Interactive Visualisation Techniques for the Web of Data

Marie Destandau
Supervised by Emmanuel Pietriga and Caroline Appert
Univ. Paris-Sud, CNRS, Inria
Orsay, France
marie.destandau@inria.fr

ABSTRACT

The RDF format offers powerful possibilities for machines, such as reasoning or federated queries over interlinked datasets. However, presenting RDF data to humans is very challenging: its very structure defeats traditionnal approaches, as it separates information into small pieces, making it difficult for users to make sense of it. My PhD work proposes an approach that presents RDF data in a context, to make them understandable by humans. We first describe S-Paths, a system to support set-based exploration of a dataset's content. We show that it works well on simple models, but that its efficiency is limited by performance issues on very abstract models. Then we lay the basis for a second project, whose aim is to take one more step back and put these sets of entities in a broader context, to give a structural overview of Linked Datasets.

CCS CONCEPTS

• **Information systems** → Presentation of retrieval results; **Browsers**.

KEYWORDS

Visualization, Linked Data, Semantic Web

ACM Reference Format:
Marie Destandau. 2019. Interactive Visualisation Techniques for the Web of Data. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3308560.3314189

1 PROBLEM

Though RDF format has been designed to be processed by machines, there is a strong need for visualisation and exploration tools: experts involved in publication and reuse of RDF data need to gain a better understanding of their data, and many of the datasets published as RDF would also be of interest to lay users. However, the very structure of RDF makes it difficult to produce efficient visualisations.

The first obvious representation, which accurately reflects the underlying directed-graph structure, is a node-link diagram. Such diagrams are very efficient to describe the data model or a limited number of entities, but they become unreadable over a few dozens of triple statements, while even very small datasets contain several thousands of them.

TheWebConf, 2019, San Francisco

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3314189

Another approach, used by most Linked Data browsers, consists in displaying one page per resource, with all statements directly related to it, as clickable links when they consist of URIs. While this allows users to hop from one resource to another across graphs and datasets, whatever their size, the first drawback is that relevant properties to describe a resource might be several hops away: when the user reaches a piece of information after several clicks, this information is displayed separately from the resource that was originally of interest. Keeping the chain of previous pages in mind requires a significant memorization effort. It is similar to following series of sentences *Peter has a son. The latter has a nephew. The latter has a wife. The latter is a pharmacist.* instead of the single sentence *The nephew of Peter's son is married to a pharmacist.* The mental effort increases as the user browses back and forth to explore other paths. He gathers crumbs of information that he must keep in memory to later make sense of them. However, to display longer chains of information, a browser would need to know where to start and stop. Only browsers with an *a priori* knowledge of the data model are able to do it. A second limitation of this navigation paradigm is that it does not allow to display a resource in the context of other similar resources, whereas this is often an essential feature for sense-making activities.

Tools dedicated to display sets of resources exist. They face the same problems, amplified by the sometimes very large number of entities in a set. In addition, they encounter a problem common to all set-based representations, especially when the range of items is very large: for readability reasons, the relevant information might depend on the number of items in the set (or in the subset in the case of an interactive tool enabling subselections). Thus, considering a collection of 50,000 books, one might want to display the century when they were published, whereas one could list the topics of 500 books, the titles and authors of 50 books, the abstracts and pictures of 10 books, or the full information of a single book. Displaying titles for 50,000 books would simply not make sense because it is not readable.

In this context the first research question for my PhD was: can we use the specificities of Linked Data to automatically identify and display sense-making visualizations at any scale?

2 RELATED WORK

Early visualization tools exposed the raw RDF graph represented as a node-link diagram [9], sometimes using stylesheets to customize the appearance of nodes and links [10]. The approach is generic and accurate, and the resulting representations can be useful to illustrate very small graphs and communicate to a relatively expert audience, but it quickly yields illegible *"big fat graphs"* [12]. In

most cases, the RDF graph is not the right level of abstraction for presentation purposes [5]. It is verbose and the details of how the model structures the data are of low interest to users.

Early linked data browsers approached the problem by producing generic (almost raw) representations of the data in HTML pages, exposing property-value pairs of the current resource of interest as text and clickable URIs. Clicking a URI would attempt to dereference it, and a new HTML page with the properties of the corresponding resource would be displayed. Some browsers featured support for Fresnel [11], a language giving high-level indications about how to display RDF data, which can be used to generate more human-friendly representations of the data. Such languages, however, require that declarative presentation rules be available to the browsers for the different RDF vocabularies involved. Tabulator [1] was one of the first linked data browsers to provide users with more relevant visual representations of RDF data without resorting to such vocabulary-specific presentation languages. However, since Tabulator only features a limited set of views (map, timeline, calendar), it frequently displays data using a generic, more triple-oriented, tree or tabular view on data which are neither spatial nor temporal.

Another significant limitation of many linked data browsers is their strong focus on the *follow-your-nose* browsing strategy. Users explore individual paths from resource to resource over the Web of Data, incrementally fetching the properties of the new resource of interest, without much context (if any) about similar resources. Context information, however, is essential to answering questions related to the distribution of values, to the identification of potential correlations, and to the direct comparison of resources forming a coherent set such as, *e.g.*, all resources of a given type in a particular dataset.

Answering questions such as these, and more generally gaining insights about collections of resources, requires generating multivariate data visualizations (*e.g.*, scatterplots [8]) of a set of properties associated with resources in the set. The Linked Data Visualization Model [4] defines a transformation workflow to dynamically associate one or more datasets with multiple visualizations. Implementations of this workflow typically support a range of data visualization techniques. The methodology defined in [3] focuses on statistical linked data as found in, *e.g.*, RDF data cubes, making it possible to combine data from different linked sources and display them in visualization dashboards. Multiple other projects have been attempting at making it easier for users to generate data visualizations from linked data. Of particular interest here are Visualbox [6] and LinkDaVis [14]. Visualbox enables its users to create different visualizations by writing SPARQL queries and populating predefined visualization templates with the queries' results. The produced visualizations can then be embedded in classic Web pages. LinkDaVis provides users with a hierarchical representation of properties, from which users can choose the ones to visualize. It then performs a heuristic analysis of the data to suggest a ranked list of visualization configurations for these properties, based on different bindings to visual encoding channels. Once a particular visualization is selected, it can further be customized, as in other mixed-initiative approaches such as that adopted in Voyager [15], a tool for the creation of multi-variate data visualizations.

Such tools are very powerful, and enable the creation of a wide variety of visualizations from linked data: maps, timelines, and a range of statistical charts (scatterplot, line plot, density plot, *etc.*). However, they produce standalone visualizations, that can be used on the spot or exported and integrated elsewhere. They do not support browsing over the Web of Data. Indeed, once created, these visualizations feature a very limited level of interactivity. They cannot be used directly to make sub-selections or continue navigation, which requires following links, fetching additional data, and displaying it.

3 PROPOSED APPROACH

Based on the earlier mentioned considerations that the very first condition to enable sense-making activities is readability, and that we would rather follow longer chains of properties than give crumbs of information, our hypothesis is that chains of triple statements in the graph can be used as aggregate steps to reach readable ranges of values. Whatever the number of entities in the set we consider, following chains of triple statements should always lead to a set of values that is either in a readable dimension or can be aggregated.

3.1 S-Paths: Browsing the Content of a Dataset, from Overview to Detail

The first project of this PhD is S-Paths. Its model is based on a *mixed-initiative* approach [14, 15] where the system automatically suggests interesting views for a set of resources, but lets users reconfigure them at will. Exploring a set of resources is an iterative process: whenever users make a selection, S-Paths defaults to what it considers to be the most relevant view on that selection, which users can, again, reconfigure at will. A demo instance is available[1].

3.1.1 Sets of Entities. To identify the initial sets of resources, we relied on the `rdf:type` property. We listed chains of property patterns leading from this set of entities to a set of values. We name these patterns *semantic paths*.

We then defined 6 categories for the paths, corresponding to different possible behaviors in terms of aggregation and display: *datetimes* can be aggregated by years, decades, centuries...; *geographical coordinates* can be aggregated at different scales, corresponding to zooming on a map; *images* can be resized, displayed at different scales, and juxtaposed in grids or mosaic; *numbers* can be aggregated by numerical groups; *text strings* can only be aggregated by similar values, and can be displayed in different layouts depending on their charlength; *URIs* can only be aggregated by similar values, share a common pattern and an approximate similar charlength, and can be used as links. We used these categories, as well as other dimensions listed in Table 1, to characterize the *semantic paths*.

3.1.2 Available Visualizations. Since the conditions of readability are not absolute, but depend on the type of visualization, we relied on a set of views, each one declaring how many dimensions it can display, and for each dimension which categories it can display, and under which conditions (minimum, maximum and optimal). The views are not exclusive, several can apply to a specific case, in this case the optimal conditions will be used to propose the most efficient one.

[1] http://s-paths.net

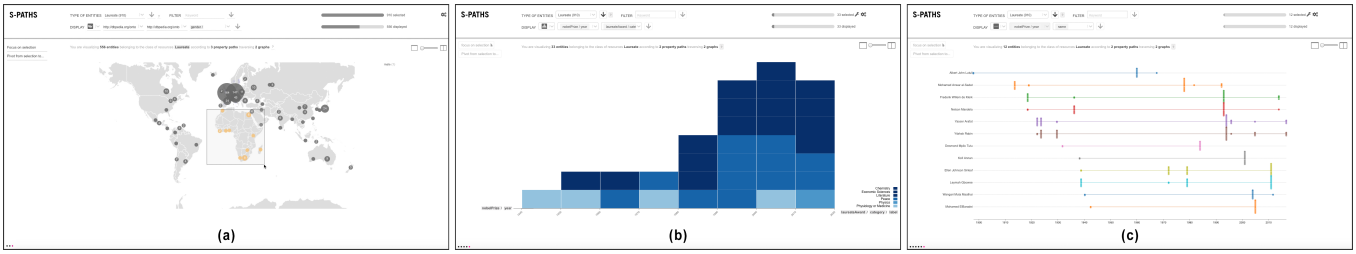

Figure 1: S-Paths can display multiple views on RDF resource sets, showing different properties along paths in the graph. Users can navigate between different resource sets by selecting a subset or pivoting. From (a) a map view of Nobel laureates, selecting those born in Africa and focusing on them yields (b) a histogram showing award year and category (Physics, Literature, *etc.*) for the corresponding laureates. (c) A timeline of events for laureates in the Peace category.

Table 1: Characteristics of semantic paths.

Characteristic	Description
category	one of: `datetime`, `geographical coordinate`, `image`, `number`, `text` or `URI`
depth	number of hops (statements) from the subject to the final property
coverage	percentage of resources in the set for which this path actually exists
count	total number of values for the property at the end of the path, over all resources
unique count	number of unique values for the property at the end of the path, over all resources

S-Paths smoothly animates transitions between views [7] when the two views have entities in common. This provides some basic level of perceptual continuity that contributes to minimizing the cognitive cost of relating the current view to the previous one.

The system also supports the juxtaposition of two consecutive views, as well as brushing and linking between those views: selecting elements in one view immediately highlights the corresponding elements in the other view, further helping users relate views. The same space-filling strategy as above is used to handle brushing & linking between aggregates.

S-Paths also keeps track of all past views and represents them as dots forming a basic timeline displayed in the bottom left corner of the interface. Clicking on one of these dots reverts to the corresponding view, enabling users to easily backtrack.

3.1.3 Matching Algorithm. After having iterated over the collection of views, S-Paths builds a list of optimally-configured views, retaining only the best-scoring paths for each view (Figure 2, 3rd column). It then assigns a score to each view using another normalized weighted average, of:

- *configuration quality*: average of associated path scores;
- *preference*: each type of view has a score which indicates preferences for some types over others based on, *e.g.*, their familiarity or concreteness. As this is subjective and application domain-dependant, these scores can be edited in a configuration file;
- *number of dimensions*: support for more dimensions implies more opportunities for the simultaneous visual mapping of properties.

S-Paths then selects the top-scoring view according to this weighted average, and configures it with the top-ranked semantic paths (Figure 2, 4th column). Lower-ranked paths matching this view can be selected using the *dimensions* menus. The *view* menu lets users switch to any other view compatible with the resource set.

The view reconfiguration capabilities let users change *what dimensions* of resources in the current set are visualized, and *how* they are visualized. Users can also restrict *what resources* to visualize by making direct selections in the currently displayed view: clicking on individual items and aggregates, performing rubber-band selections of contiguous elements, selecting ranges by, *e.g.*, clicking a particular bin on the x-axis of a histogram to select all items in that bar. They can also combine multiple, non-contiguous selections by holding a modifier key (Shift), as in popular graphics-oriented applications such as presentation programs and graphics editors. Once such a sub-selection has been made, users can turn it into the new resource set to explore. The process can be repeated iteratively. Combined with the automatic aggregation of resources along the chosen dimensions, which only occurs when the resource set is too large, this selection mechanism provides users with means to effectively zoom-in on part of the data and get details on demand [13].

3.2 Structural overview of a Linked Dataset

While S-Paths was focused on sets of resources sharing the same type, the second research project of this PhD, explores a broader context, aiming to combine the concept of semantic paths with other pseudo-structural elements, to give an overview of a dataset, and of its links to other datasets.

4 METHODOLOGY

Persona. We started by organizing two workshops, one with 9 Linked Data experts, and another with 7 lay users interested in Linked Data. From these workshop, combined with informal observations during French National Library's hackathon and other Linked Data community events in Paris, we derived 3 persona characters: a Linked Data publisher, a Linked Data reuser, and a lay user interested in the content of Linked Datasets. We used these persona as a leading thread through the development of proofs of concept, to keep in mind relevant user tasks and concerns. We also rely on them to define use cases, and identify relevant users for surveys and studies.

Interviews and Participatory Design. For our second project, we are conducting a series of interviews with experts to better characterize what they need to know about their datasets. We will follow

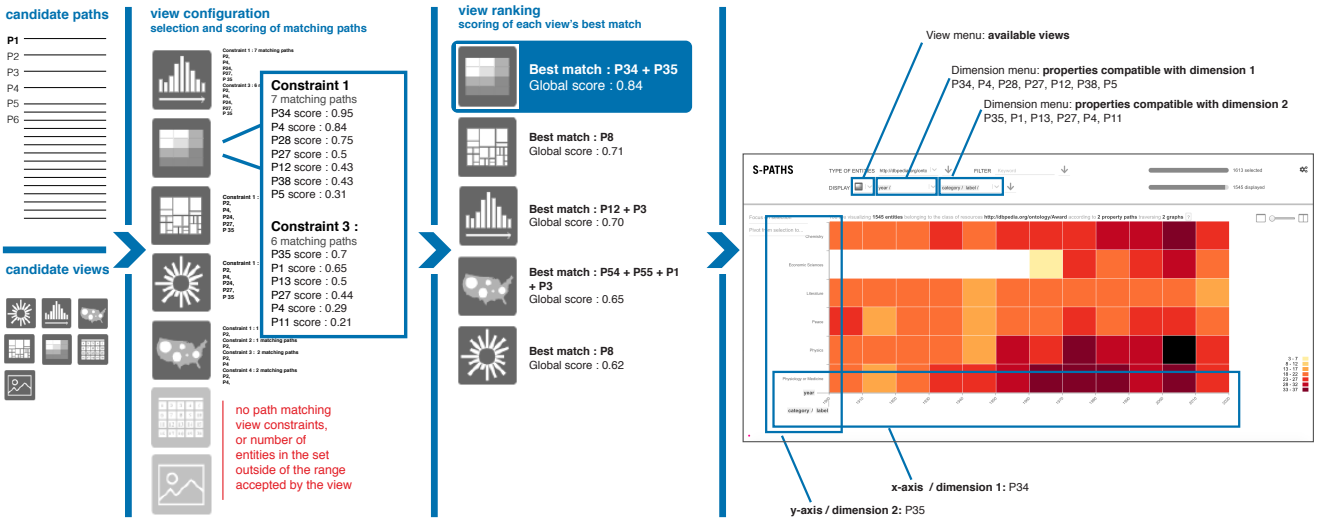

Figure 2: S-Paths' process for generating a default view.

a participatory design approach together with a few of them, who agreed to test our tools and give feedback all along the design and development process. The code will be partly distributed, and run on the endpoints of participants.

Proofs of Concept. We develop high-fidelity prototypes to give proofs of concept of our approach, and evaluate them. Their code is open source. For S-Paths we first made an API to identify and characterize sets of entities and the corresponding semantic paths. Then we started a light version of the visualisation system with two view components, to set up the generic query and transition mechanisms, and then we progressively added other views. The dataset used for development was the Nobel Prize Linked Dataset, which contain 85 797 triple statements, representing 15 classes of ressources. To demonstrate that our tool can work across several graphs, we created a small graph extracted from DBpedia, containing images and geolocations for entities mentioned in the Nobel Dataset, which represents 2 234 triple statements. In a second step we adapted the system to make it work on the French National Library's dataset. Experts at the Bnf extracted a sample that they considered representative, containing 32 106 950 statements. Then we tried the tool on several other datasets from the Linked Data cloud.

Qualitative Evaluation. We are currently running a series of qualitative evaluations, with users representative of each of our persona categories. We demonstrate the tool, and ask users to perform discovery tasks, using a think-aloud protocol. We also collect their feedback in a semi-structured interview. We wish to evaluate if the tool enables to discover new information, and to efficiently understand and remember it.

We are also deploying a version of s-paths at the French National Library to run a longitudinal observation.

5 RESULTS

User Perspective. Though the evaluation of S-Paths is still running, we can already share some observations. Lay users were able to use the tool without knowing about graph databases or what a path in a graph is. However, they were bothered in the navigation by the limitations due to Linked Data: they expected the interface to be more responsive, and complained about not getting previews and rollovers for selections. On the contrary, experts easily understood that such limitations are due to the data's structure and distributed nature, and expressed enthusiasm for the possibility to get overviews. All of them were fully focused on the navigation and 20 minutes appeared very short to them. They would have wanded more time to practice with the tool, and to discover the data. We asked data reusers to explore a dataset as they would have at the start of a hackathon. Our first participant told us that the tool could automatically do most of the work he would usually do when discovering a new dataset. *"When you engage in a hackathon, what you are looking for is irregularity in the data, and this tool finds them and points them out"*. This user noticed that the map showing the co-laureates' affiliation showed surprisingly very few entities in the USA, which revealed that most of the laureates affiliated with US institutions win the prize as a single laureate. This example is particularly interesting, since it would otherwise have demanded an important analysis work from an expert to reach this conclusion, and maybe he would never have had the idea to plot a map with co-laureates only. S-Paths, following the paths, and thus being able to relate geolocations from DBpedia with places across the two graphs, made this starightforward. While developing the tool – using it ourselves and informally demonstrating it to colleagues – we had noticed that most of its visualisations provoke high-level thoughts. For example, a user spotted that there was only one laureate born in the 90's and that she was a female. Another one remarked that there were little shared awards in literature. And indeed, the tool uses aggregation to present the most readable

overview at any step, and, as Bertin states it, *"Useful information comes with cluster"* [2]. Wondering if this would have an impact on memorising and learning, we ended our evaluation with a MCQ. All users – except one, who also reported having no interest in the topic – were able to answer general questions about Nobel prizes (namely, the number of categories, the number of laureates, the time range and the percentage of women), that they would not have been able to answer for the most part before the experiment. A side-effect of the systematic use of aggregation is that it gives the impression of presenting the whole set even when the properties displayed only partially cover that set. Although we clearly indicate coverage in the the top-right corner of the interface (see Figure 1), and even though the tool favors well covered properties, a user pointed out that this should maybe be even more visible.

Scalability and adaptability. So far we tested S-Paths with 7 datasets of various sizes and characteristics: Nobel[2], Data BnF[3], ELI[4], RISM[5], John Peel Sessions[6], Amsterdam Museum[7], Linked Movie DB[8]. Two of them, ELI and RISM did not feature any `rdf:type` statement. Since such statements are the starting point for our tool, we had to generate them in a small adjacent graph according to the model. A limitation of the tool is that it needs to find enough properties in a readable dimensions on the whole set of entities to offer views. At the moment, given the views available and their configuration, this means either dates, numbers, geolocations in any dimensions, or text or URIs with a number of unique values at the end of the path lower than 100. Several classes of resources in the John Peel Sessions dataset did not match this condition. This could be overcome by designing views that group text values and URIs by their first characters(s), thus enabling aggregation for any path. The view would be rated low, because this is not the most meaningful aggregation, but it could serve as an entry point when no other is available.

6 FUTURE WORK

S-Paths, by putting a resource in the context of other similar resources, aims at presenting Linked Data in a manner that is understandable by humans. Continuing with this approach, and taking one more step back, my second research question is: could we identify, analyze and display elements constituting a (pseudo) structure to give an overview of a RDF dataset ?

Indeed, talking with experts during workshops and more informal meetings, we realized that they had difficulties gaining a general understanding of their datasets. In particular, data producers themselves do not seem to have a clear idea about the number of entities in their dataset that are linked to another dataset. Detecting precise links between datasets is not a simple task. If the rest of the analysis can be performed on each SPARQL endpoint independently, links require to query at least two SPARQL endpoints at a time. The support for federated queries and the availability of other endpoints being uncertain parameters, link detection might have to be piloted from a central server. We are currently working on these specific questions.

Finding a way to efficiently detect links between datasets would also be of interest for S-Paths, since it would allow to turn it into a full linked data browser. For now, it is limited to the exploration of multiple graphs hosted behind a single endpoint. Paths traversing datasets could be combined with the pivoting functionality, which already exists in the tool. Pivoting means changing the focus from one set of entities to another along the current path: one can choose to keep the constraints of the current subset or not. This would enable hopping from dataset to dataset. This would imply to find a standard way to communicate the analysis performed in the endpoint.

[2]http://www.nobelprize.org/about/linked-data-examples/
[3]http://api.bnf.fr/dumps-de-databnffrwearecurrentlyworkingona10percentsample
[4]http://data.public.lu/fr/datasets/legilux-journal-officiel-du-grand-duche-de-luxembourg/
[5]https://old.datahub.io/dataset/rism
[6]http://raimond.me.uk/resources/peel.tar.gz
[7]https://bitbucket.org/biktorrr/amlod/downloads/
[8]https://old.datahub.io/dataset/linkedmdb

REFERENCES

[1] Tim Berners-Lee, Yuhsin Chen, Lydia B. Chilton, Dan Connolly, Ruth Dhanaraj, James W. Hollenbach, Adam Lerer, and David Sheets. 2006. Tabulator : Exploring and Analyzing linked data on the Semantic Web.

[2] Jacques Bertin. 1977. *La graphique et le traitement graphique de l'information.* Number 91 (084.21) BER.

[3] Adrian Braşoveanu, Martab Sabou, Arnoa Scharl, Alexandera Hubmann-Haidvogel, and Daniela Fischl. 2016. Visualizing statistical linked knowledge for decision support. *Semantic Web Journal* 8, 1 (2016), 113–137. https://doi.org/10.3233/SW-160225

[4] Josep Maria Brunetti, Sören Auer, Roberto García, Jakub Klímek, and Martin Nečaský. 2013. Formal Linked Data Visualization Model. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS '13)*. ACM, Article 309, 10 pages. https://doi.org/10.1145/2539150.2539162

[5] Aba-Sah Dadzie and Emmanuel Pietriga. 2017. Visualisation of Linked Data - Reprise. *Open Journal Of Semantic Web* 8, 1 (2017), 1 – 21. https://doi.org/10.3233/SW-160249

[6] Alvaro Graves. 2013. Creation of Visualizations Based on Linked Data. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics (WIMS '13)*. ACM, Article 41, 12 pages. https://doi.org/10.1145/2479787.2479828

[7] Jeffrey Heer and George Robertson. 2007. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1240–1247. https://doi.org/10.1109/TVCG.2007.70539

[8] Philipp Heim, Steffen Lohmann, Davaadorj Tsendragchaa, and Thomas Ertl. 2011. SemLens: Visual Analysis of Semantic Data with Scatter Plots and Semantic Lenses. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics '11)*. ACM, 175–178. https://doi.org/10.1145/2063518.2063543

[9] Emmanuel Pietriga. 2002. IsaViz: a Visual Environment for Browsing and Authoring RDF Models. In *WWW '02: the 11th World Wide Web Conference (Developer's day)*. ACM.

[10] Emmanuel Pietriga. 2006. Semantic Web Data Visualization with Graph Style Sheets. In *Proceedings of the 2006 ACM Symposium on Software Visualization (SoftVis '06)*. ACM, 177–178. https://doi.org/10.1145/1148493.1148532

[11] Emmanuel Pietriga, Christian Bizer, David Karger, and Ryan Lee. 2006. Fresnel: A Browser-independent Presentation Vocabulary for RDF. In *Proceedings of the 5th International Conference on The Semantic Web (ISWC'06)*. Springer-Verlag, 158–171. https://doi.org/10.1007/11926078_12

[12] m. c. schraefel and David Karger. 2006. The Pathetic Fallacy of RDF. In *Semantic Web User Interface Workshop ISWC*.

[13] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the Symposium on Visual Languages (VL '96)*. IEEE, 336–343. https://doi.org/10.1109/VL.1996.545307

[14] Klaudia Thellmann, Michael Galkin, Fabrizio Orlandi, and Sören Auer. 2015. LinkDaViz – Automatic Binding of Linked Data to Visualizations. In *The Semantic Web - ISWC 2015*. Springer International Publishing, Cham, 147–162.

[15] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. on Visualization and Computer Graphics* 22, 1 (2016), 649–658. https://doi.org/10.1109/TVCG.2015.2467191