

# WebVOWL Editor: Device-Independent Visual Ontology Modeling

Vitalis Wiens<sup>1,3</sup>, Steffen Lohmann<sup>1</sup>, and Sören Auer<sup>2,3</sup>

<sup>1</sup> Fraunhofer IAIS, Sankt Augustin, Germany  
vitalis.wiens@iais.fraunhofer.de  
steffen.lohmann@iais.fraunhofer.de

<sup>2</sup> L3S Research Center, University of Hannover, Germany  
auer@l3s.eu

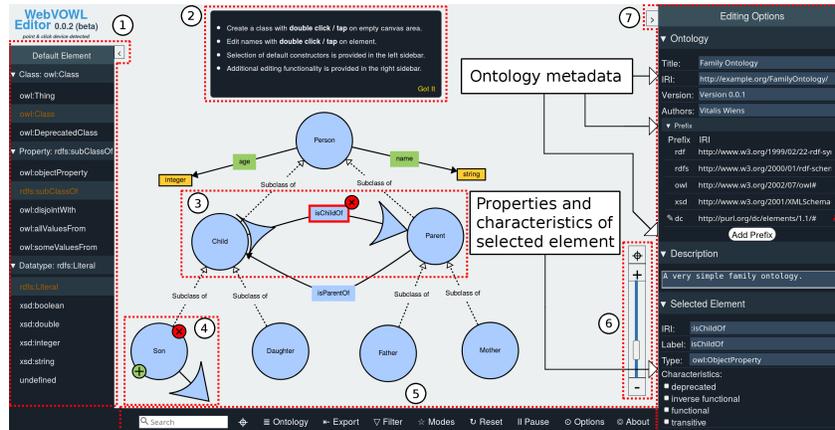
<sup>3</sup> TIB Leibniz Information Center for Science and Technology, Hannover, Germany

**Abstract.** The growing attention of ontologies in the industrial and the Semantic Web domain raises the need for approaches that are easy to use, immediately available, and independent of a particular device category and interaction context. In order to foster direct involvement of different user groups in ontology modeling, the complexity and learning processes of ontology engineering tools need to be reduced. We present WebVOWL Editor, an open source software that is 1) largely device independent, 2) uses visual modeling, and 3) provides modeling guidance. The device-independence is achieved by considering different modes of operation, including mouse and touch interactions. The visual modeling provides an intuitive interaction mode for creating and editing ontologies. Guidance during the ontology modeling and editing process reduces common pitfalls and ensures that valid ontologies are created. From discussions with domain experts, we identified a subset of OWL constructs that are considered most relevant in ontology modeling and are supported by the current implementation of the software.

**Keywords:** Visual Modeling, Device-Independence, OWL, WebVOWL

## 1 Introduction

A fundamental aspect of the Semantic Web is to create and communicate conceptualizations of information and data in various domains. Ontologies serve this purpose by providing a formal, machine-readable representation of the domain knowledge. The modeling of ontologies is often done collaboratively in joint efforts of knowledge engineers and domain experts. On the one hand, domain experts, who provide the conceptualization of the knowledge domain, are typically not familiar with semantic formalism and conceptual modeling techniques. They often find it hard to follow logical notations in OWL representation [1]. On the other hand, ontology engineers, who provide the necessary know-how for ontology modeling and logical notations in OWL, usually lack the expertise in the domain to create ontologies of sufficient quality.



**Fig. 1.** GUI overview: 1) collapsible sidebar for default element selection; 2) message box; 3) editing elements for a selected object property; 4) editing elements for a selected class; 5) tool menu; 6) zooming controls; 7) collapsible sidebar for editing options.

Visualizations are often used to support ontology development, exploration, verification, and sensemaking [3]. They can enable domain experts to get more directly involved in ontology modeling. Many ontology visualization tools have been developed in recent years, however, only few of them support direct editing of the ontology [2]. The available tools are currently mainly used for illustrating and exploring ontologies but not as an entry point for modeling. Nowadays, knowledge workers often use more than one device for their daily tasks in a myriad of interaction contexts, ranging from classical desktop settings to mobile scenarios in meetings, workshops, and on business trips. Also, digital devices for creativity techniques, such as interactive whiteboards, idea walls, and touch tables, are increasingly available and used for idea generation and conceptual modeling. This development shows the increasing need for device independent solutions for ontology modeling.

We demonstrate WebVOWL Editor, a web-based visual ontology modeling application that is independent of a particular device and interaction context. Ontologies are visualized using the VOWL notation which is a well-specified visual language for the user-oriented representation of OWL ontologies [3]. We achieve a high device-independence by considering different input and output modalities of common computing devices. Additional built-in guidance functionalities help to avoid common pitfalls and the creation of syntactically valid ontologies. The current implementation of WebVOWL Editor already supports a large share of OWL constructs, and runs in most modern web browsers. An overview of the graphical user interface is illustrated in Figure 1. Limited by device resources and web-based technologies, WebVOWL Editor is designed for visualizing and editing small and medium sized ontologies. Additionally, targeting domain experts and other user groups, it does not include any inference nor reasoning for visualization and creation of OWL ontologies and RDF(S) graphs.

## 2 WebVOWL Editor

WebVOWL Editor is designed to serve the skills and needs of domain experts with limited knowledge of ontology modeling. Its most distinctive characteristic is the device-independent mode of operation for the visual modeling of ontologies. The implementation is available under a persistent URL<sup>4</sup> and its source code is released on GitHub. The GitHub repository enables users to download and reuse the source code, open issues, and ask for help. Additionally, it is released under the MIT license in order to allow for wide-spread use and adoption.

We address the various backgrounds of different user groups by using the VOWL notation [3] for visual ontology modeling. Visual modeling is realized by device-independent modes of operation for mouse and touch interactions. The current implementation of WebVOWL Editor supports a subset of OWL constructs that were identified from discussions with domain experts and which are considered most relevant in ontology modeling.

**Visual Modeling Elements** — Visual modeling elements are provided for the nodes and links in the graphical representation of the ontology. In order to reduce visual clutter, these are presented to the user when an editing operation of a certain element is desired. With respect to device-independence, we define that the desire to do so arises when the user selects an element. Additionally, on point and click devices we show the visual modeling elements on mouse hovering in order to skip the interaction of first having to select an element. Visual modeling elements provide functionalities for creating, editing, and deleting elements such as classes, object properties, datatype properties and datatypes.

**Modeling of Classes** — Classes are created by performing a double click or double tap interaction on the empty space of the canvas area. After a class is created, we assume that the user wants to change the default label (i.e., `NewClass`) to a proper name for the conceptualization. Thus, we trigger a label editing operation on the creation of an element.

**Modeling of Object Properties** — Object properties, describing interrelations between classes, are created by dragging the arrow head from a source class to a target class, defining the domain and range of the corresponding property respectively. Additional visual elements are provided for properties on their selection in order to easily change domain and range of a property.

**Modeling of Datatypes** — Datatype properties are created with a visual modeling element that is provided only for classes (green plus button). The corresponding datatype is created based on the selection of the default element in the left sidebar. Additionally, according to VOWL notation, datatypes and datatype properties are bound together, thus datatype properties provide only an editing element for its domain.

**Label and IRI Editing** — Label editing is provided in the graphical representation (double click or double tap on a visual element) and in the right sidebar. IRI editing is provided only in the right sidebar. However, the corresponding IRI is also synchronised during label editing when it was not changed

---

<sup>4</sup> Tool and GitHub repository can be found at <https://w3id.org/webvowl/editor>

from the default initial value (as similar as how IRIs are handled in Protégé [4]), allowing the user to set the label and the IRI of an element simultaneously.

**Further Editing Options and Refinement** — The right sidebar provides further editing options for the meta information of ontology itself and the selected element in the graphical representation. Additional characteristics options are presented in the right sidebar for a selected element. Based on its type, these are filtered to only those that can be applied without invalidating the ontology. For example, for object properties *inverse functional* among other characteristics is applicable, while for datatype properties only *functional* is allowed.

**Guidance** — Guidance during the modeling and editing process plays a key role in order to involve domain experts and other user groups directly in ontology modeling. WebVOWL Editor provides built-in constraints for the visual modeling elements and additional restrictions in order to ensure the creation of syntactically correct ontologies. A message box provides feedback to the user, explaining the reason of an enforced constraint or other hints and guidance.

### 3 Demonstration Overview

In the demonstration scenario we will explain the need of WebVOWL Editor and its features. We want to provide a hands on experience for visual ontology modeling with WebVOWL Editor. A QR-Code (link) forwarding to the application will be provided, thus, allowing for independent testing of the tool. Additionally, we would like to invite the users to fill out a SUS-Questionnaire in order to gain feedback and insights for improvement of WebVOWL Editor. This tool aims for a direct involvement of different user groups by providing a software that is easy to learn, ready to use, and independent of a particular device category.

#### Acknowledgments

This work has partly been funded by the EU project GRACeFUL (grant no. 640954) as well as a scholarship from the German National Library of Science and Technology (TIB). In addition, parts of it evolved in the context of the Fraunhofer Cluster of Excellence “Cognitive Internet Technologies”.

#### References

1. Vania Dimitrova, Ronald Denaux, Glen Hart, Catherine Dolbear, Ian Holt, and Anthony G Cohn. Involving domain experts in authoring OWL ontologies. In *International Semantic Web Conference*, pages 1–16. Springer, 2008.
2. Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia Giannopoulou. Ontology visualization methods – a survey. *ACM Computing Surveys*, 39(4), November 2007.
3. Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4):399–419, 2016.
4. Stanford. Protégé. <http://protege.stanford.edu>.