A Method of Visual Presentation of Data Schemas

Lelde Lāce, Aiga Romāne, Mikus Grasmanis, and Kārlis Čerāns

Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia

Abstract

We describe and demonstrate a method of automated creation of refined visualizations of Linked data endpoint schemas, based on their pre-computed structure information. The visualization uses UML class diagram style structure with classes, associations, and attributes, as well as subclass structure. Due to the pre-computed nature of the data schemas, visualization can avoid repeating the subclass properties and links at a superclass level and present parts of larger data schemas in a compact and conceivable way.

Keywords

Linked data endpoint schema, Visual schema, UML class diagrams

1. Introduction

Visual presentation of a knowledge graph or a Linked data endpoint schema can be expected to help a user to comprehend the graph/endpoint structure and, therefore, use more efficiently the data contained therein. There are several tools allowing visualization of existing Linked data endpoint schemas, such as LD-VOWL [1] and LODSight [2], allowing to obtain on-the-fly visualizations of the class-to-property relations actually present in the suitably sized data sets.

The actual data schema information, important both for visual characterization of the data set contents and for technical assistance in handling the models based on such a data schema (e.g., by providing context-aware model auto-completion) is, however, much richer than the class-to-property correspondence alone. Such a schema can distinguish between more relevant and less relevant classes as property sources and targets (as described further on in the paper), it also can involve e.g., property domain/range information (in the ontological sense), and cardinalities.

The means for knowledge graph schema description are provided also by RDF data shape languages SHACL [3] and ShEx [4]. The concepts used in the data schema can also be described by means of OWL ontologies [5].

Due to the richness of the information the data schemas may contain, it might not (and typically would not) be possible to obtain their full information on-the-fly, with the user waiting behind the computer screen. In this demonstration we work in the setting of creating a data schema first and further on providing its visualization in the form of an extended UML class diagram, paying attention to the features we find necessary for the successful schema presentation.

The primary use case for the data schemas we consider here is serving the context-aware auto-completion in building visual queries over RDF data in the ViziQuer tool [6]. Still, data schemas are important also besides the particular tool. The ability to present the data schema visually to the users that seek to build visual data queries over the schema would be an important contribution towards a more encompassing visual experience in their data analysis work.

There exists a wealth of tools for visual presentation of OWL ontologies, including VOWL [7], OWLGrEd [8] and OntoDia [9]; [10] provides an expanded survey of such tools. There are tools for RDF data shape visualization, as well, including, e.g., RDFShape [11]. The data schema visualizations we provide here have a common conceptual understanding with the existing OWL and RDF data shape visualization methods by using graph of class and property connections as

Proceedings Acronym: Proceedings Name, Month XX–XX, YYYY, City, Country

lelde.lace@lumii.lv (L.Lāce); karlis.cerans@lumii.lv (K. Čerāns)

0000–0001–7650–2355 (L.Lāce); 0000-0002-0154-5294 (K. Čerāns)

the visual data schema image. Our presentation differs both from the OWL ontology and ShEx/SHACL shape presentations in that it concerns the actual data structure, as it is present in the data endpoint, and it involves a focus on important nuances (as the relevance of a class as a property source or target) that are not present or are present partially (e.g., as designating classes as property domains/ranges) in the existing visualization tools.

In the context of RDF data summarization (cf. [12] for a survey), our work is primarily concerned with fine-grained presentation of the entities (classes and properties) of the actual data schema that can be acted upon during the query creation.

In the rest of the paper Section 2 provides a motivational example and the data schema concept description, Section 3 describes the visualization of the data schemas and Section 4 provides brief conclusions and outlines directions for future work.

2. Data Schemas

We consider data schemas that are based on property availability at classes and connections of classes by properties, including the situations when a property can link multiple pairs of classes. Should it be typical for an instance in a data set to belong to multiple classes (including, but not limited to, the case of a subclass and a superclass), a direct class-to-property mapping may exhibit a set of links that is overly large for characterizing the actual connections of instances in the data set. This would show up immediately in the schema diagram visualization, where the extra links both overload the schema and may even obscure the actual connections.

As a simple example, consider the data set of Nobel Prizes[2] (we work with a copy[3] made for easier connection reasons). Figure 1 shows a (manually drawn) fragment of the data set schema, regarding Awards, Laureate Awards and Nobel Prizes; similar structure of links is obtained also in the LD-VOWL visualization[4]. Here only the two direct links between *nobel:LaureateAward*[5] and *nobel:NobelPrize* would be explicitly relevant; all links involving the *dbo:Award* class would actually obscure the data set presentation.

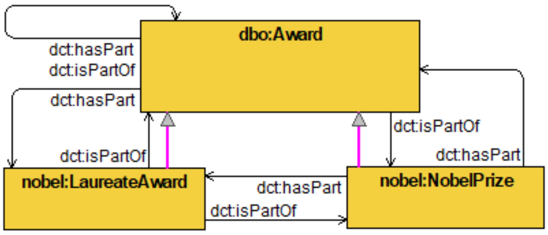


Figure 1: Naïve links in Nobel Prizes data set schema

If the data schema is used to support a modeling environment (e.g., in context-dependent data query completion, either in SPARQL form, or possibly in visual form, as in ViziQuer tool[6]), the full list of source and target classes for a property would need to be collected. However, the markers for the class importance in the context of a property would be essential also in the modeling environment. For instance, if a class were the property domain or range (the smallest domain/range, if there are several ones), this would allow to consider all other source/target classes for the property to be less relevant and even auto complete the property domain/range information. Still, the data schemas need to also cover the cases where there are more than one relevant class designated as a source or target class for a property (intuitively, the union of all "relevant" classes would need to cover the entire set of subjects or objects in the property triples).

[2] https://data.nobelprize.org/sparql

[3] http://85.254.199.72:8890/sparql, Named graph: http://nobelprizes.local

[4] http://vowl.visualdataweb.org/ldvowl/#/graph?endpointURL=http:%2F%2F85.254.199.72:8890%2Fsparql

[5] Here and further on the used namespace prefixes stand for their usual URIs. The prefix nobel: stands for http://data.nobelprize.org/terms/ and ncat: stands for http://data.nobelprize.org/resource/category/.

[6] http://viziquer.lumii.lv/. https://viziquer.app

Figure 2 outlines the structure of (an essential fragment of) the data schema used for storing the class-to-property correspondence, together with the described relevance markers (*cover_set_index* in the *cp_rels* table that relates the classes and properties) also stating the relation *type_id* (1 for incoming and 2 for *outgoing* property), cardinalities and property domain/range class information. The schema contains also further information that is used in the setting of auto-completing visual queries over RDF data in the ViziQuer tool, although the schema structure and its data can be used also independently of the visual tool.

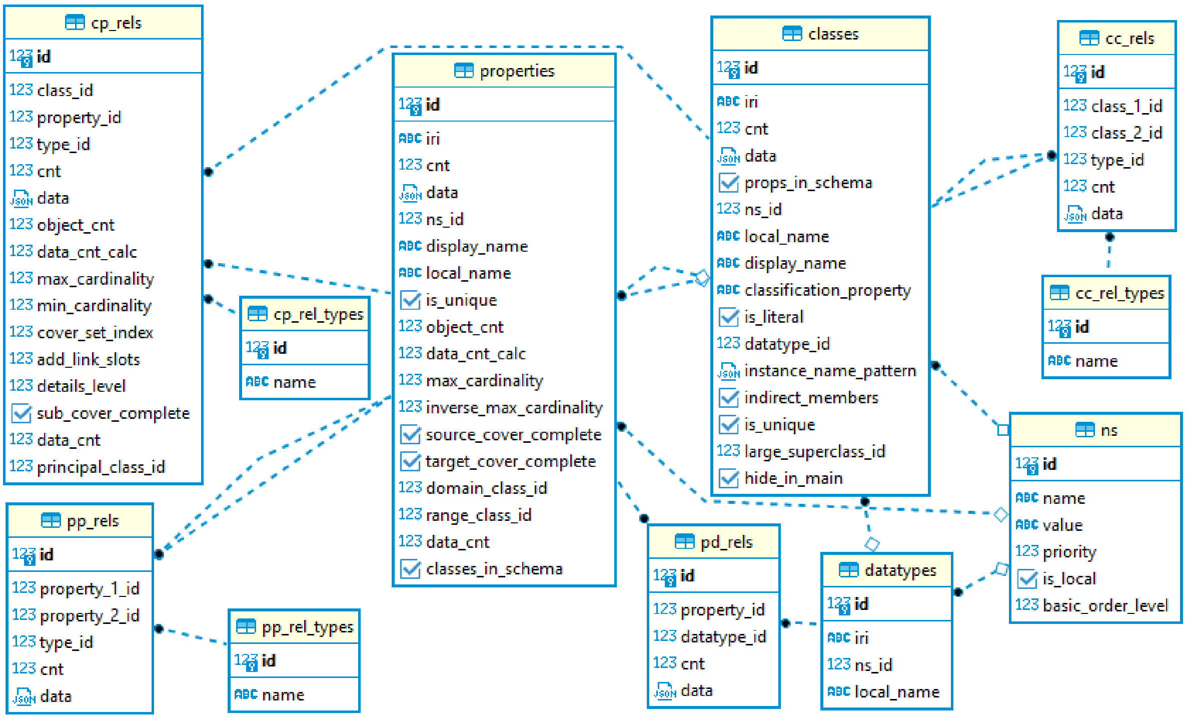


Figure 2: Data set schema structure

We offer the data schema filling by running first the open-source OBIS Schema Extractor tool[7] that issues a (possibly, large) series of SPARQL queries over a data set and creates a JSON-encoded structure of the data set schema (there are options to specify the level of granularity of the information to be retrieved from the data set at the schema extractor interface). The created JSON file is then stored into the database of the described schema by the data shape import service tool[8], followed by some manual editing of the namespace prefix abbreviations (e.g., for namespaces not found on prefix.cc site).

3. Schema Visualization

The data schemas described in Section 2 assist query auto-completion in the ViziQuer tool (with possible independent usage). The work demonstrated here shows the possibility to visualize the corresponding schema in form of extended UML class diagram thereby providing the visual schema presentation in addition to the visual ViziQuer query creation tool environment.

The idea of the schema visualization is to ascribe the properties only to those source and target classes that are relevant in the sense described in Section 2, so not ascribing a property link source/target to a superclass, if all property subjects/objects belong to a subclass, and not ascribing a property link source/target to a subclass if it is ascribed to a superclass.

[7] https://github.com/LUMII-Syslab/OBIS-SchemaExtractor
[8] https://github.com/LUMII-Syslab/data-shape-retrieval-services/tree/main/data-shape-import-service

Figure 3 shows a visualization of the Nobel Prizes data set schema in accordance with these principles. We note the single appearance of the properties *dct:hasPart* and *dct:isPartOf* connecting the *nobel:LaureateAward* and *nobel:NobelPrize* classes, in contrast to the naïve visual schema presentation in Figure 1.

Should only the object properties connecting the classes and the data properties at classes been visualized, an important property *nobel:category* would have been missing from the schema, as it connects (in the actual dataset) the instances of *nobel:LaureateAward* and *nobel:NobelPrize* classes to resources that do not have their class assertions specified. Therefore, a design is suggested to show an object property (*nobel:category* in the example) at a source class as an attribute, if some of its object instances are classless.

The created data schema is further on enriched by the property triple statistics (optional) showing the property triple count in the context (source class or source and target class combination) and total property triple count, property domain (*D*) and (local) range (*R*) information (in the ontological sense) and property max cardinality (1 or *).

To offer a further compactified data schema presentation, abstract superclasses (as *dbo:City or dbo:Country* in the example) can be introduced into the data schema, based on the shared properties for which the classes are sources or targets. An abstract class is introduced based on shared incoming and outgoing object properties, however, when introduced, it lists the data properties common to all its subclasses, as well[9]. The ontological range information for properties incoming into the abstract superclass is not shown in the example schema, however, it can be computed (approximated) from the information that is collected in the data schema (Figure 2).

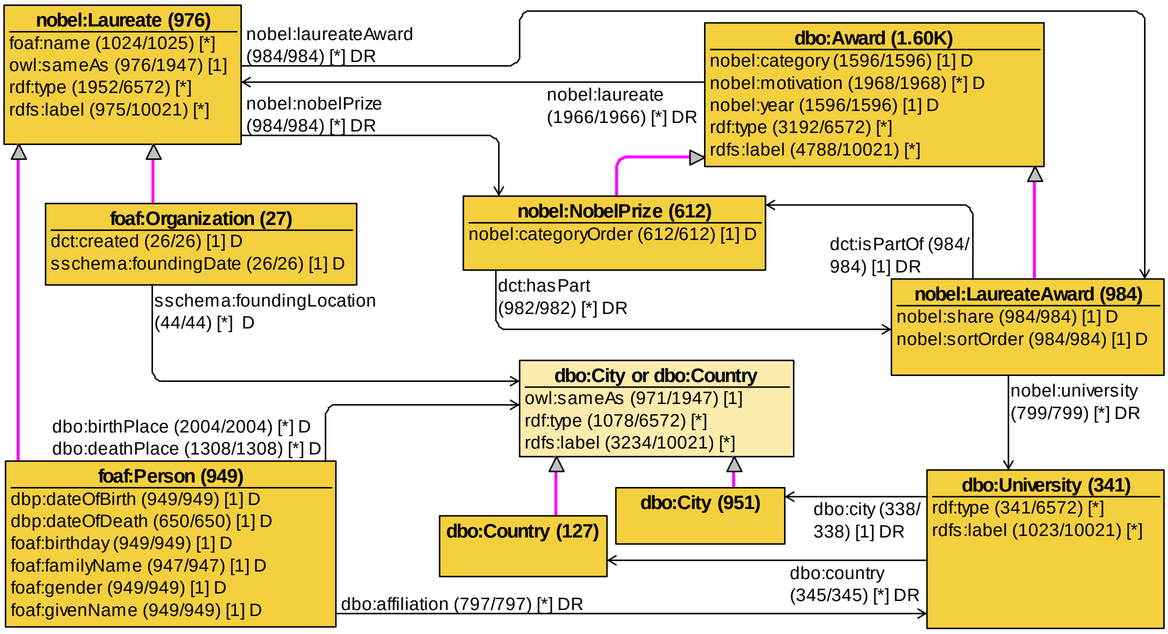

nobel:Laureate (976)
foaf:name (1024/1025) [*]
owl:sameAs (976/1947) [1]
rdf:type (1952/6572) [*]
rdfs:label (975/10021) [*]
nobel:laureateAward (984/984) [*] DR
nobel:nobelPrize (984/984) [*] DR
nobel:laureate (1966/1966) [*] DR
dbo:Award (1.60K)
nobel:category (1596/1596) [1] D
nobel:motivation (1968/1968) [*] D
nobel:year (1596/1596) [1] D
rdf:type (3192/6572) [*]
rdfs:label (4788/10021) [*]
foaf:Organization (27)
dct:created (26/26) [1] D
sschema:foundingDate (26/26) [1] D
nobel:NobelPrize (612)
nobel:categoryOrder (612/612) [1] D
dct:isPartOf (984/984) [1] DR
dct:hasPart (982/982) [*] DR
sschema:foundingLocation (44/44) [*] D
nobel:LaureateAward (984)
nobel:share (984/984) [1] D
nobel:sortOrder (984/984) [1] D
dbo:City or dbo:Country
owl:sameAs (971/1947) [1]
rdf:type (1078/6572) [*]
rdfs:label (3234/10021) [*]
nobel:university (799/799) [*] DR
dbo:birthPlace (2004/2004) [*] D
dbo:deathPlace (1308/1308) [*] D
foaf:Person (949)
dbp:dateOfBirth (949/949) [1] D
dbp:dateOfDeath (650/650) [1] D
foaf:birthday (949/949) [1] D
foaf:familyName (947/947) [1] D
foaf:gender (949/949) [1] D
foaf:givenName (949/949) [1] D
dbo:Country (127)
dbo:City (951)
dbo:city (338/338) [1] DR
dbo:University (341)
rdf:type (341/6572) [*]
rdfs:label (1023/10021) [*]
dbo:country (345/345) [*] DR
dbo:affiliation (797/797) [*] DR

Figure 3: Nobel Prizes data set schema

The data schema shown in Figure 2 (in Section 2) admits a possibility to specify a classification property (*classification_property* field in table *classes*) used for identifying a class in the data. The typical classification property is *rdf:type*, however, there are endpoints that use different classification properties (e.g., *Wikidata*[10]). In the considered Nobel Prizes data set example the classes in Figure 3 are created each with the *rdf:type* property, however, it would be possible to

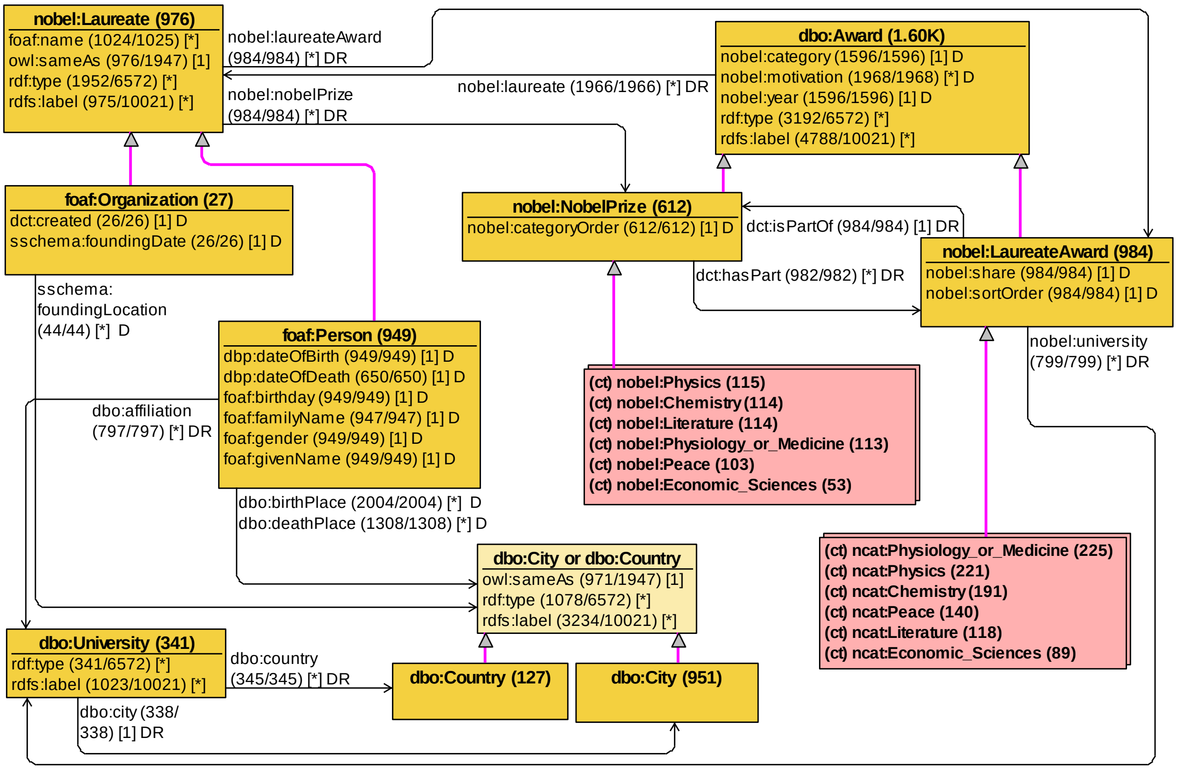
[9] Since the abstract superclasses currently are not reflected in the data schema (Figure 2), the statistics information for the properties at an abstract superclass may be higher than the actual numbers, if there are individuals shared by the subclasses. A remedy for this would be either introducing the abstract superclasses in the data schema, or consulting the actual data set during the schema creation process.

[10] https://query.wikidata.org/

introduce into the schema also the classifiers corresponding to other properties. Figure 4 shows such an extended schema, where, in addition to the "standard" classes, the classifiers, corresponding to the *nobel:category* property, are shown.

In the example the classifier values corresponding to the *nobel:category* property are marked with the prefix *(ct)*, and are shown using a simplified notation (list of multiple class names/classifier values in a single visual container) since none of them have any incoming or outgoing property characteristics that would be different from their container superclass.

Note that the categories for Nobel prizes and Laureate in the actual data set, although have the same local names, are different resources that belong to different namespaces (and therefore are to be listed separately in the extended data schema).



nobel:Laureate (976)
foaf:name (1024/1025) [*]
owl:sameAs (976/1947) [1]
rdf:type (1952/6572) [*]
rdfs:label (975/10021) [*]
nobel:laureateAward (984/984) [*] DR
nobel:nobelPrize (984/984) [*] DR
nobel:laureate (1966/1966) [*] DR
dbo:Award (1.60K)
nobel:category (1596/1596) [1] D
nobel:motivation (1968/1968) [*] D
nobel:year (1596/1596) [1] D
rdf:type (3192/6572) [*]
rdfs:label (4788/10021) [*]
foaf:Organization (27)
dct:created (26/26) [1] D
sschema:foundingDate (26/26) [1] D
nobel:NobelPrize (612)
nobel:categoryOrder (612/612) [1] D
dct:isPartOf (984/984) [1] DR
nobel:LaureateAward (984)
nobel:share (984/984) [1] D
nobel:sortOrder (984/984) [1] D
dct:hasPart (982/982) [*] DR
sschema: foundingLocation (44/44) [*] D
foaf:Person (949)
dbp:dateOfBirth (949/949) [1] D
dbp:dateOfDeath (650/650) [1] D
foaf:birthday (949/949) [1] D
foaf:familyName (947/947) [1] D
foaf:gender (949/949) [1] D
foaf:givenName (949/949) [1] D
nobel:university (799/799) [*] DR
(ct) nobel:Physics (115)
(ct) nobel:Chemistry (114)
(ct) nobel:Literature (114)
(ct) nobel:Physiology_or_Medicine (113)
(ct) nobel:Peace (103)
(ct) nobel:Economic_Sciences (53)
dbo:affiliation (797/797) [*] DR
dbo:birthPlace (2004/2004) [*] D
dbo:deathPlace (1308/1308) [*] D
dbo:City or dbo:Country
owl:sameAs (971/1947) [1]
rdf:type (1078/6572) [*]
rdfs:label (3234/10021) [*]
(ct) ncat:Physiology_or_Medicine (225)
(ct) ncat:Physics (221)
(ct) ncat:Chemistry (191)
(ct) ncat:Peace (140)
(ct) ncat:Literature (118)
(ct) ncat:Economic_Sciences (89)
dbo:University (341)
rdf:type (341/6572) [*]
rdfs:label (1023/10021) [*]
dbo:country (345/345) [*] DR
dbo:Country (127)
dbo:City (951)
dbo:city (338/338) [1] DR

Figure 4: Extended example schema, with category classifiers

The visualization module is freely available[11], together with its usage instructions in the ViziQuer tool context. The visualization starts from an existing ViziQuer project, from which the data to be loaded into the visualization tool are generated (an independent use of the visual schema presentation module would require generating this information in some other way).

The technical implementation of the visualization is currently performed within the desktop based GrTP/TDA platform [13] (the one providing the framework for the OWLGrEd visual ontology editing tool[12]). It is one of avenues of future work to integrate the visual schema presentation into the web-based environment of the ViziQuer tool.

The visualization module repository also provides links to the information on the currently visualized schemata. The current schemata list involves schemas over Nobel Prizes, Mondial[13] and ISWC 2017 metadata[14], and is kept growing.

[11] https://github.com/LUMII-Syslab/dss-schema-explorer
[12] http://owlgred.lumii.lv
[13] http://servolis.irisa.fr:3232/mondial/sparql
[14] http://servolis.irisa.fr/iswc2017/sparql

4. Conclusions and Future Work

In this work we have demonstrated that a saved data schema, in particular the one used by the ViziQuer tool environment, may contain useful information for the data schema visualization that would typically not appear in the simple class-to-property mappings that could be retrieved on-the-fly from the suitably sized data endpoints.

The concept of a data schema expansion to introduce non-standard classifiers on the level of the data schema and its visual presentation has been introduced, as well.

The principal directions of the future work include a fine-tuning of the visualization concept (e.g., by dealing with properties whose triple subjects or objects are classless). The concepts and algorithms from RDF graph summarization area (cf. [12]) can be expected to be helpful here.

Regarding the use of the visualization in the ViziQuer tool the principal future work avenues include integrating the visualization solution into the web/based tool environment, developing the options for schema fragment presentation and navigation, as well as enabling the data query creation over the data schema backbone presentation (since both the data schema presentation and a developed visual query would have similar UML-style diagram structure).

Acknowledgements

This work has been partially supported by a Latvian Science Council Grant lzp-2021/1-0389 "Visual Queries in Distributed Knowledge Graphs".

References

[1] Weise, M., Lohmann, S., & Haag, F. (2016). Ld-vowl: Extracting and visualizing schema information for linked data. In Voila!2016 (pp. 120-127).

[2] Dudáš, M., Svátek, V., Mynarz, J.: Dataset summary visualization with LODSight. In: The 12th Extented Semantic Web Conference (ESWC2015). http://lod2-dev.vse.cz/lodsight/lodsight-eswc2015-demopaper.pdf (2015).

[3] Shapes Constraint Language (SHACL), https://www.w3.org/TR/shacl/

[4] ShEx - Shape Expressions, http://shex.io/

[5] Web Ontology Language (OWL), https://www.w3.org/OWL/

[6] Čerāns, K., Šostaks, A., Bojārs, U., Ovčiņņikova, J., Lāce, L., Grasmanis, M., Romāne, A., Sproģis, A., Bārzdiņš, J. (2018). ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data, in Gangemi, A., et al. (ed.), Proceedings of The Semantic Web: ESWC 2018 Satellite Events. ESWC 2018. Lecture Notes in Computer Science, Vol. 11155. Springer, Cham, pp. 158–163. https://doi.org/10.1007/978-3-319-98192-5_30

[7] Lohmann, S., Negru, S., Haag F., Ertl, T.: Visualizing Ontologies with VOWL. In: Semantic Web 7(4), 399-419, (2016)

[8] Bārzdiņš, J., Čerāns, K., Liepiņš, R., Sproģis, A.: UML Style Graphical Notation and Editor for OWL 2. In: Proc. of BIR'2010, LNBIP, Springer 2010, vol. 64, pp. 102-113, (2010)

[9] 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: ISWC P&D 2015, CEUR, vol.1486, http://ceur-ws.org/Vol-1486/paper_77.pdf, (2015).

[10] Dudáš, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. In: The Knowledge Engineering Review, 33, (2018)

[11] Labra Gayo, E., Fernández Álvarez. D., and García González, H., RDFShape: An RDF Playground Based on Shapes, ISWC 2018 Posters &Demonstrations (2018).

[12] Čebirić, Š., Goasdoué, F., Kondylakis, H. *et al.* Summarizing semantic graphs: a survey. *The VLDB Journal* **28**, 295–327 (2019). https://doi.org/10.1007/s00778-018-0528-3

[13] Bārzdiņš, J., Zariņš, A., Čerāns, K., Kalniņš, A., Rencis, E., Lāce, L., Liepiņš, R., Sproģis, A.: GrTP: Transformation Based Graphical Tool Building Platform. In: Proc. of MDDAUI-2007, CEUR, vol. 297, http://ceur-ws.org/Vol-297/paper6.pdf ,(2007).