Visualizing Ontology Metrics In The NEOntometrics Application

Achim Reiz[a] and Kurt Sandkuhl[a]

[a] *Rostock University, 18051 Rostock, Germany*

Abstract
Metrics offer an objective assessment of the inner fabrics of ontologies. They allow us to quickly understand graph properties, logical complexity, completeness of human-centered annotations, or degree of interconnection. When analyzed historically, ontology metrics tell much about development decisions and the impact of changes and can be used for quality control measures. The NEOntometrics software allows calculating evolutional ontology metrics for git-based repositories and implements the majority of literature-proposed metric frameworks. This paper presents the recently added visualization capabilities for visualizing the differences between ontologies in a repository, assessing the change impact of the most recent commit, and examining ontology evolution.

Keywords
NEOntometrics, Ontology Metrics, Knowledge Graph, Quality Management

1. Introduction

Ontologies underpin the technology of applications such as question-answering systems, recommendation systems [1], or autonomous driving [2]. They build on subsets of first-order logic, which has little in common with traditional data modeling or imperative programming. The development team is often dispersed skill-wise, experience-wise, in their responsibilities, and geographics [3]. The complexity and collaborative development processes put quality control activities at the forefront. Users and ontology managers need to understand the impact of proposed changes and the evolutional history of the artifacts as a whole to make informed development decisions.

One way to gather these kinds of information is the calculation of metrics. Ontology metrics translate the structural attributes of ontologies into objective, reproducible measurements. The measures cover the use of RDF(S) and OWL formalisms, graph properties, or human-centered annotations.

With NEOntometrics [4], a tool is available for analyzing large amounts of historical metric data based on git repositories. NEOntometrics implements most proposed ontology metrics and allows *csv* data export or integration into custom applications and analyses using a GraphQL interface. Until recently, however, the frontend visualization capabilities were limited to a tabular metric representation.

This paper presents a new integrated visualization feature for NEOntometrics. The evolutional developments and the differences of ontologies in a repository can now be displayed using diagrams. A comparison view for the last two versions of an artifact visualizes changed measures, thus allowing a quick overview of impacted structural attributes. Integrating diagram capabilities into software for calculating ontology metrics eases the consumption and productive use of the measures and hopefully contributes to the broader dissemination of ontology metrics for quality control.

The paper is structured as follows: The next section recapitulates the state of the art regarding ontology evolution and visualization approaches. Section three presents NEOntometrics, including the newly implemented visualization features. The research concludes with an outlook and a discussion.

VOILA! 2023: 8th International Workshop on Visualization and Interaction for Ontologies and Linked Data, November 6, 2023, Athens, Greece, Co-located with ISWC 2023
EMAIL: achim.reiz@uni-rostock.de (A. 1); kurt.sandkuhl@uni-rostock.de (A. 2)
ORCID: 0000-0003-1446-9670 (A. 1); 0000-0002-7431-8412 (A. 2)
© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3508/paper8.pdf

2. Related Work

In this related work, we aim to recapitulate the current state regarding software support for ontology evolution and evolution visualization. [4] further contains a review of software support for ontology metric calculation. Hence, it is not thoroughly discussed in this paper.

Analyzing and visualizing ontology evolution is a multifaceted field with various manifold approaches. As such, there are already extensive literature reviews that collected the relevant state of the art:

Novais et al. [5] collected an extensive review of visualization approaches in the related field of software evolution and classified the existing research (among others) along their application scenarios, visual paradigms and attributes, data sources, and explanation strategy.

Lambrix et al. [6] collected software for ontology and software evolution and categorized them along their functionality, which is further assigned to evolution steps and change categories. De Leenheer and Mens [7] presented processes and tool support for single-developer and collaborative ontology development processes. They focus on the challenges of inter-organizational changes with multiple possible ontology managers and the corresponding organizational requirements. These authors also present their own collaborative ontology engineering process DOGMA, including tool support [8]. Tudorache [9] recapitulated the recent advances in ontology engineering, including software support. While she argues that the tool support is nowadays much better suited not only for research but also for commercial projects, she also argues for making access easier for newcomers and for increasing the usability of the tools.

Table 1
Software for visualizing or calculating ontology evolution

#	Software	Type	Use case	Available	Open Source
[10]	ReX	Standalone	Identify and visualize unstable ontology regions	✗	✗
[11,12]	CODEX, ContoDiff Explorer	Standalone, API	Understand and visualize ontology changes and their impacts	✗	☑ [2]
[13]	Ecco	Shell	Detect changes between two ontology versions	✗	☑ [3]
[14]	PromptDiff	Protégé Plugin	Comparing Structural changes between two ontology versions	✗	✗
[15]	OWLDiff	Standalone, Protégé Plugin	Compare and merge two OWL2 ontologies	☑	☑ [4]
[16]	ChIMP	Protégé Plugin	Dashboard view of evolved ontology metrics	☑	☑ [5]
[8]	Dogma Studio	Standalone	Manage ontology evolution in collaborative environments	✗	✗
[17]	Live Diff Taxonomy	Protégé Plugin	Summarizes the changes in taxonomical view	☑	(☑) [6]

Table 1 presents tools for detecting and visualizing ontology evolution. Many activities have been at the automatic detection of differences between two ontologies with implementations in [12–15]. These tools detect the differences between two ontology versions and return the changes mainly as a list. Graphics-wise, [11] offers a visualization of the ContoDiff algorithm, [10] allows visualization of

[2] https://github.com/dbs-leipzig/conto_diff – Only the underlying algorithm of CODEX is open source
[3] https://github.com/rsgoncalves/ecco/
[4] https://github.com/psiotwo/owldiff
[5] https://gitlab.ifi.uzh.ch/DDIS-Public/chimp-protege-plugin
[6] https://github.com/NJITSABOC/oaf-protégé – no license attached

unstable ontology subparts. To instantly understand the impact of modeling decisions in the Protégé editor, [17] calculates a taxonomy of changes, and [16] calculates numerical differences between the saved version and currently performed changes.

While the visualization view of the new NEOntometrics features (cf. Figure 4) has some similarities to [16], our approach does not calculate the metrics instantly for a change but regards the last two committed versions in a git repository. NEOntometrics also implemented a more significant number of ontology metrics, allowing us to choose from ~160 metrics from different frameworks. Compared to the current state of the art regarding tool-supported ontology evolution, the NEOntometrics approach is strictly focused on visualizing ontology metrics. Further, it not only regards two versions but calculates and visualizes the overall version history, thus allowing conclusions on overall evolutional processes and design decisions.

3. NEOntometrics

NEOntometrics is a web-based application that calculates evolutional ontology metrics of git-based repositories. It iterates through a repository and calculates the respective measures for every ontology in every commit if the file has changed. It comes with an interactive help page *Metric Explorer*, is open source[7], and is available online[8]. For more information on NEOntometrics, cf. [4,18].

As calculating the version history of ontology repositories can take a considerable amount of time, NEOntometrics works asynchronously. A new calculation can be triggered by pasting the URL to a given repository in the bottom text field of the *Calculation Engine* (cf. Figure 1). If the repository is not yet known to the system, it can be put in the calculation queue. Afterward, a separate worker application retrieves the ontology from the git repository and starts the analysis.

The ontology metrics can be retrieved as soon as the calculation is finished. Currently, NEOntometrics supports OQuaRE [19], OntoQA [20], oQual [21], the cohesion metrics by Yao et al. [22], the good ontology metrics by Fernández et al. [23], Orme et al.'s evolutional metrics [24], and the complexity metrics by Yang et al. [25]. Metrics can be run on the ontology as it is or on the inferred graph. However, as the inference engine can take a considerable time, it is only advised for small ontologies.

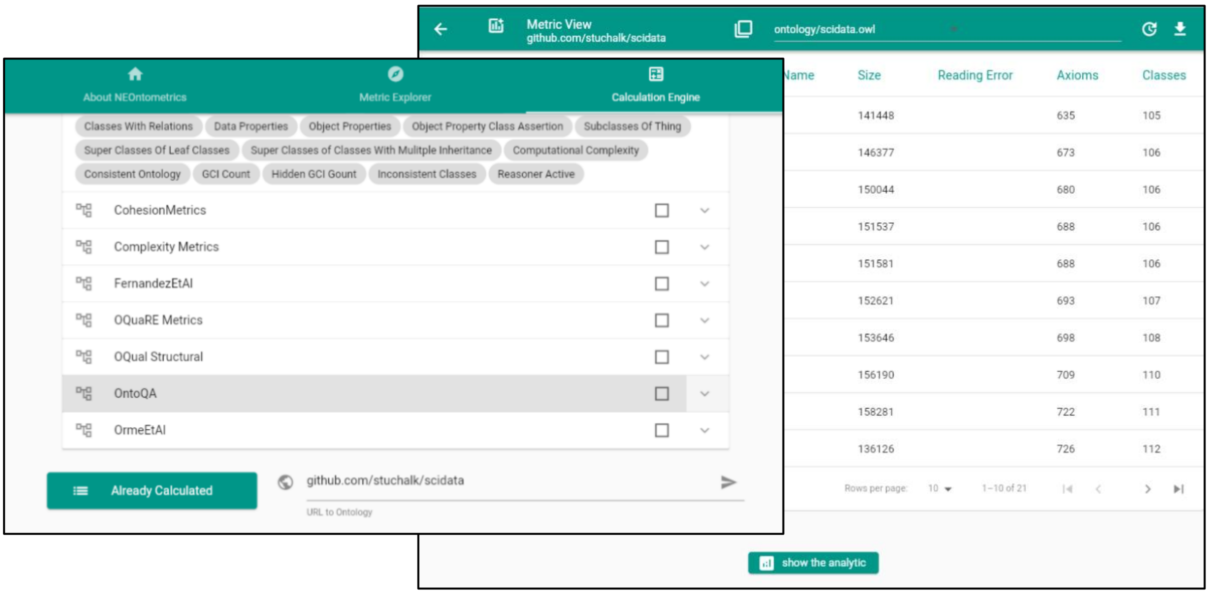

Figure 1: The entry point of NEOntometrics for the metric calculation. The new visualization feature is called using the "show the analytic" button.

[7] https://github.com/achiminator/NEOntometrics
[8] http://neontometrics.com

After selecting the desired metrics and frameworks and triggering the retrieval process, the software shows a paginated tabular view of the ontology metrics. The button *"show the analytic"* opens the visualization page. The measures shown in the diagrams are always congruent to the ones selected during the ontology retrieval process.

Three visualizations are available to examine the most recent changes, compare ontologies in a repository, and visualize the ontology evolution. The figures below exemplify these visualizations for the SciData ontology[9] for interoperable scientific data exchange [26].

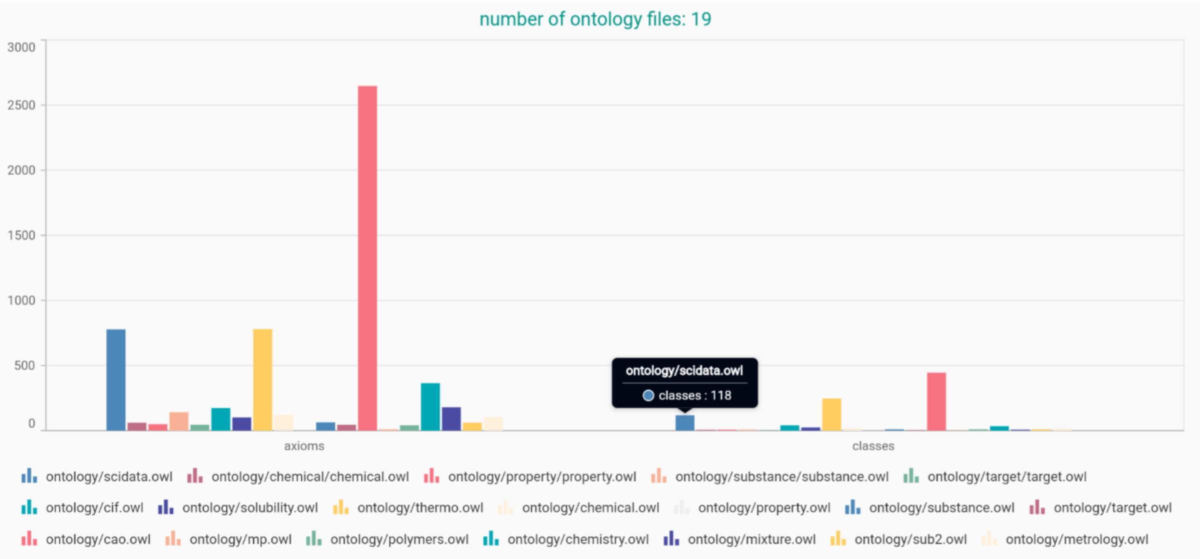

Figure 2: Bar chart for comparing the ontologies in the repository in their recent version for the selected ontology metrics.

The first chart (cf. Figure 2) displays differences between the repositories' ontologies in a bar diagram. The bar diagram presents the last version of each ontology. It allows a quick comparison of the available artifacts in a repository. The diagram can be scrolled by dragging the picture to either side of the frame. Hovering over a measure shows the detailed measures. Ontologies can be selected and deselected by clicking on their name below the chart. The scaling changes dynamically depending on the sizes of the bars visible in the frame.

The bar chart in the example visualizes the *axioms* and *classes* of the repositories' ontologies. The *thermo*, *scidata*, and *cao* files have the most classes and axioms. However, *cao* has many more axioms than *thermo*. Thus, one can conclude that the *thermo* ontology is more driven by a taxonomical class structure, while *scidata* and *cao* have more additional axioms than *thermo*.

Further examination revealed that *cao* and *scidata* are indeed more logically interconnected. In the given example, the bar chart visualizes discrete count-based measures. However, the bar chart visualization works also for ratio-based formulas. For example, the OntoQA metrics shown in Table 2 and visualized in the line chart in Figure 3 could also be used in the bar diagram.

The second visualization (cf. Figure 3) contains an evolutional view of one ontology for the selected ontology metrics. It allows an understanding of the impact of changes on the structural attributes and can be used to understand and evaluate design decisions. The drop-down menu at the top of the window changes the ontology to be visualized. The user can activate and deactivate measures by clicking their name on the legend below the chart, and the diagram scales automatically, similar to the bar chart. Hovering over a data point gives further details on the displayed value.

The diagram in Figure 3 shows four OntoQA measures for the *scidata* ontology. At first, it is evident that relationship diversity is constantly at 0, originating from the fact that no object properties are declared on classes. The class utilization and average population increase early in the lifetime of the ontology. Otherwise, they stay reasonably consistent. No individuals nor classes were added in this ontology, and the changes in classes and the subclass structure are relatively subtle. On the opposite,

[9] https://github.com/stuchalk/scidata - The authors of SciData and this paper are not affiliated with each other.

the attribute richness increases heavily. As the classes are somewhat consistent, these changes are driven by data and object properties. Also, the ontology describes a relatively high number of data and object properties compared to declared classes.

Table 2
The OntoQA measures visualized in Figure 3

Measure	Calculation
Attribute Richness	$\frac{dataProperties + objectProperties}{classes}$
Average Population	$\frac{individuals}{classes}$
Relationship Diversity	$\frac{objectPropertiesOnClasses}{subClassOfAxioms}$
Class Utilization	$\frac{classesWithIndividuals}{classes}$

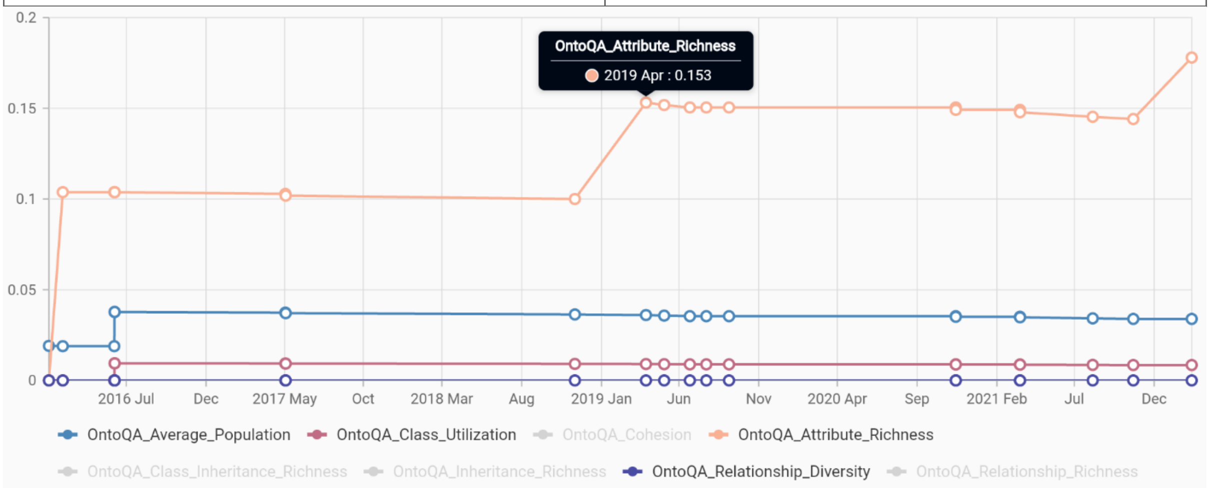

Figure 3: Line chart for one ontology showing the complete evolutional process.

The last analysis shows which measures have changed and by how much in the recent commit. It thus provides a detailed view of the ontologies' last two versions and shall allow a quick assessment of the impact of the most recent change to evaluate the usefulness and identify eventually unintended side-effects. Similar to the previous analysis, the ontology can be selected in the drop-down menu at the top of the application. In contrast to the other visualizations, this view shows not only the selected metrics but every metric that has changed. The little icon on the left indicates an increase (↑) or decrease (↓). If a metric does not change, it is not on the list.

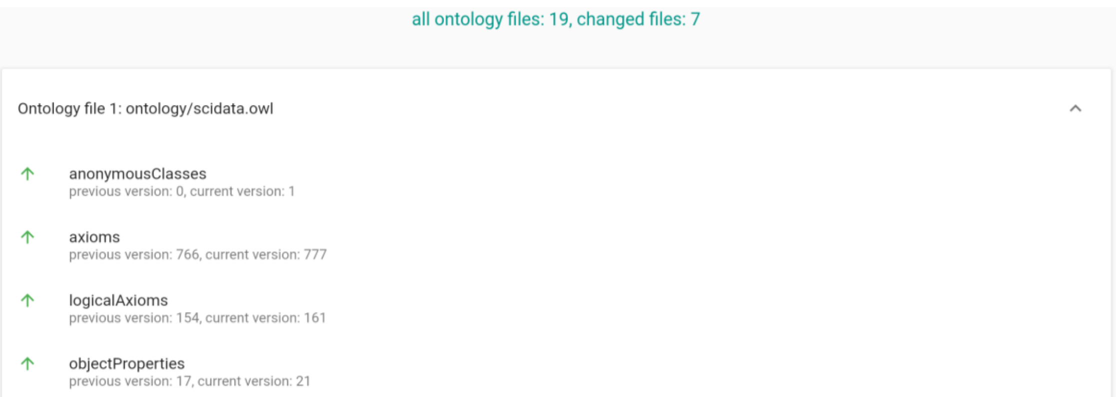

Figure 4: Summary outlining the changes between the last two versions for one ontology

4. Conclusion

The evolutional analysis of ontology metrics offers an objective insight into the evolvement of their structural attributes and can tell much about underlying design decisions. With the software NEOntometrics, tool support is available for analyzing large quantities of ontologies. The new visualization capabilities presented in this paper extend the application for an easily consumable human-oriented metric interface. We hope they ease the consumption of ontology metrics and contribute to a broader dissemination of metrics for quality control.

Application for the ontology metrics are manifold. For example, the metrics allow the user to understand evolutional processes or differences between various ontologies. Ontology metrics can aid in making better-informed reusing and development decisions as they allow quickly grasping an ontology's inner fabrics. One can set objective and reproducible goals for ontology developments and track their achievement through metrics.

Soon, we plan on integrating even more visualization capabilities to allow a better deep dive into the ontology development processes. Further on the NEOntometrics roadmap is integrating more ontology metrics, e.g., to assess SHACL-constructs and use of custom vocabularies. In that sense, we are interested in the functionalities that the community would like to see implemented. A potential evaluation of the usefulness of the visualizations and metrics is also desirable in future work.

5. References

[1] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, Knowledge Graphs: Opportunities and Challenges, Artif. Intell. Rev. (2023) 1–32. https://doi.org/10.1007/s10462-023-10465-9.

[2] R. Wickramarachchi, C. Henson, A. Sheth, Knowledge-infused Learning for Entity Prediction in Driving Scenes, Front. Big Data 4 (2021) 759110. https://doi.org/10.3389/fdata.2021.759110.

[3] E. Simperl, M. Luczak-Rösch, Collaborative ontology engineering: a survey, The Knowledge Engineering Review 29 (2014) 101–131. https://doi.org/10.1017/S0269888913000192.

[4] A. Reiz, K. Sandkuhl, NEOntometrics: A Flexible and Scalable Software for Calculating Ontology Metrics, in: Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems co-located with 18th International Conference on Semantic Systems (SEMANTiCS 2022), CEUR-WS, Vienna, 2022.

[5] R.L. Novais, A. Torres, T.S. Mendes, M. Mendonça, N. Zazworka, Software evolution visualization: A systematic mapping study, Information and Software Technology 55 (2013) 1860–1883. https://doi.org/10.1016/j.infsof.2013.05.008.

[6] P. Lambrix, Z. Dragisic, V. Ivanova, C. Anslow, Visualization for Ontology Evolution, in: CEUR Workshop Proceedings, Kobe, Japan, 2016.

[7] P. de Leenheer, T. Mens, Ontology Evolution, in: M. Hepp, P. Leenheer, A. Moor, Y. Sure (Eds.), Ontology Management, Springer US, Boston, MA, 2008, pp. 131–176.

[8] P. de Leenheer, C. Debruyne, DOGMA-MESS: A Tool for Fact-Oriented Collaborative Ontology Evolution, in: On the Move to Meaningful Internet Systems: OTM 2008 Workshops, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 797–806.

[9] T. Tudorache, Ontology engineering: Current state, challenges, and future directions, Semant. Web 11 (2020) 125–138. https://doi.org/10.3233/SW-190382.

[10] V. Christen, M. Hartung, A. Groß, Region Evolution eXplorer - A tool for discovering evolution trends in ontology regions, J. Biomed. Semantics 6 (2015) 26. https://doi.org/10.1186/s13326-015-0020-6.

[11] M. Hartung, A. Gross, E. Rahm, CODEX: exploration of semantic changes between ontology versions, Bioinformatics 28 (2012) 895–896. https://doi.org/10.1093/bioinformatics/bts029.

[12] M. Hartung, A. Groß, E. Rahm, COnto-Diff: generation of complex evolution mappings for life science ontologies, J. Biomed. Inform. 46 (2013) 15–32. https://doi.org/10.1016/j.jbi.2012.04.009.

[13] R. Goncalves, B. Parsia, U. Sattler, Ecco: A hybrid diff tool for OWL 2 ontologies, OWL: Experiences and Directions Workshop (OWLED) Vol-849 (2012).

[14] Noy, Natalya F., Musen, Mark A., PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions.
[15] P. Kremen, M. Smid, Z. Kouba, OWLDiff: A Practical Tool for Comparison and Merge of OWL Ontologies, in: 2011 22nd International Workshop on Database and Expert Systems Applications, Toulouse, France, IEEE, 29.08.2011 - 02.09.2011, pp. 229–233.
[16] R. Pernisch, D. Dell'Aglio, M. Serbak, R.S. Gonçalves, A. Bernstein, Visualising the effects of ontology changes and studying their understanding with ChImp, Journal of Web Semantics 74 (2022) 100715. https://doi.org/10.1016/j.websem.2022.100715.
[17] C. Ochs, J. Geller, M.A. Musen, Y. Perl, Real Time Summarization and Visualization of Ontology Change in Protégé, in: VOILA@ISWC, Vienna, 2017.
[18] A. Reiz, K. Sandkuhl, An Ontology for Ontology Metrics: Creating a Shared Understanding of Measurable Attributes for Humans and Machines, in: Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Valletta, Malta, SCITEPRESS - Science and Technology Publications, 2022, pp. 193–199.
[19] A. Reiz, K. Sandkuhl, A Critical View on the OQuaRE Ontology Quality Framework, in: J. Filipe, M. Śmiałek, A. Brodsky, S. Hammoudi (Eds.), Enterprise Information Systems, Springer Nature Switzerland, Cham, 2023, pp. 273–291.
[20] S. Tartir, I.B. Arpinar, Ontology Evaluation and Ranking using OntoQA, in: International Conference on Semantic Computing, 2007, Irvine, CA, USA, IEEE Computer Society, Los Alamitos, Calif., 2007, pp. 185–192.
[21] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann, R. Gil, F. Bolici, Strignano Onofrio, Ontology evaluation and validation: An integrated formal model for the quality diagnostic task, Trentino, Italy, 2005.
[22] H. Yao, A.M. Orme, L. Etzkorn, Cohesion Metrics for Ontology Design and Application, J. of Computer Science 1 (2005) 107–113. https://doi.org/10.3844/jcssp.2005.107.113.
[23] M. Fernández, C. Overbeeke, M. Sabou, E. Motta, What Makes a Good Ontology? A Case-Study in Fine-Grained Knowledge Reuse, in: The semantic web, Springer, Berlin, 2009, pp. 61–75.
[24] A.M. Orme, H. Yao, L.H. Etzkorn, Indicating ontology data quality, stability, and completeness throughout ontology evolution, Journal of Software Maintenance and Evolution 19 (2007) 49–75. https://doi.org/10.1002/smr.341.
[25] Z. Yang, D. Zhang, C. Ye, Evaluation Metrics for Ontology Complexity and Evolution Analysis, in: IEEE International Conference on e-Business Engineering, 2006, Shanghai, China, IEEE Computer Society, Los Alamitos, Calif., 2006, pp. 162–170.
[26] S.J. Chalk, SciData: a data model and ontology for semantic representation of scientific data, J. Cheminform. 8 (2016) 54. https://doi.org/10.1186/s13321-016-0168-9.