A Comparative Study of State-of-The-Art Linked Data Visualization Tools

Federico Desimoni[1][0000−0002−9051−2741], Nikos Bikakis[2][0000−0001−6859−1941],
Laura Po[1][0000−0002−3345−176X], and George
Papastefanatos[2][0000−0002−9273−9843]

[1] University of Modena and Reggio Emilia, Italy
{fdesimoni,laurapo}@unimore.it
[2] ATHENA Research Center, Greece
{bikakis,gpapas}@athenarc.gr

Abstract. Data visualization tools are of great importance for the exploration and the analysis of Linked Data (LD) datasets. Such tools allow users to get an overview, understand content, and discover interesting insights of a dataset. Visualization approaches vary according to the domain, the type of data, the task that the user is trying to perform, as well as the skills of the user. Thus, the study of the capabilities that each approach offers is crucial in supporting users to select the proper tool/technique based on their need.

In this paper we present a comparative study of the state-of-the-art LD visualization tools over a list of fundamental use cases. First, we define 16 use cases that are representative in the setting of LD visual exploration, examining several tool's aspects; e.g., functionality capabilities, feature richness. Then, we evaluate these use cases over 10 LD visualization tools, examining: (1) if the tools have the required functionality for the tasks; and (2) if they allow the successful completion of the tasks over the DBpedia dataset. Finally, we discuss the insights derived from the evaluation, and we point out possible future directions.

Keywords: Visualization Use Cases · Usability · Tools Benchmark · Ontology Visualization · RDF Graph Data · OWL · Linked Data · Semantic Web

1 Introduction

The wide adoption and availability of a very large number of Linked Data (LD) sources makes data visualization and exploration a crucial task for most LD consumers. Data scientists, domain experts, business users and citizens, wish to employ intuitive and visual, rather than programmatic, ways to interact with the data.

The exploration of LD is a particular task that differs from the classical data visualization, mainly due to the LD characteristics. The use of common (usually cross-domain) vocabularies for the description of the data (e.g., a resource is an instance of a class) or the use of typed properties for capturing relationships between resources within a dataset or across datasets, makes traditional data visualization ways, such as bar charts, incapable of capturing the complex inter-relationships within a LD dataset. Graph-based or similar approaches are usually employed. In addition, users, usually accessing a remote SPARQL endpoint, do not have an a-priori knowledge of the dataset, do not know if the dataset might be relevant for their goals and follow an exploratory way to visualize information. Finally, visualizing LD means to handle several issues: the large size, and the dynamic nature of data, the requests for exploratory searches, the variety of tasks, and the different types and needs of users.

A large number of LD visualization tools have been introduced the recent years, most of them originating from the academic sector. LD visualization tools provide graphical representations of a dataset or parts of it, with the aim of facilitating their analysis and generating insights into complex interconnected information. Visualization techniques can vary according to the domain, the type of data, the task that the user is trying to perform, as well as the skills of the user. To this end, the evaluation of the functionality that each technique offers, is necessary for users to better fit a tool/technique to their need.

Visual methods for query formulation undertake the challenge of making querying independent of users' technical skills [29]. There is a plethora of works that attempt to provide systematic reviews and surveys for several aspects of LD exploration and visualization. The recently published LD visualization tools book [24] includes an extensive review of such tools. Also, there are other surveys [10,7,17,21,1,5], which also address different aspects of visualization methods and tools. For example, [17,3] studies issues related to the entire process of LD consumption and exploration.

In contrast, in this study, we evaluate the functionality capabilities and features richness of several tools over a list of use cases, which are fundamental in the context of LD visual exploration. We define the use cases and evaluate whether each tool enables users to perform each use case over the DBpedia dataset. Our evaluation studies the following two aspects, if the tools: (1) provide the functionality/feature that is required to perform the tasks; and (2) allow the successful completion of the tasks over a dataset.

Contribution. (1) We define 16 use cases related to LD visualization, which allow us to evaluate several functionality aspects of LD visualizations tools. The use cases cover a plethora of tasks, focusing on different visualization aspect. For example, use cases related to: core LD concepts (e.g., schema, classes, instances, properties), concepts relationships (e.g., paths, common properties), navigation, visualization recommendation, filtering, statistics, etc. (2) We examine a long list of tools, and we select 10 tools which are publicly accessible and available for use. (3) We evaluate the tools by examining the defined use cases over the

DBpedia dataset. The use of DBpedia allows to study the tools over real-world tasks and scenarios.

2   An Overview of the Evaluated Tools

This section provides a brief description of the tools used in our study. Regarding the tools' selection process, we initially consider the *list of the 29 tools* presented in [11,24], then, we selected the tools that are *publicly accessible*, *available for use*, and support *SPARQL endpoint access*. As a result, *10 tools* are selected, 9 out of 10 are available online as web services, while Tarsier is an open source tool, which its code is available on GitHub. More details about the tools could be found at [11,24].

RelFinder[3] [18] is a Web-based tool that offers interactive discovery and visualization of associations (i.e., relationships) between selected LD instances. In this context, as associations are considered the paths (i.e., set of properties) that connect the instances in the LD graph.

LinkedOpenGraph[4] (LOG) [4] visualizes LD following a graph node-link layout. LOG allows the users to explore the visualized LD graph, by offering several interactive functionalities, such as, zoom, pan, filter, keyword search, and edit over nodes and edges (e.g., delete, rename, change color and shape, rearrange layout). Similarly, *Tarsier*[5] [31] tool visualizes LD using 3D node-link graph representation.

LodLive[6] [8] and *LodView*[7] are graph-based tools that adopt the incremental visualization paradigm, in which, instead of the whole graph, a starting point (e.g., node) is visualized, then, based on user interaction, more part of the graph are presented. In this context, the user starts her exploration from a given URI or a SPARQL endpoint.

LD-VOWL[8] [32] uses SPARQL queries to process RDF triples in order to infer schema information. The tool first identifies and presents the most representative concepts, using several methods and assumptions, (e.g., consider the classes with larger number of instances as representative). Then, the ontology schema is (progressively) visualized as graph, offering several interactive operations. The schema/ontology visualizations generated by the LD-VOWL follow the *VOWL* [20,19] graphical representation.

H-BOLD[9] (High-level visualization over Big Open Linked Data) [23,25] generates a representative summary of a LD source. BOLD takes as input a SPARQL endpoint and generates a visual (graph-based) summary of the LD source, accompanied by statistical and structural information of the source. Further, BOLD

---

[3] http://www.visualdataweb.org/relfinder/relfinder.php
[4] https://log.disit.org/service
[5] https://github.com/desmovalvo/tarsier
[6] http://lodlive.it
[7] https://lodview.it
[8] http://vowl.visualdataweb.org/ldvowl
[9] https://dbgroup.ing.unimo.it/hbold

offers incremental multilevel exploration, where a community detection algorithm is used to effectively construct the abstract levels.

In order to assist and guide the users in visual exploration scenarios, *LD-VizWiz*[10] (Linked Data Visualization Wizard) [2] exploits data semantics to simplify the process of setting up visualizations, providing a semi-automatic way for the production of possible visualization for LD datasets. LDVizWiz offers several visualizations types, i.e., Map, Tree/Hierarchy and Pie.

RDFSurveyor[11] [30] is a LD browser that provides class-based navigation over data retrieved from SPARQL endpoints. This tool is similar to a traditional Web browser, allowing the user to navigate over the dataset throw links that represent LD properties. Similarly, *SPARKLIS*[12] [12] is a LD browser that combines the faceted exploration paradigm (a.k.a. faceted search) with query building. Based on user's selections, the tool incrementally build structure queries, based on which the explored data is selected.

Note that, there are several tools that are available for use and cover numerous of the use cases evaluated in this paper; e.g., Balloon Synopsis [28], graphVizdb [6], LODmilla [22], Phuzzy.link [26], RDF4U [9], Rhizomer [13], QueryVOWL [15]. However, these tools are not considered in our evaluation since do not support SPARQL endpoint access.

3   Linked Data Visualization Use Cases

Users perform specific tasks while visually exploring LD and these tasks can be rendered into features implemented by visualization tools. The work in [24] has identified and categorized the possible interactions between a tool and a user into a number of use cases (UC). A use case describes how a user makes use of a system to accomplish a particular goal [16]. Use cases capture common functionalities that LD visualization tools should support, for users to gain clear and convincing visualization of the information (i.e., schema, resources and statistics), contained in a LD source. Detailed examples about the use cases can be found in [24].

Based on this list, we employ for our evaluation purposes 16 use cases, considering also generalization and special cases of some of the use cases included in [24]. The employed use cases, cover a plethora of tasks, which allow us to examine several tasks related to schema-level (e.g., classes, properties) and data-level (e.g., class' instances, values of properties) visualization, and the relationships between the visualized concepts; e.g., paths, common properties values, etc. Moreover, tasks related to exploration capabilities for navigation, visualization recommendation, filtering, and statistics presentation. The examined use cases, with more details on their intended functionalities are:

---

[10] http://semantics.eurecom.fr/datalift/rdfViz/apps
[11] http://tools.sirius-labs.no/rdfsurveyor
[12] http://www.irisa.fr/LIS/ferre/sparklis

UC1   **Visualize OWL/RDF Schema.** User selects a *dataset* to visualize schema information; e.g., classes, properties, classes/properties relations (e.g., super, equivalent, disjoint), semantics (e.g., inverse property, carnality), etc.
  This information useful for obtaining a global view of the dataset's schema. User can understand with a glimpse which classes the dataset is composed by, which properties connect each of them, and several semantics.

  – **UC1-C Visualize OWL/RDF Schema classes.** User selects a *dataset* to visualize classes' information. This is a special case of UC1, in which the information refers only to classes.

  – **UC1-P Visualize OWL/RDF Schema properties.** User selects a *dataset* to visualize (object, datatype) properties' information. This use case can be considered as a special case of UC1, in which the visualized information refers only to properties.

UC2   **Visualize the information related to a class.** User selects a specific *class* to visualize class's information This use case can be considered as a special of UC1-C, in which the visualized information refers to a specific class.
  For large T-Boxes it is useful to analyze single elements instead of the complete schema. For example, a users may be interested in discovering the different kind of relations connecting the selected class to other classes (e.g., sub-class, disjoint), etc.

UC3   **Visualize the information related to a property.** User selects a *property* to visualize property's information. This use case can be considered as a special of UC1-P, in which the visualized information refers to a specific property.
  For example, a user may be interested in examining the characteristics of a property, such as: domain, range, carnality, relations to other properties (e.g., inverse, equivalent), etc.

UC4   **Visualize instances.** User selects a *dataset* to visualize its instances. This use case is applicable for small LD datasets, allowing the user to examine data-level information; e.g., relations between the instances.

UC5   **Visualize the information related to a specific instance.** User selects an *instance* to visualize its object and datatype properties' values. Following UC4, a user may wish to focus her attention on specific instances analyzing their properties and their relations.

UC6   **Visualize the instances belonging to a class.** User selects a *class* to visualize its instances.

UC7   **Visualize the paths that connect different instances.** User selects *a set of instances* to visualize the paths (i.e., graph paths -set of properties-) among them.

This provides useful information regarding the direct or indirect ways that different instances are related.

UC8 **Navigate by traversing dataset's paths.** User selects a *class*, or an *instance* to explore its properties (i.e., paths).

For example, in a node-link graph layout, the presented part of the graph could be extended by clicking on an instance/class property (i.e., progressive visualization).

UC9 **Provide information about the vocabularies that are used in the dataset.** User selects a *dataset* to examine the vocabularies used. Vocabularies could provide useful information related to the provenance of the terms used in the dataset.

UC10 **Provide statistics about the dataset.** User selects a *dataset* to analyze various statistics; e.g., instances per class, frequencies.

UC11 **Visualize the data by selecting visualization types (e.g., map, plot) that are determined by the data contents.** User selects a *dataset*, generate or recommend different visualization types based on the types and semantics of data; e.g., instance type and properties, properties' values, class hierarchies.

For example, a tool generates a map visualization, for the instances having the geo:lat and geo:long attributes, or a treemap for class hierarchies.

UC12 **Visualize the instances that have specific properties**. User selects a *set of properties* to visualize the instances that have properties contained in the given properties' set.

UC13 **Visualize type-specific properties of an instance.** User selects an *instance* and a *set of properties* to visualize information regarding the given properties.

UC14 **Visualize range-specific datatype properties of an instance.** User selects an *instance* and a *a datatype* (e.g., integer) to visualize information of properties for which their range is of the specific datatype.

It is a common case, where instances have a long list of datatype properties. In such cases, filtering datatype properties based on its range datatype could help users to identify the information they are looking for.

UC15 **Visualize range-specific object properties of an instance**. User selects an *instance* and a *class*, to visualize information regarding the object properties which their range contained in the given class.

Similar to the UC14, a user could be interested in visualizing properties with a specific range object.

UC16 **Visualize more than one instances in parallel.** User selects *multiple instances* to visually compare information (e.g., properties values)

Use Case	Selected Resources [⋆]
UC2	Book
UC3	series
UC5	The_Lord_of_the_Rings
UC6	Book
UC7	The_Lord_of_the_Rings & Manchester
UC8	The_Lord_of_the_Rings & Manchester
UC9	The_Lord_of_the_Rings & Manchester
UC12	series
UC13	The_Lord_of_the_Rings & author
UC14	The_Lord_of_the_Rings & xsd:dateTime
UC15	The_Lord_of_the_Rings & The_Hobbit
UC16	The_Lord_of_the_Rings & Manchester

Table 1: DBpedia Resources Used in each Use Case

[⋆] Book: *class*   series, author: *object properties*
The_Lord_of_the_Rings, Manchester, The_Hobbit: *instances*

related to these instances. Particularly, this use case can be considered as a generalization of the UC5, where more than two instances are visualized.

In this use case, a user is able to compare and identify possible commonalities (e.g., same property values), relations (e.g., have a path connection), or dissimilarities between the selected instances.

4   Evaluation

In this Section, we present the evaluation of the 10 tools presented in Section 2 w.r.t. the 16 use cases described in Section 3. For each tool and use case, we examine whether the tool gives the ability to the user to accomplish the respective task over the DBpedia dataset.

4.1   Setup

Dataset. In our evaluation, we use the well-known, generic DBpedia dataset, through its SPARQL endpoint[13]. At the time of our evaluation (August 2020), DBpedia endpoint accesses about 1.6B triples.

---

[13] http://dbpedia.org/sparql

Use Cases over the DBpedia Dataset. In order to define the proposed use cases as specific tasks over the DBpedia, we use the following resources (i.e., classes, properties, instances) as described in the DBpedia dataset's description page[14].

For class-related UCs, we consider the *class* Book, which has 41 outcoming and 5 incoming properties, and about 64K instances. We also consider the *object properties* series and author. The series property has the owl:Thing as domain and range, and connects resources to the series they belong to (e.g., TV series, Book series). For example, the resource Friends is the series of the episode The_One_with_the_Proposal. The author property has as domain the class Work and range the class Person.

Finally, we consider the following three *instances*: (1) The_Lord_of_the_Rings which is an instance of the class Book, having 480 outcoming and 1.4K incoming properties; (2) Manchester which is an instance of the class City, having about 1K outcoming and 19K incoming properties; and (3) The_Hobbit which is an instance of the class Work.

The use of the resources per use case is presented in Table 1. For example, the UC6 is expressed as "*Visualize all the instances of the class* Book", whereas the UC7 is expressed as "*Visualize the paths (i.e., set of properties) between the* The_Lord_of_the_Rings *and the* Manchester".

Evaluation Setting. The evaluation has been conducted on a laptop with i7 CPU at 2.0GHz, 16G RAM and 15.6" (1366×768) screen, running Windows 10 and Google Chrome 84.0.4147. In the evaluation, the online versions of the 10 tools are used (URL are presented in Section 2). Tarsier is installed and evaluated locally on the laptop.
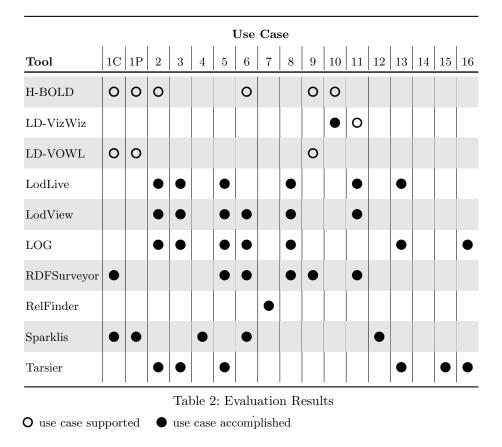
4.2   Results

Table 2 summarizes the results of our evaluation. The table indicates if a tool has the functionality to perform a use case, as well as if the tool was able to successfully complete a use case over the DBpedia.

UC1- C & UC1-P. Regarding the visualization of OWL/RDF schema information, H-BOLD and LD-VOWL provide the required functionality. H-BOLD allows users to understand the schema in a stepwise approach by showing a summarized version of it and then presents more elements incrementally. LD-VOWL, initially presents the complete schema and provides functionality that allows users to hide undesired elements and focus on the interesting ones. However, in our evaluation, in both tools, we are not able to perform schema-related tasks. Particularly, in the UC1-C and UC1-P use cases, after a long time, both tools are not responding.

Regarding the tools that follow a tabular-like data representation, Sparklis is able to successfully complete both use cases. Particularly, it presents classes and properties in two different list, where the user can select the resource of interest

---

[14] https://wiki.dbpedia.org/data-set-38

Tool	Use Case																
	1C	1P	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
H-BOLD	○	○	○				○			○	○						
LD-VizWiz											●	○					
LD-VOWL	○	○								○							
LodLive			●	●		●			●			●		●			
LodView			●	●		●	●		●			●					
LOG			●	●		●	●		●					●			●
RDFSurveyor	●					●	●		●	●		●					
RelFinder								●									
Sparklis	●	●			●		●						●				
Tarsier			●	●		●								●		●	●

Table 2: Evaluation Results

○ use case supported   ● use case accomplished

in order to further explore it. Additionally, RDFSurveyor accomplishes only the UC1-C, since it is able to display only a list of classes, without presenting any information about the properties. Finally, the rest of the tools do not provide the functionality for UC1-C and UC1-P use cases.

UC2, UC3 & UC5. Regarding the use cases that request information about a class (UC2), a property (UC3), and an instance (UC5), we report on the following.

LodLive, LodView, LOG and Tarsier are able to successfully complete all of the use cases. Particularly, in LodLive, LodView and LOG the user is able to retrieve information about a resource by giving its URI. In Tarsier, this is accomplished through SPARQL queries that are expressed by the user and evaluated over an endpoint. RDFSurveyor is able to perform UC5, in which the user is able to navigate over the hierarchy of classes in order to indicate the required instance. Note that, H-BOLD has this functionality, however, in our evaluation, we are not able to use it (see UC-1C & UC1-P). The rest of the tools do not provide the functionality required to perform any of these use cases.

UC4. Sparklis is the only tool that is able to accomplish UC4 by presenting a list of instances included in a dataset. Initially, a list of about 200 instances are presented; then, the user is able to use the search functionality in order to find the resource of interest. The rest of the tools do not provide the required functionality for this use case.

UC6. LOG, LodView, RDFSurveyor, and Sparklis are able to visualize the instances of a selected DBpedia class. Particularly, LOG follows a node-link layout, in which a class' instances can be visualized using the incoming rdf:type class' properties. LodView, RDFSurveyor, and Sparklis present data in a tabular form, in which the (web browser's) URI links of the incoming rdf:type properties can be used to explore instances. Note that, H-BOLD has this functionality, however, in our evaluation, we are not able to use it (see UC-1C & UC1-P).

UC7. RelFinder offers discovery and visualization of relations between instances. Hence, we can claim that RelFinder is mainly designed to support UC7. In our evaluation, RelFinder is the only tool that supports UC7.

UC8. In this use case we examine which tools are able to navigate over a dataset through properties, starting from a class or an instance. In this use case, beyond RelFinder, four tools are able to navigate through paths; i.e., LOG, LodLive, LodView, and RDFSurveyor. Particularly, LOG and LodLive visualize the dataset adopting a node-link layout. They follow a progressive visualization approach, in which, new parts of the graphs are presented, each time the user click on a node. This way, the user can navigate following the nodes' paths. LodView and RDFSurveyor present the information in a tabular form. In this case, the user are able to navigate through URI web browser's links.

UC9. Regarding providing vocabularies information, RDFSurveyor presents the vocabularies used in the dataset. H-BOLD and LD-VOWL have similar features, however, in our evaluation, we were not able to use these tools over the DBpedia data (see UC-1C & UC1-P).

UC10. This use case examines the capability of the tools to provide statistical information about the dataset. H-BOLD provides basic statistics regarding the datasets; e.g., the number of classes, instances, triples, properties, etc. Additionally, LD-VizWiz reports information regarding the types of the data contained in the dataset, as well as extracts some of the most common elements.

UC11. Regarding the capability of the tools to generate different visualization types based on the types of data, LDVizWiz exploits data semantics and generate/recommend different visualization types for the data. LodLive, LodView, and RDFSurveyor are able to identify the properties *geo:lat* and *geo:long* and present a map with instances' locations.

UC12. Sparklis is the only tool able to filter the instances based on a specific property.

UC13, UC14 & UC15. As described in UC 3, LodView, LodLive, LOG and Tarsier can present information for a particular property. In UC13 we want the

tools to present a set of specific instance's properties. LodLive, LOG and Tarsier are able to perform such tasks, using a filtering functionality which allows to filter out properties. There are no tool to support UC14, in which datatype properties are filtered based on its range datatype. Regarding the UC15, using the filtering functionality of Tarsier we are able to indirectly accomplish UC15.

UC16. This use case can be accomplished only by the LOG and the Tarsier tools. LOG allows the user to select and visualize multiple instances from an endpoint, by giving its URI. In Tarsier, this is accomplished through SPARQL queries that are expressed by the user and evaluated over an endpoint.

5   Discussion

The results presented in the previous section (Table 2) reveal the heterogeneity of the tools in accomplishing different user needs. Tools handle tasks by following different methods and/or workloads, or by adopting different visualization techniques. Generally, the adopted approaches vary according to the domain, the type of data, the task that the user is trying to perform, as well as the skills of the user. The evaluation's findings allow users to select the proper tool/technique based on their need. For example, we can claim the following.

If the user's goal is the visualization of the structure/schema of the dataset, then, H-BOLD and LD-VOWL can be used. H-BOLD shows a summarized version of the data, and then present more elements incrementally, while LD-VOWL presents the complete schema. Such tasks can be also accomplished by the RDF-Surveyor and Sparklis, which provide a tabular representation of the data.

If the user wishes to obtain a statistical overview of a dataset, H-BOLD or LDVizWiz can be an option. H-BOLD provides basic statistics regarding the datasets (e.g., the number of classes, instances); and LDVizWiz can extract the most common elements, as well as other statistic-related information.

LodLive and LOG may be the most suitable tools for *visualizing instances and their relations*. They offer specific features related to owl: sameAs relations and inverse properties. In addition, LOG allows to insert more elements on the screen and is able to *handle and compare many datasets simultaneously*.

Finally, If the user wishes to *explore the relations between two or more instances*, the RelFinder can effectively support such tasks.

6   Conclusion and Future Work

In this paper, we evaluated and compared the functionality capabilities and features richness of several Linked Data Visualization tools over a list of use cases. The evaluation has been conducted over the DBpedia dataset.

We are aware that this is only a first step of a deeper analysis of these tools. In the near future, we intend to include a user evaluation about effectiveness and efficiency of each tool in performing the defined use cases, following the metrics presented in [27]. Alternatively, if involving users is not feasible, we will explore

the use of usability evaluation frameworks that do not require users involvement but provide similar metrics, like [14].

References

1. Alahmari, F., Thom, J.A., Magee, L., Wong, W.: Evaluating Semantic Browsers for Consuming Linked Data. In: Australasian Database Conference, ADC (2012)
2. Atemezing, G.A., Troncy, R.: Towards a Linked-Data Based Visualization Wizard. In: International Workshop on Consuming Linked Data (COLD) (2014)
3. Bauer, F., Kaltenbapck, M.: Linked Open Data: The Essentials: A Quick Start Guide for Decision Makers. Edition mono/monochrom (2012)
4. Bellini, P., Nesi, P., Venturi, A.: Linked open graph: Browsing multiple SPARQL entry points to build your own LOD views. J. Vis. Lang. Comput. **25**(6) (2014)
5. Bikakis, N.: Big Data Visualization Tools. In: Encyclopedia of Big Data Technologies, 2nd Edition. Springer (2021)
6. Bikakis, N., Liagouris, J., Krommyda, M., Papastefanatos, G., Sellis, T.K.: GraphVizdb: A Scalable Platform for Interactive Large Graph Visualization. In: ICDE. pp. 1342–1345 (2016)
7. Bikakis, N., Sellis, T.K.: Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art. In: EDBT (2016)
8. Camarda, D.V., Mazzini, S., Antonuccio, A.: LodLive, exploring the web of data. In: I-SEMANTICS (2012)
9. Chawuthai, R., Takeda, H.: RDF Graph Visualization by Interpreting Linked Data As Knowledge. In: JIST (2015)
10. Dadzie, A., Pietriga, E.: Visualisation of Linked Data - Reprise. Semantic Web Journal **8**(1) (2017)
11. Desimoni, F., Po, L.: Empirical evaluation of Linked Data visualization tools. Future Generation Computer Systems **112** (2020)
12. Ferré, S.: SPARKLIS: a SPARQL Endpoint Explorer for Expressive Question Answering. In: International Semantic Web Conference (Posters & Demos) (2014)
13. García, R., Gil, R.: Building a Semantic Intraweb with Rhizomer and a Wiki. In: IntraWebs Workshop, World Wide Web Conference (2006)
14. González, R.G., Iranzo, R.M.G., Bakke, E., Karger, D.R.: A benchmark for end-user structured data exploration and search user interfaces. Journal of Web Semantics **65**(8), 100610 (2020). https://doi.org/10.1016/j.websem.2020.100610, `https://doi.org/10.1016/j.websem.2020.100610`
15. Haag, F., Lohmann, S., Siek, S., Ertl, T.: QueryVOWL: Visual Composition of SPARQL Queries. In: ESWC. vol. 9341, pp. 62–66 (2015)
16. Kim Hamilton, R.M.: Learning UML 2.0. O'Reilly (2006)

17. Klímek, J., Skoda, P., Necaský, M.: Survey of Tools for Linked Data Consumption. Semantic Web Journal (2018)
18. Lohmann, S., Heim, P., Stegemann, T., Ziegler, J.: The RelFinder user interface: interactive exploration of relationships between objects of interest. In: IUI (2010)
19. Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: User-Oriented Visualization of Ontologies. In: EKAW (2014)
20. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing Ontologies with VOWL. Semantic Web Journal **7**(4) (2016)
21. Marie, N., Gandon, F.L.: Survey of Linked Data Based Exploration Systems. In: International Workshop on Intelligent Exploration of Semantic Data (IESD) (2014)
22. Micsik, A., Turbucz, S., Györök, A.: LODmilla: A Linked Data Browser for All. In: SEMANTiCS. pp. 31–34 (2014)
23. Po, L.: High-level Visualization Over Big Linked Data. In: International Semantic Web Conference (P&D/Industry/BlueSky) (2018)
24. Po, L., Bikakis, N., Desimoni, F., Papastefanatos, G.: Linked Data Visualization: Techniques, Tools, and Big Data. Synthesis Lectures on the Data, Semantics, and Knowledge, Morgan and Claypool (2020)
25. Po, L., Malvezzi, D.: Community detection applied on big linked data. J. UCS **24**(11), 1627–1650 (2018), `http://www.jucs.org/jucs_24_11/community_detection_applied_on`
26. Regalia, B., Janowicz, K., Mai, G.: Phuzzy.link: A SPARQL-powered Client-Sided Extensible Semantic Web Browser. In: VOILA. pp. 34–44 (2017)
27. Sánchez, J.L.G., García, R., Brunetti, J.M., Gil, R., Gimeno, J.M.: Using SWET-QUM to compare the quality in use of semantic web exploration tools. J. Univers. Comput. Sci. **19**(8), 1025–1045 (2013). https://doi.org/10.3217/jucs-019-08-1025, `https://doi.org/10.3217/jucs-019-08-1025`
28. Schlegel, K., Weißgerber, T., Stegmaier, F., Seifert, C., Granitzer, M., Kosch, H.: Balloon Synopsis: A Modern Node-centric Rdf Viewer and Browser for the Web. In: ESWC (2014)
29. Soylu, A., Kharlamov, E., Zheleznyakov, D., Jiménez-Ruiz, E., Giese, M., Skjæveland, M.G., Hovland, D., Schlatte, R., Brandt, S., Lie, H., Horrocks, I.: Optiquevqs: A visual query system over ontologies for industry. Semantic Web **9**(5), 627–660 (2018). https://doi.org/10.3233/SW-180293, `https://doi.org/10.3233/SW-180293`
30. Vega-Gorgojo, G., Slaughter, L., von Zernichow, B.M., Nikolov, N., Roman, D.: Linked Data Exploration With RDF Surveyor. IEEE Access **7** (2019)
31. Viola, F., Roffia, L., Antoniazzi, F., D'Elia, A., Aguzzi, C., Cinotti, T.S.: Interactive 3D Exploration of RDF Graphs through Semantic Planes. Future Internet **10**(8) (2018)
32. Weise, M., Lohmann, S., Haag, F.: LD-VOWL: Extracting and Visualizing Schema Information for Linked Data Endpoints. In: VOILA@ISWC (2016)