Towards a Unified Visual Notation for OWL Ontologies: Insights from a Comparative User Study

Stefan Negru
Faculty of Computer Science, Alexandru Ioan Cuza University
General Berthelot 16
Iasi 700483, Romania
stefan.negru@info.uaic.ro

Florian Haag
Institute for Visualization and Interactive Systems (VIS)
Universitätsstraße 38
70569 Stuttgart, Germany
florian.haag@vis.uni-stuttgart.de

Steffen Lohmann
Institute for Visualization and Interactive Systems (VIS)
Universitätsstraße 38
70569 Stuttgart, Germany
steffen.lohmann@vis.uni-stuttgart.de

ABSTRACT

Browsing ontologies and making sense of their concepts and relationships are important activities in knowledge engineering. They are supported by a large number of tools that often provide visual representations of the ontologies for better illustration. However, different notations are used to represent the ontologies which can be confusing when switching tools. Furthermore, many representations provide only basic overviews of the ontologies that are not sufficient to get a deeper understanding of the concepts and relationships. A unified visual notation for OWL ontologies would be most helpful to overcome these limitations. Having this goal in mind, we compare two different notations for OWL ontologies in this paper: the UML profile of the Ontology Definition Metamodel (ODM) and the Visual Notation for OWL Ontologies (VOWL). We report on a comparative user study of these notations and discuss benefits and limitations raised by the study participants. Based on these findings, we draw some general conclusions regarding the development of a unified visual notation for OWL ontologies.

Keywords

Ontology Visualization, OWL, UML, VOWL, Semantic Web

1. INTRODUCTION

Ontologies, Linked Data and other concepts of the Semantic Web have become widely used and no longer exclusively address expert users. They are also increasingly used by non-experts in a variety of contexts and activities. Especially the Web Ontology Language (OWL)[1] has emerged as the 'lingua franca' to represent semantic data, i.e. to describe the concepts, instances, and relationships that make up the data [2].

In this context, visualizations have an appealing potential when it comes to exploring, verifying and ultimately making sense of semantic data and the ontologies that describe it. Although several solutions for the visualization of ontologies have been developed, the reusability of most of these approaches is limited by their context of use or objective. With the available solutions, one can often only explore particular pieces of information in ontologies, or they are tailored for a specific domain. Moreover, the majority of existing approaches represent concepts and instances in a mutually exclusive manner, often lacking an integrated representation of the ontology. Such a separation between concepts and instances raises the level of abstraction and requires the users to be familiar with ontology-related formalisms. All of these issues have a great impact particularly on less experienced users who would greatly benefit from an effective and uniform visual representation for ontologies [6].

In this paper, we compare two visual notations for representing OWL ontologies, one reusing the Unified Modeling Language (UML)[2] and the other being specifically designed for OWL. We analyze and compare the strengths and weaknesses of each notation in order to identify general issues and potential improvements. Furthermore, we correlate our results with the findings of related work. The ultimate goal is to get some general insights that foster the development of a unified and comprehensive visual notation for OWL ontologies.

Section 2 gives a brief overview of existing ontology visualizations and of studies that evaluate such visualizations. Section 3 introduces the notations that we compare in this work. The comparative study itself is described in Section 4, along with a presentation and discussion of the results. Based on these results, Section 5 presents a number of challenges that need to be addressed by a comprehensive visual notation for OWL.

2. RELATED WORK

While the World Wide Web Consortium (W3C) specifies OWL in several documents, it does not recommend a notation to visually represent OWL. It only provides a notation for the visualization of RDF[3] that OWL is build upon. However, as the OWL constructs are represented by multiple triples in RDF, the visualizations quickly become large with plenty of nodes and edges. Such RDF visualizations of OWL are not only hard to read but also fail to adequately reflect the semantics of the OWL constructs.

This has encouraged a broad exploration of different methods and tools for visually representing OWL ontologies. Most of them provide only a partial representation of the ontologies, focusing either on the concepts, the instances, or the relationships, but do not offer an integrated and complete view [4, 8].

[1]http://www.w3.org/OWL/

[2]http://www.uml.org

[3]http://www.w3.org/TR/rdf-primer/

ISEM '13, September 04 - 06 2013, Graz, Austria

ACM 978-1-4503-1972-0/13/09...$15.00.
http://dx.doi.org/10.1145/2506182.2506192.

To appear in:
Proceedings of the 9th International Conference on Semantic Systems (I-SEMANTICS 2013)
New York, NY, USA: ACM, 2013.

2.1 Visualization Methods

Looking at the multitude of approaches for ontology visualization, some common characteristics stand out. Most of them employ graph-based layouts, are in 2D and propose visualizations especially tailored to provide a basic overview of the ontology that focuses on key parts, such as the concept hierarchy.

A decent number of visualizations are implemented as part of ontology editors. For instance, there are several plugins for Protégé[4] that offer ontology visualizations. While some of them are centered around the visualization of concept hierarchies (e.g. OWLViz[5]), others are capable of representing different kinds of property relations (e.g. OntoGraf[6]). There are also several stand-alone tools dedicated to the visualization of ontologies. Examples include GrOWL [10], CropCircles [18], or OOBIAN Insight[7]. All these approaches vary in the complexity of their visualization techniques, the visual notation used to graphically represent ontologies – which is often not explicitly described –, and the context of use.

In an effort to create a standard graphical notation to enhance communication of OWL to a larger audience, the Object Management Group (OMG) formally specified mappings between elements from OWL and UML class diagrams, as part of the Ontology Definition Metamodel (ODM)[8]. Related approaches have been proposed by Brockmans et al. [3], Djurić et al. [5], and Parreiras et al. [15], including corresponding metamodeling frameworks and transformation rules. Such UML-based approaches allow the creation of OWL visualizations with common UML editors. Yet, there are also stand-alone editors specifically for the UML-based visualization of OWL ontologies, such as OWLGrEd [1].

2.2 Comparative Studies

Most studies that evaluate and compare ontology visualization methods, techniques, or tools focus primarily on assessing quantitative values, such as issues concerning the performance of the visualization tools (and the graphical representation they made use of) in terms of scalability, efficiency, and available features. Only few studies address qualitative aspects of ontology visualizations and/or perceived (dis)advantages of the graphical representations for OWL.

Lanzenberger et al. [11] reviewed a number of ontology tools along with the visualization techniques they employ. Their survey provides a good overview of the techniques, methods, graphical elements, and distinct features of these tools, and they conclude that "most tools require ontology expert knowledge". A more in-depth analysis of different 2D and 3D ontology browsing tools, their visualization techniques, characteristics, features, and issues has been conducted by Katifori et al. [8]. They provide valuable insight into the different visualization types, user interactions, and scalability issues. Regarding task support, they point out that visualization methods maintaining a fixed positioning of the nodes and allowing for quick browsing prove to be most effective. Moreover, they emphasize that interaction and navigation techniques play an important role in visualization methods, as they allow for an advanced exploration of the ontology.

In a different study, Katifori et al. [7] investigated the effectiveness of visualization plugins for Protégé. They compared the Protégé Class Browser (class hierarchy) with the visualization plugins Jambalaya, TGVizTab, and OntoViz. Their results indicate that the Protégé Class Browser performs best in terms of effectiveness (as perceived by users) and time for task completion. Another study [17] focused on a performance analysis of cognitive support issues for the Protégé plugins DL Query, OntoGraf, and OWL2Query, with the result that users solved the tasks of the study faster with OntoGraf than with the other tools.

Related to this, Motta et al. [13] compared the KC-Viz plugin of the NeOn Toolkit with the *OWLViz* plugin of Protégé. They assessed task performance and usability among other measures. Overall, KC-Viz proved effective for displaying "key concepts" when users were confronted with large and unfamiliar ontologies.

Dzbor et al. [6] conducted an observational user study which investigates shortcomings of ontology engineering and visualization tools developed within the NeOn project. They point out that the majority of the investigated visualizations are "too generic to be useful in the users' problems (e.g. seeing ontology dependencies or term occurrences in an ontology)".

3. VISUAL NOTATIONS

As we have seen in the previous section, there are many studies that evaluate tools used to visualize ontologies but only little that investigate the actual graphical representations. Moreover, most studies focus on quantitative rather than qualitative issues.

In order to address these shortcomings, we compared two visual notations for OWL in a qualitative user study. The first is the aforementioned UML profile of the Ontology Definition Metamodel (ODM). We took version 1.0 of the ODM specification that was published in 2009. The second is the "Visual Notation for OWL Ontologies (VOWL)" [14]. Version 1.0 has been released in early 2013 by a subset of the authors of this paper.[9]

We decided for these two notations, as they provide a comparatively complete and comprehensive set of graphical elements for representing OWL ontologies. Furthermore, they are well-specified and have also been designed for users who are less familiar with the Semantic Web and related concepts. Another reason was that they follow quite different approaches: While one reuses the well-established UML notation, the other has been designed specifically for OWL. Still, the notations are comparable, as they both employ some kind of node-link diagram.

3.1 Used Ontology

The ontology that we made use of throughout our evaluation is the Modular Unified Tagging Ontology (MUTO) [12]. It is an ontology focusing on the domains of tagging and folksonomies that reuses several concepts from other ontologies. We used version 1.0 which has been released in late 2011.[10]

We selected the ontology for its manageable size and because it is relatively easy to understand for users who have not come in contact with ontologies before. On the other hand, the MUTO ontology is not as popular as other ontologies (such as FOAF[11] or the wine ontology[12]) that some of the study subjects might have seen before. Thus, we could avoid effects that may result from certain study participants being familiar with the ontology. Furthermore, the visual representations of the MUTO ontology fitted on a screen and were easily readable without a need for scrolling or zooming (compare the figures in this paper). This was important, as we were purely interested in the visual notation and not in the interaction with the ontology in this work.

[4]http://protege.stanford.edu/
[5]http://protegewiki.stanford.edu/wiki/OWLViz
[6]http://protegewiki.stanford.edu/wiki/OntoGraf
[7]http://dbpedia.oobian.com/
[8]http://www.omg.org/spec/ODM/
[9]http://purl.org/vowl
[10]http://purl.org/muto
[11]http://xmlns.com/foaf/spec/
[12]http://www.w3.org/TR/owl-guide/wine.rdf

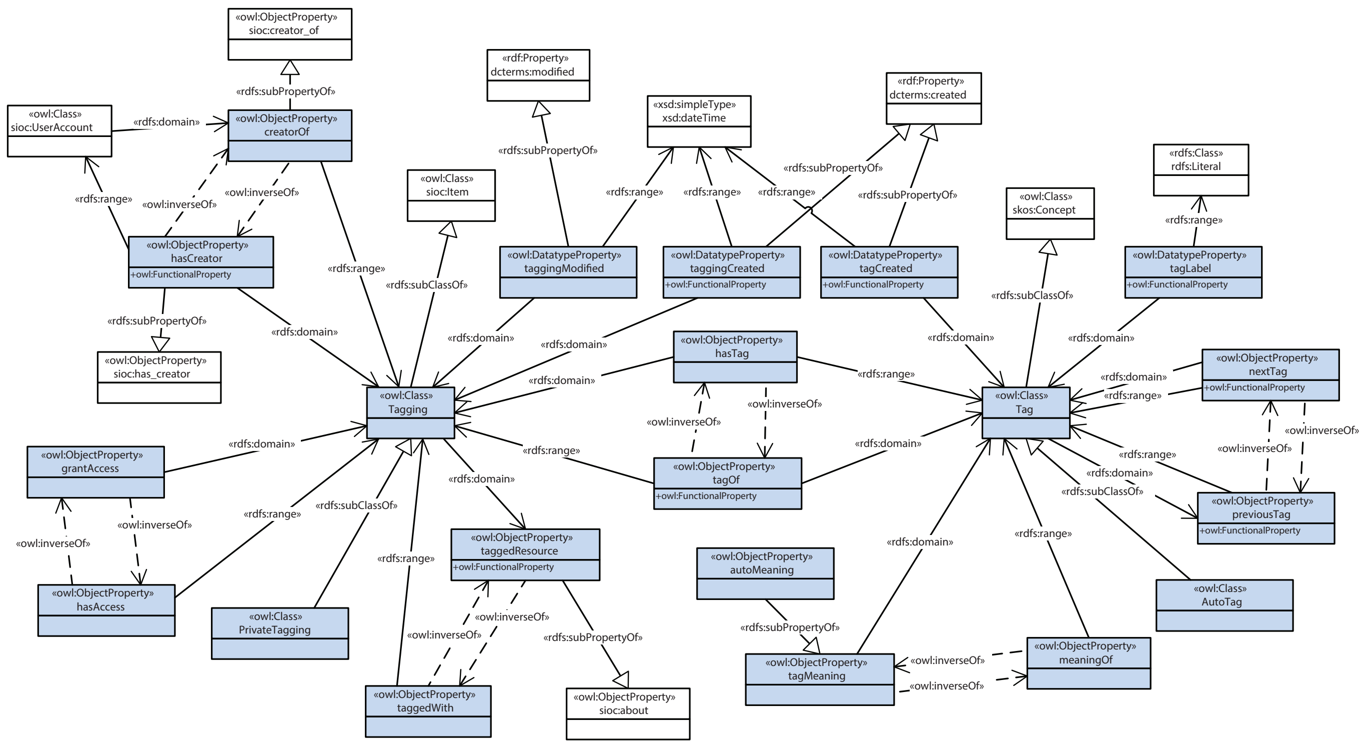

Figure 1: UML notation applied to the MUTO ontology (Conceptual View)

3.2 UML Notation for OWL Ontologies

The UML profile for OWL (as part of the ODM specification) has been designed to assist modelers developing ontologies through the reuse of the UML notation and standard UML tools [9]. It supports users familiar with UML by:

- Making use of UML constructs when they have the same semantics as OWL;
- Defining custom stereotypes of existing UML constructs consistent with the OWL semantics;
- Offering additional combinations of stereotyped constructs – when suitable UML constructs do not already exist – in order to provide a notation for the OWL semantics.

Figure 1 shows a graphical representation of the MUTO ontology that uses this notation. The *owl:Class* construct is mapped to the *UML::Class* element, while subclass relations are visualized via *UML::Generalization*. Properties are represented as association classes, i.e. they are also mapped to *UML::Class* and combined with *UML::Association* to indicate their domain and/or range. Subproperty relations are visualized like subclass relations via *UML::Generalization*.

Overall, the notation makes use of two types of graphical elements: *rectangles* for concepts, properties, and instances (the latter is not shown in Figure 1), and *lines* (with different arrow heads and stroke patterns) to indicate the domain and/or range of properties as well as special property types (subclass, subproperty, and inverse relations).

3.3 VOWL Notation for OWL Ontologies

The Visual Notation for OWL Ontologies (VOWL) [14] allows for a comprehensive visualization of ontologies by providing graphical representations for most OWL elements. It distinguishes between three visualization layers: *Conceptual Layer*, *Instance Layer* and *Integrated Layer*. The latter provides an integrated view on the ontology, depicting the concepts along with their instances (visually represented as sections of a circle). Figure 2 shows the visual representation of the *Conceptual Layer* of the MUTO ontology using the VOWL notation, while Figure 3 depicts the *Integrated layer*, including a sample set of instances.

The notation separates the OWL elements into two sets: *primary elements* (e.g. *owl:Class*, *owl:ObjectProperty*, *owl:DatatypeProperty*, *owl:cardinality*, etc.) that are crucial for representing and understanding the ontology and *additional elements* (e.g. *owl:DeprecatedClass*, *owl:sameAs*, etc.) which either define constraints or provide supplementary details of the ontology. The graphical representations for elements such as *owl:intersectionOf*, *owl:unionOf*, and *owl:ComplementOf* reuse corresponding Description Logic symbols ($\sqcup$, $\sqcap$ and $\neg$), which may only be familiar to expert users.

In total, the VOWL notation is based on four types of graphical elements:

- *circles* for concepts;
- *sections in the circles* for instances;
- *lines* (with different arrow heads and stroke patterns) for properties;
- *rectangles* for literals, data types and values.

Concepts can 'contain' instances in the *Integrated Layer* that are either defined by *rdf:type*, part of an enumeration (e.g. *oneOf*), or inferred from their subclasses. Instances of the latter type are shown in gray to avoid confusion or false attribution (cp. Figure 3). A detailed description of the notation on the example of the FOAF ontology is given in [14].

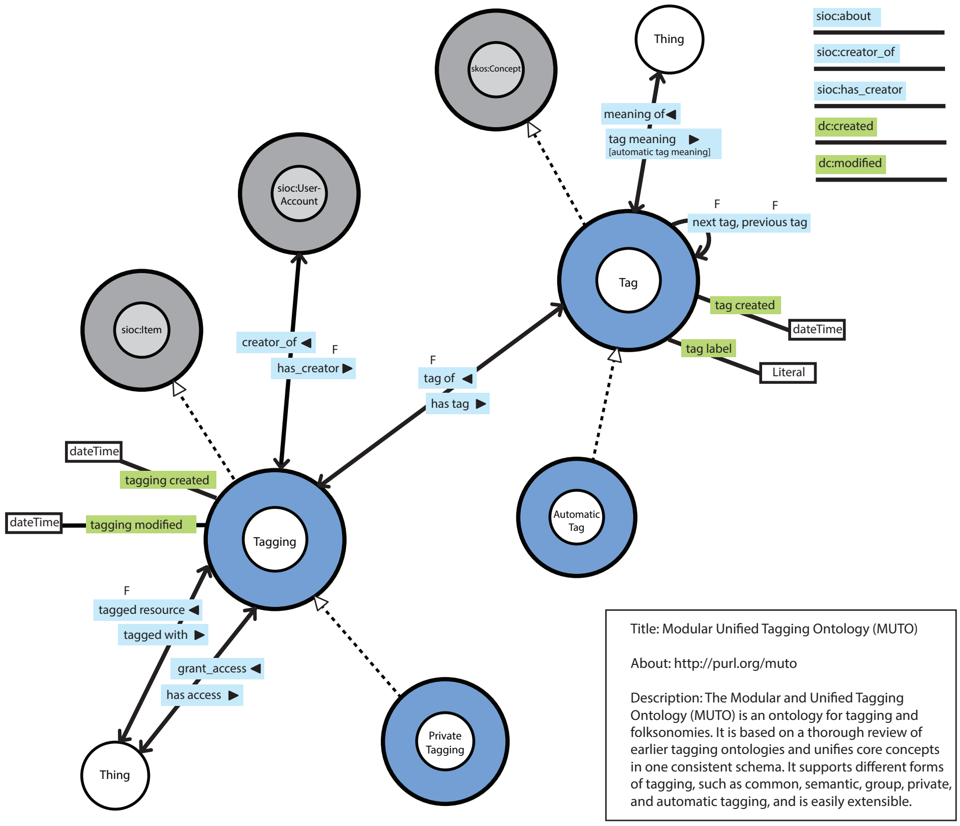

Figure 2: VOWL notation applied to the MUTO ontology (Conceptual View)

4. USER STUDY

The user study was conducted with the purpose of evaluating the two visual notations and finding useful information regarding:

- Advantages and disadvantages of the UML notation;
- Advantages and disadvantages of the VOWL notation;
- Insights on issues and improvements to both visual notations;
- General findings and insights on the visual representation of OWL ontologies.

This section provides an overview of the performed user study, providing details regarding the used materials, the participants, and the tasks performed by them. Our focus was on the overall clarity and readability of the graphical representations offered by the notations.

4.1 Study Design

In order to compare the two visual notations, we recruited a group of 6 participants – between 23 and 38 years of age – whose knowledge and familiarity with UML and OWL varied. The user group was composed of both graduate students and researchers in the fields of computer science and visualization. This ensured that the users were at least somewhat familiar with one of either UML or OWL.

Participants were asked to specify their familiarity with UML and OWL using the following options: *Never used it*, *Passing Knowledge*, *Practical Knowledge*, or *Deep Understanding*. The majority of the participants had either practical knowledge or a deep understanding of UML, while their familiarity with OWL ranged from passing knowledge to practical knowledge.

Participants were provided with an explanation of the concepts and terminology of ontologies at the beginning of each session in order to make sure that tasks and representations were sufficiently understood. The participants were not provided with a tutorial or warm up task for each notation, because we wanted to include observations of the very first practical experience of the users with each of the visual notations.

The materials consisted of four visual representations (presented as static images) depicting the MUTO ontology. We took up the idea of VOWL and created two views for each notation: one *Conceptual View*, depicting the concepts and the relationships between them, and one *Integrated View*, showing the instances and their relationships with the concepts. In order to prevent any bias towards a notation, we gave both notations neutral labels.

Figure 1 shows the *Conceptual View* created with the ODM UML profile, while Figure 2 depicts the *Conceptual View* realized with the VOWL notation. Figure 3 depicts the *Integrated View* of the VOWL notation.

During the study, each participant was asked to fill out a questionnaire consisting of the following parts:

- *General Questions* – This part asked for demographic data and familiarity with UML and OWL;
- *Tasks* – The tasks were presented in two groups, each containing four tasks to be performed by the participants;
- *Notation Assessment* – This part asked for information on the preferred visual representation along with advantages, disadvantages, observations, and potential improvements, as suggested by each participant.

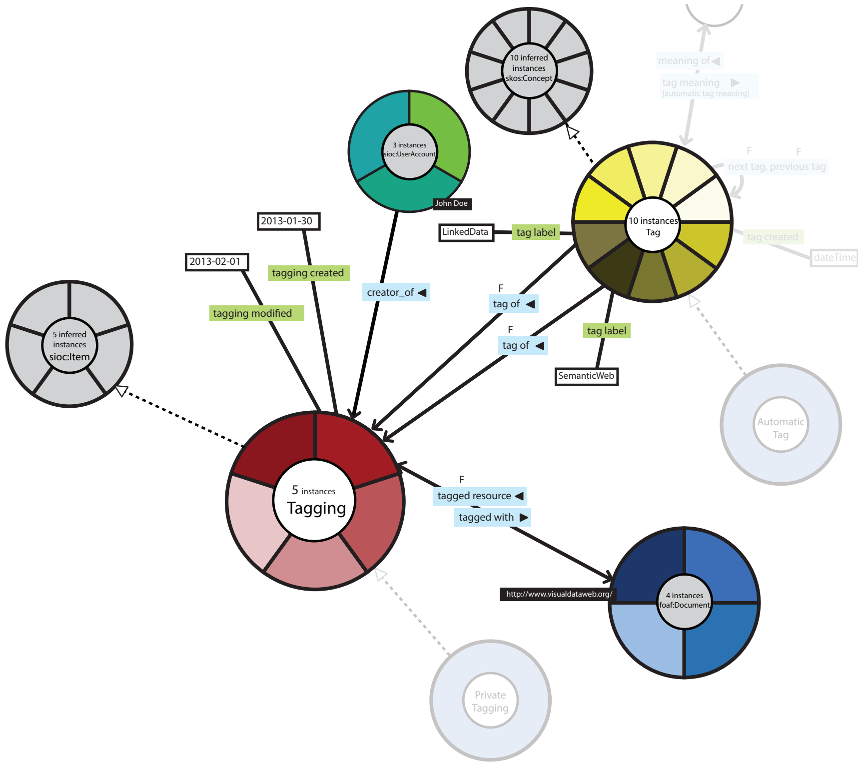

Figure 3: VOWL notation applied to the MUTO ontology (Integrated View)

4.2 Tasks and Procedure

At the beginning of each test session, the objectives of the study were stated to the participants: "You will be presented with two types of visual notations for representing ontologies: type A and type B. Please consider both the overall graphical representation and particular elements of the representation". We asked the participants to 'think aloud' and informed them that there is no time limit, so that they can take all the time they need to solve the tasks. The visual representations were presented on a 24-inch monitor in full screen. The questionnaire was provided on paper and the participants had to write down their answers.

The tasks focused primarily on the identification of key elements that make up an ontology: concepts, instances, and relationships between them. The first group of tasks was done on one of the notations, while for the second group we switched to the other notation. The order of the tasks remained the same, but we alternated the two visual notations, i.e. one half of the participants first saw the UML-based visualization, while the other half first got the visualization created with VOWL. The tasks were designed to ensure coverage of frequently encountered visual elements for each of the notations. At the same time, we strived to keep the tasks at a beginner level as far as possible, being aware that some of the participants were only roughly familiar with ontologies.

Table 1: One group of tasks performed by the participants.

Task 1: What are the domain and range of the property *tag of*?
Task 2: Name the properties that have the concept *Tagging* as domain.
Task 3: Which property is the inverse of the property *tag meaning*?
Task 4: How many instances does the concept *Tagging* have?

One group of tasks given to the participants is shown in Table 1. The first three tasks of this group are focused on the conceptual representation of the ontology, while the last task considers the instances. Accordingly, we presented the *Conceptual View* for the first three tasks, and the *Integrated View* for the last, using the corresponding visual notation.

After the participants had finalized all tasks, the visual representations of both notations were shown again to each participant, and they were asked to state which notation they prefer and why. Moreover, they had the opportunity to write down any perceived advantages and disadvantages of each notation in the questionnaire, and to suggest potential improvements.

4.3 Results

The findings of our observations and the notes from the think-aloud protocol showed no serious problems regarding the tasks or the understanding of the ontology – all participants managed to correctly complete the majority of the tasks. Moreover, based on the fact that none of the participants got an introduction in any of the two visual notations, they needed relatively few hints during task completion. The advantages and disadvantages of both notations, as pointed out by the participants, are summarized in Table 2 and Table 3.

Participants more familiar with UML found it easier to read the direction of the properties using the UML notation for OWL. On the other hand, some participants who had less experience with UML and/or OWL were confused by the direction of the arrows and required hints in order to correctly identify the domain and range of the properties. Many participants found the information given by the UML notation visually redundant. They suggested, for example, to represent the domain and range with just one arrow (as in VOWL) instead of two (as in the UML diagrams).

Table 2: Reported advantages and disadvantages of the UML notation

Advantages	Disadvantages
Similarity with the representation used by UML.	Participants found it difficult to get an overview of the ontology.
Labels are more meaningful (contain more information).	In some cases, labels seem redundant.
Exact type of each property is visible (e.g. *rdf:Property*).	Too many individual elements which makes the representation seem chaotic and cluttered.
Names of instances are easy to depict and identify.	Searching for particular information can be time consuming and difficult.
–	Participants found it difficult to distinguish between elements (concepts, properties, and instances) as they had similar graphical representations.

Table 3: Reported advantages and disadvantages of the VOWL notation

Advantages	Disadvantages
Clean and compact overview of the ontology (conceptual structure).	Direction of domain and range and inverse property labels is not always clear.
Spotting the differences between datatype and object properties is easier than in the UML notation.	Limited scalability of the integrated view due to the representation of instances as sections in circles.
Instances and concepts are visually integrated.	Meaning of some visual elements (e.g. *F* for functional) is unclear and difficult to interpret without additional knowledge.
Good impression of the rough number of instances per concept.	The names and other details of individual instances are not immediately visible.

Furthermore, most participants found that a better separation between concepts and instances is required for the UML notation. They also expressed concern on how this notation handles large numbers of instances.

The VOWL notation was preferred by the majority of the participants (5 out of the 6 participants), because they found it "less cluttered" and they could more easily solve the tasks. When asked to provide more details regarding their preference, the main reasons given by participants were that "one immediately sees the concepts of the ontology" and that "the grouping of the instances in the circles is intuitive" (as two representative statements).

Despite the fact that the participants preferred the VOWL notation, many of them found it troublesome to read the directional indicators of the *inverseOf* type properties, such as the properties *creator_of* and *has_creator* (see Figure 2). Moreover, some of the subjects had trouble understanding the dashed arrows (the graphical representation of the *subclassOf* property) and the meaning of datatype properties, which, unlike object properties, had no arrow at either end of the line (for instance, *tagging created* in Figure 3).

Another aspect worth mentioning is that, although participants identified the existence of two concepts having the label *Thing* (see Figure 2), they correctly counted it as one. However, some participants with more practical knowledge about ontologies expressed their dislike for this visual multiplication, as they found the existence of two *Thing* concepts confusing.

Several participants stated that for the VOWL notation the grouping of the instances (Integrated View, Figure 3) might become problematic for a large number of instances. At the same time, it might become hard to distinguish what the instances are, as only the labels of a limited number of instances can be reasonably shown at a time.

All participants suggested that both notations would benefit from interaction, which could provide details on demand and ease the visual exploration of the ontology. For instance, they proposed to use visual highlighting to interactively indicate inferred instances or domain and range information.

4.4 Discussion

The results of the user study reveal several strengths and weaknesses of the visual notations. Although the participants preferred the VOWL notation over the UML one, both were successfully used to complete the tasks. If we correlate the findings with related work, we can draw more general conclusions.

One explanation of the participants' preference for the VOWL notation is that it allows for a more compact representation of the ontology. This finding corresponds to the study results of Purchase et al. [16] who found that 73% of the subjects prefer a narrower layout over a wider one because of "the closeness which allows you to view more association which helps" in task completion.[13]

While participants preferred to find the connecting edge and read its property label (VOWL notation) rather than finding boxes which show properties (UML notation), they also pointed out that the UML notation is more comprehensive with regard to some graphical elements. Among others, this refers to elements like arrows that explicitly define domain and range, or to explicitly showing the type of a property – object or datatype, functional, inverse functional, etc. Still, the information might become redundant and – as the findings from [16] show – 90% of their study participants preferred "having a single edge connecting the nodes, rather than an additional directional arc alongside the connecting edge".

Concerning the directional indicators used by the VOWL notation for representing inverse properties, [16] points out that in their study 60% of the subjects "preferred having directional indicators associated with every labeled relationship".

In addition, the participants of our study pointed out that coloring the concepts, property types, and instances had a positive impact on task solution. It helped to identify elements and find information in the graphical representation. Though color is considered important, it is not explicitly used and specified by the visual notations.

5. RESULTING CHALLENGES

Based on the results of the user study, we can derive a number of challenges that need to be addressed by a unified and comprehensive visual notation for OWL. We grouped these challenges into four categories: 1) relating concepts and instances, 2) representing different property types, 3) indicating domain and range, and 4) deciding which parts of the visual notation are static and which are interactive.

5.1 Relating Concepts and Instances

Clearly showing the relationships between concepts and instances by depicting instances inside concepts (as in VOWL) or closely related to concepts was preferred by the study participants. However, this becomes problematic in both notations if there is a high number of instances. While the individual sections of the circles can get

[13] Although the study of Purchase et al. [16] served another purpose, the findings regarding graph layout aesthetics and particular graphical elements could be generalized and are partly also applicable in our context.

very small in case of VOWL, placing instances in close proximity to their concepts is not always possible in UML due to space restrictions in the graph layout.

One approach to overcome this problem would be to group instances by certain criteria (e.g. alphabetically, by semantic relatedness, etc.), as has been proposed by some of the participants. However, finding some general criteria besides class membership that allow for an intuitive grouping of instances is difficult. Related to this, it was not clear to all participants how inheritance and multi-membership are represented in the notations.

The concepts *owl:Thing* and *owl:Nothing* have special meaning in OWL. Including them in the visualization can be irritating. Some participants were confused by the fact that *owl:Thing* appeared twice in the VOWL-based visualization of Figure 2, while others liked this visual multiplication. Hiding *owl:Thing* as in the UML example of Figure 1 might be less irritating, yet expert users could expect to see this information.

Similarly, the distinction between *rdfs:Class* and *owl:Class* might be relevant in certain cases but in many others it is not. Finding an adequate representation for such special constructs that does not confuse or distract the users is challenging.

5.2 Different Property Types

With regard to functional, inverse functional, transitive, and symmetric properties, we can observe that the usage of simple letter abbreviations as in VOWL is not sufficient. The UML notation implements a better approach in using the full label for each type of property.

Besides a different arrow head, the UML notation uses a label to indicate subclass relations. Some participants found this approach redundant in comparison with VOWL, where this kind of relationship is indicated purely by graphical elements. Others had trouble understanding the meaning of the graphical element if no label is provided.

Finding a good representation for subproperty relations is even more difficult. Because properties are represented in the same way as classes in the UML notation, they can simply be connected by an edge showing the subproperty relation. VOWL groups subproperties along with their superproperty if they have the same domain and range (as e.g. the subproperty *automatic tag meaning* in Figure 2). If the properties have a different domain and/or range, the representation of the subproperty relation is less compact and clear in VOWL.

However, using the same representation for both concepts and properties (as in the UML notation) seems to have negative effects on the users' ability to perform certain tasks and to easily distinguish between elements. We therefore recommend to use different representations, not only for concepts and properties but also for object and datatype properties. At least a different coloring as in the VOWL notation was reported to be helpful.

5.3 Indicating Domain and Range

The UML notation clearly indicates the direction of the domain and range of properties. However, our study participants had partly problems reading and interpreting this information correctly.

On the other hand, even though participants could easily find inverse properties in the VOWL notation, they had difficulties to identify the domain and range in the graphical representation of inverse properties (as e.g. for the properties *creator_of* and *has_creator* in Figure 2). A possible cause for this issue might arise from the positioning of the arrowheads associated which each label, as they do not point towards the domain but instead keep a horizontal orientation.

5.4 Interactive vs. Static Representation

Many study participants mentioned that they would like to interactively explore the ontology. They suggested to use interaction to visually emphasize certain information or to show details on demand. However, an interactive environment is not always available but ontology visualizations are often presented as static images (as in our user study).

While the UML notation for OWL does not require any interaction for its exploration, the VOWL notation was developed with an interactive exploration in mind. For instance, the Integrated Layer displays detailed information about instances only on demand. Showing all details at once, as in the UML diagram, can be problematic especially for larger ontologies. Therefore, another challenge is to decide which parts of the visual notation are static and which are interactive.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a comparative user study of two visual notations for representing OWL ontologies. The study focused on qualitative aspects of the notations with the goal to identify strong and weak points for each of them. In addition, we correlated the results with insights from related studies in order to understand how these notations could be further improved. We finally derived some key challenges that need to be faced when visualizing OWL ontologies. We hope that our findings will contribute to the development of a unified visual notation for OWL ontologies. Furthermore, we plan to use the insights gained in this study for developing a tool aimed at providing users with an easy to use and interactive graphical representation of OWL ontologies.

Acknowledgments

This work was partially supported by the European Social Fund in Romania, under the responsibility of the Managing Authority for the Sectorial Operational Program for Human Resources Development 2007-2013 [grant POSDRU/107/1.5/S/78342].

7. REFERENCES

[1] J. Barzdins, G. Barzdins, K. Cerans, R. Liepins, and A. Sprogis. OWLGrEd: a UML style graphical notation and editor for OWL 2. In *Proceedings of the 7th International Workshop on OWL: Experiences and Directions*, OWLED '10. CEUR-WS.org, 2010.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[3] S. Brockmans, R. Volz, A. Eberhart, and P. Löffler. Visual modeling of OWL DL ontologies using UML. In *Proceedings of the 3rd International Semantic Web Conference*, ISWC '04, pages 198–213. Springer, 2004.

[4] A.-S. Dadzie and M. Rowe. Approaches to visualizing linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.

[5] D. Djurić, D. Gašević, V. Devedžić, and V. Damjanović. A UML profile for OWL ontologies. In *Proceedings of the 2003 European conference on Model Driven Architecture: foundations and Applications*, MDAFA'03, pages 204–219. Springer, 2005.

[6] M. Dzbor, E. Motta, C. Buil, J. M. Gomez, O. Görlitz, and H. Lewen. Developing ontologies in OWL: an observational study. In *Proceedings of the OWLED 2006 Workshop on OWL: Experiences and Directions*, OWLED '06. CEUR-WS.org, 2006.

[7] A. Katifori, E. Torou, C. Halatsis, G. Lepouras, and C. Vassilakis. A comparative study of four ontology visualization techniques in Protégé: Experiment setup and preliminary results. In *Proceedings of the International Conference on Information Visualisation*, IV '06, pages 417–423. IEEE, 2006.

[8] A. Katifori, E. Torou, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods – a survey. *ACM Computer Surveys*, 39(4), 2007.

[9] E. F. Kendall, R. Bell, R. Burkhart, M. Dutra, and E. K. Wallace. Towards a graphical notation for OWL 2. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions*, OWLED '09. CEUR-WS.org, 2009.

[10] S. Krivov, R. Williams, and F. Villa. GrOWL: A tool for visualization and editing of OWL ontologies. *Web Semantics*, 5(2):54–57, 2007.

[11] M. Lanzenberger, 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. IEEE, 2009.

[12] S. Lohmann, P. Díaz, and I. Aedo. MUTO: the modular unified tagging ontology. In *Proceedings of the 7th International Conference on Semantic Systems*, I-SEMANTICS '11, pages 95–104. ACM, 2011.

[13] 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.

[14] 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.

[15] F. S. Parreiras, T. Walter, and G. Gröner. Visualizing ontologies with UML-like notation. In *Proceedings of the 2nd International Workshop on Ontology-Driven Software Engineering*, ODiSE '10, pages 4:1–4:6. ACM, 2010.

[16] H. C. Purchase, J.-A. Allder, and D. Carrington. Graph layout aesthetics in UML diagrams: User preferences. *Journal of Graph Algorithms and Applications*, 6(3):255–279, 2002.

[17] V. Swaminathan and R. Sivakumar. A comparative study of recent ontology visualization tools with a case of diabetes data. *International Journal of Research in Computer Science*, 2(3):31–36, 2012.

[18] 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.