OWLGrEd: Web-Based Visual Ontology Management

Jūlija Ovčiņņikova[1[0000-0002-5884-763X]], Mikus Grasmanis[1[0000-0002-0668-0970]], Lelde Lāce[1[0000-0001-7650-2355]] and Kārlis Čerāns[1[0000-0002-0154-5294]]

[1] Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia
julija.ovcinnikova@lumii.lv

Abstract. We present a major re-work of the visual diagrammatic OWL ontology visualization and editing tool OWLGrEd into a web environment. OWLGrEd uses UML class diagram style for ontology presentation by depicting OWL classes as UML classes and OWL properties as UML associations and attributes, while OWL Manchester syntax is used for compact presentation of deeper ontology constructs. The web-based OWLGrEd supports full ontology engineering round-trip workflows between the visual and textual formats, as well as collaborative ontology management. In addition to the technological re-work, the new editor introduces node-based visualization options for object and data properties, with a visual notation for property qualifiers.

Keywords: Visual ontology notation, OWL 2, Web-based ontology editor.

1 Introduction

Presenting OWL ontologies in a comprehensible form is important for ontology designers and users alike. Graphical representations help developers obtain an overview of concepts and relations and facilitate communication with users and domain experts [1]. The strength of the visual diagrammatic notation lies in presenting the related ontology elements as related also in the diagram.

Several visual tools support graphical modeling and exploration of OWL 2 ontologies. OWLGrEd [2, 3] combines UML-style diagrams with Manchester syntax to compactly represent OWL 2 constructs, while OntoDia [4] and WebVOWL [5] focus more on simplified interactive graph-based views. Diagrammatic environments such as Chowlk [1] and GRAPHOL [6] support ontology diagram creation in the visual environment and represent properties as explicit visual nodes. Support for ontology import and export is available in several tools, including classical OWLGrEd, GRAPHOL, and WebVOWL. However, these tools vary in how ontology constructs are represented visually, what users can create or modify in the visual environment, and whether they support shared and extensible ontology-engineering workflows. The technological limitations, including the platform and the licence conditions behind the desktop-based OWLGrEd has precluded its wider adoption, as well as slowed down its development efforts.

In this paper, we address the earlier existing technological limitations on the OWLGrEd editor by providing a new web-based implementation in ajoo [7] modelling

tool building platform and a web-based graphical diagram presentation engine [7]. The implementation in the ajoo opens the platforms' collaborative management capabilities of visual ontology presentations. The OWLGrEd web version maintains the principle of compact UML-style and OWL Manchester syntax presentation of OWL ontologies.

In addition to the technological re-work, the new OWLGrEd editor introduces node-based visualization options for object and data properties, with a visual notation for property qualifiers, as well as it adopts RDF-style namespace prefix notation. The re-worked open-source ontology management environment opens the possibilities of further generic and domain-specific enhancements.

The idea of a web-based OWLGrEd was investigated already in [8] by migrating the desktop editor while preserving its domain logic within a web-based operating-system-like infrastructure. Although this enables rapid web adoption through logic reuse, the approach primarily reflected a mechanical migration that does not fully exploit native web architecture. In contrast, our work develops a new web-based OWLGrEd that preserves the notation semantics while introducing an updated web-native design.

2 OWLGrEd Notation and Editor

OWLGrEd [2, 3] provides a UML-based graphical notation for OWL 2, where classes are shown as class nodes, object properties as associations between domain and range classes, and data properties as attributes; individuals and cardinalities are represented using UML object and multiplicity notation. To cover constructs without direct UML counterparts, the notation adds Manchester-syntax fields for complex class axioms and property-level fields for property axioms and characteristics, and it visualizes equivalence/disjointness (binary and n-ary) via connectors and grouping boxes, as well as object property restrictions (some, only) and cardinalities.

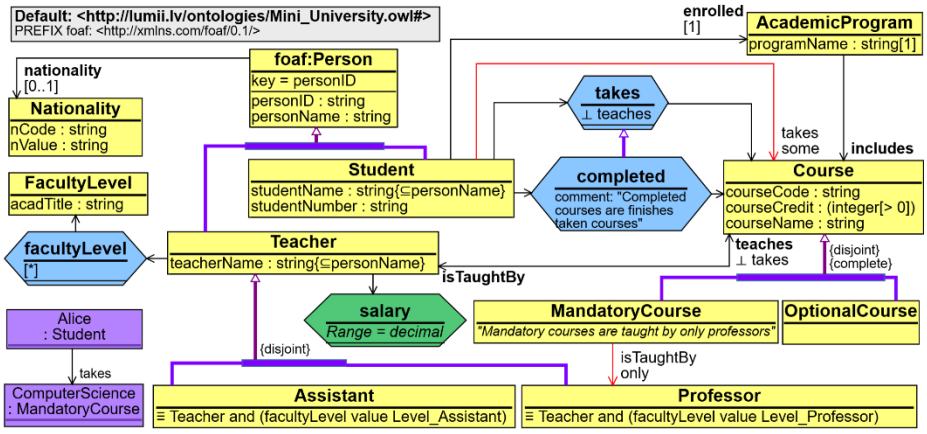

Fig. 1. Mini University ontology example

Compared with the original desktop-based OWLGrEd editor, the proposed approach, besides the technological re-work, introduces several notation improvements.

It adopts namespace prefix notation that is closer to standard RDF/OWL practices (see Fig. 1, class *foaf:Person*). In addition, it introduces a separate node notation for object properties, data properties, and property assertions, which is particularly useful when attaching annotations to properties. An example of Mini University ontology, together with its description and usage instructions, is available in the project documentation[1].

In the proposed notation, property nodes include property declaration axioms. The domain and range may be represented either graphically, using outgoing and incoming arrows that link the property node to the corresponding classes, or textually as values within the property node (see Fig. 1, *salary* data property node). Sub property relationships can be expressed either by a graphical generalization line (see Fig. 1, generalization line between *takes* and *completed* property nodes) or by a textual representation inside the property node.

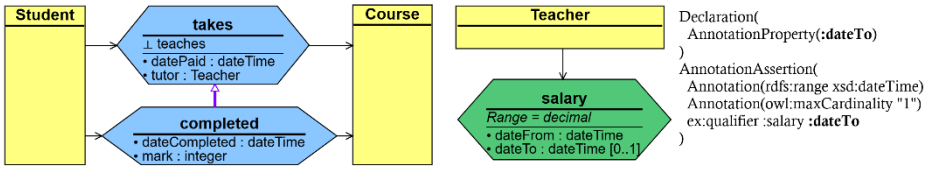

Fig. 2. Examples of object and data property nodes with qualifiers.

Property nodes can also be enriched with qualifiers [9], i.e., labelled fields for contextual information such as temporal validity, provenance, or confidence (see Fig. 2). Qualifiers are encoded using OWL 2 annotation properties. The connection between a qualifier and the corresponding property is represented by an *AnnotationAssertion* axiom with the annotation property *ex:qualifier*[2] which links a qualifier annotation property (e.g., *:dateTo*) to the qualified property (e.g., *:salary*). Additional annotations nested within the *AnnotationAssertion* may further describe the qualifier, for example by specifying its expected value type (via *rdfs:range*) or multiplicity constraints (e.g., *owl:maxCardinality*). In this way, qualifiers provide structured contextual metadata for property axioms and property assertions while preserving standard OWL semantics. Once defined, the same qualifier annotation properties, such as *:dateTo*, can be reused to annotate instance-level property assertions by attaching the corresponding *Annotation* axioms to the property assertion.

3 Implementation and Availability

The editor supports ontology creation and editing with import and export in multiple RDF serializations, including Turtle, N-Triples, N3, RDF/XML, and JSON-LD. Import ensures interoperability with ontologies produced by external tools, while export to

[1] https://github.com/LUMII-Syslab/owlgred/wiki/Mini-University-demo-project
[2] ex: <http://owlgred.lumii.lv/2011/1.1/extensions#>

standard OWL/RDF syntaxes enables reuse in ontology editors and reasoning environments. Since the notation supports alternative presentations of the same semantic elements, ontology import is accompanied by configurable visualization parameters [10].

The tool is implemented on top of Ajoo, a generic platform for developing web-based diagrammatic tools [7], with support for automatic diagram layout. Within this platform, OWLGrEd is defined through JSON-style configuration that specifies diagram elements such as nodes, edges, and compartments, complemented by JavaScript functions that implement tool-specific behavior. The platform relies on the Meteor framework[3] and MongoDB for diagram storage, exchange, and user management.

The platform architecture supports collaborative editing, allowing multiple users to work on the same project within a shared workspace. The tool is available as open-source software in a GitHub repository[4]. It can be deployed locally or accessed via an online playground environment[5]. The system also supports project upload and download, enabling users to prototype in the online environment and later migrate projects to a self-hosted installation.

4 Conclusions and Future Work

We presented a web-based OWLGrEd editor that preserves the semantics of the established OWLGrEd notation while introducing improvements such as namespace prefix notation and node style visualization options for properties and assertions. The tool supports collaborative work and interoperable ontology import/export, providing a practical basis for round-trip OWL 2 modeling in the browser.

The open-source availability of the configurable editor and notation enable further improvements through additional fields and visual effects, particularly for domain-specific information captured via OWL annotations (cf. [2]). We intend to consider the future options for adding user-friendly notations for classifier modelling in the editor.

Acknowledgments. This work was supported by activity 1.1.1.9 Research application No 1.1.1.9/LZP/1/24/037 of the Activity “Post-doctoral Research” “Visual methods and tools for ontology management”, and the Latvian Council of Science, project “What’s in Your Knowledge Graph?” (Project No. lzp-2024/1-0665).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Chávez-Feria, S., García-Castro, R., Poveda-Villalón, M.: Chowlk: From UML-Based Ontology Conceptualizations to OWL. In: Groth, P., Vidal, M.-E., Suchanek, F., Szekley, P.,

[3] Meteor Documentation: https://docs.meteor.com/
[4] OWLGrEd GitHub: https://github.com/LUMII-Syslab/owlgred
[5] OWLGrEd online playground environment: https://owlgred.semtech.lv/

Kapanipathi, P., Pesquita, C., Skaf-Molli, H., Tamper, M. (eds.) *The Semantic Web*. Springer, Cham, pp. 338–352 (2022)
2. Čerāns, K., Ovčiņņikova, J., Liepiņš, R., Grasmanis, M.: Extensible Visualizations of Ontologies in OWLGrEd. In: Hitzler, P. et al. (eds.) *The Semantic Web: ESWC 2019 Satellite Events*. Springer, Cham, pp. 191–196 (2019)
3. Bārzdiņš, J., Bārzdiņš, G., Čerāns, K., Liepiņš, R., Sproģis, A.: Uml style graphical notation and editor for owl 2. In: *P. Forbrig, H. Günther (Eds.), Perspectives in Business Informatics Research*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 102–114. (2010)
4. Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The Simple, Web-Based Tool for Visualization and Sharing of Semantic Data and Ontologies. In: *Proceedings of the ISWC 2015 Posters & Demonstrations Track*. CEUR Workshop Proceedings, vol. 1486 (2015)
5. Wiens, V., Lohmann, S., Auer, S.: WebVOWL Editor: Device-Independent Visual Ontology Modeling. In: *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks*. CEUR Workshop Proceedings, vol. 2180 (2018)
6. Lembo, D., Santarelli, V., Savo, D.F., De Giacomo, G.: GRAPHOL: A Graphical Language for Ontology Modeling Equivalent to OWL 2. *Future Internet* 14, 1–29 (2022)
7. Sprogis, A.: ajoo: WEB Based Framework for Domain Specific Modeling Tools. // In: *Frontiers of AI and Applications, Vol. 291, Databases and Information Systems IX*, IOS Press, pp. 115-126. (2016)
8. Kozlovičs, S.: Towards an Extensible Web-Based Open-Source Graphical Ontology Editing Framework. In: *Proceedings of the ISWC 2019 Posters & Demonstrations, Industry, and Outrageous Ideas Tracks (ISWC-P&D-Industry-Outrageous 2019)*. CEUR Workshop Proceedings, vol. 2456, pp. 157–160. CEUR-WS.org (2019)
9. Ovčiņņikova, J., Čerāns, K.: Adding Context in OWL Ontologies: Visual Qualifiers in OWLGrEd. In: Chaves-Fraga, D., Heibi, I., Garijo, D., Collarana, D., Salatino, A., Vahdati, S. (eds.) *Posters, Demos, Workshops, and Tutorials at SEMANTiCS 2025 (SEMANTiCS-PDWT 2025)*. CEUR Workshop Proceedings, vol. 4064. CEUR-WS.org (2025)
10. Ovčiņņikova, J., Čerāns, K.: Advanced UML Style Visualization of OWL Ontologies. In: *Proceedings of VOILA 2016*. CEUR Workshop Proceedings, vol. 1704, pp. 136–142. CEUR-WS.org (2016)