

VOWL 2: User-Oriented Visualization of Ontologies

Steffen Lohmann¹, Stefan Negru², Florian Haag¹, and Thomas Ertl¹

¹Institute for Visualization and Interactive Systems (VIS),
University of Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany
{steffen.lohmann, florian.haag, thomas.ertl}@vis.uni-stuttgart.de

²Faculty of Computer Science, Alexandru Ioan Cuza University,
Strada General Henri Mathias Berthelot 16, 700483 Iasi, Romania**
stefan.negru@info.uaic.ro

Abstract. Ontologies become increasingly important as a means to structure and organize information. This requires methods and tools that enable not only ontology experts but also other user groups to work with ontologies and related data. We have developed VOWL, a comprehensive and well-specified visual language for the user-oriented representation of ontologies, and conducted a comparative study on an initial version of VOWL. Based upon results from that study, as well as an extensive review of other ontology visualizations, we have reworked many parts of VOWL. In this paper, we present the new version VOWL 2 and describe how the initial definitions were used to systematically redefine the visual notation. Besides the novelties of the visual language, which is based on a well-defined set of graphical primitives and an abstract color scheme, we briefly describe two implementations of VOWL 2. To gather some insight into the user experience with the new version of VOWL, we have conducted a qualitative user study. We report on the study and its results, which confirmed that not only the general ideas of VOWL but also most of our enhancements for VOWL 2 can be well understood by casual ontology users.

Keywords: Ontology, visualization, owl, vowl, visual language, semantic web, protégé, prefuse, d3, svg, user study.

1 Introduction

Ontologies describe the concepts and relationships in an area of knowledge using a logic-based language that enables automated reasoning. They are no longer exclusively used by ontology experts but also by non-expert users in various domains. However, especially these casual users often have difficulties to understand ontologies.

Ontology visualizations can help in this regard by assisting in the exploration, verification, and sensemaking of ontologies [15,26]. They can be particularly useful for casual users, but may also give expert users a new perspective on ontologies. While several ontology visualizations have been developed in the last couple of years, they either focus on specific aspects of ontologies or are hard to read for casual users. Furthermore, many visualizations are tailored for a specific task or use special types of diagrams that must first be learned to understand the visualization.

** Stefan Negru is now with MSD IT Global Innovation Center.

In order to fill this gap and provide a more intuitive and user-oriented visualization for ontologies, we developed the Visual Notation for OWL Ontologies (VOWL). An early version of VOWL [34] has been compared to the UML-based visualization of ontologies [33]. Based on insights from the comparison, we have completely reworked the VOWL notation and developed version 2 with significant improvements and more precise mappings to OWL. One of the main goals of VOWL 2 is to define a visual language that can also be understood by casual users with only little training. In this paper, we present the considerations and concepts of VOWL 2 in detail. Two implementations of VOWL 2 demonstrate its applicability and usability. In addition, we report on a user study that compares VOWL 2 to the two related ontology visualizations GrOWL [24] and SOVA [25].

2 Related Work

Quite a number of visualizations for ontologies have been presented in the last couple of years [10,15,23]. While some of them are implemented as standalone applications, most are provided as plugins for ontology editors like Protégé [3].

2.1 Graph Visualizations of Ontologies

Many approaches visualize ontologies as graphs, which is a natural way to depict the structure of the concepts and relationships in a domain of knowledge. The graphs are often rendered in force-directed or hierarchical layouts, resulting in appealing visualizations. However, only few visualizations show complete ontologies, but most focus on certain aspects. For instance, OWLViz [21], OntoTrack [27], and KC-Viz [31] depict only the class hierarchy of ontologies. OWLPropViz [37], OntoGraf [11], and FlexViz [12] represent different types of property relations, but do not show datatype properties and property characteristics required to fully understand ontologies.

A smaller number of approaches provide more comprehensive graph visualizations that represent all key elements of ontologies. Unfortunately, the different ontology elements are often hard to distinguish in the visualizations. For instance, TGViz [5] and NavigOWL [22] use very simple graph visualizations where all nodes and links look the same except for their color. This is different in GrOWL [24] and SOVA [25], which define more elaborated notations using different symbols, colors, and node shapes. However, as the notations of both GrOWL and SOVA rely symbols from description logic [6] and abbreviations, they are not perfectly suited for casual users. Furthermore, the visualizations created with GrOWL and SOVA are characterized by a large number of crossing edges which has a negative impact on the readability.

Other graph visualizations focus on specific tasks. The RelFinder [17], for instance, visualizes relationships between individuals described by ontologies and makes these relationships interactively explorable. GLOW uses a radial tree layout and hierarchical edge bundles [19] to depict relationships within ontologies [20]. Both approaches provide some insight into links between certain classes and individuals, but they do not give an overview of the complete ontology.

2.2 Ontology Visualizations based on Specific Diagram Types

There are also a number of works that use other types of diagrams than graph visualizations to represent ontologies. For instance, Jambalaya [36] and OWL-VisMod [14] use treemaps to depict the class hierarchy of ontologies. Jambalaya additionally provides a nested graph visualization called SHriMP that allows to split up the class hierarchy into different views [36]. CropCircles is a related visualization technique that visualizes the class hierarchy of ontologies with the goal to support the identification of “undermodeled” ontology parts [38]. All these approaches visualize once again mainly the class hierarchy, without considering other property relations.

Cluster Maps use a visualization technique that is based on nested circles and has also been successfully applied to ontologies [13]. Instead of showing the class hierarchy, Cluster Maps visualize individuals grouped by the classes they are instances of. Similar techniques are used in VisCover [28] and OOBIAN Insight [2] that additionally provide a number of interactive filtering capabilities. Another related approach is gFacet [18], where individuals are grouped by their classes and can be filtered by selecting linked individuals or data values. While using appealing visualizations, these tools show only a selection of classes along with their instances but do not provide complete visualizations of ontologies.

A powerful type of diagram related to OWL and often reused to visualize ontologies is the class diagram of the Unified Modeling Language (UML) [4]. Precise mappings between OWL and UML class diagrams are specified in the Ontology Definition Metamodel (ODM) [1], among others. A major drawback of such attempts is that they require some knowledge about UML class diagrams. Although many people with an IT background are familiar with these types of diagrams, people from other domains have difficulties interpreting them correctly, as we also found in the aforementioned comparative study [33].

2.3 Discussion of Related Work

Looking at the related work, some common characteristics stand out: Most visualizations utilize a well-known type of diagram for ontology visualization (graph visualization, treemap, UML) and focus on specific aspects of ontologies. Only few attempts aim for a comprehensive ontology visualization. Even less approaches provide an explicit description of the visual notation, i.e. a specification that precisely defines the semantics and mappings of the graphical elements. Often, there is no clear visual distinction between different property types or between classes, properties, and individuals.

Furthermore, many works implement a stepwise approach of ontology exploration, where only a root class is shown at the beginning and the user has to navigate through the visualization. With VOWL, we rather aim for an approach that provides users with an overview of the complete ontology and let them subsequently explore parts of it in depth, following the popular Visual Information Seeking Mantra of “overview first, zoom and filter, then details-on-demand” [35]. We chose this approach as we consider it important to give users a visual impression of the size and topology of the ontology before they start to explore it any further.

Most importantly, we aim for an intuitive visualization that is also comprehensible to users less familiar with ontologies, while most of the related work has rather been designed for ontology experts.

3 VOWL 2: Visual Notation for OWL Ontologies

Based upon our review of related work and the comparative evaluation [33], we decided to retain numerous traits of the initial VOWL version (VOWL 1). As already mentioned, graphs seem to be a natural and intuitive way to represent the structure of ontologies, which is confirmed by many of the related work reported above. VOWL is based on a mapping of OWL elements to graphical depictions that are combined into a graph representing the ontology. For VOWL 2, we have reworked these mappings and taken into account the exact semantics of all definitions from OWL that were considered. In particular, we have broken down the components of VOWL to a set of basic building blocks consisting of shapes and colors that express specific aspects of the OWL elements (datatype or object properties, different class and property characteristics, etc.), also considering possible combinations thereof.

VOWL 1 included an integrated view that would display the TBox of an ontology along with information from the ABox. Comments from the initial user study on VOWL 1, however, led us to conclude that concerns about the scalability of the integrated view were justified. Even with few instances per class, additional information, such as property values of instances, would be difficult to show without creating lots of clutter. Therefore, VOWL 2 focuses on displaying primarily the TBox and only optionally integrates some ABox information in the visualization itself, but rather recommends to display this information in another part of the user interface.

3.1 Basic Building Blocks of VOWL

The basic building blocks of VOWL 2 are a clearly defined set of graphical primitives and a color scheme. In addition, VOWL 2 uses a force-directed graph layout along with splitting rules that specify which elements are multiplied in the visualization.

Graphical Primitives VOWL 1 defined graphical representations for a number of OWL concepts. For VOWL 2, we took into account those definitions, but based the visualization upon a more abstract and systematic approach. The alphabet of the visual language is now formed by only a handful of graphical primitives and features. Table 1(a) lists these primitives and the ontology elements they are applied to.














As the visualization of individual instances was not considered crucial for most contexts, classes are simply depicted as a circle in VOWL 2, without any additional elements to accommodate instance information. Where available and desired, the number of instances may still be implied by modifying the radius of the circle compared to the default radius. VOWL 2 does not specify a particular scaling method for the circle sizes, but good results may be achieved with a logarithmic or square-root scaling in most cases. The class representation of `owl:Thing` has a fixed size, as it usually does

not carry any particular domain information and as all individuals in an ontology are instances of `owl:Thing` according to the OWL specification.

Like in VOWL 1, property relations are expressed by labeled arrows. The labels do not have individual arrowheads any more, as the user study revealed that their intended direction was often ambiguous, particularly for vertical edges. Other edges, like those for subproperty relations, are completely left out and replaced by additional texts in the labels or interactive features to reduce the number of edge crossings and to facilitate the implementation of VOWL.

As the use of description logic and other symbols to express concepts such as the union or the intersection of classes can be unnecessarily confusing to lay users, VOWL 2 avoids such symbols or combines them with graphical representations reminiscent of Venn diagrams to more clearly communicate the underlying set operations.

Table 1. Graphical primitives and color scheme forming the basic building blocks of VOWL.

(a) Graphical primitives		(b) Excerpt of the VOWL color scheme		
Primitive	Application	Name	Color	Application
	classes	General		classes, object properties, disjoints
	properties	Rdf		elements of RDF and RDF Schema
	property directions	External		external classes and properties
	datatypes, property labels	Deprecated		deprecated classes and properties
	special classes and properties	Datatype		datatypes, literals
	labels, cardinalities	Datatype property		datatype properties
		Highlighting		highlighted elements

Color Scheme Since the results of the user study on VOWL 1 showed that colors are very helpful in identifying the different elements, a stronger focus was placed on creating systematic coloring rules for VOWL 2. We therefore created a color scheme that clearly defines foreground and background colors for all elements based on a number of attributes. While colors in the VOWL 1 specification were statically linked to the visual elements, VOWL 2 defines colors by their function, for example, to mark deprecated or external elements. Where several of the color mappings may apply, priority rules are specified. For instance, the deprecated color has priority over the external color according to the VOWL 2 specification.

VOWL 1 and most related approaches rely either on a concrete set of colors or do not specify colors at all. We acknowledged that a specific color scheme may not always be adequate and therefore defined the VOWL 2 color scheme in a more flexible way by using abstract color names, such as the ones listed in Table 1(b). Although the VOWL 2 specification recommends a particular color mapping, it is meant merely as a default suggestion and any references to colors in the specification can be treated as variables. In addition, the color scheme comes with guidelines on how the colors should relate to each other in order to encode the VOWL semantics.

3.2 Visual Elements and Graph Visualization

The VOWL 2 specification includes graphical representations for the most common OWL elements based on the aforementioned primitives and colors. A selection thereof is shown in Figure 1.¹ In several cases, information is redundantly encoded. For example, equivalent classes both carry a double border as well as several class names in their label, and external classes both adhere to the external color as defined in the color scheme and sport the hint “external” beneath their name. While the colors improve the overview in most situations, the additional texts ensure that even in the absence of colors, the important aspects of the ontology are still intelligible. Moreover, we wanted to make the visualization more self-explanatory with the text representations.

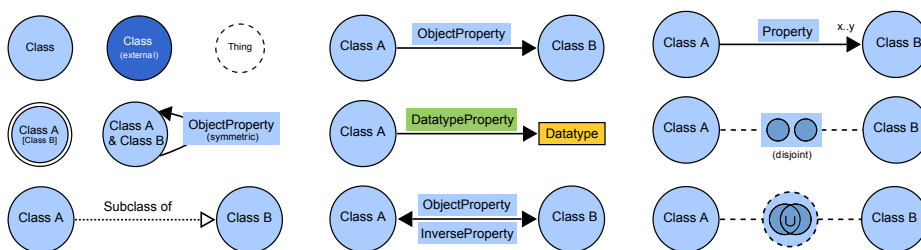


Fig. 1. A selection of visual elements from the VOWL 2 specification.

As participants of the VOWL 1 user study stated the desire for an interactive highlighting of certain information and relationships, some of the visual element definitions in VOWL 2 include guidelines on how to design interactive implementations. For example, while the aforementioned subproperty relation is no longer expressed by an arrow in VOWL 2, hovering over a subproperty is defined to automatically highlight its parent property and vice-versa in interactive contexts.

The visual elements are combined to a graph representing the entire ontology. By default, VOWL graphs are visualized using a force-directed layout. In order to relax the energy of that layout and to reduce the visual importance of certain frequently referred to but generic ontology elements, some elements can be multiplied to appear more than once in a VOWL graph. While this was already the case for `owl:Thing` in VOWL 1, we have systematically defined generic splitting rules in VOWL 2. These rules determine that there may be no multiplication for elements, multiplication across the entire graph, or multiplication for each connected class. The VOWL 2 specification relies on such rules to define the splitting of `owl:Thing` into several representations, as well as that of certain other elements, such as datatypes or literals.

¹ The graphical representations of further OWL elements are defined in the VOWL 2 specification that is publicly available on the web at the persistent URL <http://purl.org/vowl>. Most of the remaining representations are variations of the ones presented in Figure 1 and visualize specific property characteristics (e.g. functional, transitive) or other set operators (e.g. intersection, complement), among others.

Figure 2 shows two visualizations of the main component of the Friend of a Friend (FOAF) vocabulary, one created with VOWL 1 and the other with VOWL 2 [34]. The visualizations contain several of the visual elements described above and exemplify some of the differences between VOWL 1 and VOWL 2. Another important difference of the two visualizations is that the VOWL 1 visualization has been manually created with a vector graphics editor (cf. [34]), while the VOWL 2 visualization has been automatically generated and only slightly adapted using the WebVOWL implementation that is presented in the next section.

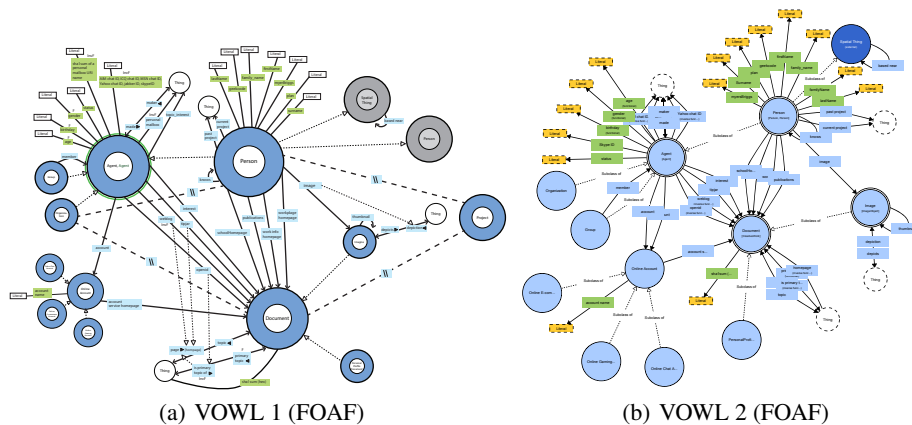


Fig. 2. Friend of a Friend (FOAF) vocabulary visualized with (a) VOWL 1 and (b) VOWL 2.

3.3 Implementations of VOWL 2

We have implemented VOWL 2 in two different tools that demonstrate its applicability and usability: ProtégéVOWL and WebVOWL. ProtégéVOWL is a Java-based plugin for the aforementioned ontology editor Protégé and makes use of the visualization toolkit Prefuse [16].² It implements the VOWL 2 specification to a large extent and uses the data model supplied by the underlying Protégé API.

WebVOWL is a standalone application based on web technologies and the visualization library D3 [7]. Figure 3 shows a screenshot of WebVOWL where it is used to visualize the Personas Ontology [32]. Instead of implementing an OWL parser itself, it defines a JSON structure that the ontology needs to be converted into. This makes WebVOWL independent from a particular OWL parser and broadly applicable. WebVOWL additionally allows to export the entire ontology visualization or any portion of it as SVG image that can be opened in other programs, edited, shared, and printed.

Both implementations use physics simulations provided by the visualization toolkits to generate the force-directed graph layouts. The forces are iteratively applied in these simulations, resulting in an animation that dynamically positions the nodes. The users

² A demo of ProtégéVOWL has been presented at ESWC 2014 [30].

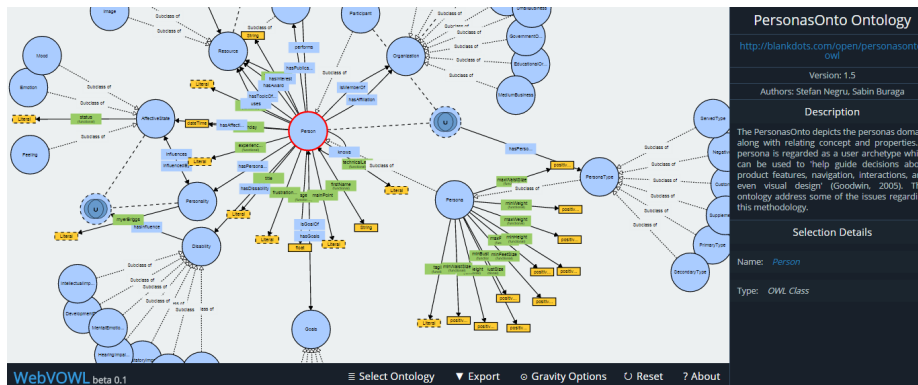


Fig. 3. Screenshot of WebVOWL with a visualization of the Personas Ontology.

can smoothly zoom in to explore certain ontology parts in detail or zoom out to analyze the global structure of the ontology. They can pan the background and move elements around, which results in a repositioning of the nodes by animated transitions, triggered by the force-directed layout. Both implementations support interactive highlighting and display additional information on the selected elements on demand (in Figure 3, the class “Person” is selected). Moreover, the force-directed layout can be adapted, as the attraction forces between nodes can be modified and the automatic layout can be suspended in favor of a manual repositioning of nodes. Since datatypes have a separate attraction force, they can be placed in close proximity to the classes they are connected with, to emphasize their radial arrangement and increase readability.

The user interfaces of both tools consist of three parts (see Figure 3): A *viewer* displaying the VOWL visualization, a *sidebar* listing details about the element that is selected, and the *controls* allowing to adapt the force-directed graph layout and providing further options, such as a function to export the VOWL visualization as SVG image. We developed the tools, in particular WebVOWL, so that they can be used in different interaction contexts, including settings with touch interaction. For instance, zooming can not only be performed with the mouse wheel but also with a double click-/touch or two fingers zooming gestures on the canvas. As some interactive features, such as mouseover effects, may not be available in all interaction contexts (e.g. when using touch interfaces), we took care that they are not crucial for the interaction or for understanding the ontology.³

To the best of our knowledge, WebVOWL is the first tool for comprehensive ontology visualization that is completely based on open web standards. Related tools running in web browsers, such as FlexViz [12] or OOBIAN Insight [2], are implemented with technologies like Adobe Flex or Microsoft Silverlight that require proprietary browser plugins. The tool LodLive [9] is technically more related to WebVOWL but focuses on the visual exploration of Linked Data and does not visualize ontologies.

³ Both tools are released under the MIT license and are publicly available at <http://vowl.visualdataweb.org>.

4 Evaluation

We evaluated how well users could handle VOWL by comparing it to the ontology visualizations SOVA [25] and GrOWL [24]. We chose those two, as they can both be used to gain a general overview of ontologies. In addition, they are based on a systematic mapping between ontology concepts and graphical elements. This makes them most related in purpose and scope to VOWL (cf. Section 2), even though both SOVA and GrOWL include some formal symbols.

We presented three ontologies to the participants of the user study: Questions were asked about the Friend of a Friend (FOAF) vocabulary [8] (version 0.99) as an example of a smaller ontology, and the Personas Ontology [32] (version 1.5) as an example of a more extensive one. In addition, we used the Modular Unified Tagging Ontology (MUTO) [29] (version 1.0) as a small training ontology, to give participants an opportunity to familiarize themselves with each visualization approach and the user interface of the respective implementation.

The FOAF vocabulary visualized with SOVA and GrOWL is shown in Figure 4, the VOWL representation of FOAF was already depicted in Figure 2(b). While both WebVOWL and ProtégéVOWL are, in principle, conformant with the VOWL 2 specification, the implementation of WebVOWL had progressed further at the time of the study, which is why the VOWL visualizations were shown in WebVOWL. The SOVA visualizations were generated with the respective plugin for Protégé [25], while for the GrOWL visualizations, we used the Java tool presented in [24]. The Java tool supports either the creation of a new ontology or the visualization of an existing one in a force-directed layout, which was what we used. Note that the force-directed layouts arrange the graphs differently each time the ontologies are loaded into the tools, as these layouts are inherently non-deterministic.

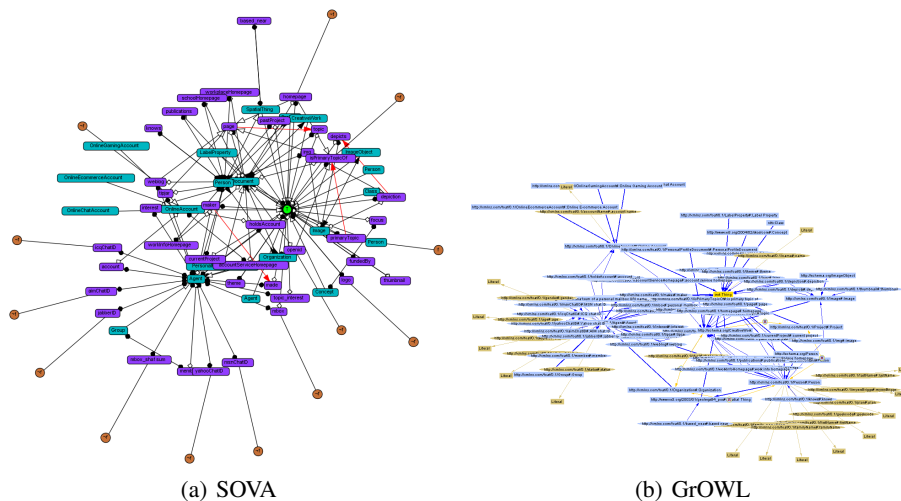


Fig. 4. Friend of a Friend (FOAF) vocabulary visualized with (a) SOVA and (b) GrOWL.

4.1 Tasks

We prepared 18 comprehension tasks, six per visualization, and among those, three for each of the two ontologies (FOAF and Persones Ontology). The tasks reflected questions that users working with ontologies would have to pose in order to identify, use, or edit parts of the ontologies. Some of the tasks featured the same or similar questions, referring to different ontologies. A full list of the tasks is provided in Table 2 at the end of this paper.

The tasks also forced study subjects to analyze the visualizations and recognize relationships between ontology elements, just like actual users of ontologies would have to. While we did check the correctness of the answers, we were primarily interested in the comments made by the participants during task completion. Therefore, we asked participants to adhere to the “think-aloud” method and state everything they were thinking, feeling, considering, or doing with respect to the visualizations.

4.2 Other Material

We also prepared a brief printed introduction to the topic of ontologies. It included a quick overview of the concepts relevant to the study, comprising classes, properties, subclassing, equivalent classes, set operators, and imported ontology elements. Moreover, we printed a table for each of the evaluated visualizations, showing how the concepts are depicted in the given notations and how a very small exemplary ontology that combines some of the elements could look like. Finally, a questionnaire about the experienced ease-of-use was prepared for each visualization, and an additional questionnaire asked for the prior knowledge of the participants.

Implementations of the three visualizations—the SOVA plugin (version 0.8.1) for Protégé, the GrOWL Java application (version 0.02), and WebVOWL (version 0.1) in a Mozilla Firefox browser—were installed and running on the test computer. In the case of SOVA, the plugin version 0.8.1 was the latest available version at the time of conducting the user study. Unfortunately, that version did not support the display of datatype properties, which is why tasks related to datatype properties had to be skipped in SOVA. The three ontologies (MUTO, FOAF, and Persones) were loaded in each of the visualization applications and displayed on a 24” TFT monitor at a resolution of 1920×1200 pixels.

4.3 Participants

We chose to conduct the study with participants who may have to work with ontologies at some point, though not necessarily with formal ontology languages like OWL—in short, casual ontology users. We recruited six researchers from various fields of information technology (*not* including Semantic Web technologies) between 29 and 57 years of age. All participants were roughly familiar with the idea of ontologies and/or had some knowledge in related topics, such as object-oriented class structures, UML class diagrams or ER diagrams for database modeling. Two of them had worked with ontologies in the past, but only to a limited extent and without making use of formal representations.

The tested ontologies were unknown to the study participants. While two participants had already heard about the FOAF project, they had never seen the FOAF ontology before. Another two of the participants had some passing knowledge about the topic of ontology visualization, but none of them had ever used any of the three visualizations evaluated in the study.

4.4 Design and Procedure

The study had a mixed design, so that participants could compare the different visualizations. The presentation of the three visualizations was counterbalanced to avoid order effects. The order of the tasks remained fixed, as some of them would require incremental knowledge. In combination with the counterbalancing of the visualizations, this resulted in a setting where each task had to be solved for each of the visualizations by some of the participants. Likewise, the FOAF ontology was always shown before the Personas Ontology, as the latter has a larger size and is more difficult to grasp.

Participants first had to complete the questionnaire about their prior knowledge before being shown the introduction to ontologies. Meanwhile, it was announced that in the case of any doubt, questions for clarification could be asked at any point before or during the study. After briefly familiarizing themselves with the relevant concepts, participants had to use the three visualizations, one at a time.

For each visualization, participants were first provided with the printed table explaining the notation and had an opportunity to explore the MUTO ontology on screen. When they felt sufficiently familiar with the visualization, they had to perform the three tasks for each of the two ontologies, i.e. for FOAF and the Personas Ontology. The study supervisor would note down any of the participants' statements as well as the solutions they arrived at.

After completing the tasks for all three visualizations, participants received the questionnaire on which they had to rate each visualization by the criteria of clarity, learnability, ease of finding elements, mappings between visual and conceptual elements, use of colors, and use of shapes. Each criterion was accompanied by a short explanation, as given in Figure 5.

4.5 Results

In general, participants could solve most of the tasks correctly (84%). Two participants gave up solving some of the tasks in GrOWL, as the implementation lacked a feature to search for specific elements and the participants could not find these elements by visually scanning the graph. Figure 5 illustrates how participants rated the three visualizations based on the criteria of the questionnaire.

Comments were generally favorable toward VOWL (“properties and classes can be distinguished very well”, “not so much tangle and less edges [compared to the other visualizations]”), which was especially praised for the clear visual distinction of classes, properties, and other ontology elements. Several participants pointed out that classes could easily be recognized due to their different shape and size compared to other visual elements. The text labels indicating the meaning of various VOWL elements were mostly thought of as beneficial; one participant even stated that there was no need to

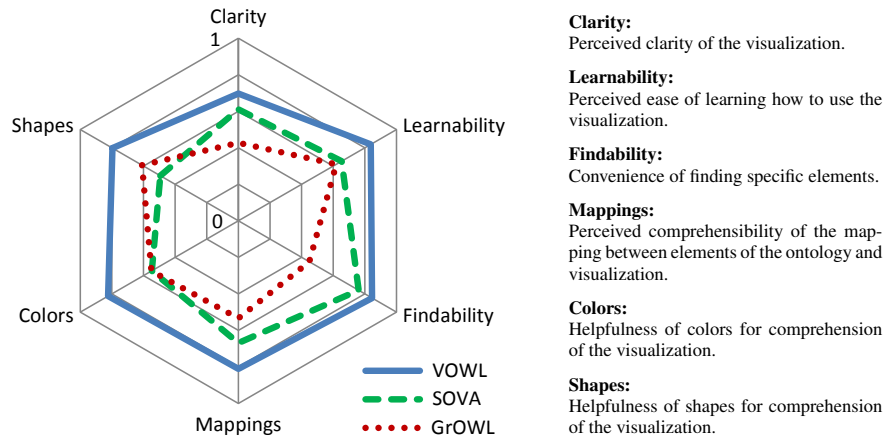


Fig. 5. User ratings of the visualizations based on different criteria.

use the printed table as the VOWL visualization was very self-explanatory. On the other hand, one participant with UML knowledge found some of the labels to be unnecessarily cluttering the visualization, such as the *subclass* text on the UML-inspired class inheritance arrows. Participants considered the animated force-directed layout in all three visualizations to be beneficial, as nodes would to some extent arrange themselves. One participant remarked that circles are a good choice for representing classes in VOWL, as arrow heads of edges pointing to a class would align nicely around its circle with minimal overlapping.

Overall, VOWL 2 was assessed to be well readable and the participants emphasized the comparably low number of edges and, in particular, edge crossings. This effect was achieved both by avoiding several edges present in other visualizations, such as between equivalent classes, and by applying splitting rules for node multiplication, as described in Section 3.2. All but one of the participants understood the way equivalent classes are displayed. Two more participants asked for a clarification, as they wondered whether there would be multiple copies of the double-ringed equivalent class circles, one for each equivalent class, or only one in all.

As opposed to that, the general base class `owl:Thing` was indeed displayed several times according to the splitting rules of VOWL 2 (cf. Section 3.2). Interestingly, this was instantly understood and pointed out whenever participants encountered `owl:Thing` in one of the tasks solved in VOWL. While multiple copies of `owl:Thing` at first create slightly more nodes, the edges linking to those `owl:Thing` nodes are overall shorter and with less edge crossings, again contributing to the aforementioned impression that VOWL does not feature a ‘maze of edges’.

Many of the other comments dealt with implementation-specific issues. The most important features asked for (when they were missing) or praised (when they were present) were a continuous zooming feature—for which most participants preferred the mousewheel, as offered in VOWL and SOVA—and a search facility to quickly find particular classes (or other elements) by name, which was present in SOVA, and in a

limited fashion also available in WebVOWL due to the built-in browser text search. Furthermore, a highlighting feature that changes the color of all directly connected nodes upon hovering the mouse cursor over one node, like the one available in SOVA, was stated to be useful. It was, however, noted, that such a highlighting should not completely replace the original node colors.

5 Conclusions and Future Work

In an earlier effort to create a uniform visual notation for OWL ontologies, we developed a first version of VOWL [34]. Based upon that work, related endeavors, as well as findings from a user study [33], we have redesigned large parts of the notation to create VOWL 2, a visual language that can also be understood by casual ontology users. We have described the key considerations, features, and capabilities of VOWL 2 and have presented two implementations of the visualization, a plugin for the widely used ontology editor Protégé and a responsive web application. Moreover, we have reported on a qualitative user study that compared VOWL 2 to two related ontology visualizations to get a better understanding of its readability and usability. We found that users were able to solve a majority of the tasks correctly and received overall favorable comments concerning VOWL 2.

5.1 Summary of Study Results

The qualitative user study provided us with some insights into how users perceive various features of VOWL 2:

- Structurally, the design of the graph visualization that allows moving nodes with the pointing device was welcomed, and the force-directed layout would cause highly connected nodes to be easily recognizable. Study participants would frequently use the opportunity to reposition single nodes in the graph manually in order to get a clearer understanding of the connections present.
- Due to the force-directed layout, disjoint subgraphs can get pushed out of sight. Hence, the question of adding some invisible link to create a slight attraction to the main part of the graph still needs to be tackled.
- The user study revealed that the idea of multiplying general nodes (e.g. `owl:Thing` or `rdfs:Literal`) so they do not appear overly important can be understood by users.
- The unification of equivalent nodes causes a cleaner appearance for the graph visualization. However, although the equivalence relationship was correctly recognized in the user study, several users were unsure whether equivalent classes are displayed as several nodes, each referring to the other nodes, or as only one node.
- The fact that many elements are explicitly labeled evoked mostly positive feedback among the study participants. One of the study participants stated that VOWL 2 is so self-explanatory that no notation reference was needed at all, while others commented that some of the labels could be removed from the visualization without loss of clarity. Thus, it seems to be advisable to give users a choice whether such labels are displayed.

5.2 Open Issues

The ontologies used in the study, which were of a relatively moderate size (comprising 7, 22, and 53 classes), could still be handled in VOWL 2. However, there is no upper limit for the size of ontologies, both because a vast number of topics can be covered in one ontology, and because an ontology can be modeled down to an arbitrary level of detail. Graph visualizations are, on the other hand, viable only up to a certain graph size, at which point the overview is lost and the graph is not easily usable any more. VOWL 2 mitigates this only slightly based on its splitting rules and the unification of equivalent classes. Ongoing research on VOWL will have to look into means of dealing with this problem, at latest when VOWL is used to analyze the connections and alignments across several ontologies. Solutions will incorporate both considerations on the automatic and manual detection of ontology parts that carry context-specific importance, so parts of the ontology can be temporarily hidden or bundled, as well as general strategies for handling large graphs used in the field of graph visualization.

A related issue exists due to the fact that ontology elements have no inherent relative location information. Therefore, all elements are initially placed in a random manner in the force-directed layout. While this does not influence a single session of work with the VOWL depiction of an ontology, it prevents the users to create a “mental map” of the visualization that is valid for several sessions, as the elements are at different locations every time the VOWL graph is rendered. Future work will have to develop reasonable guidelines on how to best place ontology elements so their positioning follows a reproducible pattern.

References

1. Ontology Definition Metamodel. <http://www.omg.org/spec/ODM/>.
2. OOBIAN Insight. <http://dbpedia.oobian.com>.
3. Protégé Wiki. <http://protegewiki.stanford.edu/wiki/Visualization>.
4. Unified Modeling Language. <http://www.uml.org>.
5. H. Alani. TGvizTab: An ontology visualisation extension for protégé. In *2nd Workshop on Visualizing Information in Knowledge Engineering, VIKE '04*, 2003.
6. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
7. M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.
8. D. Brickley and L. Miller. FOAF Vocabulary Specification 0.99. <http://xmlns.com/foaf/spec/>, 2014.
9. D. V. Camarda, S. Mazzini, and A. Antonuccio. Lodlive, exploring the web of data. In *Proceedings of the 8th International Conference on Semantic Systems, I-SEMANTICS '12*. ACM, 2012.
10. A.-S. Dadzie and M. Rowe. Approaches to visualizing linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.
11. S. Falconer. OntoGraf. <http://protegewiki.stanford.edu/wiki/OntoGraf>, 2010.
12. S. M. Falconer, C. Callendar, and M.-A. Storey. A visualization service for the semantic web. In *Proceedings of the 17th International Conference on Knowledge Engineering and Management by the Masses, EKAW '10*, pages 554–564. Springer, 2010.

13. C. Fluit, M. Sabou, and F. van Harmelen. Ontology-based information visualization: Toward semantic web applications. In *Visualizing the Semantic Web*, pages 36–48. Springer, 2002.
14. F. J. García-Peñalvo, R. Colomo-Palacios, J. García, and R. Therón. Towards an ontology modeling tool. a validation in software engineering scenarios. *Expert Systems with Applications*, 39(13):11468–11478, 2012.
15. V. Geroimenko and C. Chen. *Visualizing the Semantic Web: XML-Based Internet and Information Visualization*. Springer, 2nd edition, 2006.
16. J. Heer, S. K. Card, and J. A. Landay. Prefuse: A toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 421–430. ACM, 2005.
17. P. Heim, S. Lohmann, and T. Stegemann. Interactive relationship discovery via the semantic web. In *Proceedings of the 7th Extended Semantic Web Conference, Part I*, ESWC '10, pages 303–317. Springer, 2010.
18. P. Heim, J. Ziegler, and S. Lohmann. gFacet: A browser for the web of data. In *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web*, IMC-SSW '08, pages 49–58. CEUR-WS.org, vol. 417, 2008.
19. D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sep. 2006.
20. W. Hop, S. de Ridder, F. Frasinca, and F. Hogenboom. Using hierarchical edge bundles to visualize complex ontologies in glow. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 304–311. ACM, 2012.
21. M. Horridge. OWLViz. <http://protegewiki.stanford.edu/wiki/OWLViz>, 2010.
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), Nov. 2007.
24. 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.
25. P. Kunowski and T. Boiński. SOVA – Simple Ontology Visualization API. <http://protegewiki.stanford.edu/wiki/SOVA>, 2012.
26. M. Lanzemberger, J. Sampson, and M. Rester. Visualization in ontology tools. In *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems*, CISIS '09, pages 705–711, 2009.
27. 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.
28. T. Liebig, O. Noppens, and F. W. von Henke. Viscover: Visualizing, exploring, and analysing structured data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, VAST '09, pages 259–260. IEEE, 2009.
29. S. Lohmann. Modular unified tagging ontology (MUTO). <http://purl.org/muto/core#>, 2011.
30. S. Lohmann, S. Negru, and D. Bold. The ProtégéVOWL plugin: Ontology visualization for everyone. In *Proceedings of ESWC 2014 Satellite Events*. Springer, 2014, to appear.
31. 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 *Proceedings of the 10th International Conference on the Semantic Web, Volume I*, ISWC '11, pages 470–486. Springer, 2011.
32. S. Negru. PersonasOnto. <http://blankdots.com/open/personasonto.html>, 2014.

33. S. Negru, F. Haag, and S. Lohmann. Towards a unified visual notation for owl ontologies: Insights from a comparative user study. In *Proceedings of the 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 *Proceedings of the 9th International Conference on Web Information Systems and Technologies, WEBIST '13*, pages 308–315. SciTePress, 2013.
35. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages, VL '96*, pages 336–343. IEEE, 1996.
36. M.-A. Storey, N. F. Noy, M. Musen, C. Best, R. Ferguson, and N. Ernst. Jambalaya: An interactive environment for exploring ontologies. In *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI '02*, pages 239–239. ACM, 2002.
37. L. Wachsmann. OWLPropViz. <http://protege.wiki.stanford.edu/wiki/OWLPropViz>, 2008.
38. T. D. Wang and B. Parsia. CropCircles: topology sensitive visualization of OWL class hierarchies. In *Proceedings of the 5th International Conference on the Semantic Web, ISWC '06*, pages 695–708. Springer, 2006.

Table 2. Tasks of the user study (translated from German to English). Groups of six tasks were applied to the three visualizations (column *V*), which were presented in a balanced order (i.e. $V = 1$ was either GrOWL, SOVA, or VOWL). In each of these groups, there were three tasks per ontology (column *O*), FOAF (*F*) and PersonasOnto (*P*).

V	O	Question
1	F	Which classes appear to be crucial for the ontology?
		Which data properties does the class <i>Person</i> have?
		Is the class <i>Spatial Thing</i> equivalent to any other class? If so, to which?
	P	How many <i>different</i> (i.e. not equivalent) classes are approximately visible in total?
		Which classes are subclasses of the class <i>Resource</i> ?
		Do the classes <i>Place</i> and <i>Organization</i> have any properties in common? If so, which ones?
2	F	Which classes have been imported (from other ontologies)?
		Which data properties does the class <i>Document</i> have?
		Is the class <i>Document</i> equivalent to any other class? If so, to which?
	P	Which classes appear to be crucial for the ontology?
		Which classes are subclasses of the class <i>Persona</i> ?
		The property <i>isPartOfTest</i> refers to the class <i>UsabilityTest</i> . For which classes is this property defined?
3	F	How many <i>different</i> (i.e. not equivalent) classes are approximately visible in total?
		Which data properties does the class <i>Image</i> have?
		Is the class <i>PersonalProfile</i> equivalent to any other class? If so, to which?
	P	Can you recognize any classes that serve as a generalization of many subclasses?
		Which classes are subclasses of the class <i>Scenario</i> ?
		Do the classes <i>Participant</i> and <i>Goals</i> have any properties in common? If so, which ones?