

# RDF Playground: An Online Tool for Learning about the Semantic Web

Bastián Inostroza  
DCC, Universidad de Chile  
Santiago, Chile

Raúl Cid  
DCC, Universidad de Chile  
Santiago, Chile

Aidan Hogan  
IMFD; DCC, Universidad de Chile  
Santiago, Chile

## ABSTRACT

We present RDF Playground: a web-based tool to assist those who wish to learn or teach about the Semantic Web. The tool integrates functionalities relating to the key features of RDF, allowing users to specify an RDF graph in Turtle syntax, visualise it as an interactive graph, query it using SPARQL, reason over it using OWL 2 RL, and to validate it using SHACL or ShEx. The tool further provides the ability to import and explore data from the Web through a graph-based Linked Data browser. We discuss the design and functionality of the tool, its implementation, and the results of a usability study considering students from a Web of Data course that used it for lab assignments. We conclude with a discussion of these results, as well as future directions that we envisage for improving the tool.

## CCS CONCEPTS

- **Information systems** → **Semantic web description languages**;
- **Applied computing** → **Interactive learning environments**.

## KEYWORDS

Semantic Web, education, RDF, RDFS, OWL, SPARQL, SHACL, ShEx

### ACM Reference Format:

Bastián Inostroza, Raúl Cid, and Aidan Hogan. 2023. RDF Playground: An Online Tool for Learning about the Semantic Web. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion)*, April 30-May 4, 2023, Austin, TX, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3543873.3587325>

## 1 INTRODUCTION

It can be intimidating for newcomers to learn about the Semantic Web [13]. The standards themselves – RDF [7], RDFS [4], OWL [11], SPARQL [9], SHACL [16], etc. – focus on definitions. While there are primers and textbooks available [1, 10, 12, 14, 17], it can be difficult to find systems where one can try these standards out. Although there are individual online and offline systems that implement or demonstrate specific standards, there are few systems that support various standards in an integrated way [8].

In this paper, we present *RDF Playground*: an online system for teaching and learning about the Semantic Web in a hands-on manner. The goal of the system is to provide an integrated interface that allows users to run small-scale illustrative examples of key

Semantic Web standards. The system is designed to make an RDF graph the central element of the interface. The user can then select from a number of tabs where they can apply RDFS or OWL 2 RL reasoning, run SPARQL queries, and/or apply SHACL/ShEX validation over the RDF graph. The user can further use a graph-based Linked Data browser in order to import RDF data from the Web.

In Section 2, we first discuss and compare related systems with RDF Playground. Section 3 describes the design, layout and functionality of the user interface. Section 4 discusses the implementation of the back-end, including the reasoning, querying and validation engines. Section 5 presents the results of a usability evaluation conducted within a “Web of Data” course. Section 6 concludes with a discussion of results, limitations, and future work.

## 2 RELATED WORK

In Table 1, we summarise the systems available online that allow for testing out small examples involving the main Semantic Web standards, and could thus be useful in the context of teaching and/or learning about the Semantic Web. We include systems that were online at the time of writing, and provide an interactive system online (i.e., do not need to be installed locally); below the table we provide links. We remark that not all such systems are designed for educational purposes, but could be used in such settings. In the **Graph** column we include systems that support a graph visualisation of RDF graphs. The **LD** column refers to systems that allow for exploring Linked Data through dereferencing.

We categorise the systems as follows. **Graph visualisations** (EasyRDF, LodLive, LODmilla, and WebVOWL) enable viewing RDF data and/or OWL ontologies via a graph-based visualisation. **Linked Data browsers** (LodLive, LODmilla, Lodview, and Q&D RDF Browser) allow users to browse Linked Data through either an entity-centric or graph-centric interface. **Query interfaces** (SPARQLer, SPARQL Playground, and YASGUI) focus on offering users the ability to query the dataset of their choice via SPARQL. **Ontology editors** (WebProtégé and WebVOWL) allow for browsing, visualising and editing RDFS/OWL ontologies. **Reasoning interfaces** (OWL RL reasoner and WebProtégé) provide reasoning services over RDFS/OWL ontologies and datasets. **Validation interfaces** (SHACL playground) allow for testing SHACL/ShEx validation. **SW playgrounds** (RDFShape and RDF Playground) support a range of the aforementioned features and standards.

The goal of RDF Playground is then to enable users to try small-scale interactive examples for the key Semantic Web standards in a single online system. Per Table 1, the closest system to the one we propose is RDFShape [8], which supports a similar set of features, and which served as inspiration for RDF Playground. However, there are a number of key differences. Primarily, RDFShape provides distinct interfaces for RDF data, for generating a graph, for

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9419-2/23/04.

<https://doi.org/10.1145/3543873.3587325>

**Table 1: Online systems annotated with the Semantic Web standards and features they support**

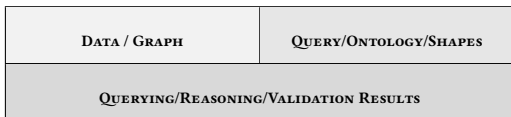
System	Graph	RDFS	OWL	SPARQL	SHACL	ShEX	LD
EasyRDF <sup>a</sup>	✓	✓					
LodLive [6] <sup>b</sup>	✓						✓
LODmilla [18] <sup>c</sup>	✓						✓
LodView <sup>d</sup>	✓						✓
OWL RL reasoner <sup>e</sup>		✓	~				
Q&D RDF Browser <sup>f</sup>							
RDF Explorer [21] <sup>g</sup>							
RDFShape [8] <sup>h</sup>	✓	✓	~	✓	✓	✓	
SHACL Playground <sup>i</sup>					✓		
SPARQLer <sup>j</sup>				✓			
SPARQL Playground [3] <sup>k</sup>	~			✓			
WebProtégé [15, 20] <sup>l</sup>		✓	✓				
WebVOWL [23] <sup>m</sup>	✓	✓	~				
YASGUI [19] <sup>n</sup>				✓			~
RDF Playground <sup>o</sup>	✓	✓	~	✓	✓	✓	✓

<sup>a</sup> <http://www.easyrdf.org/>  
<sup>b</sup> <http://en.lodlive.it/>  
<sup>c</sup> <http://lodmilla.sztaki.hu/lodmilla/>  
<sup>d</sup> <https://lodview.it/>  
<sup>e</sup> <http://www.ldf.fi/service/owl-rl-reasoner>  
<sup>f</sup> <http://graphite.ecs.soton.ac.uk/browser/>  
<sup>g</sup> <http://www.rdfexplorer.org/>  
<sup>h</sup> <http://rdfshape.weso.es/>  
<sup>i</sup> <https://shacl.org/playground/>  
<sup>j</sup> <http://sparql.org>  
<sup>k</sup> <http://sparql-playground.sib.swiss/>  
<sup>l</sup> <http://webprotege.stanford.edu/>  
<sup>m</sup> <http://service.tib.eu/webvowl/>  
<sup>n</sup> <http://yasgui.triply.cc/>  
<sup>o</sup> <http://rdfplayground.dcc.uchile.cl/>

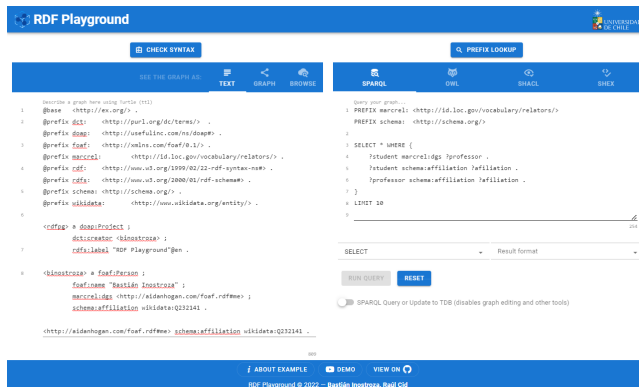
reasoning, for querying, for validation. Each distinct interface has its own separate data entry. Conversely, in the design of RDF Playground, we wanted to make a unified interface centred around a single RDF graph, with tabs available to query, reason over and validate that graph. Furthermore, we wanted to make the graph visualisation a central element of the interface, and to add a Linked Data browser in order to navigate RDF data on the Web.

### 3 USER INTERFACE

The user interface of RDF playground is implemented with Vue and VisJS, and is based on three main panes with the following layout:



The top two side-by-side panes are used for input. The top-left pane refers to the input data: the RDF graph. The top-right pane refers to the input query, ontology or shape definition that applies over the data; the user can switch between options by selecting one of the icons on top of the pane. Putting these two panes side-by-side enables users to, for example, view the data as they are writing the query over it, easily copy and paste elements between both panes, etc. Once a query, reasoning or validation process is executed, the user can scroll down to see the results. In Figure 1, we provide a screenshot of the RDF Playground user interface showing the top two panes. On the left we see an RDF graph in Turtle syntax,

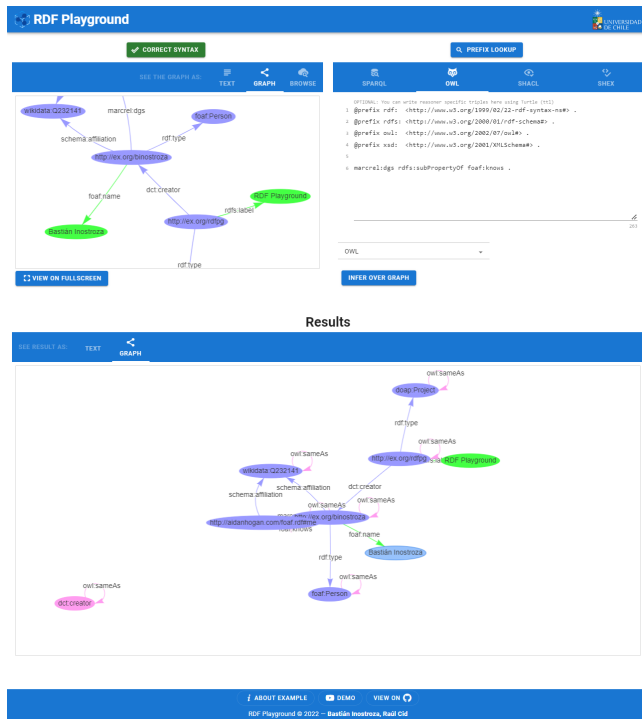


**Figure 1: The RDFPlayground User Interface with RDF data in Turtle syntax (left) and a SPARQL query (right)**

while on the right we see a SPARQL query that we can evaluate against this RDF graph. At the top of the panes, we can see the icons to switch between the different features. In case the screen being used is not wide enough to fit both panes side-by-side (e.g., the user is on a smartphone), the interface reflows in order to put the three panes in one full-width column, with data/graph on top, query/ontology/validation in the middle, and results at the bottom.

A key design goal of RDF Playground is to make graphs a central element of the interface, thus emphasising to newcomers the graphical nature of the RDF data model. Viewing the graph can also reveal errors that commonly occur when students edit Turtle by hand; for example, while the difference between the IRIs `dbr:Dwarf_Star` and `dbr:Dwarf_star` can be difficult to spot in a text-based serialisation, the graph visualisation will show two noticeably distinct nodes. Visualisations are not only applicable for the input RDF graph, but also, where applicable, for output RDF graphs. For example, in case the user runs a CONSTRUCT or DESCRIBE SPARQL query, or executes reasoning with respect to RDFS or OWL definitions, the result can be viewed in Turtle or graphical format. In Figure 2, we provide an example of how an input RDF graph and an output RDF graph – resulting from reasoning – can be visualised. The visualisation is fully interactive, where users can zoom in, zoom out, pan, highlight nodes and their edges by clicking on them, reveal more details (e.g., datatypes of literals) by hovering over them, move nodes by clicking and dragging, etc. Different types of RDF terms are distinguished with different colours, and prefixes are used to provide more concise node names. In the case of the output graph, we further highlight novel materialised edges and nodes with a distinctive colour, indicating which edges are part of the input, and which are generated through inference by the reasoner.

The final key feature is a Linked Data browser. The top-left pane includes a BROWSE option and associated icon. This interaction requests a Linked Data IRI, which is dereferenced. The RDF content is retrieved from the indicated location, parsed, and visualised as a graph. However, initial experiences with this interface created some complications, where certain Linked Datasets – particularly Wikidata [22] – provide very cluttered RDF graphs with tens or hundreds of thousands of triples upon dereferencing, often making the visualisation unresponsive. We thus shifted towards interactive



**Figure 2: Graph visualisations of data and reasoning results with an RDF graph (top left), an OWL axiom (top right), and an RDF graph highlighting inferred triples (bottom)**

browsing, similar to LodLive [6]. Browsing begins from the principal resource described by the dereferenced document, defined to be the most common subject term in the dereferenced RDF graph. We provide a left-hand pane that introduces two key facets: documents and properties. Expanding a document, the user can select properties in order to only display triples/edges with the corresponding predicates. The user can double click on any IRI node and request to dereference it. If the node is a valid Linked Data IRI – i.e., if it successfully dereferences to an RDF document – the document is added to the facets. We provide a screenshot of the Linked Data browser in Figure 3 with a graph formed from two Geonames documents. The left-hand pane contains the facets that select what to show. Data from this graph can be imported back into the local querying, reasoning and validation features of RDF Playground.

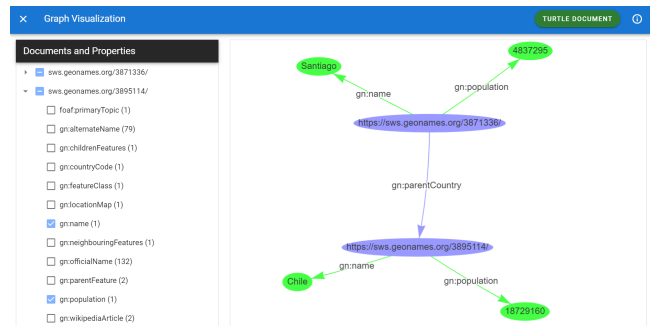
#### 4 BACK-END

We implemented a back-end in Kotlin that coordinates tasks with Jena<sup>1</sup> (for handling RDF parsing, SPARQL querying, SHACL validation and Linked Data dereferencing), the OWL RL reasoner<sup>2</sup> (for handling RDFS and OWL reasoning), and ShExJava<sup>3</sup> (for ShEX validation). The back-end, Jena and ShExJava operate in the JVM, while coordination with the Python-based OWL RL reasoner is via the command line. The back-end further implements methods

<sup>1</sup><https://jena.apache.org/>

<sup>2</sup><https://github.com/RDFLib/OWL-RL>

<sup>3</sup><http://shexjava.lille.inria.fr/>



**Figure 3: Incremental, graph-based Linked Data browser with facets (left) and visual RDF graph (right)**

to convert from RDF graphs to the .dot graph format needed for visualisation in the front-end. A RESTful API is implemented using Spring Boot that the front-end can invoke.

#### 5 EVALUATION

We conducted a preliminary usability study with students of the postgraduate Web of Data course in the Department of Computer Science, University of Chile, whose requirements inspired the creation of RDF Playground. Students were given various lab assignments to complete using RDF Playground, spanning several weeks. A summary of the weekly labs involved was as follows:

- Given a paragraph from Wikipedia (in natural language), create an RDF graph to represent as accurately as possible the claims expressed in the paragraph [1 lab].
- Given an input RDF graph, provide axioms in RDFS [1 lab] and OWL [2 labs] to derive specified entailments.
- Given an input RDF graph, provide SPARQL queries to return specified answers [2 labs].
- Given an input RDF graph, provide SHACL constraints in order to detect specified errors in the graph [1 lab].
- Create a FOAF profile in RDF, publish it on the Web linking to classmates, and use the Linked Data browser to map out the decentralised social network formed by the class [1 lab].

Each of the five items listed here corresponds to a different feature of RDF Playground: RDF, RDFS/OWL, SPARQL, SHACL and Linked Data browsing. We asked students to (voluntarily) respond to two Standard Usability Survey (SUS) questionnaires [5]: one after the lab corresponding to the graph generation, and the second after the labs corresponding to RDFS/OWL reasoning labs. We received 11 responses for the first questionnaire, and 10 responses for the second questionnaire. Students taking the course had no prior familiarity with the Semantic Web, and varied from third year undergraduates with intermediate programming skills (from engineering disciplines other than Computer Science) to doctoral students of Computer Science. We show the mean and standard deviation results in Table 2; scores vary from 1 to 5. By convention, an overall SUS score greater than 68 is considered a better-than-average score. In the first questionnaire, the results are close to this score, reaching 67.5. In the second questionnaire, the results drop, which we attribute to the more technical nature of the reasoning lab. In terms of aspects to improve, students noted some minor issues

**Table 2: SUS evaluation results for two questionnaires ( $m$  = mean;  $s$  = std. deviation)**

Claim to evaluate	Q1		Q2	
	$m$	$s$	$m$	$s$
I think that I would like to use this system frequently.	4.0	0.78	3.8	0.42
I found the system unnecessarily complex	1.6	0.50	1.7	0.48
I thought the system was easy to use.	4.6	0.52	4.6	0.51
I think that I would need the support of a technical person to be able to use this system.	2.2	0.60	2.7	1.25
I found the various functions in this system were well integrated.	3.9	0.83	4.1	0.74
I thought there was too much inconsistency in this system.	1.6	0.67	1.8	0.92
I would imagine that most people would learn to use this system very quickly.	4.2	0.90	4.3	0.95
I found the system very cumbersome to use.	1.7	0.65	1.7	0.67
I felt very confident using the system.	4.1	0.54	3.4	1.17
I needed to learn a lot of things before I could get going with this system.	1.5	0.52	1.5	0.52
<b>SUS Points:</b>	67.5	7.24	64.5	9.34

that we later resolved, such as visual bugs. However, they also noted some more general issues, such as a lack of syntax highlighting, large graphs being cluttered, etc.

## 6 CONCLUSIONS

We have presented RDF Playground: an online teaching tool that combines a number of Semantic Web standards and functionalities into a unified user interface. We have been successfully using this tool for laboratories within a Web of Data course. We likewise think the tool will be of use to others who wish to teach or learn about the Semantic Web. In terms of future improvements, we would like to create a Docker version of RDF Playground so that it can be more easily deployed within universities that need it. We would also like to add some “quality of life” improvements, such as adding syntax highlighting, automatic detection of the SPARQL query type, support for syntaxes other than Turtle, integration with more advanced editors for RDF and SPARQL, richer examples, a preview of the SPARQL algebra, etc. We would also like to support OWL 2 DL reasoning with the OWL API. With such improvements, it would also be of interest to conduct further usability studies, perhaps in other courses, following rubrics for e-learning tools [2].

*Supplemental material.* An RDF Playground demo is available at <http://rdfplayground.dcc.uchile.cl/>. Source code is available at <https://github.com/BastyZ/RDFPlayground>.

## ACKNOWLEDGMENTS

This work was supported by Fondecyt Grant No. 1221926 and ANID – Millennium Science Initiative Program – Code ICN17\_002. We thank the students who answered our questionnaire anonymously.

## REFERENCES

- [1] Dean Allemang and James A. Hendler. 2008. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann/Elsevier.
- [2] Lauren Anstey and Gavan Watson. 2018. A rubric for evaluating e-learning tools in higher education. *Educause Review* (2018).
- [3] Paolo Bottoni and Miguel Ceriani. 2015. SPARQL Playground: A Block Programming Tool to Experiment with SPARQL. In *Proceedings of the International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pennsylvania, USA, October 11, 2015 (CEUR Workshop Proceedings, Vol. 1456)*. CEUR-WS.org, 103. <http://ceur-ws.org/Vol-1456/paper12.pdf>
- [4] Dan Brickley, R.V. Guha, and Brian McBride. 2014. RDF Schema 1.1. W3C Recommendation. <https://www.w3.org/TR/rdf-schema/>.
- [5] John Brooke. 1996. SUS – A quick and dirty usability scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.
- [6] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. 2012. LodLive, exploring the web of data. In *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*. ACM, 197–200. <https://doi.org/10.1145/2362499.2362532>
- [7] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. <https://www.w3.org/TR/rdf11-concepts/>.
- [8] José Emilio Labra Gayo, Daniel Fernández-Álvarez, and Herminio García-González. 2018. RDFShape: An RDF Playground Based on Shapes. In *ISWC 2018 Posters & Demonstrations (CEUR Workshop Proceedings, Vol. 2180)*. CEUR-WS.org.
- [9] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. 2013. SPARQL 1.1 Query Language. W3C Recommendation. <https://www.w3.org/TR/sparql11-query/>.
- [10] Tom Heath and Christian Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space (1st Edition)*. Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 1. Morgan & Claypool. 1–136 pages. Issue 1. Available from <http://linkeddatabook.com/editions/1.0/>.
- [11] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. 2012. OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation. <https://www.w3.org/TR/owl2-primer/>.
- [12] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. 2010. *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC Press. <http://www.semantic-web-book.org/>
- [13] Aidan Hogan. 2020. The Semantic Web: Two decades on. *Semantic Web* 11, 1 (2020), 169–185. <https://doi.org/10.3233/SW-190387>
- [14] Aidan Hogan. 2020. *The Web of Data*. Springer. <https://doi.org/10.1007/978-3-030-51580-5>
- [15] Matthew Horridge, Rafael S. Gonçalves, Csongor I. Nyulas, Tania Tudorache, and Mark A. Musen. 2019. WebProtégé: A Cloud-Based Ontology Editor. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 686–689. <https://doi.org/10.1145/3308560.3317707>
- [16] Holger Knublauch and Dimitris Kontokostas. 2017. Shapes Constraint Language (SHACL). W3C Recommendation. <https://www.w3.org/TR/shacl/>.
- [17] José Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, and Dimitris Kontokostas. 2017. *Validating RDF Data*. Morgan & Claypool Publishers. <https://doi.org/10.2200/S00786ED1V01Y201707WBE016>
- [18] András Micsik, Sándor Turbucz, and Zoltán Tóth. 2015. Exploring publication metadata graphs with the LODmilla browser and !editor. *Int. J. Digit. Libr.* 16, 1 (2015), 15–24. <https://doi.org/10.1007/s00799-014-0130-2>
- [19] Laurens Rietveld and Rinke Hoekstra. 2017. The YASGUI family of SPARQL clients. *Semantic Web* 8, 3 (2017), 373–383. <https://doi.org/10.3233/SW-150197>
- [20] Tania Tudorache, Csongor Nyulas, Natalya Fridman Noy, and Mark A. Musen. 2013. WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the Web. *Semantic Web* 4, 1 (2013), 89–99. <https://doi.org/10.3233/SW-2012-0057>
- [21] Hernán Vargas, Carlos Buil Aranda, and Aidan Hogan. 2019. RDF Explorer: A Visual Query Builder for Semantic Web Knowledge Graphs. In *ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) (CEUR Workshop Proceedings, Vol. 2456)*. CEUR-WS.org, 229–232. <http://ceur-ws.org/Vol-2456/paper60.pdf>
- [22] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. <https://doi.org/10.1145/2629489>
- [23] Vitalis Wiens, Steffen Lohmann, and Sören Auer. 2018. WebVOWL Editor: Device-Independent Visual Ontology Modeling. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018 (CEUR Workshop Proceedings, Vol. 2180)*. CEUR-WS.org. <http://ceur-ws.org/Vol-2180/paper-75.pdf>

Received 3 February 2023