◼ VISUALIZATION OF TYPED LINKS IN LINKED DATA

by Georg Neubauer

Abstract: *The main subject of the work is the visualization of typed links in Linked Data. The academic subjects relevant to the paper in general are the Semantic Web, the Web of Data and information visualization. The Semantic Web, invented by Tim Berners-Lee in 2001, was announced as an extension to the World Wide Web (Web 2.0). The actual area of investigation concerns the connectivity of information on the World Wide Web. To be able to explore such interconnections, visualizations are critical requirements as well as a major part of processing data in themselves. In the context of the Semantic Web, representation of information interrelations can be achieved using graphs. The aim of the article is to primarily describe the arrangement of Linked Data visualization concepts by establishing their principles in a theoretical approach. Putting design restrictions into context leads to practical guidelines. By describing the creation of two alternative visualizations of a commonly used web application representing Linked Data as network visualization, their compatibility was tested. The application-oriented part treats the design phase, its results, and future requirements of the project that can be derived from this test.*

Keywords: *Visualization; Visual representations; Linked Open Data; Analytics; Semantic Web*

VISUALISIERUNG VON TYPISIERTEN LINKS IN LINKED DATA

Zusammenfassung: *Das Themengebiet der Arbeit behandelt Visualisierungen von typisierten Links in Linked Data. Die wissenschaftlichen Gebiete, die im Allgemeinen den Inhalt des Beitrags abgrenzen, sind das Semantic Web, das Web of Data und Informationsvisualisierung. Das Semantic Web, das von Tim Berners Lee 2001 erfunden wurde, stellt eine Erweiterung zum World Wide Web (Web 2.0) dar. Aktuelle Forschungen beziehen sich auf die Verknüpfbarkeit von Informationen im World Wide Web. Um es zu ermöglichen, solche Verbindungen wahrnehmen und verarbeiten zu können sind Visualisierungen die wichtigsten Anforderungen als Hauptteil der Datenverarbeitung. Im Zusammenhang mit dem Sematic Web werden Repräsentationen von zuhammenhängenden Informationen anhand von Graphen gehandhabt. Der Grund des Entstehens dieser Arbeit ist in erster Linie die Beschreibung der Gestaltung von Linked Data-Visualisierungskonzepten, deren Prinzipien im Rahmen einer theoretischen Annäherung eingeführt werden. Anhand des Kontexts führt eine schrittweise Erweiterung der Informationen mit dem Ziel, praktische Richtlinien anzubieten, zur*

Vernetzung dieser ausgearbeiteten Gestaltungsrichtlinien. Indem die Entwürfe zweier alternativer Visualisierungen einer standardisierten Webapplikation beschrieben werden, die Linked Data als Netzwerk visualisiert, konnte ein Test durchgeführt werden, der deren Kompatibilität zum Inhalt hatte. Der praktische Teil behandelt daher die Designphase, die Resultate, und zukünftige Anforderungen des Projektes, die durch die Testung ausgearbeitet wurden.

Schlüsselwörter: *Visualisierung; Visuelle Repräsentation; Linked Open Data; Analyse; Semantic Web*

The aim of this work is to construct and evaluate different graphic representations of Linked Data that can be compared in terms of better handling and simultaneously as a main question have a more suitable design as a result.

This leads to three main research questions:

1. *How can typed links in Linked Data be visualized for a broader number of users?*
2. *Which graphic representation do typed links have in Linked Data?*
3. *Is there an interrelation between the visualization and the functionality of graphs when it comes to interpretation?*

1. Data and Information

There are two important parts of processes of data transformation concerning the exchange of information[1]:

– *Formalization in the beginning*
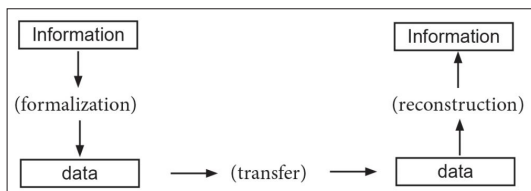– *Reconstruction after data transfer*

Fig. 1: Exchange of information[2]

Data preservation necessitates data storage. Nowadays data can be stored digitally using computers, magnetic memory media and the Internet.

2. Ontology

Data formalization is generally referred to as an ontology. It derives from the Greek *onto* (being) and *logia* (written or spoken discourse). In information technology, ontology is the **working model of entities and interactions** in a particular domain of knowledge or practices, such as electronic commerce. A good definition is the following sentence. "Ontology is a formal specification of a shared conceptualization"[3].

The construction of ontologies is a cyclic process and consists of the following steps[4].

– *Definition of area and domain*
– *Analysis of available ontologies for possible reutilization*
– *Identification of relevant terms*
– *Production of the draft hierarchy*
– *Definition of relations and formalization of classes*

3. The Web

The web builds on a gigantic amount of information and is accessible via the Internet. It is made up of three components:

– *HTML (Hyper Text Markup Language)*
– *HTTP (Hyper Text Transfer Protocol)*
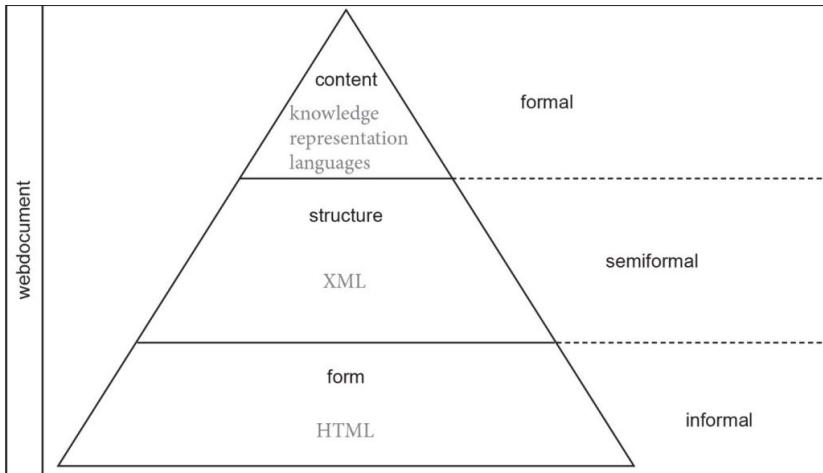– *URL (Uniform Resource Locator)*

Fig. 2: Levels of structuralization of information on the web[2]

The Semantic Web is a suggestion for unorganized growing on the web. The idea of the Semantic Web is that information depending on content and relations can be processed. In order to bind information it is necessary to put the information in a confirmed, structured form.

4. RDF

The Resource Description Framework (RDF) was developed for representing information on the web. RDF is a basic language for the description of structured information on the web and of course for the exchange of information[5][6]. URI = scheme ":" authority [path]["?" query]["#" fragment]

RDF is used to formulate logical statements. ***Every resource is assigned a URI***. RDF is described by a number of triples (RDF graphs) which is often indicated by arrows[6].

There are three components of these arrows:
  – *The subject (is a URI or a "Black Node")*
  – *The predicate (is a URI)*
  – *The object (can be a URI, a "Black Node" or a "Literal")*

A Literal is used to identify a simple data type (numbers, data and alphanumeric characters).

A superset of RDF is RDF N3 (Notation 3). It extends the RDF data model by adding formulae, variables, logical implications, functional predicates and provides a textual syntax alternative to RDF/XML.

5. SPARQL

SPARQL and the RDF Query Language are semantic query languages for databases and constitute key technologies for the Semantic Web. They can retrieve and manipulate data stored in the RDF format[7].

SPARQL allows for a query to consist of ***triple patterns, conjunctions, disjunctions, and optional patterns***[8].

Implementations for multiple programming languages exist. There are tools that allow connection and semi-automatic construction of a SPARQL query for a SPARQL endpoint (i.e.: ViziQuer). Additionally, there are tools that are able to translate SPARQL queries to other query languages (e.g.: SQL).

6. OWL

The W3C Web Ontology Language (OWL) is a computational logic-based

Semantic Web language and specification of the World Wide Web consortium that is intended to *represent complex knowledge about things, groups of things, and relations* between things. OWL documents (ontologies) can be published in the World Wide Web and may refer to or be referred to from other OWL ontologies. OWL is part of the W3C's Semantic Web technology area, which includes RDF, RDFS, SPARQL, etc.

The current version of OWL is OWL 2 (published in 2012) and is defined by five core specification documents describing its conceptual structure: the primary exchange syntax (RDF/XML), two alternative semantics (direct and RDF-based), and conformance requirements. In addition, three specification documents contain optional features that may be supported by some implementations: the language profiles and two other syntaxes (OWL/XML, Manchester).

7. Typed Links

The term "typed link" can be explained as a link relation and has a descriptive feature connected to a hyperlink that defines the type of the link or the relationship between the source and target resources[9].

RDF typed links are essential in Linked Open Data (LOD) datasets to identify the relationship type of RDF-triples and thus contribute to automatic processing of machine-readable statements of the Semantic Web. Typed links in RDF are referred to as the value of the rdf:type property, defining the relationship type using vocabulary terms or definitions from LOD datasets[9].

The mentioned *URIs of a relation are connected in the form of graphs*. Such implementations therefore preserve the semantic variability of content types by using the same syntax in a similar way. In Linked Data, a set of information uses this syntax, which provides an interconnection via the main three components of natural languages (subject, predicate, object). Figure 3 describes the main dependencies.

To offer further prospects of the functionality, *the predicate defines the relation of an RDF-triple*. This will be treated later on.
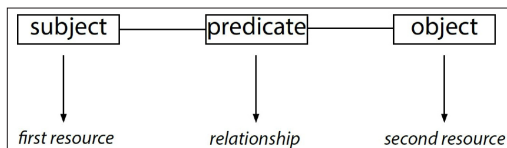

Fig. 3: RDF-triple (general)

8. Visualization

For the process of analysis, evaluation and identification of datasets visualizations are based on computational graphic systems[10].

Computer-based visualization systems provide visual representations of datasets intended to help people carry out some tasks better. These visualization systems are often – but not always – interactive. Resource limitations include the *capacity of computers, of humans, and displays*. The area of possible visualization system design is huge and full of tradeoffs[11].

9. Information visualization

Information visualization, a term coined by the User Interface Research Group at Xerox PARC, concentrates on the use of computer-assisted tools to explore a large number of abstract data. Useful applications of information visualization in computer programs involve selection, transformation and representation of abstract data in a form that facilitates human interaction for exploration and understanding. For this reason, information has to be *prepared, condensed and filtered* to enable effective utilization. In addition to this, perception, psychology and linguistics have to be integrated to deliver a genuine increase in surplus value for users. Central aspects of information visualization are the dynamics of visual representation and interactivity. Advanced technologies enable the user to modify visualization in real-time.

There are many tools available to create graphic representations of data nowadays. The integration of different scientific disciplines – like linguistics, information design, psychology of perception – necessitates preparation to render it generally understandable[12].

10. Knowledge visualization

Knowledge visualization is used to represent and interpret complex information, text in particular, at the intersection of knowledge, art and cultural heritage. The use of visual representations to transfer knowledge between humans aims to improve the transfer of knowledge by using computer and non-computer-based visualization methods in a complementary fashion[13].

While information visualization concentrates on the use of computer-assisted tools to gain new insight, knowledge visualization focuses on transferring insight and creating new knowledge in groups.

11. Visual communication

Visual communication is the communication of ideas through the visual display of information. It includes alphanumeric or artistic signs and electronic resources. Latest research in the field mainly concerns web design and graphically-oriented usability.

Visual analytics focuses on human interaction with visualization systems as part of a larger process of data analysis. Its focus is on human information discourse within dynamically changing information spaces. Visual analytics research concentrates on support for perceptual and cognitive operations that allow users to become aware of the expected and find out the unexpected in complex information spaces.

The concept visualization type deals with the meaning of a picture within a specific social and/or cultural context.

Visual perception laws include "Gestalt laws" which are commonly used.

Position as the most important part of design principles was researched by McKinlay[24].

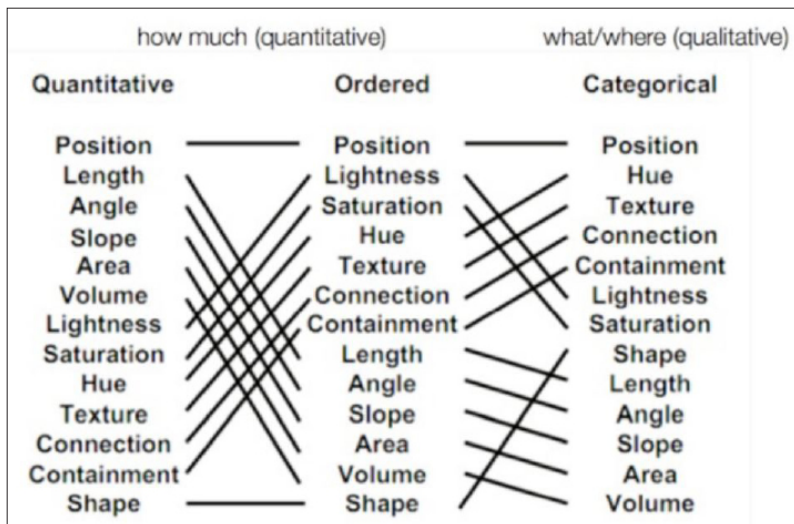The figure shows a distinct change in the ranking of these tasks from quantity to quality.

Fig. 4: Ranking of perceptual tasks[14]

To display the relevance of information components, positioning is highly important and therefore was examined in connection with the ranking of

the layout of all other graphical elements. As the figure shows, this layout changes when the quality of content increases.

12. Conceptual Models in Visualization

The science of visualization is in its infancy and involves the combination of graphics, imaging, data management and human perception. Two divergent trends in visualization have emerged. While visualization is often available and practical, the supporting technology is usually not designed for these applications. The second trend stems from the data glut problem. This means that *typical ad hoc approaches do not scale up to accommodate large, complex problems*. Despite competing requirements, access to data is the common barrier. A first step is to decompose visualization into a set of transformations that can highlight these limitations by defining a conceptual model and develop a taxonomy.

13. Linked Data Visualization Model

Applying information visualization techniques to the Semantic Web helps users to explore large quantities of data and interact with them. The main objectives of information visualization are to *transform data into and present data as a visual representation* in such a way that users can obtain a better understanding of it[15].

Most existing work related to visualizing RDF is focused on concrete domains and concrete data types. The Linked Data Visualization Model (LDVM) is a formal base that exploits the Linked Data principles to ensure interoperability and *compatibility of compliant analytic and visualization components*. In short, LDVM allows users to create data visualization pipelines that consist of four stages:
- *Source data*
- *Analytical abstraction*
- *Visualization abstraction*
- *View*

The aim of LDVM is to provide means of creating reusable components at each stage that can be put together to create a pipeline even by non-expert users who do not know RDF. The *typical use case for visualization abstraction is to facilitate reuse* of existing analyzers and existing visualizers that work with similar data, only in different formats. For that purpose, LDVM uses a transformer. In the view stage, data is passed to a visualizer that creates

a user-friendly visualization. The components, when connected, create an analytic and a visualization pipeline which, when executed, takes data from a source and transforms it to produce a visualization at the end.

Linked Data browsers such as Tabular or Explanator allow users to navigate the graph structures and usually display property-value pairs in tables, but offer no broader view of the dataset[16][17].

Fresnel is a vocabulary for rendering RDF to HTML, but its focus is on rendering instance data rather than on creating visualizations[18].

Rhizomer provides an overview of the datasets and allows interacting with data through information architecture components such as navigation menus, breadcrumbs and facets[19]. It also provides visualizations such as maps and timelines.

DERI Pipes is an engine and graphical environment for general web data transformations and mashups[20]. It is not intended for lay-persons and requires software expertise.

The Linked Data Visualization Model (LDVM) uses the Data State Reference Model (DSRM) proposed by Chi as a conceptual framework.
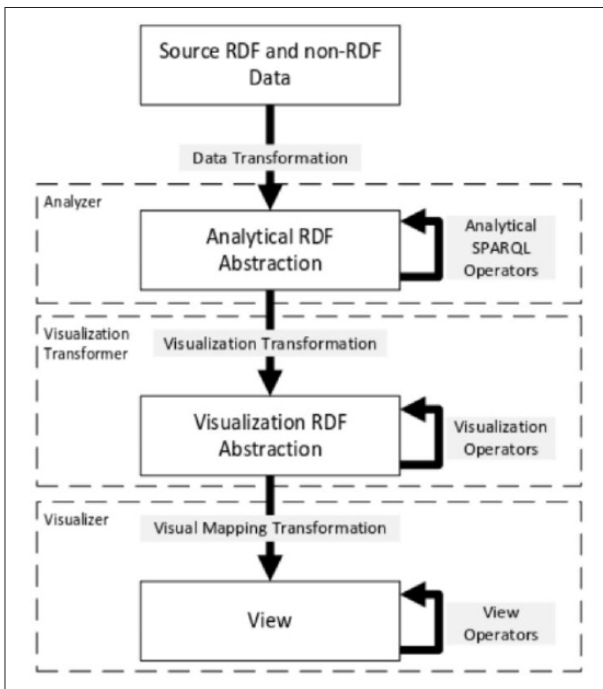

Fig. 5: High level LDVM overview[15]

DSRM describes the visualization process in a generic way[15].

14. Prototyping the new visualization

To gain drafts of graph visualizations it is necessary to analyze why the existing layout standards are commonly used. The most important insights to construct visualization are gained by the components a visualization exhibits, and by their *noticeable design restrictions*. Components and restrictions concerning the visualization of Linked Data relations are itemized in the following list.

Components
   – *Nodes:* labeled components (signifiers that are extracted parts of the URLs of RDF-triples)
   – *Edges:* visual lines between these labeled components (graphs that show the dependencies of a relation by orientation)

Restrictions
   – *Node size:* restricts the number of nodes displayed
   – *Node position:* restricts the arrangement of nodes displayed
   – *Edge length:* restricts the space between nodes
   – *Edge formation:* restricts readability of information
   – *View size:* restricts the space of viewable information
   – *Node control:* restricts operations of manipulating nodes
   – *View control:* restricts operations of manipulating view

Heuristics of general design principles treating application views give vertices for possible visualizations.

Figure 6 shows the process of such a refinement of one of these drafts.

The figure shows the first draft of the adapted visualization. The relations are arranged in a scrollable list here. The draft also includes an option to toggle the content of the abstract of each subject and object in the representation view.

From the input nodes (subjects) or the given nodes view additional found nodes are similar to objects in grammar. It is then possible to *filter identical predicates for direct relations* of two or more given input nodes while preserving the completeness of the path.

To normalize the relations I suggested limiting *identical predicates to a unique representation* that replaces several appearances of nodes having similar predicates in the relations. Semantically this method preserves the

meaning of information given by the relations. If the predicates pertaining to the relation have identical terms *these predicates can be normalized and the information is preserved*.
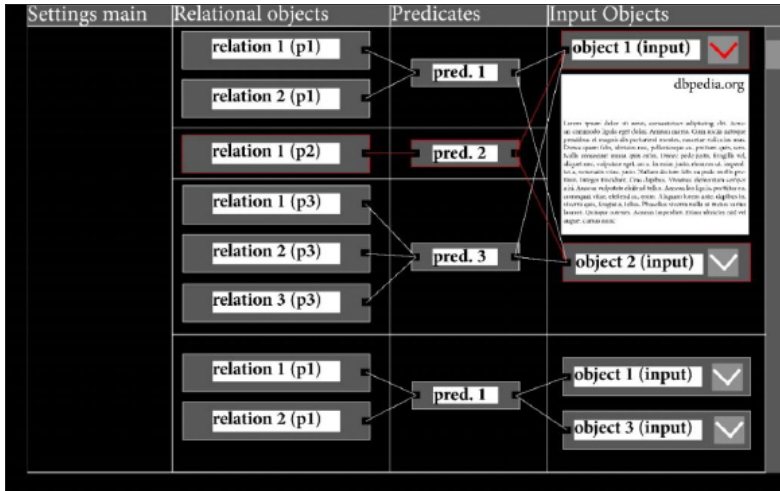

Fig. 6: Draft visualization

15. Introducing the views

Node-link view (RelFinder) – http://www.visualdataweb.org/relfinder.php

RelFinder is a graph visualization software based on "Adobe Flash" showing relations to specific objects or subjects of SPARQL queries using simple input fields. It calculates the relationships between two or more input phrases.

The software is based on flex framework 3.5 including the Spring graph visualization core. The Spring graph visualization enables software to link nodes (components of data) providing a network view.

Relations in RelFinder consist of RDF-triples representing the subject, predicate and object using the string component of the rdfLabel already described as extracted part of these URLs.

Linear view (RelFinder)

The linear view visualizes the relations in a list. Each node leads to a unified predicate and connects the given nodes over this predicate with the found nodes of a search query.
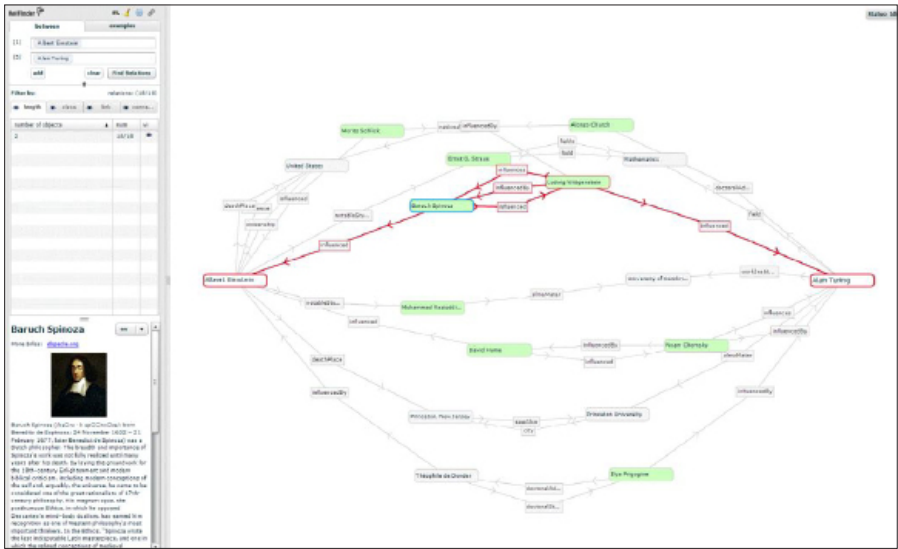
Fig. 7: RelFinder (node-link view)

Fig. 8: RelFinder (linear view)

Three main columns stand for three parts of a relation. The first column shows the given nodes, the second one the predicates while the third column is preserved for the found nodes of the relations. The button with the arrow next to the nodes opens an abstract (a box including some short information about the object).

Radial view (RelFinder)

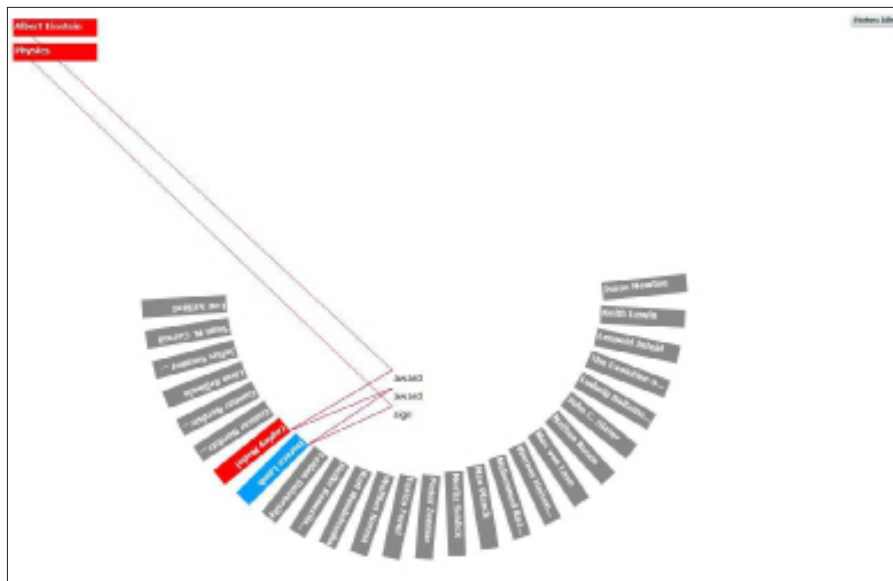The radial view visualizes the found nodes' positions in a (semi)circle.

Fig. 9: RelFinder (radial view)

To describe the main points of reprogramming an equal list for the implementation of the list view is given below.

- *Normalizing predicates:* **Unification of similar predicates**
- *Toggleable abstracts:* **Show and hide abstracts in the view**
- *Repositioning nodes as list/(semi)circle:* **Set ascendant x- and y-positions of nodes**
- *Highlighting specific paths:* **Hiding paths of unselected sub-relations**
- *Controllable components:* **Applying scrollbars to the view and drag nodes vertically**
- *Layout of components:* **Changing the visualization of labels and nodes**

Both views differ from the standard visualization of a node-link view common in Linked Data visualizations and therefore had to be tested in a prepared workflow in order to gain knowledge about interaction advantages and corresponding design principles.

16. Methods of testing

To find relevant methods concerning these questions the paper "*Task taxonomy for graph visualization*"[21] proved very helpful since it provided *essential tasks for the specification of results*. On the basis of the following listing the executed tasks describe the propositions for the methods.

Topology-Based Tasks
  – *Adjacency (direct connection):* **Find the set of nodes adjacent to a node**
  – *Connectivity:* **Identify connected components**

Attribute-Based Tasks
  – *On the nodes:* **Find the nodes having a specific attribute value**
  – *On the links:* **Given a node, find the nodes connected only by certain types of links**

Browsing Tasks
  – **Follow a given path**

Test subjects

20 test subjects participated in the test. To describe their general characteristics a survey was held in addition to the test tasks. Exactly 10 of the test subjects were male and 10 female. 15 of the test subjects where between 21 and 39 years old.

11 of them had a general qualification, a bachelor's or a master's degree.

Materials

The user test took place in the usability laboratory of the University of Applied Science St. Pölten, using an eye tracking device from SMI (Sensomotoric Instruments).

It included the 2 different types of questions: one for navigation and one for information included in the abstract. To guide the test subjects

a textual briefing was given for each of the tasks before showing the test screen. The maximum duration of any task was set to 3 minutes, after which the next briefing faded in. The format of the eye tracking data was a *screen recording which included the scan path of the test user and the mouse clicks on the test screen*.

17. Test results

Each test subject had to learn to understand the visualization given and was asked to provide answers in text form. The results showing the correctness of the answers of all test subjects are shown below.
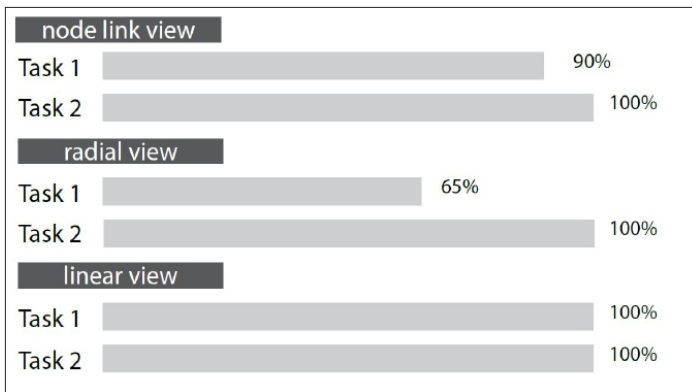

Fig. 10: Correctness of task answers

The second task depending on the information of the abstracts was always answered correctly.

The information specified by each test subject was denoted as true or false. Therefore the results clearly show that the answers given using the linear view were always right while the answers using the node-link view were wrong in 2 cases. The radial view had 7 wrong answers.

The results of task 1 that the following figure shows include the time the test subjects needed to find and click one node. The results as quartiles show that the *nodes were found fastest using the radial view and the linear view*.

Task 2 shows the time a person needed to find the information provided by the abstract. The linear view was the only view where the abstract had to be opened by clicking a button. Nevertheless the linear view could be handled very quickly and the concept was understood very easily.
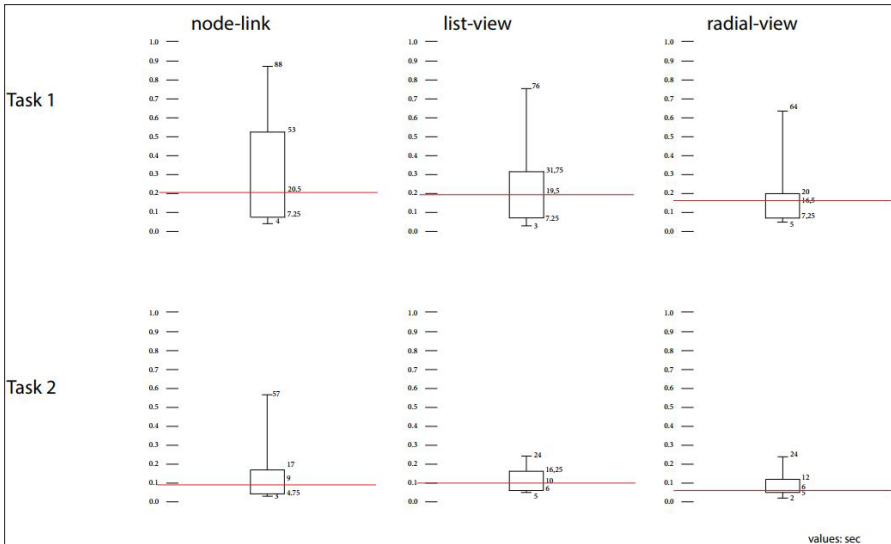
Fig. 11: Times (sec) to solve task 1 (find a node) and task 2 (abstract info)

In comparison to the node-link view we clearly see that the time needed for solving the first task in the *linear view and the radial view has very similar minima*. In the third quartile we see that the test subjects needed less time in both new views.

The main survey treated the three given visualizations in terms of preference, clearness and understanding and handling. According to this information *the visualization type preferred by the test subjects could be evaluated in relation to its clarity and handling*. To inform the test subjects about the visualization types a printed poster including a figure for each visualization type and a label naming these views was offered in order to make the test subject's choice easier. The charts below describe the results.

Question 1: Did you know Semantic Web and Linked Data before? (War Ihnen Semantic Web und Linked Data vor der Testung ein Begriff?)

Fig. 12: Semantic Web publicity

Question 2: Which version of RelFinder would you prefer? (Welche Version von RelFinder würden Sie bevorzugen?)

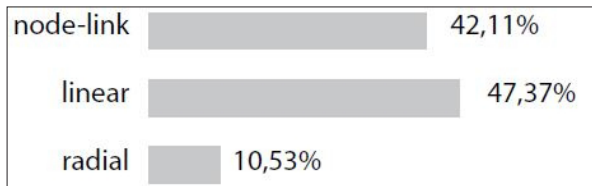|	
node-link	42,11%
linear	47,37%
radial	10,53%

Fig. 13: Preference of visualization

Question 3: In your opinion, which view of RelFinder is more clearly arranged? (Welche Version von RelFinder ist Ihrer Meinung nach bezüglich der Darstellung übersichtlicher?)

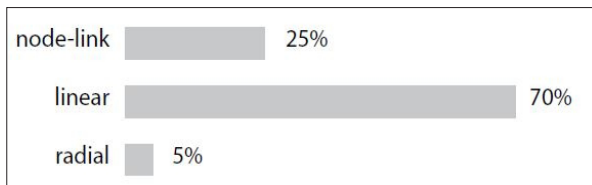|	
node-link	25%
linear	70%
radial	5%

Fig. 14: Clarity of visualization

Question 4: In your opinion, which version of RelFinder is easier to handle? (Welche Version von RelFinder ist Ihrer Meinung nach leichter handhabbar?)

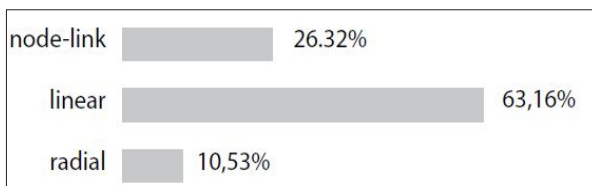|	
node-link	26.32%
linear	63,16%
radial	10,53%

Fig. 15: Clarity of visualization

18. Conclusion and future work

The criteria of good design were utilized and compared with the test and the answers of the survey. In *terms of handling, preference and clarity, the line-*

ar view's design was evaluated as best. This may be due to its *minimalistic concept, which reduces the noise of unfiltered paths and nodes by repositioning them*.

Faster handling for a broader number of users could be demonstrated for the radial and the linear view via the time the tasks took in comparison. The nodes were found very much faster than in the node-link view. Proof of concept, however, would demand a larger sample of test subjects.

The placement of the abstract as a toggleable container of the main view was often recognized intuitively and needed no further explanation. In times of responsive design, the vast number of at times very large screen resolutions profits immensely from toggleable information boxes to organize nodes.

Another find is that *no statement referring to the normalized predicates* of the linear view was given, which might indicate that this feature was not recognized. One interesting point is that the users construed Linked Data in a very abstract way. Most of the time they had difficulties to put the visualized sentence into words. In terms of the translation of information into another language some common Linked data predicates were misunderstood.

The *unordered loading of the nodes found was also criticized* by some test subjects. The test subjects suggested that the nodes have to be ordered alphabetically in the linear and the radial view especially.

The visualization of *nodes in the radial view should be turned 180° so that readability is better*. As mentioned, the performance of findability is very good in the radial view, but the visualization space for nodes positioned in such a way is limited.

Nevertheless a *graph oriented visualization of data presented as a list seems to have synergies with the presentation of regular lists* when it comes to user habits and user demands.

An improvement already implemented is the finalization of the radial view by adding graph directions represented as arrows. All data of the prototypes developed and the most important information is hosted on a github repository available here: https://github.com/geonb/Visualization-of-typed-links-in-Linked-Data.

Where the Semantic Web in general is concerned, the *grammar of relative clauses visualized as oriented graphs cannot be used exclusively in dependence on certain vocabularies that represent equal exchange of information*. A group of predicates like "communicate", "meet", "talk" *could be linked without including the direction of the graphs*.

To outline future work on the prototypes of RelFinder, it is necessary to follow the rules of visual representation as well as the results of this evaluation. Better design seems to include *hiding of unnecessary informati-*

on and better positioning of the nodes. To briefly illustrate what could be improved concerning the linear and the radial view, one might say that the found nodes (objects) of a *normalized list of predicates should be grouped around similar normalized predicates and could appear in an alignment centered around a clicked found node.* One topic in need of further investigation is optimizable graph length, i.e. the distance of the node lists between the nodes.

From the perspective of a developer, all the modern visualizations of Linked Data in connection to a model view controller concept should *prepare the result set of a query inside the controller completely* and draw the entire visualization after this calculation.

Finally, to preserve the semantic sense of the relationships of typed links, I suggest a presentation where the *composition of all relationships, including subjects and normalizable predicates dependent on sortable and filterable object groups, represents a result set as a vertically positioned node-list view*.

Dipl.-Ing. Georg Neubauer, BSc
Danube University Krems, Library
E-Mail: georg.neubauer@donau-uni.ac.at

Notes

1 N. Bartelme, *„Einführung in Geoinformatik".* Springer Berlin Heidelberg, 2005, pp. 1–42. DOI: http://dx.doi.org/10.1007/3-540-27201-1_1

2 Hernando Sáenz-Sánchez, *„Visualisierung von Linked Open Data mit Orts- und Zeitbezug",* Institut für Informatik der Freien Universität Berlin, 2012. http://www.inf.fu-berlin.de/inst/ag-se/theses/Saenz-Sanchez13-visual-geo.pdf

3 W. Borst, *"Construction of Engineering Ontologies for Knowledge Sharing and Reuse",* 1997, Centre for Telematics and Information Technology, University of Twente, Enschede-Noord, the Netherlands, 2006. http://eprints.eemcs.utwente.nl/17377/01/t0000004.pdf

4 Heiner Stuckenschmidt, *„Erstellen von Ontologien",* in Ontologien Konzepte, Technologien und Anwendungen, ser. Informatik im Fokus. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. 5, pp. 155–205.

5 T. Berners-Lee, R. Fielding, and L. Masinter, RFC 3986, *"Uniform Resource Identifier (URI): Generic Syntax",* 2005.

6 F. Manola and E. Miller, *"Resource description framework (RDF) primer",* W3C Recommendation, vol. 10, 2004.

7   Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak, *"Linking open data cloud diagram"*, 2014. http://lod-cloud.net/

8   Robin Cover, *"XML and Web Services In The News"*, xml.org. 6 October 2006. http://www.xml.org/xml/news/archives/archive.10062006.shtml

9   Dubost Karl, *"Web site metadata profile*: *favicon",* World Wide Web Consortium. October 2005. http://www.w3.org/2005/10/profile

10 Thomas A. Runkler, „*Visualisierung in Data Mining: Methoden und Algorithmen intelligenter Datenanalyse"*, 2010, pp. 35–55. http://link.springer.com/content/pdf/10.1007%2F978-3-8348-9353-6_4

11 T. Munzner, *"Visualization Design and Analysis: Abstractions, Principles, and Methods"*, 2012. http://web.cse.ohio-state.edu/~raghu/teaching/CSE5544/ClassLectures/PDF-old/02-Chapter1.pdf

12 C. Moser, *"Informationsdesign, in User Experience Design"*, ser. X.media.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 105–120. http://www.springerlink.com/index/10.1007/978-3-642-13363-3; http://www.springerlink.com/index/10.1007/978-3-642-05404-4

13 Remo Aslak Burkhard, *"Tube map: Evaluation of a visual metaphor for interfunctional communication of complex projects"*, 2004. https://www.researchgate.net/publication/229100758_Tube_Map_Evaluation_of_a_Visual_Metaphor_for_Interfunctional_Communication_of_Complex_Projects

14 J. D. Mackinlay, *"Automating the Design of Graphical Presentations of Relational Information"*, ACM Transactions on Graphics 5(2):110–151, 1986. https://www.researchgate.net/publication/220184008_Automating_the_Design_of_Graphical_Presentations_of_Relational_Information

15 M. Brunetti, S. Auer, R. Garcia, J. Klímak, M. Neceský, *"Formal Linked Data Visualization Model"*, ACM New York, NY, USA 2013. http://ceur-ws.org/Vol-914/paper_29.pdf

16 T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets, *"Tabulator: Exploring and analyzing linked data on the semantic web"*, In 3rd Int. Semantic Web User Interaction WS, 2006.

17 S. Araujo, D. Shwabe, and S. Barbosa. *"Experimenting with explorator: a direct manipulation generic rdf browser and querying tool"*, In WS on Visual Interfaces to the Social and the Semantic Web (VISSW2009), 2009.

18 E. Pietriga, C. Bizer, D. R. Karger, and R. Lee, *"Fresnel: A browser-independent presentation vocabulary for rdf"*, In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, The Semantic Web - ISWC 2006, 5th International Semantic

Web Conference, ISWC 2006, Athens, GA, USA, November 5–9, 2006, Proceedings, volume 4273 of Lecture Notes in Computer Science, pp. 158–171. Springer, 2006.

19 J. Brunetti, R. Gil, and R. Garcia, *"Facets and Pivoting for Flexible and Usable Linked Data Exploration"*, In Interacting with Linked Data Workshop, ILD'12, Crete, Greece, May 2012.

20 Danh Le Phuoc, Axel Polleres, Giovanni Tummarello, Christian Morbidoni, *"DERI Pipes: visual tool for wiring Web data sources"*, 2008. https://www.researchgate.net/publication/228733128_DERI_Pipes_visual_tool_for_wiring_Web_data_sources

21 Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry, *"Task taxonomy for graph visualization"*, 2006. In Proceedings of the 2006 AVI workshop on Beyond time and errors: novel evaluation methods for information visualization (BELIV '06). ACM, New York, NY, USA, pp. 1–5. DOI: http://dx.doi.org/10.1145/1168149.1168168