Mendez, and F. Zablith. A novel approach to visualizing and navigating ontologies. In *10th International Semantic Web Conference (ISWC '11), Part I*, volume 7031 of *LNCS*, pages 470–486. Springer, 2011.
33. S. Negru, F. Haag, and S. Lohmann. Towards a unified visual notation for OWL ontologies: Insights from a comparative user study. In *9th International Conference on Semantic Systems (I-SEMANTICS '13)*, pages 73–80. ACM, 2013.
34. S. Negru and S. Lohmann. A visual notation for the integrated representation of OWL ontologies. In *9th International Conference on Web Information Systems and Technologies (WEBIST '13)*, pages 308–315. SciTePress, 2013.
35. S. Negru, S. Lohmann, and F. Haag. VOWL: Visual notation for OWL ontologies. `http://purl.org/vowl/`, 2014.
36. OMG. Ontology Definition Metamodel, Version 1.1. `http://www.omg.org/spec/ODM/1.1/`, 2014.
37. S. Peroni. Graffoo specification. `http://www.essepuntato.it/graffoo/specification/current.html`, 2013.
38. M. Sintek. OntoViz. `http://protegewiki.stanford.edu/wiki/OntoViz`, 2007.
39. B. Vatant. GeoNames Ontology. `http://www.geonames.org/ontology/`, 2012.
40. L. Wachsmann. OWLPropViz. `http://protegewiki.stanford.edu/wiki/OWLPropViz`, 2008.