

Customised Visualisations of Linked Open Data

Alice Graziosi, Angelo Di Iorio, Francesco Poggi, and Silvio Peroni

DASPLab - Digital and Semantic Publishing Laboratory
DISI - Department of Computer Science and Engineering
University of Bologna, Italy

`alice.graziosi@studio.unibo.it, angelo.diiorio@unibo.it,
silvio.peroni@unibo.it, francesco.poggi5@unibo.it`

Abstract. This paper aims to tackle on Linked Open Data (LOD) customised visualisations. The work is part of an ongoing research on interfaces and tools for helping non-programmers to easily present, analyse and support sense making over large semantic dataset. The customisation is a key aspect of our work. Producing effective customised visualisations is still difficult due to the complexity of the existing tools and the limited set of options they offer, especially to those with little knowledge in LOD and semantic data models. How can we give users full control on the primitives of the visualisation and their properties, without requiring them to master Semantic Web technologies or programming languages? The paper presents a conceptual model that can be used as a reference to build tools for generating customisable *infoviews*. We used it to conduct a survey on existing tools in terms of customisation capabilities. Starting from the feedback collected in this phase, we will implement these customisation features into some prototypes we are currently experimenting on.

Keywords: Semantic Web · Linked Open Data Visualisation · RDF · SPARQL.

1 Introduction

Along with the spread of the Semantic Web techniques, the amount of data available on the Web as RDF and, in particular, as Linked Open Data (LOD) increased. A lot of rich datasets have been released, therefore visualising large semantic dataset content in a comprehensible way has become a major challenge.

Generating and customising visualisations is needed for different classes of users. First of all, for *data producers*: after making available their data in a machine-readable format, these users need to make these data easily readable for humans and to hide the complexity of the underlying ontological model. Second, for *data readers* who access existing datasets and need clear views and summaries. Clear visualisations are fundamental, for instance, when exploring a new dataset in order to understand its structure and content. This is very common on DBpedia—in fact, different visualisation tools have been proposed by the community to explore such a large knowledge base[23][18] and then generalized. Explorative tasks are also common on other datasets that are continuously added to the LOD cloud.

Clear visualisations are also helpful for users who need to interrogate multiple LOD sources and to produce mashups; these are also needed for users who investigate statistical data and aggregated information on datasets.

Thus, we use the term *RDF view producers* for referring to all those users who need to represent and view RDF datasets effectively.

There are two options nowadays for view producers. The first one is using tools for building visualisations, for instance LD-VOWL[36]. Just a few clicks and the system gets data from data sources and produces a clear visualisation. Though, most of existing tools give the user a limited set of predefined representations and configurable options (for instance shapes, colors and metaphors). Overall, the creation process is easy and intuitive but the flexibility could be improved.

Instead, the flexibility is the main strength of the second approach: users combine graphic libraries to implement *ad hoc* data visualisations. There are many frameworks focused on data visualisations, like D3.js¹ and Cytoscape.js². These frameworks enable *programmers* to build very sophisticated views, but obviously technical competencies are still required. Most view producers, who are non-programmers, still have difficulties in using these frameworks directly.

Thus, this paper presents a preliminary research about interfaces and tools for helping *non-programmers*, both data producers and data readers, to easily produce custom LOD *infoviews*.

The contribution is twofold. First, we present a conceptual model and a workflow which can be used as reference to build tools for creating customisable, dynamic and interactive visualisations. Second, we use the model to survey the existing tools and to evaluate how they support views customisation. We would like to foster discussion in the research community in order to get feedback for future development.

The paper is then structured as follows. Section 2 presents some motivational case studies that lead us to devise the model introduced in Section 3. The survey is discussed in Section 4, before the conclusions in Section 5.

2 Generating LOD Views for Non-Programmers

This work started from our own experience in displaying LOD datasets. Our first case study was the Semantic Lancet Project[2]. The goal of the project was to create an experimental LOD on scholarly publications. In particular, we took as input all papers published in the Journal of Web Semantics by Elsevier and produced a dataset that contains, for each paper, bibliography, abstract and citations compliant with the Semantic Publishing and Referencing (SPAR) Ontologies³. These ontologies describe the publishing domain in detail and allow designers to build complex and expressive structures. The problem we faced was to make such information understandable for average users. For instance, we were interested in producing a bird-eye overview of the dataset that just shows the citation network and then details on demand.

One way was to use tools that produce a graph-based visualisation. The VOWL-based toolset is the most representative and powerful. VOWL (Visual Notation for OWL Ontologies)⁴[22] is a visual language that uses graphical primitives in combination with

¹ <https://d3js.org/>

² <http://js.cytoscape.org/>

³ <http://www.sparontologies.net/>

⁴ <http://vowl.visualdataweb.org/>

color scheme to define graphical objects for most of the OWL elements. Those objects are combined together in a force-directed graph view to represent ontologies. VOWL-based tools like WebVOWL[19] and LD-VOWL[21] provide users with interfaces and wizards to build visualisations incrementally, starting from a publicly available data source or a local dataset. The results are very good but, to the best of our knowledge, the level of customisation is still limited (for instance, the shapes and the colours of the notation are fixed). Furthermore, the VOWL specification focuses on the representation of TBox information (classes, properties and datatypes), though it also includes recommendations on how to portray ABox information (instances and data values).

The navigation of ontologies and data models (TBox) is widely supported and users can count on powerful stand-alone or Web-based environments. As far as we know, the visualisation of instances (ABox) is much less supported. There are obviously more difficulties in such navigation because of the large amount of data to display; also, this is usually combined with the TBox navigation and the resulting interfaces are more appropriate for technical users.

The VOWL notation is widely designed for RDF graphs: circles stand for classes, arrows for property relations and rectangles for property labels and datatypes. This notation maps directly the RDF graph structure: it looks clear to Semantic Web experts but it is still difficult for others.

The main problem we saw is that the final data depiction is still complex: it is needed to make things simple and to select just a few data to show. This was evident in the Semantic Lancet Project because of the complex structure of the SPAR ontologies. To represent a citation, for instance, SPAR requires a complex chain of objects and relations. Such complexity is needed to allow a precise description of each aspect of the publication domain, and meaningful and automatic reasoning on such data; on the other hand, such complexity should be hidden to most people.

A further problem we experienced was the customisation: the tools that we used provided us a limited set of options. For instance, there was no possibility to represent some entities with personalised icons or to change shapes and colours.

Eventually we decided to implement some visualisations in order to gain full control on the output. Apart from the required effort of this second solution, we ended up having visualisations designed as task specific solutions, quite difficult to adapt to other contexts. In the same project, for instance, we wanted to show information about co-authorship and affiliations, but we had to basically re-implement visualisations from scratch.

The same problems were evident in another project promoted by the Italian Academic And Research Network (GARR Consortium⁵), in which we were investigating the availability of datasets that describe computers networks and developing mechanisms to display these datasets. One of the issues we had to face from the beginning was related to the background and skills of our users. In fact, most of them showed no experience in (Semantic) Web technologies and complained about the difficulties in generating clear summaries and visualisations of LOD datasets.

This second scenario leads us to discuss two more points. First of all, we noticed that users wanted to display some ‘dynamic’ graphic objects, whose properties change

⁵ <https://www.garr.it/en>

according to the attributes of the entities they represent. One of the first request of the stakeholders, in fact, was to make a clear distinction between the nodes of the network that are reachable, from those that are not. This information is usually expressed by a property: our users expected this property to be clearly mapped into the final visualisation. In other words, they are asking for the ability to map some properties of the data to *visual variables*⁶ of the graphical representation.

They also pointed out another issue: the visualisation of a subnetwork as a graph of entities and properties mirroring the RDF structure is not clear. Usually the subnetwork is represented by a node and the fact that a node belongs to that subnetwork is represented as an object property connecting the two nodes. Our users, on the other hand, did not expect to find a node representing the subnetwork but they expected to see a cluster in which all nodes of the subnetwork are close to each other, in a group whose boundaries are visible and clear.

These problems are also shared with many other LOD resources. The solution is to couple datasets with *ad hoc* user-friendly interfaces that provide sophisticated functionalities and mashup tools. The problem is that programmers have to build these interfaces manually, by combining existing modules (for instance D3.js) in a process that requires a lot of time and technical competencies.

3 Customised LOD Views: Workflow and Conceptual Model

The objective of our research is enabling users to produce customised visualisations of LOD so that programming skills are not necessary. In this section we discuss a conceptual model we derived from the two projects introduced in Section 2. Both projects concerns the use and development of LOD datasets. Moreover, given their heterogeneity both in domain (i.e. scholarly and network datasets) and in types of people involved (e.g. *data readers* and *data producers*), they are a representative sample for our investigation.

We interviewed eight people who are involved in the life-cycle of the aforementioned projects in order to find both their needs and their requirements. In particular, we considered four persons for each project: two data producers (one with programming skills, and one without technical skills) and two data readers (a programmer and a non programmer).

We structured the interview in two parts. First, we collected their problems and requirements related to their personal experience in building visualisations to communicate their data. Then, we performed some design meetings in which we presented them some problems and related tasks, and asked to sketch the structure of the representations they would develop and to discuss the proposed solution.

The results of our survey are distilled into a reference workflow shown in Figure 1. Our approach breaks down into seven phases organised in two macro-phases: the first macro-phase concerns the activities related to data preparation, and the second one on the generation of the views of data.

⁶ With the term visual variables we refer to visual properties used to convey meanings (e.g. position, size, shape, colour, orientation, texture, etc.). The term was first introduced by the French cartographer Jacques Bertin in [4]

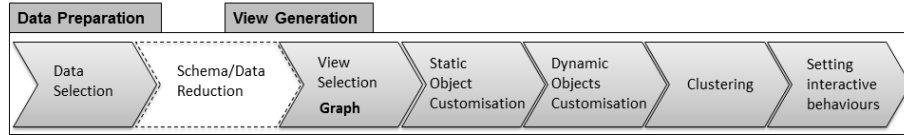


Fig. 1. A workflow for customising LOD views.

3.1 Data Preparation

The Data Preparation (DP) macro-phase concerns all the data processing tasks to prepare the data for the visualisation.

DP1: Data Selection. The users should have simplified interfaces to select the desired data from a knowledge base, without having to know neither semantic query language nor the dataset structure. This process can be incremental: first the users choose a list of entities, then explore their properties and then moves to other entities. The ability to filter data and to show only parts of the dataset is particularly important.

DP2: Schema/Data Reduction. The users also expressed a strong need for tools that make visualisations simple, even when built on top of complex data model.

Complex ontological structures in LOD datasets are very common, as they are needed to describe domains in a precise and powerful way. On the other hand, users often need overviews and subparts of these models. For instance, two entities in a graph may be linked by a long chain of properties so that some schema reduction technique may be needed. SPARQL property path feature⁷ deals exactly with that issue of property summarization. Some simplification might also happen directly on the visualised data, without impacting the underlying schema. For instance, a visualisation might produce charts or textual summaries of some extracted and/or grouped data. For this reason, we foresee a ‘Schema/Data Reduction’ phase that overlaps Data Preparation and View Generation. Actually, this area is still underinvestigated. We plan to further explore it in the future, though we believe it is useful to have it already in the overall model. This is also the reason why the diagram shows the Schema/Data Reduction in white background and dashed borders.

3.2 View Generation

The main objectives of the View Generation (VG) macro-phase is to define how the selected data should be mapped into visual objects in a consistent and complete visualisation, and how users can interact with.

In order to capture and describe the different items of the visualisations we adopt a terminology summarised as follows:

- *primitives*: the atomic building blocks of a visualisation systems. The main elements are visual marks, such as points, lines, areas, etc.;

⁷ <https://www.w3.org/TR/sparql11-query/#propertypaths>

- *properties*: the attributes of each visual mark that can be used to convey meaning. Typical examples are positional properties (where the mark is in space), retinal properties (which define how the mark is rendered such as color, shape, size, saturation, texture, and orientation), etc.;
- *views*: are conceptually cohesive sets of visual marks or other views which collectively represent a part of the data designated to be represented. Example views include bar charts, scatterplots and treemaps.
- *behaviours*: define any kind of interaction between users and view elements, mapping user intents into modifications of the views and the properties of the contained elements.

It will be used in the rest of the section to better explain each step of our workflow:

VG1: View Selection (Graphs). The first step consists in selecting the kind of visualisation users want to produce. Visualisations are described in terms of atomic building blocks and compounded objects as mentioned before. Popular examples of views are barcharts, linecharts, piecharts, etc., which are usually categorised using Shneiderman’s taxonomy [31].

Since the underlying model of LOD datasets is a graph, in this work we focus on explicit node-link visualisations such as network diagrams. Nonetheless, the proposed approach is not limited to this views and can be generalised to different kinds of visualisation.

VG2: Static Objects Customisation. The users should then be able to associate graphic objects (i.e. the primitives and properties in our terminology) to each element selected from the dataset. It is common to show datasets in graphs whose edges correspond to the entities and arcs to the properties. These objects should be customisable, allowing users to vary items in shape, colour and size. It would also be interesting to represent objects with images (for instance dynamically loaded from DBpedia⁸ dataset) rather than with a 2D shape.

VG3: Dynamic Objects Customisation. There is a further interesting point: the ability to change *dynamically* the graphical properties of the objects according to the properties of the entities they represent. Consider for instance a network infrastructure, shown as a graph, in which all hosts are represented with a green icon, while a red one is used for those which are unreachable.

VG4: Clustering. Our interviews also suggested that tools should also support alternative ways of organising groups of entities. Consider now a band and all its band members. The ‘traditional’ way of showing this group is to have a list or to have one central node for the band connected to one node for each band member. A more intuitive visualisation is to group all members in a rectangular area, representing the band that contains all nodes of the members. To the best of our knowledge, such clusters have not been used in combination with graph-based views to display LODs. Having the chance of interactively group data into cluster by selecting different properties is a further customisation that users could benefit from.

VG5: Setting interactive behaviours. The final aspect we take into account is the ability to produce interactive visualisations that readers can update with little effort. The

⁸ <http://dbpedia.org>

point is to specify the behaviours of the visualisation by providing a set of clear options to the final users, without burdening the interface and without asking them to know the details of the model and the dataset. The process of loading data should obviously be smooth and efficient, even in presence of large and complex datasets.

Closely related to the topic of interactive behaviours is the distinction between exploration and visualisation as argued by Nunes and Schwabe[26]. In this paper we only focus on visualisation tools, without going into the details of explorative tasks and processes performed by the users during infovis-based system generation and use, and leave a more in-depth analysis as future work of our research.

4 Building and Customising LOD Views: a Survey

In this section we present a comprehensive study of the related work in LOD visualisation and exploration, focusing on *customisation* features. Other surveys[10][27] have been done in the same research field, but most of them concern on scalability and performance issues in handling huge datasets.

We analyse the tools using the reference model and the workflow presented in Section 3 as a basis of comparison. Our findings are summarised in Table 1. It shows one row for each tool and one column for each phase, indicating whether that personalisation feature is supported.

Before going on, it is worth underlining that the survey review only web-based tools. The tools that concern solely ontologies (or born as plug-in for ontology editor) are not taken into consideration, except if they provide interesting interactivity and/or customisation features.

Note also that we consider visualisations of RDF datasets only. Generic visualisations tools, like Tableau⁹, are extremely powerful for end-user visualisations of raw data but they are out of scope for our analysis.

It is also interesting to point the types of statements managed by each tool. A knowledge base can be represented conceptually as a combination of terminologies (TBox) and assertions (ABox). TBox statements describe a set of concepts and their properties, while ABox statement describe entities and values. ABox information is certainly more relevant for our work.

Let us now go into details of each phase of our model.

DP1: Data Selection. The tools we surveyed provide simple interfaces to get data from local or external SPARQL endpoint and select a visualisation type among a set of predefined options. This is the case, for instance, of LOD Visualization¹⁰[8], Payola¹¹[16] and LDVizWiz¹²[1]. The degree of customisation of such tools (grouped at the top of the table) is rather low; on the other hand, they all require little effort to select sources and display data.

⁹ <https://www.tableau.com>

¹⁰ <http://lodvisualization.appspot.com/>

¹¹ <http://live.payola.cz/>

¹² <http://semantics.eurecom.fr/datalift/rdfViz/apps/>

Table 1. The table shows an overview on existing LOD visualisation tools for quick comparison.

Tool Name (release year)	TBox- ABox	Data Selection	Data Reduction	View Selection*	Static Objects Customi- sation	Dynamic Objects Customi- sation	Clustering	Setting inter- active be- haviours
LOD Visualiza- tion(2012)	ABox			tm,tr				
Payola(2013)	ABox			c,g,m,tm tl,tr				
LDVizWiz(2014)	ABox			pc,m				
LinkedPipes(2016)	ABox			cu,tr				
LODWheel(2011)	ABox			c,g,gw				
LODeX(2014)	TBox ABox			g				
ViCoMap(2015)	ABox			m				
CubeViz.js(2016)	ABox	✓		hm,pc,cc				
RelFinder(2010)	ABox	✓		g				
LodLive(2012)	TBox ABox	✓		g	✓			✓
WebVOWL(2017)	TBox	✓		g	✓			✓
LD-VOWL(2016)	TBox	✓		g	✓			✓
GraphVizDb(2016)	ABox	✓		g	✓			✓
VizBoard(2013)	ABox	✓		c,m,sp sb,g,bb sc,ac	✓			✓
PrEVIew(2016)	ABox			cmn		✓	✓	✓
BallonSynopsis(2014)	ABox	✓		g,m			✓	✓
rdf:SynopsViz(2014)	ABox	✓		bc,tl,tm,pc			✓	✓
Rhizomer(2012)	TBox ABox	✓		c,m,tm,tl,cc				✓
SemLens(2011)	ABox	✓		sp,sl				✓
Vis Wizard(2014)	ABox	✓		bc,pc,m sp,tb,bb plc,lc,sg				✓
LinkDaViz(2015)	ABox	✓		bb,c,sp,m,pc				✓
VIZ-VIVO(2016)	ABox	✓		cmn,kc,sb				✓
Geolink(2016)	ABox			tb,m,g				✓
Cesium(2016)	ABox			m				✓
Linked Data Reac- tor(2016)	ABox	✓		tb,m,g bc	✓	✓	✓	✓
Uduvudu(2015)	ABox	✓		m,bc	✓	✓	✓	✓

* Legend: hm = heatmap, pc = pie chart, cc = column chart, tm = treemap, tr = tree, sp = scatter plot, cmn = concept map network, g = graph, plc = parallel coordinates, kc = keyword cloud, sb = sunburst, cu = cube, tb = table, m = map, bb = bubble chart, tl = timeline, bc = bar chart, sc = stacked chart, ac = area chart, lc = line chart, c = chart, sg = streamgraph.

LinkedPipes¹³[17] also lets users start by selecting a data source (as an external or a local SPARQL endpoint); the system then automatically inspects the selected source to find data patterns and converts them in views. LODeX¹⁴[3] is a further tool able to easily load and display heterogeneous data. It is actually a visual editor for SPARQL query that can be used to interrogate the selected endpoints. LODWheel¹⁵[32] also provides the possibility of highlighting data categories (since it is specialised for DBpedia) by applying distinct colors and display them in different ways.

Tools can be compared also on the ability to filter data to display — for instance, a subset of elements, values and data properties. Table 1 indicates the tools providing such a feature in the ‘Data Selection’ column. Two of these are CubeViz.js¹⁶[25] and RelFinder¹⁷[18].

CubeViz.js is a tool for statistical data analysis targeted to both non-technical users and technical users. In the section for non-technical users, data are displayed using data cubes in which some suggested measure of interests are highlighted. CubeViz includes a faceted browsing widget that can be used to filter interactively observations to be visualised in charts. The user can select data sets, measures, attributes and dimension properties.

RelFinder is specialised on showing relations. It is a DBpedia featured project that extracts and displays relationships between entities described by ontologies; users are provided an intuitive interactive interface to filter relations and to produce customised views, that can be exported in multiple formats too.

DP2: Schema/Data Reduction While the data selection phase is supported by most of the tools we surveyed, the schema and data reduction (in particular the possibility of simplifying the data model for the final visualisation, or modifying - e.g. grouping - some data) is not supported yet. In fact, the corresponding column of Table 1 is empty for all tools.

VG1: View Selection (Graphs) The main difference we found among the tools is the type of visualisations (*views* in our model) they provide. For instance, LDVizWiz uses graphs, LOD Visualisation produces visual hierarchies using treemaps and trees starting from live access to SPARQL endpoints; Payola supports graphs, charts, treemaps and many other options (it provides a wide range of domain-specific visualisation plugins that can be shared and reused among the users).

The views are very heterogeneous, as shown by the table. Data types include numeric, temporal, spatial, hierarchical, cube, statistical and graph data. Data can be displayed as graph, map, treemap, heatmap, column chart, concept map network and many more. Due to the graph-like structure of the RDF data model, the graph-based (node-link) visualisations are considered the more relevant for the LOD domain.

Some of these views work very well for statistical analysis. ViCoMap¹⁸[29] allows automatic correlation analysis and enrichment of imported statistical data with infor-

¹³ <http://visualization.linkedpipes.com/>

¹⁴ <http://dbgroup.unimo.it/lodex>

¹⁵ <http://opendata.computas.no:7001/lodwheel/>

¹⁶ <http://cubevizjs.demo.aksw.org/>

¹⁷ <http://www.visualdataweb.org/relfinder.php>

¹⁸ <http://vicomap.informatik.uni-mannheim.de/>

mation from external LODs. The correlated values are distinguished by colour and displayed in diagrams or advanced visualisation maps generated using Google Maps API or custom geographical shape data. Maps are in fact the main objects in VicoMap. The tool supports users in customising these maps as well.

VG2: Static Objects Customisation. After selecting the type of visualisation, some tools also allows users to customise static objects. LodLive¹⁹[9] displays data in an excellent graph visualisation which can be expanded incrementally to explore in details the desired resource. The resources imported into LodLive can be connected to each other, allowing the user to pass from one endpoint to another by making use of LODs interconnections. Users have control on the properties of these connections.

WebVOWL²⁰[20] is based on the VOWL visual notation that maps each of OWL items to a graphical object, composed together in a force-directed graph layout.

WebVOWL[19] offers sophisticated views of data but users can choose among a limited number of graphic objects and properties. They are also able to customise a limited set of properties of these objects. LD-VOWL²¹[21] is another implementation of VOWL that extracts schema information from SPARQL endpoint's data and display it in an overview graph visualisation. LD-VOWL allows personalization in terms of Class-node/Class-node and Class-node/Type-node distance, layout and external elements colours. The classes and properties to be rendered can be filtered too (as shown in the first column of the table).

GraphVizDb²²[5] also offers some personalisation of the static objects. Users can select if they want to show edge labels or node values; they can also set which type of zoom or focus apply to the final visualisation.

VizBoard²³[35] is particular powerful in the customisation of static properties. It allows users to configure LOD views in dashboard-like perspective. The goal of VizBoard is making flexible and context-aware *infoviews* of LODs for non expert users. Many visualisation types are implemented in VizBoard, and user are allowed to configure them changing how they are rendered (e.g. item size, position, overall view complexity, etc.). The interface is very rich and complex.

VG3: Dynamic Objects Customisation. PrEVIEW²⁴[11] was the only system we found providing some personalization of the dynamic behavior of the objects. PrEVIEW is a tool that for browsing scientific publications in biomedical domain data from PumMed²⁵ resources. The tool shows detailed information about the medical literature searched concept using concept map approach. The interesting point is that PrEVIEW makes the graphic object dimension proportional to the number of the occurrences of certain concept. Such a behavior is embedded in the system though.

VG4: Clustering. PrEVIEW also leads us to the next phase of our reference model: view variance and, in particular, support for alternative clustering. In fact it clusters

¹⁹ <http://lodlive.it/>

²⁰ <http://vowl.visualdataweb.org/webvowl.html>

²¹ <http://vowl.visualdataweb.org/ldvowl.html>

²² <http://83.212.97.26:8080/graphVizdb/>

²³ <http://www.vizboard.de/>

²⁴ <http://srvgal78.deri.ie/PrEVIEW/>

²⁵ <https://www.ncbi.nlm.nih.gov/pubmed/>

data according to different criteria and allows users to select these criteria. Ballon Synopsis²⁶[30] is also relevant here. The system is a *node-centric* RDF data viewer and browser available as a jQuery plug-in embeddable in a website. Ballon uses colours to highlight similar entities and clusters. This is very peculiar of the system that proved to be very effective with the users of this domain. A further peculiar aspect of Ballon with respect to customisation is the use of configurable templates. These are preloaded layouts that can be adjusted and applied to the data. Not all the properties of these layouts are customisable but such a solution is very flexible and powerful.

The support for clusters in rdf:SynopsViz²⁷[6] is also worth mentioning. The aim of rdf:SynopsViz is to support the hierarchical LOD exploration based on the creation of multiple levels of hierarchically related groups of resources based on the values of one or more properties. This hierarchical approach provide effective data summarisation. Rdf:SynopsViz also allows user interaction over data and manipulation of charts, timelines and treemaps.

VG5: Setting interactive behaviours The last column of Table 1 in fact shows the tools that produce interactive visualisations in which users can easily analyse data from different perspectives, filter, reorder, move and so on. The support for interactivity is very heterogeneous.

Rhizomer²⁸[7][12] provides LOD visualisation based on "Overview first, zoom and filter, then details-on-demand" Information-Seeking Mantra[31]. First, the dataset overview is automatically generated. User can then zoom in on items of interest and filter out the uninteresting ones. At this point additional details are provided on user's demand.

With SemLens²⁹[13] data are shown as scatter plots and can be analysed through user-defined semantic lenses. The semantic lenses provide filters for data collections based on object properties. Filters are configurable depending on the datatype (number, date, string) of the different objects. Vis Wizard³⁰[34] and LinkDaViz³¹[33] allow users to group data and aggregate values, and provide multiple interactive widgets.

Rich sets of widgets are also provided by many domain-specific projects. Some of them are listed in the final rows of the table. These systems are meant to be used with specific data and are strongly tied with these data. The support for selecting input is quite limited — that is why the corresponding column of the table is empty — but the final visualisation is very rich and expressive.

Just to cite a few examples: VIZ-VIVO³²[14] produces concept map networks, word clouds and sunburst visualisations about scholarly works of research institutions; Geolink³³[24] offers tabular, map and graph-based visual interfaces over the major scien-

²⁶ <http://schlegel.github.io/balloon/balloon-synopsis.html>

²⁷ <http://synopsviz.imis.athena-innovation.gr/>

²⁸ <http://rhizomik.net/html/rhizomer/>

²⁹ <http://www.visualdataweb.org/semlens.php>

³⁰ <http://code.know-center.tugraz.at/vis>

³¹ <http://eis.iai.uni-bonn.de/Projects/LinkDaViz/>

³² <http://www.vivoweb.org/>

³³ <http://demo.geolink.org/>

tific geo-data LOD sources; Cesium³⁴[28] uses the globe to visualise instances from geospatial ontologies with sophisticated map views.

The bottom part of the table shows some tools that are extremely powerful and produce customised views but still required some programming skills.

UDUVUDU³⁵[23] provides users with a template-based language to match patterns in input datasets and generate HTML visualisations. The language is still quite difficult for non-programmers but highly flexible and powerful. In fact, it was used to produce views with customised static and dynamic objects, as well as for aggregated views of data.

Linked Data Reactor³⁶[15] is equally powerful. It is a framework to develop flexible and reusable widgets for Linked Data applications. Some coding is still required for the users but the system comes with a wide range of reusable UI web components and widgets which can be assembled in new visualisations. New widgets are also continuously added by the developers.

In conclusion, several tools exist that provide great interactivity features to navigate through data but there is still room for great improvement in customisation and infoviews generation.

5 Conclusion

In this paper we discuss semantic data rendering and customisation issues required to be handled by Linked Open Data (LOD) visualisation tools. We also propose a data visualisation workflow that could be adopted as a reference model while developing a tool for LOD customisable *infoviews*. A survey was conducted in order to compare existing tools using the reference model as a basis of comparison.

Currently, we are developing a Web-based application as a proof of concept in accordance with the reference model presented above.

Acknowledgments

This research has been supported by a grant of the Consortium GARR - Italian Academic And Research Network and by MIUR PRIN 2015 GAUSS Project.

References

1. Ghislain Auguste Atezing and Raphaël Troncy. Towards a linked-data based visualization wizard. In *COLD*, 2014.
2. Andrea Bagnacani, Paolo Ciancarini, Angelo Di Iorio, Andrea Giovanni Nuzzolese, Silvio Peroni, and Fabio Vitali. The semantic lancet project: a linked open dataset for scholarly publishing. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 101–105. Springer, 2014.

³⁴ <http://home.iitb.ac.in/~abhishekvpotnis/voila2016/1.1/>

³⁵ <http://dbpedia.exascale.info/>

³⁶ <http://ld-r.org/>

3. Fabio Benedetti, Sonia Bergamaschi, and Laura Po. Lodex: A tool for visual querying linked open data. In *International Semantic Web Conference*, 2015.
4. Jacques Bertin. *Semiology of graphics: diagrams, networks, maps*. 1983.
5. Nikos Bikakis, John Liagouris, Maria Krommyda, George Papastefanatos, and Timos K. Sellis. graphvizdb: A scalable platform for interactive large graph visualization. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1342–1345, 2016.
6. Nikos Bikakis, George Papastefanatos, Melina Skourla, and Timos K. Sellis. A hierarchical framework for efficient multilevel visual exploration and analysis. *CoRR*, abs/1511.04750, 2015.
7. Josep Maria Brunetti, Rosa Gil, and Roberto Garcia. Facets and pivoting for flexible and usable linked data exploration. Citeseer, 2012.
8. Josep Maria Brunetti and Sören Auer and Roberto García. The linked data visualization model. In *International Semantic Web Conference*, 2012.
9. Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. Lodlive, exploring the web of data. In *I-SEMANTICS*, 2012.
10. Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2:89–124, 2011.
11. Syeda Sana e Zainab, Qaiser Mehmood, Durre Zehra, Dietrich Rebholz-Schuhmann, and Ali Hasnain. Preview: Clustering and visualising pubmed using visual interface. In *VOILA@ ISWC*, pages 17–27, 2016.
12. Roberto García, Josep Maria Brunetti, Rosa Gil, and Juan Manuel Gimeno. Rhizomer: overview, facets and pivoting for semantic data exploration. *University de Lleida*, 2012.
13. Philipp Heim, Steffen Lohmann, Davaadorj Tsendragchaa, and Thomas Ertl. Semlens: visual analysis of semantic data with scatter plots and semantic lenses. In *I-SEMANTICS*, 2011.
14. Muhammad Javed, Sandy Payette, Jim Blake, and Tim Worrall. Viz-vivo: Towards visualizations-driven linked data navigation. In *VOILA@ ISWC*, pages 80–92, 2016.
15. Ali Khalili, Antonis Loizou, and Frank van Harmelen. Adaptive linked data-driven web components: Building flexible and reusable semantic web interfaces - building flexible and reusable semantic web interfaces. In *ESWC*, 2016.
16. Jakub Klímek, Jirí Helmich, and Martin Necaský. Payola: Collaborative linked data analysis and visualization framework. In *ESWC*, 2013.
17. Jakub Klímek, Jirí Helmich, and Martin Necaský. Linkedpipes visualization: Simple useful linked data visualization use cases. In *ESWC*, 2016.
18. Steffen Lohmann, Philipp Heim, Timo Stegemann, and Jürgen Ziegler. The refinder user interface: interactive exploration of relationships between objects of interest. In *IUI*, 2010.
19. Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. Webvowl: Web-based visualization of ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 154–158. Springer, 2014.
20. Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. Webvowl: Web-based visualization of ontologies. In *EKAW*, 2014.
21. Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. Extraction and visualization of tbox information from SPARQL endpoints. In *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016)*, volume 10024 of *LNAI*, pages 713–728. Springer, 2016.
22. Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with vowl. *Semantic Web*, 7(4):399–419, 2016.
23. Michael Luggen, Adrian Gschwend, Bernhard Anrig, and Philippe Cudré-Mauroux. Uduvudu: a graph-aware and adaptive ui engine for linked data. In *LDOW@WWW*, 2015.
24. Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, and Grant McKenzie. A linked data driven visual interface for the multi-perspective exploration of data across repositories. In *VOILA@ ISWC*, pages 93–101, 2016.

25. Michael Martin, Konrad Abicht, Claus Stadler, Axel-Cyrille Ngonga Ngomo, Tommaso Soru, and Sören Auer. Cubeviz: Exploration and visualization of statistical linked data. In *WWW*, 2015.
26. Thiago Nunes and Daniel Schwabe. Frameworks for information exploration-a case study. In *IESD@ ISWC*, 2015.
27. Oscar Peña and Diego López de Ipiña. Linked open data visualization revisited: A survey. 2014.
28. Abhishek Potnis and Surya S Durbha. Exploring visualization of geospatial ontologies using cesium. In *VOILA@ ISWC*, pages 143–150, 2016.
29. Petar Ristoski and Heiko Paulheim. Visual analysis of statistical data on maps using linked open data. In *ESWC*, 2015.
30. Kai Schlegel, Thomas Weissgerber, Florian Stegmaier, Michael Granitzer, and Harald Kosch. Balloon synopsis: A jquery plugin to easily integrate the semantic web in a website. In *ISWC Developers Workshop*, 2014.
31. Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
32. Magnus Stuhr, Dumitru Roman, and David Norheim. Lodwheel - javascript-based visualization of rdf data. In *COLD*, 2011.
33. Klaudia Thellmann, Michael Galkin, Fabrizio Orlandi, and Sören Auer. Linkdavis - automatic binding of linked data to visualizations. In *International Semantic Web Conference*, 2015.
34. Gerwald Tschinkel, Eduardo E. Veas, Belgin Mutlu, and Vedran Sabol. Using semantics for interactive visual analysis of linked open data. In *International Semantic Web Conference*, 2014.
35. Martin Voigt, Stefan Pietschmann, and Klaus Meissner. Towards a semantics-based, end-user-centered information visualization process. 2012.
36. Marc Weise, Steffen Lohmann, and Florian Haag. Extraction and visualization of tbox information from sparql endpoints. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 713–728. Springer, 2016.