Visualization of typed links in Linked Data

Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades
Dipl.-Ing. für technisch-wissenschaftliche Berufe

am Masterstudiengang Digitale Medientechnologien an der Fachhochschule St. Pölten,
Masterklasse Grafik Design

von:
Georg Neubauer, BSc
dm131520

Betreuer/in und Erstbegutachter/in: FH-Prof. Priv.-Doz. Dipl.-Ing. Dr. Wolfgang Aigner, MSc
Zweitbegutachter/in: FH-Prof. Mag. Dr. Tassilo Pellegrini

St. Pölten, 27.12.2016

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

....

Ort, Datum Unterschrift

Abstract

The main subject of the work is the visualization of typed links in Linked Data. The scientific subjects relevant to the thesis in general are the Semantic Web, the Web of Data and information visualization. The Semantic Web, founded by Tim Berners-Lee in 2001, was announced as an extension to the World Wide Web (Web 2.0). The actual area of investigations 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 main part of processing data. In the context of the Semantic Web representation of information relations can be managed by graphs. The aim of the thesis is primarily to describe the arrangement of Linked Data visualization concepts by establishing its principles as a theoretical approach. Successively a net of design restrictions is elaborated that lead to practical guidelines. By describing the creation of two alternative visualizations of a commonly used web application representing Linked Data as network visualizations their compatibility was tested. The application-oriented part treats the design phase its results and future requirements of the project that are derived from this test.

Kurzfassung

Das Themengebiet der Arbeit behandelt Visualisierungen von typisierten Links in Linked Data. Die wissenschaftlichen Gebiete, die im allgemeinen den Inhalt der Diplomarbeit abgrenzen, sind das Semantic Web, das Web of Data und Informationsvisualisierung. Das Semantic Web, dass 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, indem in einer theoretischen Annäherung in deren Prinzipien eingeführt wird. Schrittweise wird ein Netz von Gestalungskonzepten, mit dem Ziel praktische Richtlinien anzubieten, ausgearbeitet. 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 hat. Der praktische Teil behandelt daher die Designphase, die Resultate und zukünftige Anforderungen des Projektes, die durch die Testung ausgearbeitet wurden.

Table of contents

1 Introduction

The World Wide Web is a gigantic information system. The administration of data is a great problem of the web. In principle, the web contains sites of pictures, text and video, tied together with links. The publication of any information, apart from the advantages of equal rights also has disadvantages like the difficulty of checking the reliability of the published information. Finding information is complicated by different formats and different meanings of headwords.

The Semantic Web as a way to reduce the disorganization of data was proposed by Tim Berners-Lee [01].

The formalization of data structures should not only be machine-readable but also machine-understandable. This formalization enables that new applications are automatically connected to make information easily accessible.

The Semantic Web is meant to bring structure to meaningful content of web pages, creating an environment where software agents by roaming from page to page can readily carry out sophisticated tasks for users [01].

The beginning of an attempt to implement the Semantic Web is Linked Open Data (or short LOD). On the regular web, it is usual to navigate through documents. In contrast, LOD suggests navigating through data by newly offered techniques.

There are two converging groups of users: the producers and the consumers. Theoretically, the implementation of the Semantic Web should make the search for information easier for consumers but the Semantic Web requires technical knowledge, which is not necessarily available for all consumers. Tools have been developed to allow users to profit from the Semantic Web. Unfortunately, there are limitations, for example the lack of generic diagrams. Depending on how information is displayed it is easier or more difficult to compare, value, evaluate, analyze and understand the information. One aim of visualization is to simplify the data analysis. This master thesis concentrates on the visual representation of typed links in Linked Data.

1.1 Motivation 1.1

Referring to the visualization of data and information, which is not a simple subject, there are many disciplines with different attachments, that tried to investigate which graphic representation best fits a certain kind of information. I wrote about different semantic models and read a lot of papers about this special subject. It was very theoretical, but I gradually understood the practical benefit. The main goal of this work is to show the connection between visualization and Linked Open Data.

1.2 Problem formulation 1.2

The right choice of data visualization should guarantee a coherent representation of complicated datasets via different artifacts (pictures, graphics). To find the right visualization, it is necessary to consider specific factors that depend on heuristics concerning good view types. To simplify this choice, it requires classifications of visualization, which makes it possible to find a suitable visualization type depending on available data.

The aim of the present master thesis is to construct and evaluate different graphic representations of Linked Data that can be compared in terms of preference and simultaneously have a clearer design as result.

1.3 Research questions 1.3

To make these coherent representations suitable for Linked Data, several aspects must be considered: It must be established whether the structure of content is syndetic and it must be guaranteed that these relations are recognizable as visible information. Based on these dependencies the main questions:

CX *01-001* ***How can typed links in Linked Data be visualized for to a broader number of users?***

CX *02-001* ***Which graphic representations are suitable for typed links in Linked Data?***

CX *03-001* ***Is there an interrelation between the visualization and the functionality of graphs concerning the interpretation of data?***

To this end the representation standards of Linked Data where scrutinized concerning the state of the art and possibilities of optimizations where worked out to construct alternatives of visualizations. To mention the affixing of the main context gradient a guide of numbered context fields are shown on the left side.

1.4 Structure of the thesis

First, basic terms are explained, like data, ontology, Linked Open Data and typed links. Then I will present different types of visualization models. Furthermore, a method to solve the problem formulation is described. The construction of ontologies is a cyclic process and consists of following steps [08].

- *Definition of the area and the domain*
- *Analyses of the available ontologies for possible reutilizations*
- *Identification of relevant terms*
- *Production of the hierarchy*
- *Definition of relations*
- *Formalization of classes*

The following chapters deal with my modification of a model. Per this I compare this model with the ones described before. Advantages and disadvantages are also discussed in these chapters. Finally, the graphics are evaluated and the results are discussed.

Basics

2

2.1 Data and Information

2.1

cx 03-002 ***Data is a set of values*** of qualitative (describing values, not collected) and quantitative variables (numerical values) with information content. Data is measured, collected, reported, analyzed and can be visualized using graphs or images to create information suitable for making decisions [02].

cx 01-002 Data is seen as formal and ***predefined representations of information*** that are encoded by sign marks. Examples for data are the written and the spoken language. Referring to the data structure there are three types:

- *structured*
- *semi-structured*
- *non-structured*

"Data becomes information by interpretation" [03].

There are two important parts of processes of data transformation if we talk about the exchange of information [04]:

- *the formalization at the beginning*
- *the reconstruction after the data transfer*

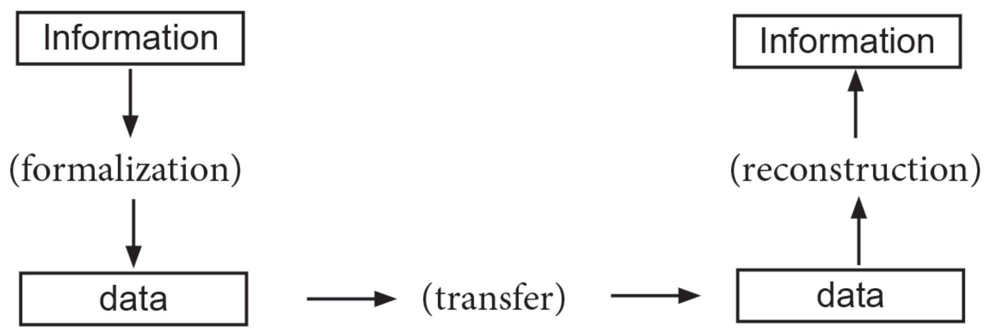


Figure 01: Exchange of information [05]

For the preservation of data it is necessary to store it. Nowadays data can be digitally stored with computers, magnetic memory media and the Internet. The general agent is the World Wide Web. Data can be described by nominal scale (compare values), rank or ordinal scale (values can be ordered), motor scale with the subgroups interval scale (enables differences between values) and ration scale (enables the comparison of values by identity and size).

Frequently, data can be distinguished in dependence on the continuity of possible values. There are discrete (vote able out values, integers) and steady evaluators (continuous number sets).

If we speak about data organization, there are certain kinds of terms to define minimal units of types. A data field (i.e.: register number) is a minimal unit of data and datasets (i.e.: article number) are ***content coherent data fields***. There are still many divisions, which I would not like to state any further.

CX *02-002*

2.2 Ontology

2.2

For the formalization of data, the term ontology is used. Generally, ontology is the study about which kinds of things exist - which entities there are in the universe. 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 domain of knowledge or practices, such as electronic commerce. A good definition is the following sentence.

CX *01-003*

"An ontology is a formal specification of a shared conceptualization" [06].

CX *03-003* Per this, an ontology is a set of concepts - such as ***things, events, and relations*** that are specified in some way (such as specific natural language) to create an agreed-upon vocabulary for exchanging information. In summary ontology is an abstract model which facilitates explaining objects of a domain.

Ontology is distinguished in two subgroups: the general ontology which describes basic concepts of the world and the domain ontology which describes a special knowledge domain. To the second subgroup belongs the user ontology (describes special aspects of the usage) and the task ontology (describes given settings of tasks).

The ontologies consist of following components:

- *Terms or classes:*	Representation of objects from reality.
- *Relations:*	Interactions between terms which belong to a special domain, these are characteristics which have values from objects of other classes.
- *Attributes:*	Characteristic terms which serve to describe and differ terms.
- *Limitations:*	Determine values for attributes and relations.
- *Category systems:*	These are objects which are classified by characteristic attributes.
- *Stamping development:*	Realization of a class.
- *Subsumption:*	Logical implication between two concept definitions.

Gruber designed five criteria to produce ontology [07].

- *Clearness:*	Ontology should describe clearly the meaning.
- *Coherence:*	There should be coherence between definition and interference.
- *Extendibility:*	It should be possible to create new terms from old ones.
- *Minimal encoding bias:*	The descriptions should be independent of the syntax.
- *Minimal Limitations:*	For modelling, there should be as few restrictions as possible.

2.3 The Web

The web exists of a gigantic amount of information and works on the Internet. It is based on three components:

- *HTML (Hyper Text Markup Language)*
- *HTTP (Hyper Text Transfer Protocol)*
- *URL (Uniform Resource Locators)*

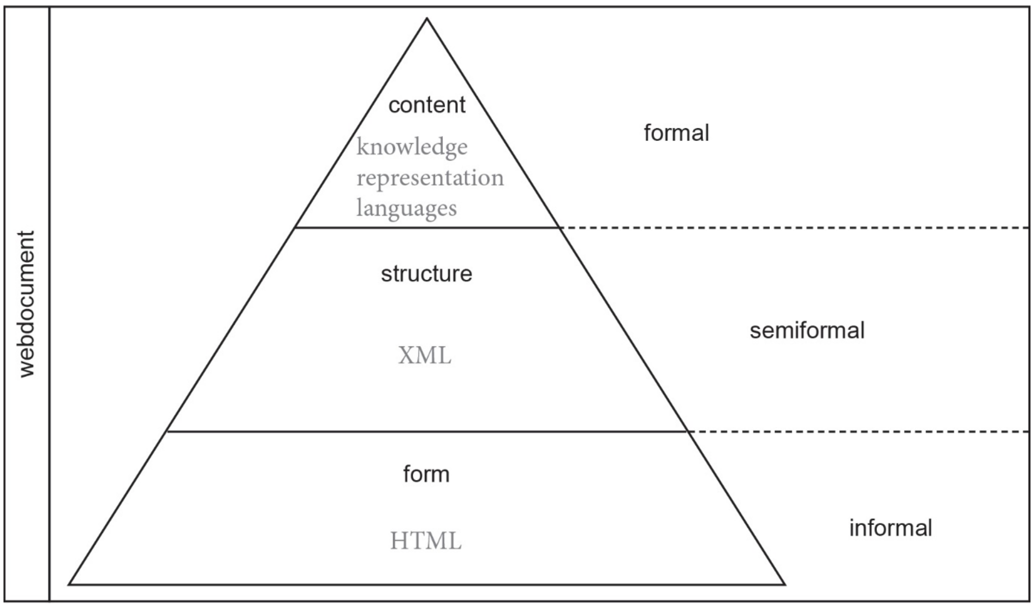


Figure 02: Levels of structuralizing of information on the web [05]

On the first level (informal) there is no rule for the publication of information. The advantages are that everyone can publish his own site. Information is available for a wide mass. The semiformal level is described by XML and enables to exchange data. Usually, DTDs4 and XML-schemata5 are used to express information on a syntactical level [09].

On the third level, there are knowledge representation languages.

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. To bind information, it is necessary to put the information in a confirm and structured form.

Tim Berners-Lee, director of the World Wide Web consortium, gives the following explanation:

" The Semantic Web is not a separate Web but an extension of the current one, in which information is given a well-defined meaning, better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing web are already under way.
CX ***03-004*** *In the near future, these developments will usher in* ***significant new functionality*** *as machines become much better able to process and understand the data that they merely display at present ".*

For the description of data, the Semantic Web is based on the open standards URI, RDF, RDFS, SPARQL and OWL. Per these standards, it is possible
CX ***03-005*** le to ***exchange between different platforms***.

2.3.1 Open standards

2.3.1

URI

"The Uniform Resource Identifier (URI) is a succession of signs for the identification of abstract and physical resources". [10]

The term resource can be defined as every object which should be described and identified.

```
URI = scheme „:“ authority [path][„?“ query][„#“ fragment]
```

Listing 01: Scheme of URIs [11]

Legend:

- *Scheme: I.e.: http, ftp*
- *Authority: The host*
- *Path: The path in the host*
- *Query: The sequence for identification of not hierarchical resources*
- *Fragment: Reference of a part within a resource*

Everyone can build an URI per the specification.

RDF

The Resource Description Framework (RDF) was developed for representing information on the web. The RDF is a basic language for the description of structured information on the web and of course in the exchange of information [11] [12].

CX 03-006 In RDF, logical statements are formulated. For ***every resource an URI*** is assigned. The Meta model of the RDF is described by several triples (RDF graphs) which stand for a relation and often is shown by arrows [12].

There are three components of these arrows:

- the subject (is an URI or a ″Black Node“)

- the predicate (is an URI)

- the object (can be an URI, a ″Black Node“ or a ″Literal“)

A Literal is used to identify a simple data type (numbers, data)

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.

SPARQL

SPARQL Protocol and RDF Query Language are semantic query languages for databases and belong to the key technologies of the Semantic Web. It can retrieve and manipulate data stored in RDF format [13].

CX 03-007 SPARQL allows for a query to consist of ***triple patterns, conjunctions, disjunctions and optional patterns*** [14].

Implementations for multiple programming languages exist. There exist tools that allow to connecting and semi-automatically constructing a SPARQL query for a SPARQL endpoint (i.e.: ViziQuer). Additionally, there are tools that can translate SPARQL queries to other query languages (i.e.: SQL).

OWL

The W3C Web Ontology Language (OWL) is a computational logic-based Semantic Web language and specification of the World Wide Web con-
CX 03-008 sortium, planned 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 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 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 articulate optional features that may be supported by some implementations: the language profiles and two other syntaxes (OWL/XML, Manchester).

2.3.2 Typed Links

2.3.2

The term typed links can be explained as a link relation and has a descriptive feature closed to a hyperlink to define the type of the link or the relationship between the source and target resources [15].

The RDF typed links are essential in LOD datasets for identifying the relationship type of RDF-triples, contributing to the automatic process skill of machine-readable statements of the Giant Global Graph on the Semantic Web. The typed links in RDF are spoken as the value of the rdf:-type property, defining the relationship type using vocabulary terms or definitions from LOD datasets [15].

CX ***03-009*** The mentioned ***URIs of a relation tie functionally together as graph***. Such implementations therefore preserve the semantic variability of content types by using similarly the same syntax. Treating Linked Data, a set of information consists of this syntax which provides an interconnection by the main three components of natural languages (subject, predicate, object) linking these components which are these URIs.

The following figure describes this main dependency.

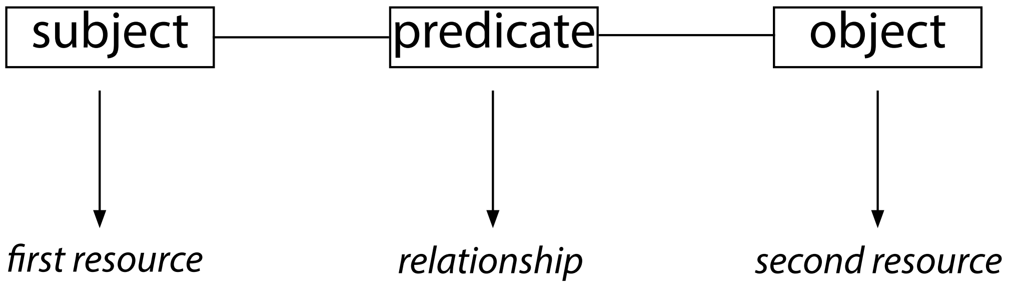


Figure 03: RDF-triple (general)

CX ***02-003*** To offer further prospects of the functionality, ***the predicate defines the relation of an RDF-triple***. As main part of the process the therefore provided options will be treated more specifically in the practical part.

2.3.3 LOD (Linked Open Data) 2.3.3

On the web, Linked Data is a method of publishing structured data so that it connects related data that was not previously linked.

LOD extends to share information in a way that can be read automatically by computers. Starting in 2007 the Linked Open Data Cloud is a project that visualizes a graph of interlinked Linked Data [13]. Linked Open Data is the open content of Linked Data which provides room for open source software.

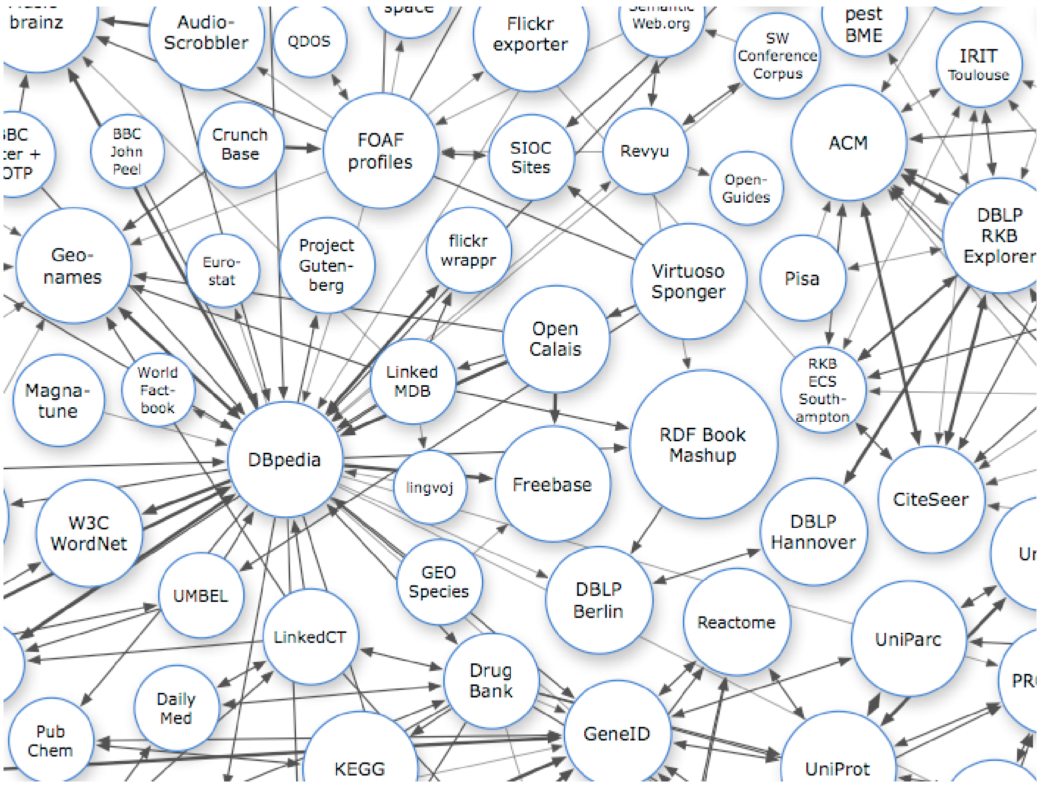


Figure 04: Part of the LOD Project Cloud [13]

2.4 Visualization 2.4

For the process of analysis, evaluation and identification of datasets there exist visualizations based on computational graphic systems [16].

CX *02-004* ***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
CX *01-004* limitations include the ***capacity of computers, of humans, and displays***. The space of possible visualization system design is huge and full of tradeoffs [17].

Visualization is a term for computer graphics and contains any technique for creating images, diagrams or animations to deliver a message. By means of visual imagery abstract and concrete ideas can be well communicated. Visualization is very common in computer graphics. The recent emphasis on visualization started in 1987 with the publication of visualization in scientific computing, a special topic of computer graphics [18].

The development of animation also helped to advance visualizations. The abstract visualizations show totally conceptual constructs in 2D or 3D. The model-based visualizations either place overlays of data on real or digitally constructed images of reality or create a digital production of a real object directly from the scientific data. Scientific visualizations are usually done with specialized software (open source software), but there are also many proprietary software packages of scientific visualization tools.

Friendly (2008) sees scientific visualization as a subject in computer science with the use of interactive, sensory representations, typically
CX *01-005* visual, of abstract data to emphasize ***cognition, hypothesis generating and reasoning***. Data visualization is a subcategory of visualization dealing with statistical graphics and geographic data (cartography) that is abstracted in schematic form [18].

There are different subgroups of visualization without a clear demarcation. I want to describe just a few of them.

2.4.1 Information visualization **2.4.1**

Information visualization, coined by the User Interface Research Group at Xerox PARC, concentrates on the use of computer-supported tools to explore a large amount of abstract data. Useful applications of information visualization in computer programs involves selecting, transforming and representing abstract data in a form that facilitates human interac-
CX *02-005* tion for exploration and understanding. Thus, information must be ***prepared, condensed and filtered*** for an effective utilization. In addition to this, there is the need to integrate perception, psychology and linguistic. Central aspects of information visualization are dynamics of visual representation and the interactivity. Strong techniques enable the user to modify the visualization in real-time.

To create graphic presentations of data nowadays there are many tools. For the integration of different scientific disciplines – like linguistic, information design, psychology of perception – it is important to prepare the data generally understandable. [19].

Moser [19] defined the task of information visualization as follows:

"Die Informationsvisualisierung beschäftigt sich damit, numerische Daten so aufzubereiten, dass ein Betrachter daraus möglichst einfach die gewünschten Erkenntnisse ableiten kann. Dabei kommen oft Interaktionstechniken zum Einsatz, die ein zusätzliches Filtern und Hervorheben gewisser Daten ermöglichen."

There is no precise classification between visualization disciplines because of the huge spectrum. Hence it is difficult to find a border between information visualization and scientific-technical visualizations [20].

2.4.2 Knowledge visualization 2.4.2

Knowledge visualization is used to represent and interpret complex information and particularly text, at the intersection of knowledge, art and cultural heritage. The use of visual representations to transfer knowledge between persons aims to improve the transfer of knowledge by using computer and non-computer-based visualization methods complementary [21].

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

2.4.3 Visual communication 2.4.3

Visual communication is the communication of ideas through the visual display of information. It includes alphanumeric, art, signs and electronic resources. Latest research in the field has aimed on 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 means what sense a picture with specific, social and or cultural context has.

2.4.4 Visual perception

As I said before, there are several factors, which play an important role for the right graphic presentation. The perception as a factor is relevant for the data interpretation and the attention that the recipient the information adjoins. Bosch et al. [22] describes the perception as a relation between the outside and the inside world of an individual with the content of the visual recording of stimulus, smell, vision, hearing, feeling and taste. The psychology of perception established laws for order the different components of perception [17]:

	- law of similarity:	Objects are perceptually grouped together if they are like each other. This similarity can occur in the form of shape, color, shading or other qualities.
	- law of proximity:	The law of proximity states that when individuals perceive an assortment of objects they perceive objects that are close to each other's by forming a group.
	- law of closure:	Individuals perceive objects such as shapes, letters, pictures, etc., as being whole when they are not complete. Specifically, when parts of a whole picture are missing, our perception fills in the visual gap.
	- law of continuity:	Elements of objects tend to be grouped together, and therefore integrated into perceptual wholes if they are aligned within an object.
cx ***01-006***	*- law of experience:*	Under some circumstances visual stimuli are categorized per experience. ***If two objects tend to be observed within proximity, or small temporal intervals, the objects are more likely to be perceived together.***
cx ***01-007***	*- law of symmetry:*	The mind perceives objects as being symmetrical and forming around a center point. Similarities between symmetrical ***objects increase the likelihood that objects are grouped to form a combined symmetrical object***.
	- law of common fate:	Objects are perceived as lines that move along the smoothest path. The law of continuity implies the grouping together of objects that have the same trend of motion and are therefore on the same path.
cx ***02-006***	*- law of good shape:*	Elements of objects tend to be perceptually grouped together if they form a pattern that is regular, simple, and orderly. This law implies that as individuals perceive the world, they ***eliminate complexity and unfamiliarity*** so they can observe a reality in its most simplistic form.

Visual perception laws include "Gestalt laws" which term is commonly used. Under consideration of the listed laws the main appearances in use are given by the following figure.

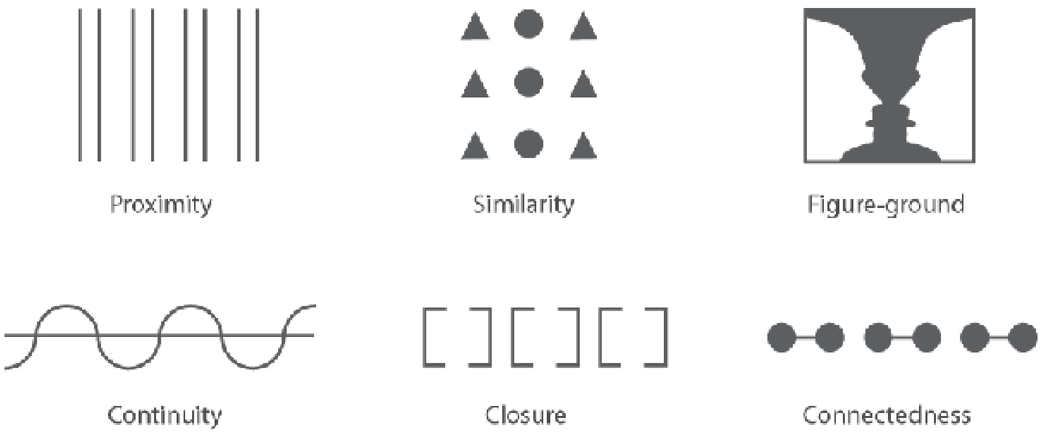


Figure 05: Gestalt laws [23]

Position as most important part of design principles was probed by Mackinlay [24]. In terms of the exploration of subordinated options of
CX 01-008 design he ***treated perceptual tasks based from the position of the components of information*** given. Figure 06 shows a distinct change in the ranking of these tasks from quantity to quality.

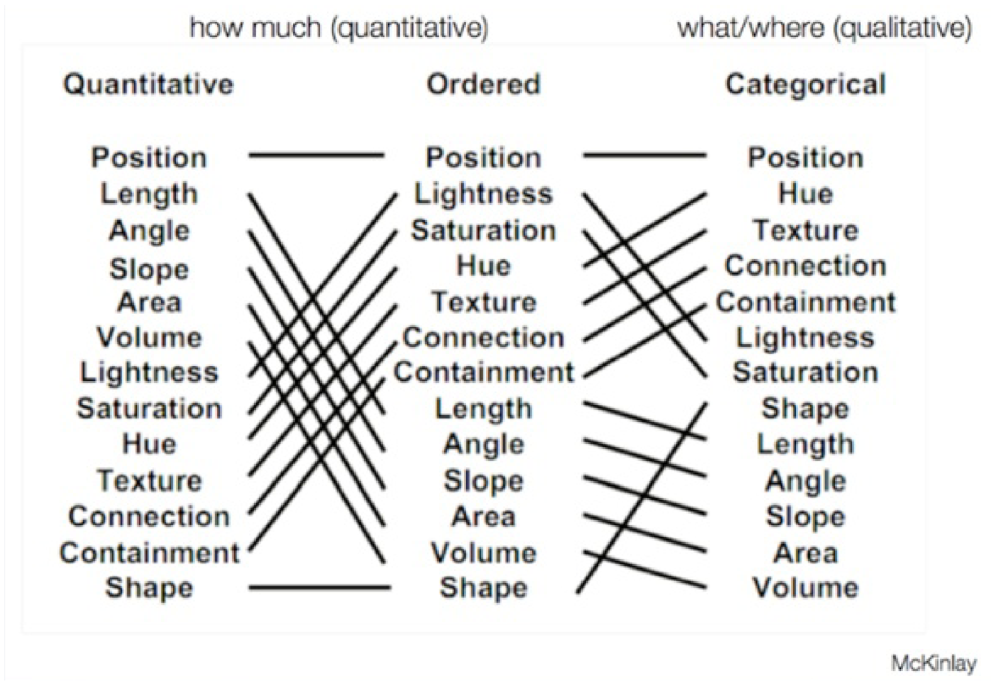


Figure 06: Ranking of perceptual tasks [24]

CX 03-010 It is also apparent that ***the rank of connection rises in conjunction to the quality of the order***.

2.4.5 The aim of visualization

The knowledge about the information process dedicates to many factors which are necessary to be recognized by the selection of the information visualizations, like the target group, the accentuation, usefulness, display format, task and data source. Depending on the question it is possible to use different types of visualization for one dataset. For the success of the visualization it is important to think about the target audience and if it is heterogeneous. Unequal values, cultures, conditions of socialization, etc. can favorite different views. The accentuation of relevant characteristics reduces and simplifies information [20]. By the way, it is to recognize if data visualization is useful. The ***display format is another factor*** which must be recognized. Therefore, a decision must be made, which type of visualization is used to show the number of data elements so that the result of the visualization can be interpreted clearly. The choice of the visualization type can be identified by means of the characteristic features and relations of separated variables. CX ***02-007***

A usage example of these attributes is the reduction that provides the usage of a certain scale. Only specific given diagrams or visualizations can be chosen, that have ***reliable relations in terms of the level of the scale***. [19] [22]. CX ***03-011***

In the same way, it is efficient to think about the reason therefore the data should be visualized.

If data and results are known, it is only demanded to present data. But if just the data is known without results, there is a need of visual media or rather interaction techniques. The origin of data (data source) should also be shown.

Seven remarkable points for a better comprehension of data visualization by Moser [19]:

- *The simple presentations are the best ones.*
- *The meridional section should be chosen in a way, that the data fill the visualization.*
- *Reduce elements which are not data related.*
- *Avoid a perspective presentation.*
- *Select a convenient excerpt for a correct interpretation of data.*
- *Offer different views in place of different designs.*
- *For associated graphs use the same scale all the times.*

2.4.6 Visualization and interaction techniques

Diverse visualizations permit the interaction of the user. This enables to zoom or to change the representation on demand, to bring non-relevant data not to view, to order data upon certain criteria's or the explicitly bring specific data to expression. Interaction is maintained through the application of some well-known technics, which have their basic prin-
CX 02-008 cipal of information research. (″***Overview first, zoom and filter, then details-on-demand***“ [25]).

1996, Schneidermann [25] described basic tasks, that should support all visualizations. Taxonomy of solutions to information visualization was announced (Task by Data Type Taxonomy).

There are seven tasks:

- *Overview:*	The overview on the whole room of information supports global measures, to recognize trends to gain understanding therefore.
- *Filter:*	Filter support the easement of the reduction of amounts of data.
- *Zoom:*	To gain view about other (global) levels of details.
- *Details-on-demand:*	After the selection of specific data objects, details of the object will be shown.
- *Relate:*	Relations between data should be reflected and connect the views.
- *History:*	Support of ″do“ and ″undo“, (progressive refinement).
- *Extract:*	Extraction of subsets of the information.

Nevertheless, there are many other interaction techniques, that provide important and in most cases standardized interactions.

- *Mark:*	Offers selectable, values of data, detailed data or filters. Labeled values should be highlighted in vision.
- *Drill-down:*	The Drill-down functionality permits the user to start with an aggregated view that offers a fast overview.
- *Perspective distortion:*	Focused data becomes enlarged while the rest is downscaled data.

2.4.7 Data type driven visualization

There are much taxonomies for the classification of types of visualization. Many authors use data types to classify solutions of visualization and to construct the taxonomies. ***Within help of the data types a model can be built***, which can basically be used for any visualization. This model of data represents a slice of the real world and consists of objects of information with their attributes and relations.

CX ***01-008***

Following Keim [26] the following datatypes should be considered:

- Dimension of data objects:	In dependence of the attributes that will be visualized.
- One-dimensional data:	Data that only has one attribute as dependency, is related to this term. (for example, acceleration). The visualization of one-dimensional data will often be used to compare data among each other's then its usage as view with only one value (for example, acceleration of more cars). Therefore, there exist several visualization technics (for example bar charts, pie charts, histograms and tag-clouds).
CX ***02-009*** *- Two-dimensional data:*	***Data with two dependent attributes***. In most of the cases data correlation is considered in comparison.
- Multi-dimensional data:	Multidimensional data is two-dimensional data as an outlier. In relation to the author of the classification we talk of multi-dimensional data, when more or equal than three [26] or four [18] attributes are given. A well-known example with a multivariate visualization are scatterplots, in which besides two dimensions following attributes are used like form, size, texture and transparency.
- Relations:	Information that only makes sense, if the relations are shown. Two sub descriptions are given therefore.
- Hierarchies (trees):	Is the hierarchical relation between the data. Visualizations like tree-visualizations and Venn-diagrams are used.
- Mashes (graphs):	If the relations between the data have no dependencies or the data is not necessary in the visualization, these visualizations will be used (mention map, node-link-diagram or tree-ring).
- Software systems:	Visualization with diagrams and UML.

2.5 Components of creation

Visual programming interfaces expose computational components as modules and allow the creation of complex visualization pipelines which combine these modules in a dataflow, where connections between modules express the flow of data through the pipeline. The palette of visualization components and their design reflect the need to provide an instant utility to users on a broad range of datasets. The core components were reconstructed to allow object-oriented programming, encapsulating all the global variables related to crystal data and graphic settings into a scene class.

Ontology visualization and exploration is not a trivial task as many issues can affect the effectiveness of interactions. As ontologies are, in the
CX ***03-012*** general case, quite connected graphs where ***concepts are the nodes and semantic relationships the edges***, the problems include space allocation, edge superposition, scene over-crowding, etc.

- Control data:	Data that activates and controls all the modules in the system.
- User Input/Output:	All forms of human - computer interaction. Input might be keyboard, mouse, light pen, etc. Output might be screen, hard copy device, sound, etc. This is converted into the metadata for the system modules.
- Internal data:	The data that can flow through the system.
- External data:	Data that can be imported/exported to the system. This might be asci data (from observation or simulation) or images.
- Sortable data:	Data that can be stored and retrieved within the system.
- Graphics data:	A reduced form of internal data that represents graphics primitives (2D or 3D)
- Picture data:	A reduced form of graphics data (pixel map or 2D primitives for display or hard copy)
CX ***01-009*** *- Limiting data:*	***A reduced form of similar labeled data***

Visualization models

"A visualization system is not only a system to create an image of the data. It can be used to manipulate the data to create different types of images. A model of a visualization system should link the system with the model of scientific investigation. Visualization can help from the connection between hypothesis and experiment and between insight and revised hypothesis." [27]

The different data types describe the dataflow between the different modules [27]:

Within these listed components Owen describes general model types that can be arranged by plotting a simple block visualization system that's components are used as input and output channels gaining different states of visualizations.

In dependence to the preparation level of data the models are pointed below.

- *User model*
- *User interface model*
- *Base graphic system model*
- *Visualization technique model*
- *Data manipulation model*
- *Data access model*

To refine this, the following sub points will specify, which facets there exist.

Conceptual Models in Visualization

The science of visualization is just beginning, which involves the combination of graphics, imaging, data management and human perception. Two divergent trends in visualization have become common recently. Although that visualization is often available and practical, the supporting technology often is not open-minded for these applications. The second trend emerges from the conversing data glut problem. This means
CX 02-010 that ***typical ad hoc approaches do not scale to 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 taxonomy.

Although this idea has been suggested before [28], there is a limitation when they are considered as a set of layers. On the left side, interactions
CX 02-011 between layers for ***typical visualization operations are shown as a set of arrows***, under the view that the data model is most fundamental. This then suggests that the organization on the right is a better illustration of the role of a data model.

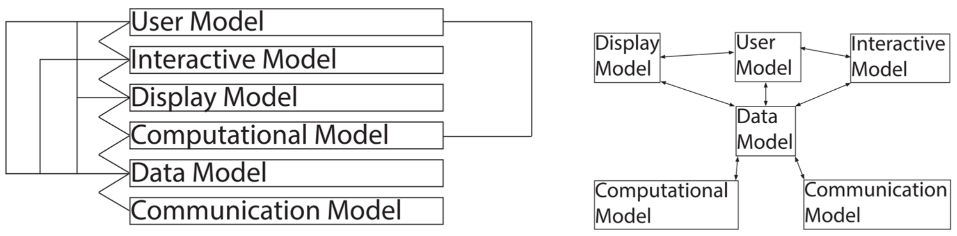
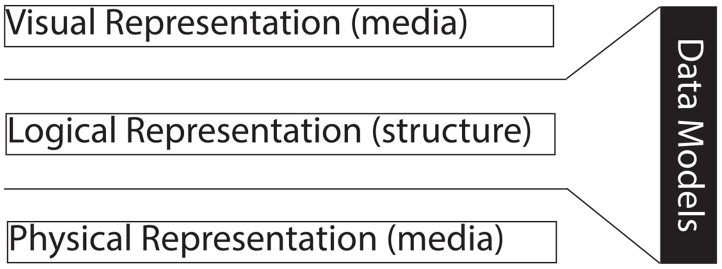


Figure 07: Conceptual models of visualization [29]

Data Models

A data model is a representation of data, that is how data is described (e.g., abstract data type) and how data is used (e.g., applications programming interface). The data base community provides a physical representation (format - media), a logical representation (data structures, schema) and a visual media (object and conceptual representation, user view), more simply illustrated in figure 07. To serve as a lexicon of data (i.e., a lingua franca for software and users), a data model must include a
CX 03-013 ***formal definition and algebra*** to express the organization and manipu-

lation of data. In this context, visualization itself is not treated as special - just another consumer and generator of data. Another way to view
CX 02-012 this idea is ***a layer that provides a logical link between the concepts*** that scientists use to think about their field (e.g., particle trajectory, cerebral cortex shape, plasma temperature profile, or gene) and the underlying data from simulation, experiment or storage system.



Visual Representation (media)
Logical Representation (structure)
Physical Representation (media)
Data Models

Figure 08: Role of Data Models in visualization software [29]

This layer provides tools that are common to all applications for data definition, metadata support, and query formulation and execution. This layer supports computational, analysis and visualization tools and provides the infrastructure for data representation and access (data model definition, metadata support, query formulation, etc.). Like a data base management system, it would reside above the operating system and enable the building of applications. Hence, data models hide the complexity of underlying computational systems for simulation, analysis and visualization, freeing scientists to focus on data comprehension by providing a common mechanism for access, utilization and interchange.

CX 03-014 The visualization is complex. Therefore, there are ***different types of components which are to classify and to shape***. The whole process of visualization needs the interaction of many elements with the aim to dedicate comprehensible information.

Duke et al. [30] describes a cyclic process of interaction between various components of the visualization. Consisting of four parts, (human to human, human to system, system to system, system to human), each sub process of the visualization cycle has two components that should communicate with each other's. To keep the sense of the visualization they should be of same language.

Next I want to describe classic visualization models.

3.1 Classic models 3.1

The visualization pipeline of Haber & McNabb [31] describes the (stepwise) process of creating visual representations of data. Haber & McNabb described a conceptual visualization process in three major transformations. These transformations occur in most visualization processes and convert raw simulation data into a displayable image. The goal of these transformations on the data is to convert the information (e.g. gained from a simulation) to a format amenable to understanding by the human perceptual system while maintaining the integrity of the information. The steps of the visualization process can be explained as:

- *Data Analysis:* Data is prepared for visualization (e.g., by applying filter, interpolating missing values, or correcting erroneous measurements) - usually computer-centered, little or no user interaction.
- *Filtering:* Selection of data portions to be visualized - usually user-centered.
- CX *02-013* *Mapping:* ***Focus data is mapped to geometric primitives (e.g., points, lines) and their attributes (e.g. color, position, size)***; most critical step for achieving expressiveness and effectiveness.
- *Rendering*: Geometric data is transformed to image data.

3.1.1 The Pipeline Model

3.1.1

In the 90s Haber and McNabb [31] described a conceptual visualization process that performs three big transformations between data, converting raw data to a viewable picture without losing the integrity of information. The model consists of four components and three transformations like shown.

Components

- *Raw data:* The source of the graphic
- *Focus data:* Chosen data
- CX *03-014* *Geometric data:* ***Visual variables that describe the attributes of the graphic***
- *Image data:* The result

Transformations

- *Filter:* Converting data, so that the other components of the models can work on the data
- CX *03-015* *Mapping:* ***Transformation of data tables in visual variables***
- *Rendering:* Generation of graphics out of the geometric and visual attributes

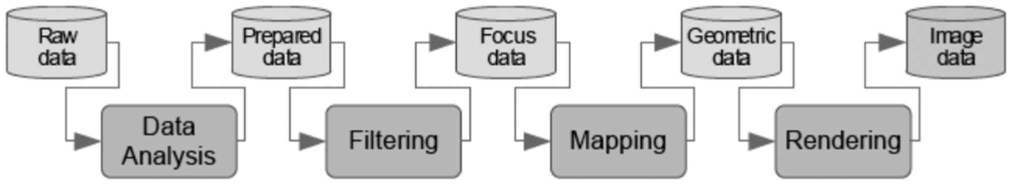


Figure 09: Pipeline Model [32]

3.1.2 The Data State Model

In the year 1998 Chi [33] presented the Data State Model, a modification version of the information visualization pipeline model. This model consists of seven components that are divided in two groups, transformations and steps.

	- data:	The source of the graph
	- data transformation:	Generate data sets from an amount of raw data
	- analytic abstraction:	Indicator sets of data
	- visualization transformation:	Classification the data sets to visualization set
	- visualization abstraction:	Abstract model of visualization
CX *02-014*	*- visual mapping-transformation:*	***Highlight relevant characteristics***
	- view:	The result

He showed that the framework successfully modelled a large collection of visualization applications; the model was functionally equivalent to the dataflow model used in existing graphics toolkits [33].

CX *01-010* The Data State Model helps researchers to ***understand the space of design*** and it makes possible to implementers to recognize how information visualization techniques can be applied more generally. As shown in figure 10 it consists of seven steps in two groups.

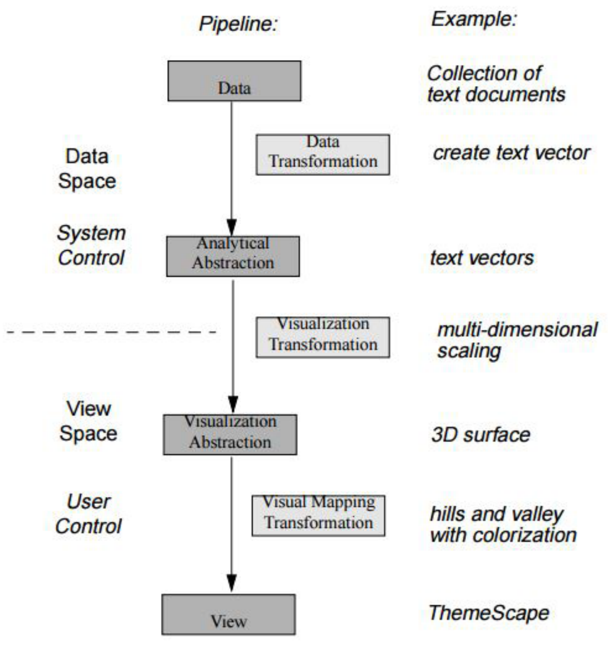

Pipeline:
Example:
Data
Collection of text documents
Data Space
Data Transformation
create text vector
System Control
Analytical Abstraction
text vectors
Visualization Transformation
multi-dimensional scaling
View Space
Visualization Abstraction
3D surface
User Control
Visual Mapping Transformation
hills and valley with colorization
View
ThemeScape

Figure 10: The information visualization pipeline [33]

3.1.3 Info Vis Data Stage Reference Model

Then, in the Year 2000 Chi [33] upgraded the Data State Model. This model can classify 36 visualization techniques in different areas, like scientific visualization, geographic based visualization or the visualization of trees.

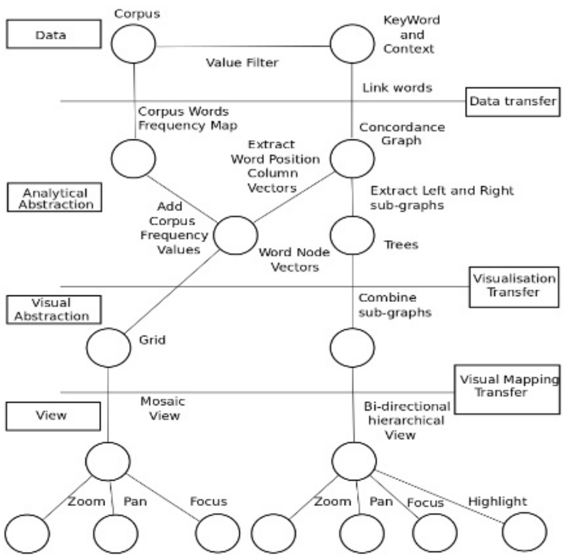


Figure 11: The Data State Reference Model [33]

Kard et al. (1999) published the book "*Reading in information visualization:* ***using vision to think***" [34]. Shneiderman developed a reference model that exists of three transformations and four steps. It differs from the Information Visualization Data Stage Reference Model that the user controls and triggers the visualization process. There are seven components:

CX ***01-011***

- raw data:	The source of the graph
- data charts:	Combination of relevant data and metadata (metadata describe the relation between data)
- visual structures:	Coding information by description of graphic qualities
- views:	The results
- transformation:	Data transformation (from raw data in data charts)
- visual mapping:	Transformation data from charts in visual structure
CX ***02-015*** *- view-transformation:*	***Generating views by specification of graphic qualities like position, scaling and sections***

3.1.4 The Grammar of Graphics - GOG

3.1.4

Wilkinson (1999) [35] purposed a further model with two types of components. The first one is the mathematic component and the other component deals with the aesthetic of graphic design. The first model consists of eleven components:

	- data:	Source of graphic
	- data view:	Generating data sets from raw data
	- data set:	Index sets of data
	- references:	Mechanism to support the object localization
	- variable mapping:	Classification of the data set to a variable set
CX **03-016**	*- algebra:*	***A Collection of operators and rules***
	- variable set:	Set of discrete or continuous variables
	- graphic:	Producing a diagram from a certain graphic function
	- position:	Pattern for the localization of a point in each room
	- aesthetic:	Aesthetical qualities of a graph
	- graphic:	Results of graph

Later with the second edition of the book, the new GOG-model was presented which consists of seven orthogonal interconnected components [36].

3.2 VISO (Visualization Ontology)

3.2

In the year of 2011 Voigt and Polowinski [37] suggested a semantic ontology for the description of visualization systems. VISO is a modular ontology for the formalization of the visualization that ***provides an RDF-S/ OWL vocabulary*** for the construction of the components of visualization and data sources. This ontology contains knowledge about the facts of visualization domains and should be utilized as a framework to save context information [37]. VISO consists of seven modules (data, graphics, action, users, systems, domains and facts) that stand for the different components of the data visualization domain. Figure 12 shows these seven domains. VISO Semantics-based Information Visualization Workflow Voigt et al. [37] suggest a visualization process based on semantics, where all members interact to ease the suggestion of a good visualization. The basic of the whole process is called Ontology VISO, which provides concepts and ***relations between diverse facts*** of the visualization.

CX **03-017**

CX **02-016**

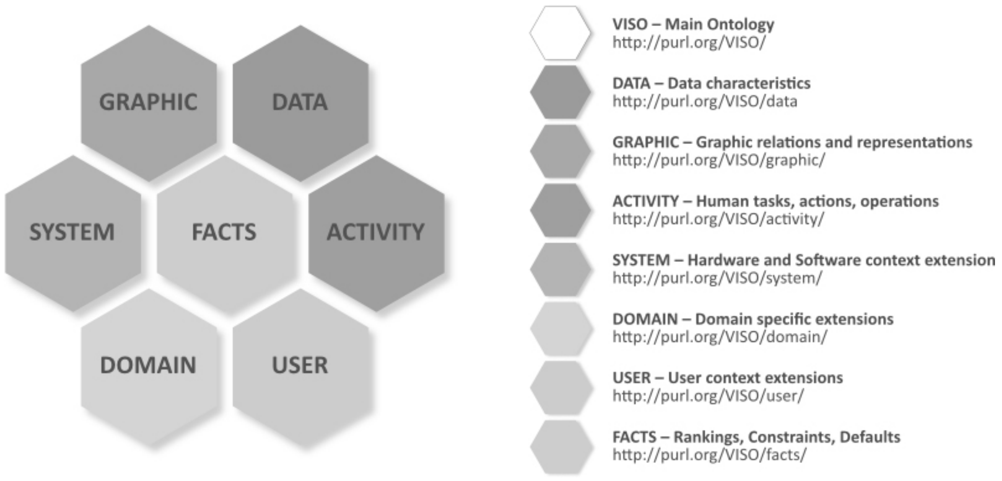

GRAPHIC
DATA
SYSTEM
FACTS
ACTIVITY
DOMAIN
USER
VISO – Main Ontology
http://purl.org/VISO/
DATA – Data characteristics
http://purl.org/VISO/data
GRAPHIC – Graphic relations and representations
http://purl.org/VISO/graphic/
ACTIVITY – Human tasks, actions, operations
http://purl.org/VISO/activity/
SYSTEM – Hardware and Software context extension
http://purl.org/VISO/system/
DOMAIN – Domain specific extensions
http://purl.org/VISO/domain/
USER – User context extensions
http://purl.org/VISO/user/
FACTS – Rankings, Constraints, Defaults
http://purl.org/VISO/facts/

Figure 12: VISO modules [37]

The workflow of the visualization process is close coupled with the interaction of the users and has five levels:

VISO Semantics-based Information Visualization Workflow [37]

- *fetch the data record*
- *overview upon the chosen data*
- *selection of relevant and fitting visualization components*
- *configuration of visual components*
- *interaction with the visualization*

To formalize the knowledge of the visualization the components are
CX *01-012* grouped in data, graphic, human interactions and the ***integration of the user-, system- and domain ontologies*** [37].

3.3 Network visualization 3.3

This chapter mentions the visualization of data using graph visualiza-
CX *02-017* tions. Per Battista [38] there originated a ***necessity to draw two-dimensional graphs*** to solve data presentations. Networks consist of ***vertices***
CX *02-018* (points or nodes) and ***lines*** (edges) which bind these nodes that contain and stand for reliable information. A network consists of these relations. Network visualizations cover a huge amount of representation duties. In example software engineering, database design and visual interfaces,
CX *03-018* that ***represent interrelated information***. Layouts for such visualizations
CX *02-019* are ***node-link or matrix visualizations*** as one alternative. Main utilizations of graph visualization represent information of ***social networks,***
CX *02-020* ***software systems or traffic networks***. The following list describes the features a network visualization should have by O'Madadhain [39].

Graph type

CX 01-013 The graph type offers support for a variety of ***representations of entities and their relations including directed and undirected graphs***, multi-modal graphs (graphs which contain more than one type of vertex or edge), graphs with parallel edges (also known as multigraphs), and hypergraphs (which contain hyper edges that may connect any number of vertices).

Annotation mechanism

CX 02-021 A mechanism for ***annotating graphs, entities and relations with metadata***. These capabilities facilitate the creation of analytic tools for complex datasets that can examine the relations between entities, as well as the metadata attached to each entity and relation.

Algorithms

Are implementations of several algorithms from graph theory, exploratory data analysis, social network analysis, and machine learning. These include routines for clustering, decomposition, optimization, random graph generation, statistical analysis, and calculation of network distances, flows, and ranking measures (centrality, PageRank, HITS, etc.)

Visualization framework

CX 03-019 A visualization framework makes it easy to construct tools for the ***interactive exploration of network data***. Users can choose among the provided layout and rendering algorithms, or use the framework to create their own custom algorithms.

Filtering

CX 02-022 ***Is a filtering mechanism which extracts subsets of a network***. This allows users to focus their attention, or their algorithms, on specific portions of a network.

CX 03-020 ***Network visualizations can be optimized in terms of minimizing overlaps and edge-crossing***. There are four main options of optimization:

- *minimize edge-crossing*
- *uniform edge-length*
- *prevent overlap*
- *symmetry*

CX 02-023 Optimized edges of a network only show ***lines from source to target***. Such networks are called ***directed***.

CX 01-014 In consideration to the practical part it is noted that the arrangement and the drafts of ***graph visualizations are necessarily to be seen depending on their perception***. Therefore, a table describes the heuristics to be considered following Bennett [40].

"Research in this area has led to aesthetic heuristics for drawing graphs. Recently, these efforts have expanded to empirically evaluate the effectiveness of these heuristics and to examine their basis in perceptual processing" [40].

CX 01-015	***- node metrics:***	Cluster similar nodes
		Distribute nodes evenly
		Keep nodes apart from edges
		Maximize node orthogonality
		Nodes should not overlap (except for nested nodes)
CX 01-016	***- edge metrics:***	Minimize edge crossings
		Keep edge lengths uniform
		Minimize edge length (total and maximum)
		Minimize edge bends
		Keep edge bends uniform (angle/position)
		Maximize edge orthogonality
		Maximize minimum edge angles
CX 01-017	***- overall layout metrics:***	Maximize consistent flow direction
		Keep correct aspect ratio
		Minimize area
		Maximize convex faces
		Maximize global symmetry
		Maximize local symmetry
	- domain specific UML:	Join inheritance edges
		Use directional indicators
		Avoid separate arrows on edges

Force-Directed graph layouts

Force-directed graph drawing algorithms are a class of algorithms for drawing graphs in an aesthetically pleasing way. Their purpose is to position the nodes of a graph in two-dimensional or three-dimensional space so that ***all the edges are of equal length and there are as few crossing edges as possible***, by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their tension [41]. CX 02-024

Force-directed methods in graph drawing date back to the work of Tutte [42] who showed that polyhedral graphs may be drawn in the plane with all faces convex by fixing the vertices of the outer face of a planar embedding of the graph into convex position, placing a spring-like attractive force on each edge, and letting the system settle into an equilibrium. Because of the ***simple nature of the forces*** in this case, the system cannot get stuck in local minima, but rather converges to a ***unique global optimum configuration***. CX 02-025 CX 02-026

While graph drawing can be a difficult problem, force-directed algorithms, being physical simulations, usually require no special knowledge about graph theory such as planarity.

Once the forces on the nodes and edges of a graph have been defined, the behavior of the entire graph under these sources may then be simulated as if it were a physical system. In such a simulation, the forces are applied to the nodes, pulling them closer together or pushing them further apart. This is repeated iteratively until the system comes to a mechanical equilibrium state; i.e., their relative positions do not change anymore from one iteration step to the next. The positions of the nodes in this equilibrium are used to generate a drawing of the graph.

Advantages are good-quality results, flexibility, intuitive, simplicity, interactivity, strong theoretical foundations. The main disadvantages are high running time and poor local minima.

3.4 Linked Data Visualization Model 3.4

Applying information visualization techniques to the Semantic Web helps users to explore large amounts of data and interact with them. The main objectives of information visualization are to ***transform and present data into a visual representation*** in such a way that users can obtain a better understanding of the data [43]. CX *03-021*

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: CX *03-022*

- *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 LDVM uses the transformer. In view stage, data is passed to a visualizer, which 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. CX *03-023*

Linked Data browsers such as Tabular or Explanator allow users to navigate the graph structures and usually display property-value pairs in tables, but no broader view of the dataset [44][45].

Fresnel is a vocabulary for rendering RDF to HTML, but its focus is on rendering instance data rather than on creating visualizations [46].

Rhizomer provides an overview of the datasets and allows to interact with data through information architecture components such as navigation menus, breadcrumbs and facets [47]. It also provides visualizations such as maps and timelines.

DERI Pipes is an engine and graphical environment for general web data transformations and mashups [48].

However, it is not intended for lay-users and requires SW expertise.

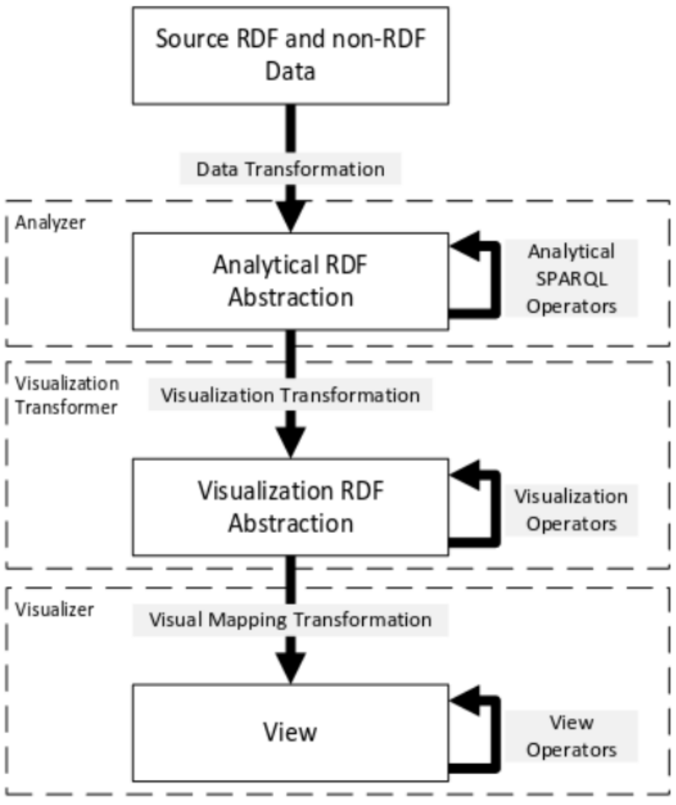

Source RDF and non-RDF Data
Data Transformation
Analyzer
Analytical RDF Abstraction
Analytical SPARQL Operators
Visualization Transformer
Visualization Transformation
Visualization RDF Abstraction
Visualization Operators
Visualizer
Visual Mapping Transformation
View
View Operators

Figure 13: High level LDVM overview [43]

The Linked Data Visualization Model (LDVM) uses the Data State Reference Model (DSRM) proposed by Chi as a conceptual framework. DSRM describes the visualization process in a generic way [43].

In principle, the web exists of an amazing number of pages with videos, pictures, etc. which are associated with links. Because of different dimensions of the webpages it is difficult to find the main information for humans and machines. Since 2001 Tim Berners-Lee [01] found a solution for the unsystematic growth. The purpose was the Semantic Web with the demand to improve the advancement of the internet and the mechanism for the data-extraction.

The main idea is to formalize the data structure of webpages, so that the content is machine-readable and –understandable. The implementation of the Semantic Web offered new services like Linked Open Data (LOD). Linked Data is about using the web to connect related data that was not previously linked, or using the web to lower the barriers to linking data currently linked using other methods.

There are users and developers of data. There are diverse tools that ordinary users can profit of the Semantic Web. In most cases, people try to present data in a browser via graph based visualization ontologies and RDF-triples, but these illustrations are limited and do not have the whole capacity of the data visualization (several views, generic diagram). The meaning of visualization is how a human perceives information in support of the information-presentation. Per graphic rendition, it is easy or difficult to compare, analyze, value or understand information. Therefore, it is necessary to find the best possible option for data visualization. There are many ways for data visualization. The relation between visual presentation and the associated data is not always clear. The right choice
CX *01-018* of visualization can help to ***show complex datasets in a simple graphic and offers an easier data analysis***.

The visualization of data [34] is based on PC generated graphic systems and enables us with the better apprehension of data.

"Computer-based visualization systems provide visual representations of datasets intended to help people carry out some task better. These visualization systems are often but not always interactive. Resource limitations include the capacity of computers, of humans, and of displays. The space of possible visualization system designs is huge and full of tradeoffs." [22]

The definition of Preim [20] can be presented as follows:

"Visualisierung ist der Prozess der Transformation von Informationen in eine
CX *01-019* *visuelle Form, die es dem Benutzer auf visuelle Weise gestattet, **verborgene Aspekte in den Daten zu entdecken, die für Exploration und Analyse wesentlich sind**."*

It is very important for the data visualization how humans construct the meaning of pictures.

The "Linked Data Visualization Model" (LDVM) was suggested by Brunetti et al. [43]. The LDVM should permit it to connect datasets with different visualizations in a dynamic way. To be flexible, the LDVM model describes several levels for this visualization. Each level consists of a declaration of a defined pipeline of transformations. The basic of the model is the Data Stage Reference Model of Chi [33] and a generic library for the extraction and the visualization.

The components are suitable with the model of Chi for LOD. Brunetti et al. [43] describes the components as:

	- RDF-data:	Raw-data (taxonomies, vocabularies, ontologies, etc..)
	- Analytical extraction:	Receiving the data extractions out of the raw-data calculation of aggregated values
	- Abstraction of visualization:	Visible information with a visualization technique
	- View:	The result of the visualization process
	- Transformation of data:	Converts raw-data with SPARQL Query Templates
	-Transformation of visualization:	Processing the data, to make them suitable for the visualization
CX ***01-020***	*- Visual mapping-transformation:*	***Visual view of the visualization abstraction to gain one view***
	- SPARQL operators:	SPARQL-functions (SELECT, FILTER, COUNT, GROUP BY, etc.)
	- Analytical operators:	Functions that process the extraction of data
CX ***01-021***	*- Visualization operators:*	***Visual variables (size, color, position)***
	- View operators:	To process the view actions, i.e. rotate, scale, zoom

On this basic model a prototype "LOD Visualization" was implemented that runs on App Engine24.

The LDVM is a very helpful guidance for understanding the programmatically way of arranging Linked Data software. It describes the model view controller principal in a more general way which often is used in modern software development. Specific components of programs can be inserted into this partitioning. By giving attention to the dependencies of these components among each other's it is easy to make drafts of such software.

Demands for the visualization of LOD 4

The general demands to be able to implement visualizations of Linked Data are [05]:

- *The expertise of the user*
- *The principles of information research*
- *The possibility to process multidimensional data:*
- *The possibility to identify and highlight relational data*
- *The possibility to extract data*

The users should generally be aware of concepts and keywords of Linked Data. Principles of information research like background, methodology or dissemination should be in common use.

Demands for expert users to be able to implement visualizations for Linked Data are [05]:

- *Navigation through Linked Data structures*
- *Exploration of graph data (content and connections)*
- *Verification of false content or a false syntax*
- *Queries with formal syntax*
- *Publication*
- *Extraction of data for various usages*

Expert users should be able to build a connection to the endpoint da-
cx 02-028 tabases of LOD and to ***visualize entries of LOD as labeled graph data*** as
they should be able to filter false data or data consisting of false syntax.
Experts should either be aware of generating SPARQL-queries out of in-
put data or prepare raw content data for publication use or for exporta-
ble reports and further usage.

About the theoretical part which describes the guidelines of graphic en-
gineering in specification of the synergies of Linked Data visualization
the concept of the work done is demarcated. The attached description
shows the practical appropriation of these concepts in a formal way.
Split able in three main key phrases the methods can be named ***proto-***
cx 01-022 ***typing layouts, applying programmatically changes and gaining results***
by testing.

Respective to the number of users that should prefer one visualization
cx 01-023 the presentation of data must be conditioned in a simple form to ***cover***
most of the exploitation methods.

Prototyping the new visualization **5**

To gain drafts of graph visualizations it is necessary to analyze why the existing layout standards are commonly used. The most important details to construct visualizations are gained by the components visualizations exhibit as like their interplay and therefore ***noticeable design restrictions***. The components and restrictions concerning the visualization of Linked Data relations are itemized in the following list. *cx* ***01-024***

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 amount of nodes in the view
- *node position:* restricts the arrangement of nodes in the view
- *edge length:* restricts the space between the nodes
- *edge formation:* restricts the readability of the information
- *view size:* restricts the space of viewable information
- *node control:* restricts the operations of manipulating the nodes
- *view control:* restricts the operations of manipulating the view

Per elaborated heuristics of the general design principles the restrictions treating application views gave the vertices for possible visualizations.

One part of the design decisions that primarily influenced the result was a research and a trial phase with the requisition to gain clearness of how much data relations a query result includes on an average to back decisions about how the data relations can be visualized in exclusion of seamless transitions of visualizations that had to be filtered out.

CX 01-025 Concisely it can be said that the supported designs remaining as an ***alternative to network visualizations*** that are state of the art endows a main decision in terms of view control. About the web application RelFinder that will be introduced in the next chapter the handling of such a network view is managed by mouse drag functionalities on the nodes as like on the view and therefore uses the empty space between the nodes. Drafts of different visualizations have therefore to be analyzed concerning if the usability of other visualizations have equal excellence reusing the drag functionality for the view. As an alternative with a similar mouse
CX 02-029 device another question treated if ***graph visualizations could be scrolled as a list*** and therefore as main assignment of tasks if semantically linked
CX 02-030 ***relations can be arranged in another way***.

Additionally, the successive order of the RDF-triples as typed links had to be visualized in reason to keep their comprehensibility.

In relation to the list of restrictions it turned out that the visualization within its 3 components can be optimized. These main components are
CX 02-031 the ***enclosed paths*** of a result set, the ***labeled node triples*** of a relation and the ***dimension of the view*** considering data oversize. A result set is defined by the results one query gives.

As shown in figure 03 for me, the best way to arrange the labeled node
CX 01-026 triples appeared to use them as one related record per ***the law of good shape*** and additionally to the direction of reading and writing a grammatical clause.

By means of the enclosed paths which had to be kept unchanged in terms of the grammar of Linked Data only one modality offered the possibility of optimization. This optimization could be described as a
CX 01-027 sort of semantic filtering process where labeled data which has ***identical information could be unduplicated by preserving the matter*** of the information as normalization.

Mentioning the left possible transitions of visualizations, it can be said
CX 01-028 that a minimalistic data view is mainly ligated with the ***positioning of the nodes given as a data triple***.

CX 01-029 With hindsight to these two restrictions an ***interference taking account of the positioning in line of one relation*** was developed.

Network views have dynamic node positions that change on each single dragged node. That is often a disadvantage for the users and demands maintaining an overview of the components changed. It is also difficult to order these components at will.

To enable these features, positioning had to be questioned most of all.

To describe the process of such a refinement one of these drafts is dealt with below.

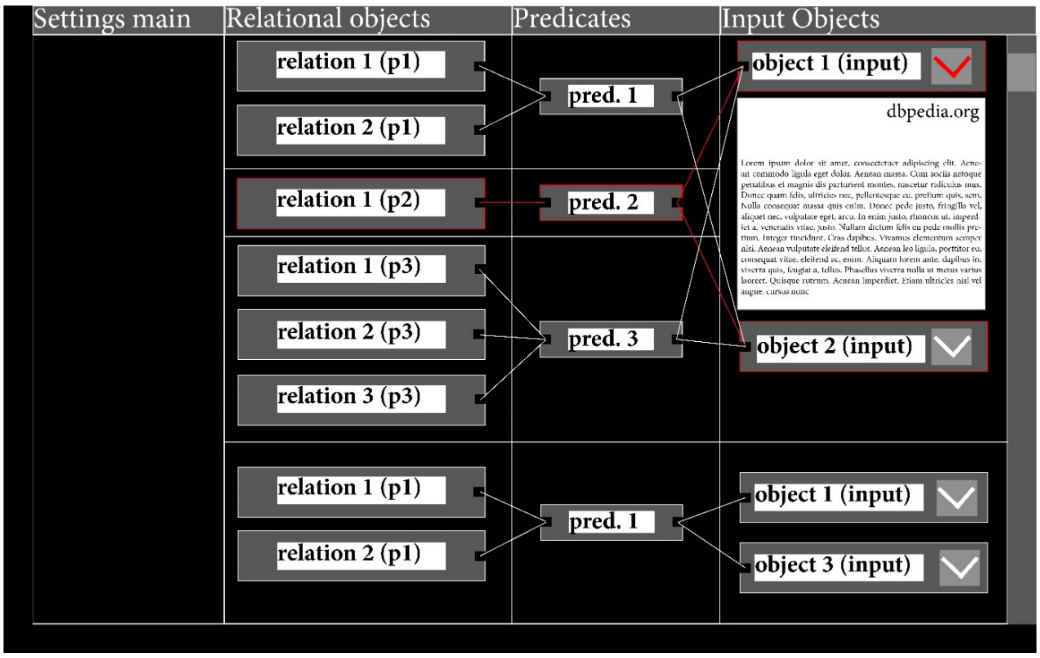


Figure 14: Draft visualization

The figure shows the first draft of the adapted visualization. The relations are arranged here in a scrollable list. 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 each additional
cx 01-030 found nodes are objects in grammar. ***To filter identical predicates is then possible for direct relations*** of two or more given input nodes preserving the completeness of the path.

cx 01-031 The constructive idea for the adaptation of the visualization was the ***normalization of the calculated relations*** in RelFinder as equally ***the positioning of the nodes as a list***. The results of the default version of RelFinder
cx 02-032 shows the ***relations ordered in circle or ellipse paths as a node-link diagram***. The given nodes build the start and the end of the path. Within there are found relations connected by graphs using the force-directed layout algorithm. If a node is clicked, the graphs that depend on the relation highlight these paths. An abstract of the clicked node can also be seen in an info box positioned in the left corner of the main application. Each relation consists of the RDF data scheme that includes the subject, the predicate and the object of the relation.

cx 01-032 To normalize the relations I suggested to ***limit identical predicates to a unique representation*** that replaces several appearances of nodes using similar predicates in the relations. Semantically this method preserves

http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

(12.08.2015; 01:49 pm)

the sense of information given by the relations. To add critics to the draft the alignment of the predicate of a relation is centered horizontally and vertically but concerning the enclosed paths of a result set the validity of
CX 03-024 this method is false, because ***the paths of the relations have dependencies among each other's*** which must be followed. A formal explication for this giving respect to this grammar law is described in mentioning that each graph that empties into the predicate nodes of a relation can be oriented. If the predicates of the concerned relations have identical
CX 01-033 terms ***these predicates can be normalized. The sense of information is preserved then***.

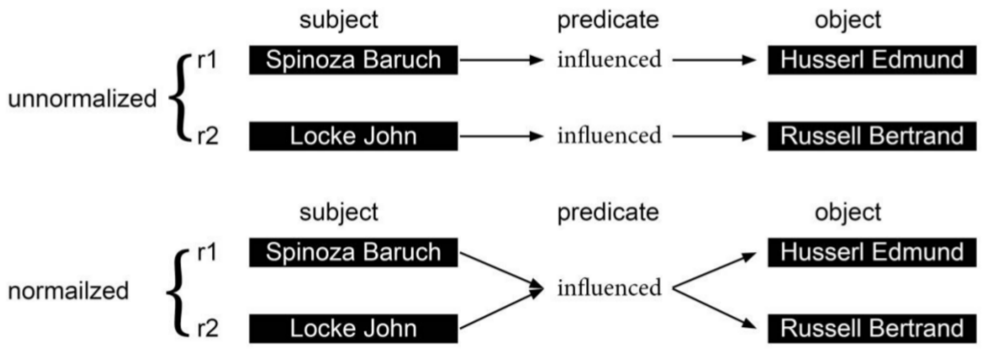


Figure 15: Normalized predicates

The idea of the normalization of the predicates in graph data primarily appeared there. The paths of a clicked subject or object should be highlighted too. Another idea was to provide blank nodes for direct input which was not applied then.

CX 01-034 Each clicked ***relation*** was ***intended to be able to be highlighted separately***. Therefore, the idea appeared to hide all other paths.

The final arrangement of the labeled nodes therefore could have been trashed and concerns the mentioned restrictions. The three columns of
CX 01-035 the draft were kept and ***horizontal space between these columns was***
CX 01-036 ***added for better readability of the directed graphs*** as like the ***first column was transcribed for representing the given nodes*** (subjects) separately and unrepeated by vertical spacing. To repeat the relations their self-depending on their in common, unique and normalized predicate and found nodes would also have been possible for example in duplicating only affected subjects.

6 Introducing the views

6.1 node-link view (RelFinder)

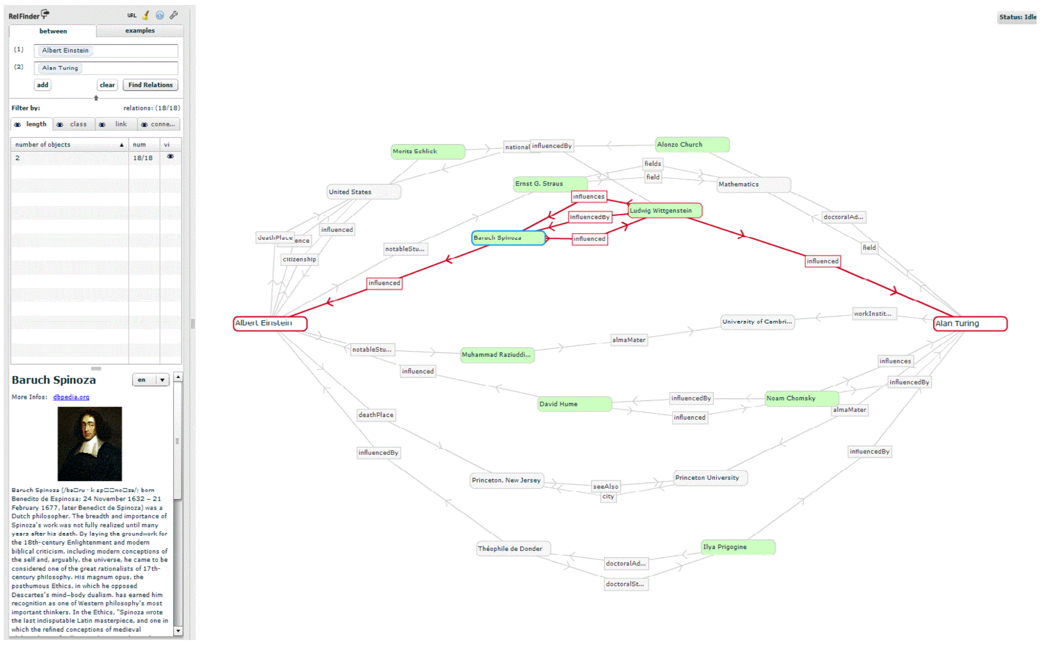
RelFinder is graph visualization software based on "Adobe Flash" showing relations to specific objects or subjects of SPARQL queries by simple input fields. Within it is possible to put in two or more input phrases. Their relations are calculated then.

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

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

http://www.visualdataweb.org/relfinder.php

(12.08.2015; 01:49 pm)



Baruch Spinoza

More Infos: dbpedia.org

Baruch Spinoza (/bə□ru · k sp□□no□zə/; born Benedito de Espinosa; 24 November 1632 – 21 February 1677, later Benedict de Spinoza) was a Dutch philosopher. The breadth and importance of Spinoza's work was not fully realized until many years after his death. By laying the groundwork for the 18th-century Enlightenment and modern biblical criticism, including modern conceptions of the self and, arguably, the universe, he came to be considered one of the great rationalists of 17th-century philosophy. His magnum opus, the posthumous Ethics, in which he opposed Descartes's mind–body dualism, has earned him recognition as one of Western philosophy's most important thinkers. In the Ethics, "Spinoza wrote the last indisputable Latin masterpiece, and one in which the refined conceptions of medieval

Figure 16: RelFinder

CX 02-033 By a click on the subject or the object the path between the given search phrases is highlighted then. The ***subjects and objects are implemented as nodes while the predicates represent the relation itself as a relation node***. The relation includes the paths linking the RDF-triples.

CX 02-034 The software additionally has information based on the results of the calculated relations and ***provides a browse able overview of a textual abstract***.

To make it simple for beginners there is a list of examples which can be explored.

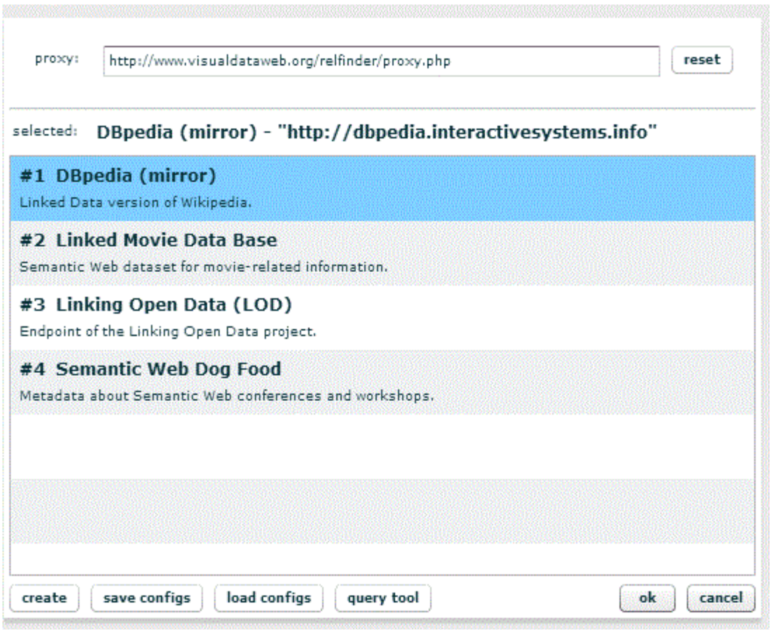
The connection to the endpoints of the LOD cloud can be edited, removed or added.

http://wiki.goodrelations-vocabulary.org/Public_endpoints

(12.08.2015; 01:49 pm)

https://www.ebi.ac.uk/rdf/documentation/SPARQL-endpints

(12.08.2015; 01:34 pm)


proxy: http://www.visualdataweb.org/relfinder/proxy.php

selected: DBpedia (mirror) - "http://dbpedia.interactivesystems.info"

#1 DBpedia (mirror)
Linked Data version of Wikipedia.

#2 Linked Movie Data Base
Semantic Web dataset for movie-related information.

#3 Linking Open Data (LOD)
Endpoint of the Linking Open Data project.

#4 Semantic Web Dog Food
Metadata about Semantic Web conferences and workshops.

Figure 17: Endpoints RelFinder

The default configuration of RelFinder includes:

- *DBpedia (mirror):*	Linked Data version of Wikipedia
- *Linked Movie Data Base:*	Semantic Web dataset for movie-related information
- *Linking Open Data (LOD):*	Endpoint of the Linking Open Data project
- *Semantic Web Dog Food:*	Metadata about Semantic Web conferences and Workshops

For the exploration of the content that lead to the commitment of the chosen search requests preparing the user test a specific endpoint of the DBpedia was used with several settings like limited amount of relations and so forth.

Endpoint URL: http://dbpedia.interactivesystems.info

This endpoint was used for all processing of this work.

For experts the concept of the implementation of RelFinder which language type is actionscript3 including an introduction to object-oriented programming and further references describing the construction of the software in particular Appendix I is attached.

The following chapter introduces two views that were implemented in adaption to the original node-link view: A linear view and a radial view. Both are described below.

6.2 Introducing two adapted views

6.2

6.2.1 linear view (RelFinder)

6.2.1

The linear view is a sort of list view that 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.

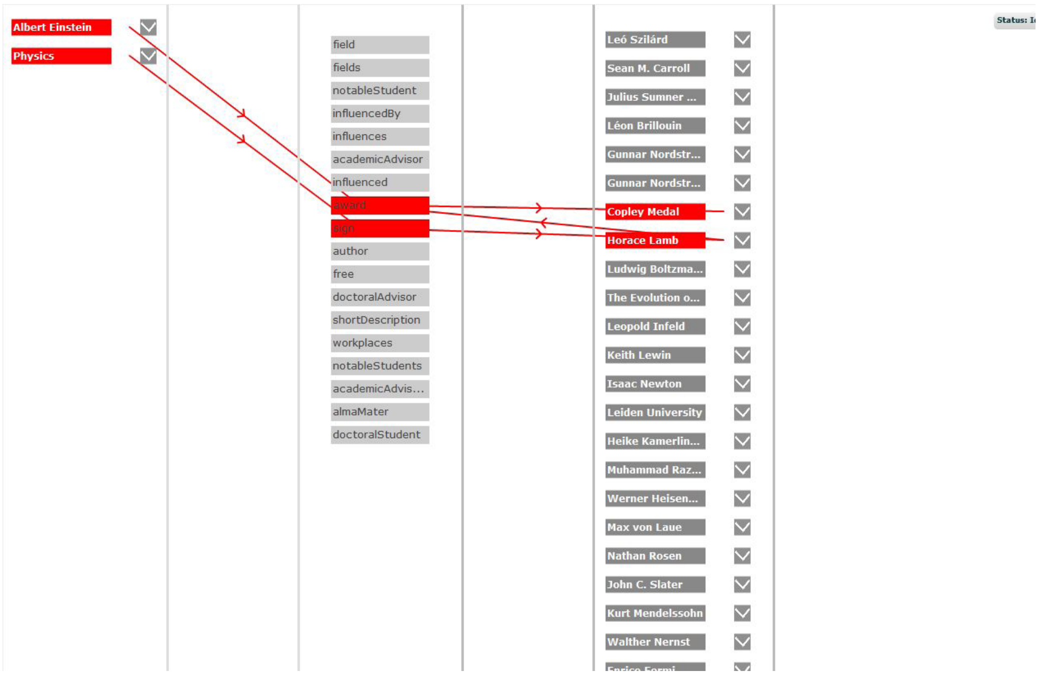
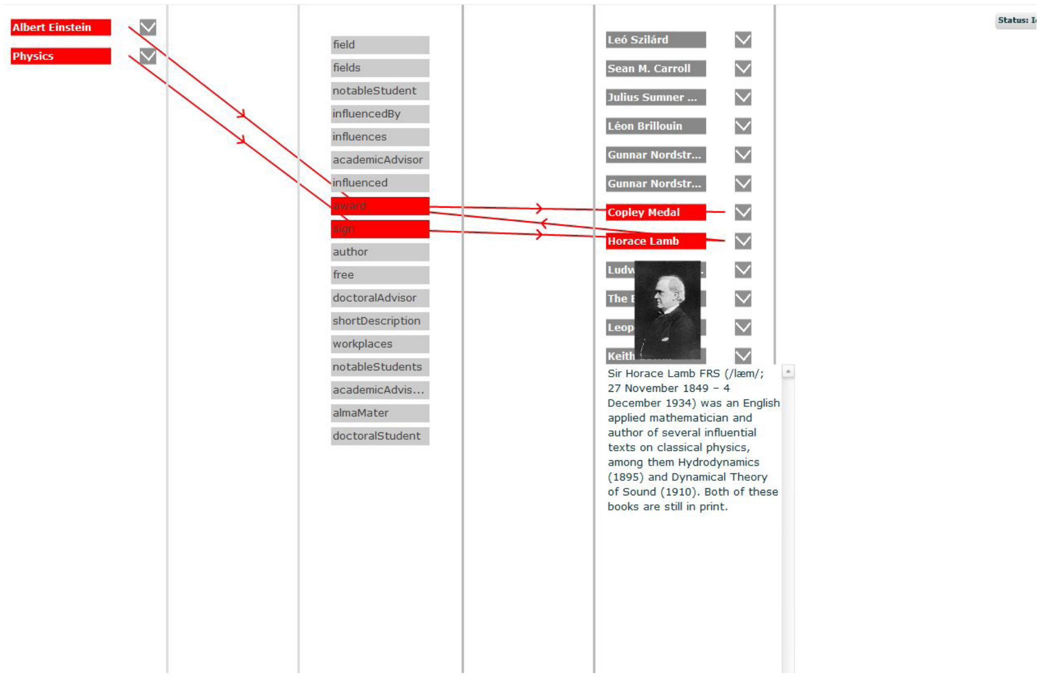


Figure 18: Adapted visualization of RelFinder (linear)

As shown in figure 18 three main columns stand for three parts of a relation. The first column is preserved for the given nodes and the second
CX 01-037 one for the predicates. The ***nodes of the columns can only be dragged vertically***. The third column includes the found nodes. This enables the user to place them along the y-axis for example to order the nodes when the abstracts are already read or a specific context wants to be visualized by giving the found nodes a special order.

CX 01-038 As main part of design principle ***the positioning of the nodes appears as vertical ordered list***. The main difference to an ordinary list findable in websites or excel-sheets is that first the graphs connecting the nodes require spacing to show a good visualization. Secondly in comparison to normative lists including more than one column the predicates and objects of a result set that are connected should appear on similar horizontal positions making use of the normalization. This conclusion is handled as future work and will be described at the end and technically in Appendix I.

CX 01-039 To ***toggle the abstracts a button with an arrow downwards can be clicked*** as shown in figure 19. If the content of the abstracts cannot be shown fully in a view it can be scrolled with the mouse wheel if the cursor is put in focus to the panel or with the vertical scrollbar. To stop the main panel from scrolling when an abstract panel is scrolled with the scrollbar the focus of the node must be resized so that the mouse is not operating on both panels. The graphs have arrows for the visualization of dependencies in showing the direction to or away from a node to determine the semantic unambiguous assignment of the relation.



Sir Horace Lamb FRS (/læm/; 27 November 1849 – 4 December 1934) was an English applied mathematician and author of several influential texts on classical physics, among them Hydrodynamics (1895) and Dynamical Theory of Sound (1910). Both of these books are still in print.

Figure 19: Applied abstract view in RelFinder (linear)

The objects of the found nodes are grouped successively depending to the path as initially calculated and therefore have no sorting step. To offer an explanation here, the word path describes the connection of all graphs of a result set. ***As mentioned the representation of the predicates in one horizontal line to the found nodes was not applied.*** This feature is not possible concerning the single appearance of subjects because multiple paths of objects can lead to one normalized predicate. This research will be investigated in the conclusion part of the work. *CX* ***01-040***

While a click on a given node shows the full path, several ***found nodes only show the appropriate part of the path while all other graphs are hidden***. This makes a clear understandable piece of directed information. *CX* ***01-041***

Appendix I describes in detail the main components that had to be reprogrammed. To describe an integrative process of the coding of these components following list gives an overview.

- *normalizing predicates:*	Unification of similar predicates
- *toggle able abstracts:*	Show and hide abstracts in the view
- *repositioning nodes as list:*	Set ascendant 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

The main difficulties of the reprogramming can be announced as changing a force directed layout to a linear list organization in observance to the dependencies of an already optimized code.

6.2.2 radial view (RelFinder) 6.2.2

CX 01-042 ***The radial view visualizes the found node's positions around a circle***.

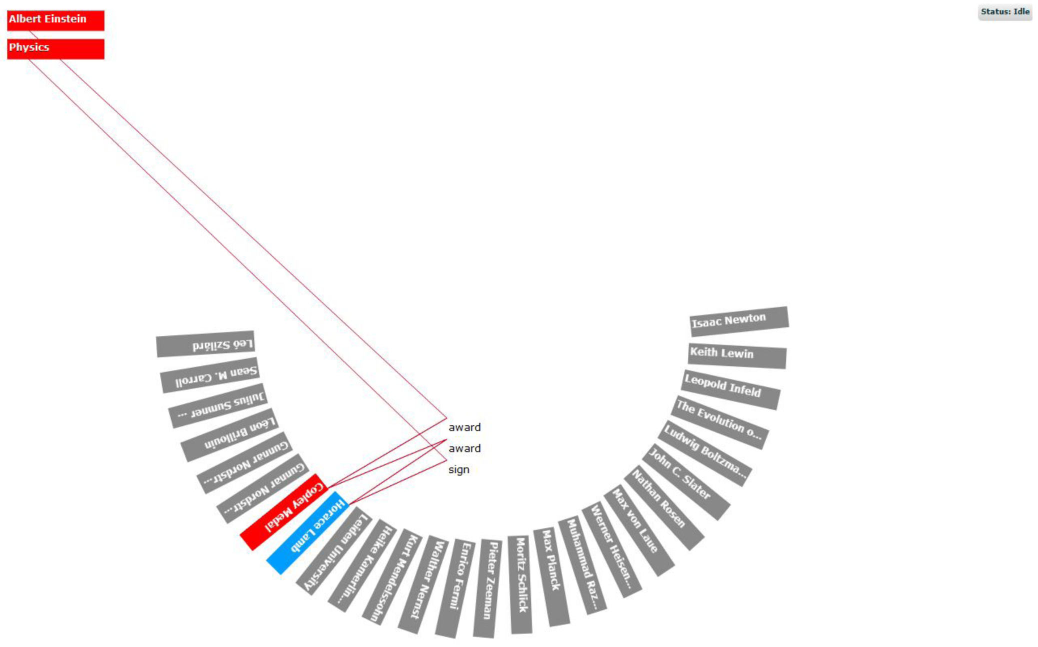


Figure 20: Adapted visualization of RelFinder (radial)

CX 01-043 The ***predicates are placed in the middle of the circle***. The difficulty of the visualization of the predicates lies in the space their visualization takes and is solved by repeating the y-values after several steps, starting at 0 again. This could be managed by a mod-operator in the counted values

of the positioning function. For some reasons, it was not possible to display normalized and unified predicates in this view. The reason therefore was that the implementation of the normalized predicate list interfered the blending of the filtered predicates concerning the clicked relation. The radial view was implemented without having arrows for the visualization of the graph's directions. Therefore, as a main result of my work
CX 01-044 it can be recorded that ***the principles of semantic information visualization gain out of the possibility to connect nodes by adding oriented graphs*** by means of defining sematic dependencies a relation can have by its predicate concerning the functionality of this vocabulary. This relevant change was not considered in the phase of reprogramming the visualization and was identified in the phase of testing. On this a research result is noted in the conclusion part.

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

- *hide unselected predicates:*	hiding predicates of unselected sub-relations
- *repositioning nodes as circle:*	set positions of nodes around a circle
- *highlighting specific paths:*	hiding paths of unselected sub-relations
- *controllable components:*	applying clickable nodes
- *layout of components:*	changing the visualization of labels and nodes

Both views are different to the standard visualization of a node-link view which is common in Linked Data visualizations and therefore had to be tested in a prepared workflow. To gain knowledge about the interaction advantages and the correlating design which concerns the main question of the master thesis.

Methods of testing 7

7.1 Design 7.1

To primarily establish the design phase of the testing, the section that described the part of prototyping the layouts is necessary in addition to the main questions of the thesis in repeat; The test design can be described in detail.

How can typed links in Linked Data be visualized for to a broader number of users?

Which graphic representations are suitable for typed links in Linked Data?

Is there an interrelation between the visualization and the functionality of graphs concerning the interpretation of data?

To find relevant methods concerning these questions the usage of the paper "*Task taxonomy for graph visualization*" [49] was very helpful, from which ***essential tasks for the specification of the results were chosen***. Based on the following listing the executed tasks describe the propositions for the methods. *CX* ***01-045***

Topology-Based Tasks

CX ***01-046*** - *Adjacency (direct connection):* ***Find the set of nodes adjacent to a node***

CX ***01-047*** - *Connectivity:* ***Identify connected components***

Attribute-Based Tasks

CX ***01-048*** - *On the nodes:* ***Find the nodes having a specific attribute value***

CX ***01-049*** - *On the links:* ***Given a node, find the nodes connected only by certain types of links***

Browsing Tasks

CX 01-050 - ***Follow a given path***

The methods of testing consist of the interaction of the selected tasks applied to the check ability and the perception of the given visualizations of RelFinder and can be specified in more detail reusing these task taxonomies.

Further explanations depend to functionalities that match to a task are
CX 01-051 the ***adjacency represented as 2 given nodes in relation to the found nodes***, the connectivity represented as graphs of the result set, labeling on
CX 01-052 the nodes, graph orientation on the links as semantic task and ***further information about the visualized relation in following a given path***.

CX 01-053 To find out in which of the views a node could be found faster, the ***time was stopped from the beginning of a test set to the correct click on the node***.

CX 01-054 A second method was to stop the ***time a test subject needed to find information he was asked to gain out of the abstract***. To evaluate the results the quartile of the time all users took in comparison to the different views was computed. The test subjects had simultaneously to find nodes of a relation given and identify connected components by being asked to
CX 01-055 explain the sense of the relation as a sentence. So, the ***specific attribute value of the relation was the predicate of the RDF-triple to complete the matter of it***. The graphs had therefore to be followed. Finally, an information part of the abstract as an additive information was questioned. Thinking aloud and voice recording was also done during the test.

To evaluate the results concerning the questions that pointed out the
CX 01-056 quality in the test where ***valued as false or true*** and each additive comment was written down.

Finally, a survey had to be filled out concerning general data about the gender, the age and the educational status of each test subjects and led
CX 01-057 into the questions about the ***usability, the handling and the preference of the views*** by having to choose one of the views as option.

7.2 Test subjects

20 test subjects participated in the test. To describe their general characteristics a survey was held in addition to the test tasks. The gender of the test subjects was exactly 50% male and 50% female. 75% of the test subjects where between 21 and 39 years.

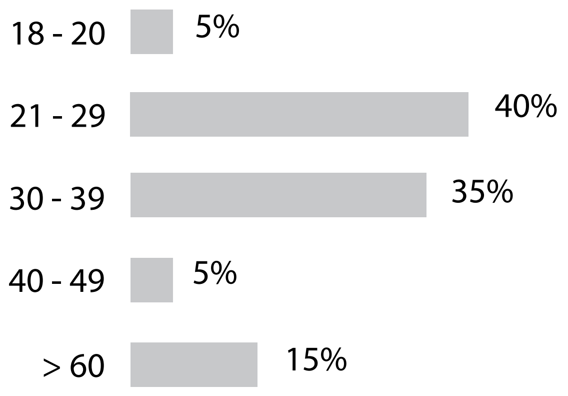


Figure 21: Age in years

Most of them had a general qualification, a bachelor's or a master's degree. For a more specific evaluation the expertise of the test subjects a multiple-choice selection could have been given. The results are shown as follows.

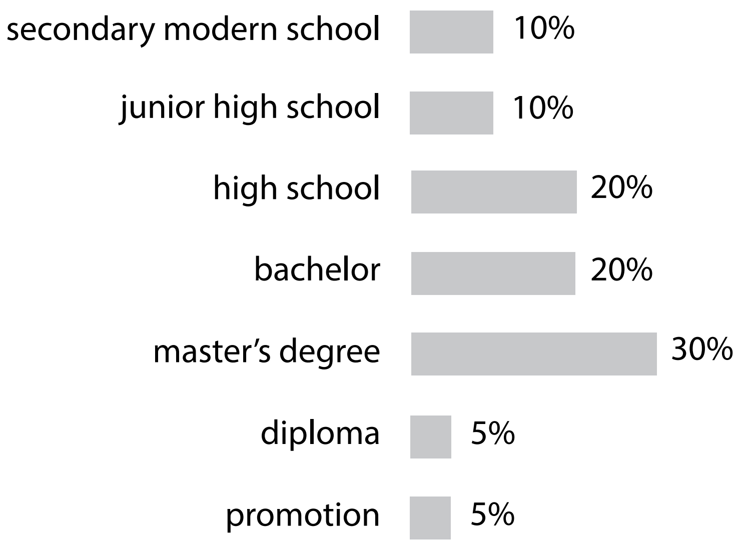


Figure 22: Education

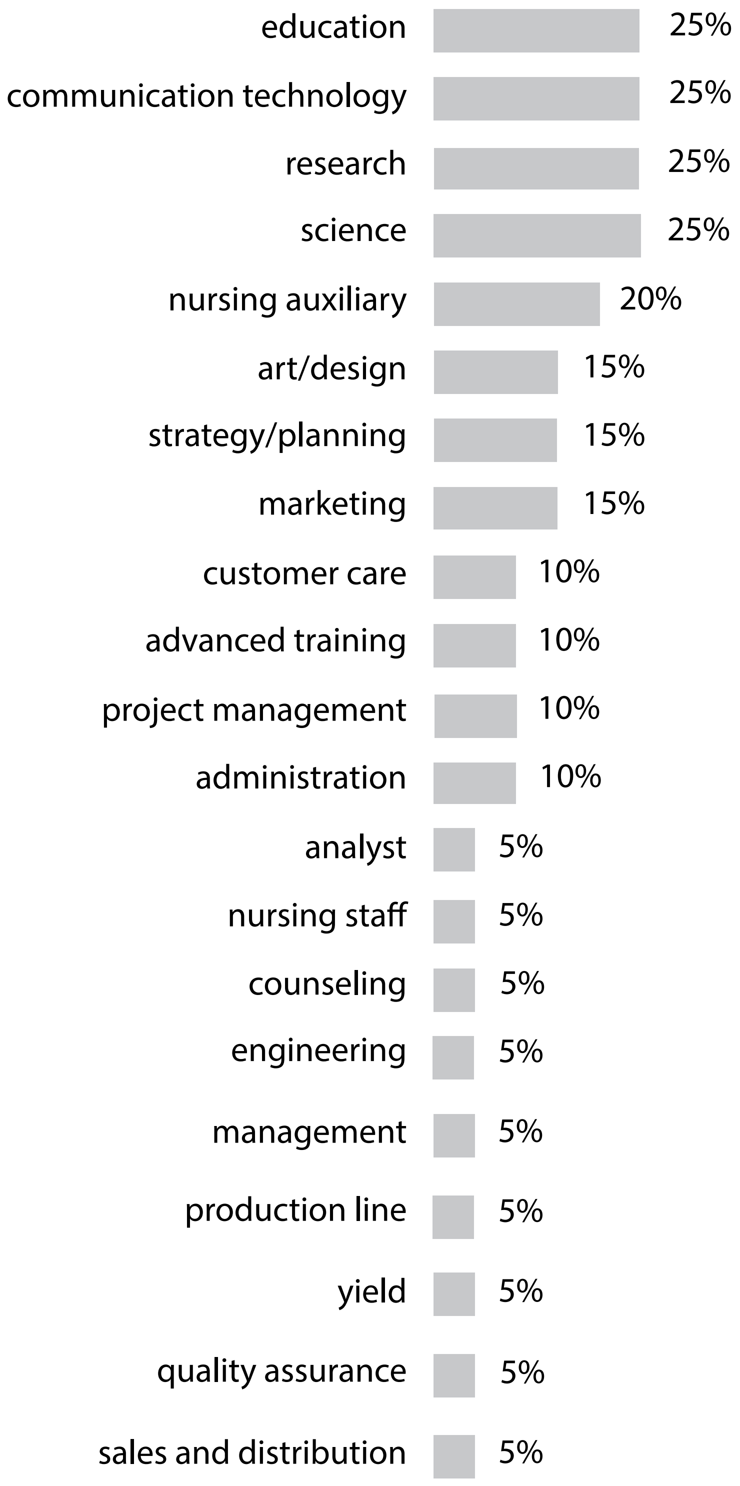


Figure 23: Apprenticeship (multiple)

7.3 Materials

7.3

The user test took place in the usability labor of the University of Applied Science St. Pölten. An eye tracking device from SMI (Sensomotoric Instruments) therefore was used.

http://www.smivision.com/en.html

(28.05.2016; 12:44 pm)

It included the 2 different types of questions for a navigation task and for an information task that contained reading out of an 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 one task was set to 3 minutes then the next briefing faded in. The format of the gaze data
CX 01-058 was a ***screen recording including the scan path of the test user and the mouse clicks on the test screen***, to gain the results concerning the required durations.

The three views of RelFinder were managed always having different orders and different question sets of same quality per test subject. The question about the navigation contains as task to click a node of the
CX 01-059 relations. The ***main attention was given to the visible graphs of the node that had to be clicked***. The number of related graphs had to be less than two in reason of keeping a basic complexity. The second task included a word which had to be searched in the abstract. The setup had been defined as three different views and three relations given as a pair of given nodes. For each of the relations two tasks had to be fulfilled. To explain further details the tasks are described below. The test was held in German language and was translated to English then.

7.3.1 Test procedure

7.3.1

Within this point a description of the briefing of each test setup a test subject had to manage is described. The names appearing in the test setup concern persons in real life taken from the DBpedia that lived from the 18th to the 20th century that got famous for their work.

The scheme of the description can be explained in first the two tasks treating navigational and informational questions and then the solutions in each view as a diagram taken out of the program. (The abstract‘s views of the node-link and the radial view where only plotted once in reason to have similar design and have the same position as origin).

Task setup 1: David Hume and John Locke

Task 1: Search and click the node Arthur Schopenhauer! Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Arthur Schopenhauer! Read in the abstract, when Arthur Schopenhauer was born!

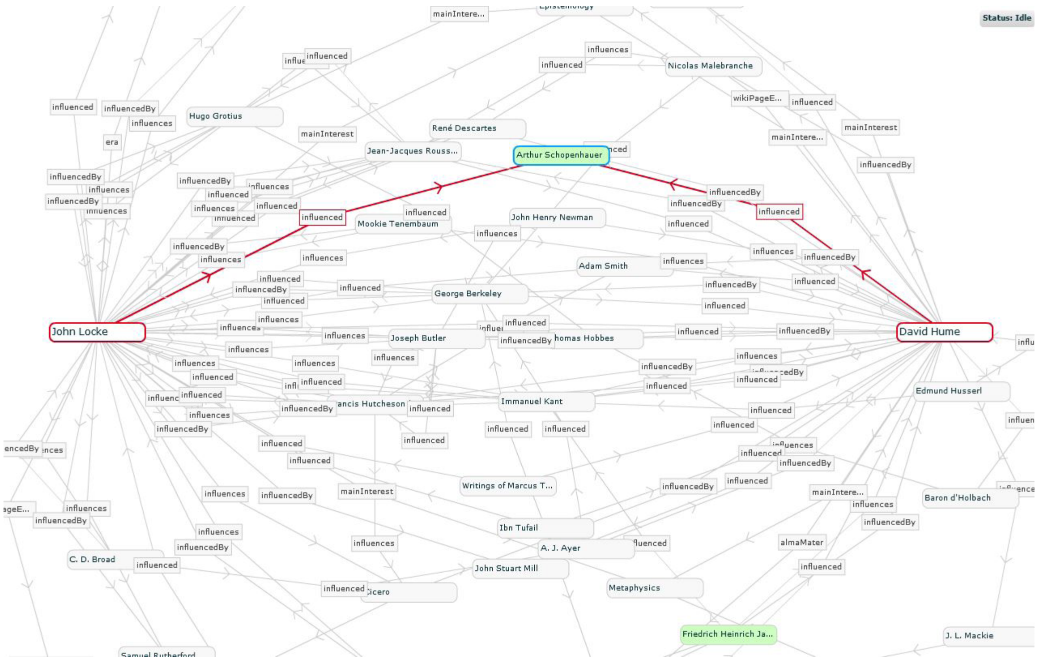


Figure 24: Node-link view Arthur Schopenhauer

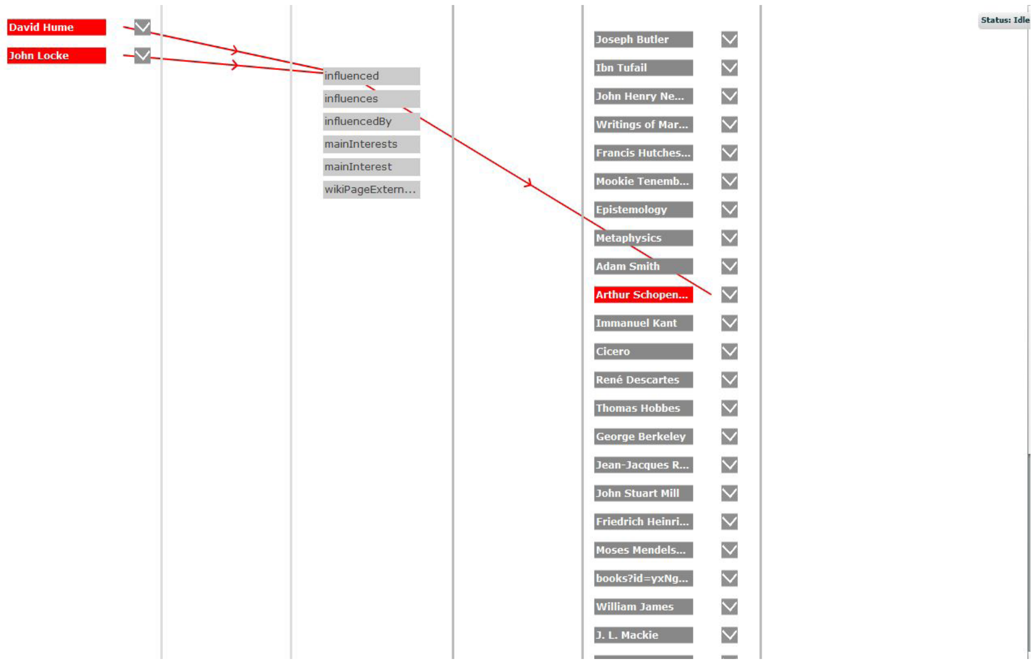


Figure 25: Linear view Arthur Schopenhauer

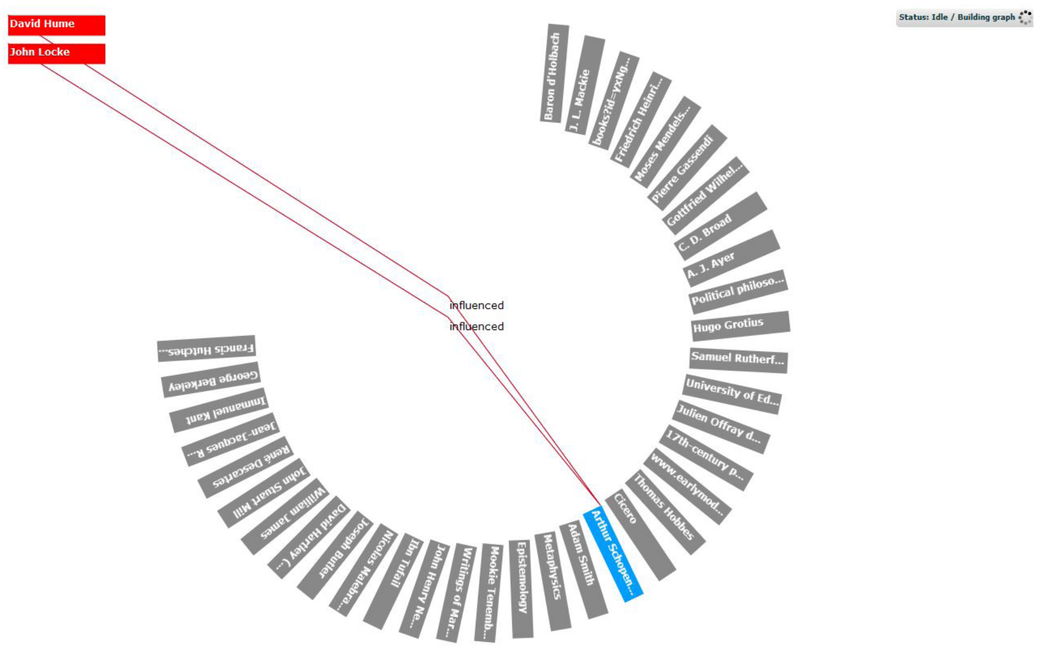
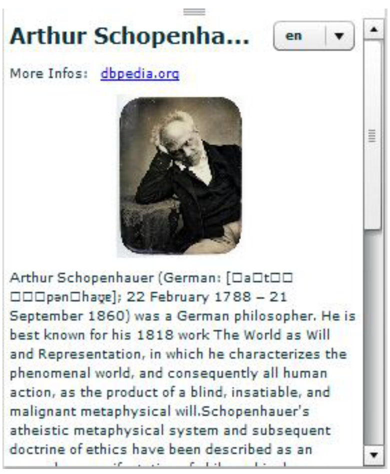


Figure 26: Radial view Arthur Schopenhauer



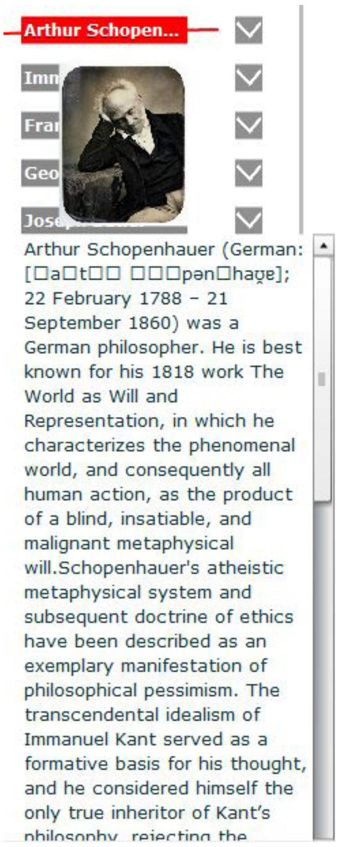
Arthur Schopenha...

en

More Infos: dbpedia.org

Arthur Schopenhauer (German: [□a□t□□ □□□pən□haʊɐ]; 22 February 1788 – 21 September 1860) was a German philosopher. He is best known for his 1818 work The World as Will and Representation, in which he characterizes the phenomenal world, and consequently all human action, as the product of a blind, insatiable, and malignant metaphysical will.Schopenhauer's atheistic metaphysical system and subsequent doctrine of ethics have been described as an

Figure 27: Abstract Arthur Schopenhauer (node-link, radial view)



Arthur Schopenhauer (German: [□a□t□□ □□□pən□haʊ̯ɐ]; 22 February 1788 – 21 September 1860) was a German philosopher. He is best known for his 1818 work The World as Will and Representation, in which he characterizes the phenomenal world, and consequently all human action, as the product of a blind, insatiable, and malignant metaphysical will.Schopenhauer's atheistic metaphysical system and subsequent doctrine of ethics have been described as an exemplary manifestation of philosophical pessimism. The transcendental idealism of Immanuel Kant served as a formative basis for his thought, and he considered himself the only true inheritor of Kant's philosophy, rejecting the

Figure 28: Abstract Arthur Schopenhauer (linear view)

Task setup 2: Sigmund Freud and Carl Jung

Task 1: Search and click the node Karl Robert Eduard v. Hartmann!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Karl Robert Eduard v. Hartmann! Read in the abstract, when Karl Robert Eduard v. Hartmann was born!

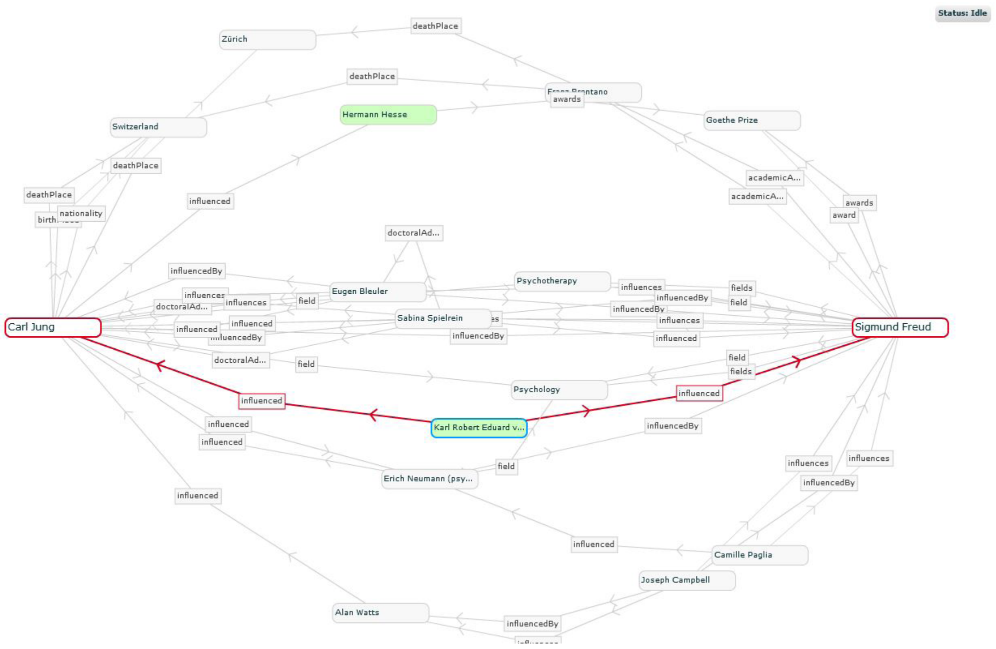


Figure 29: Node-link view Karl Robert Eduard v. Hartmann

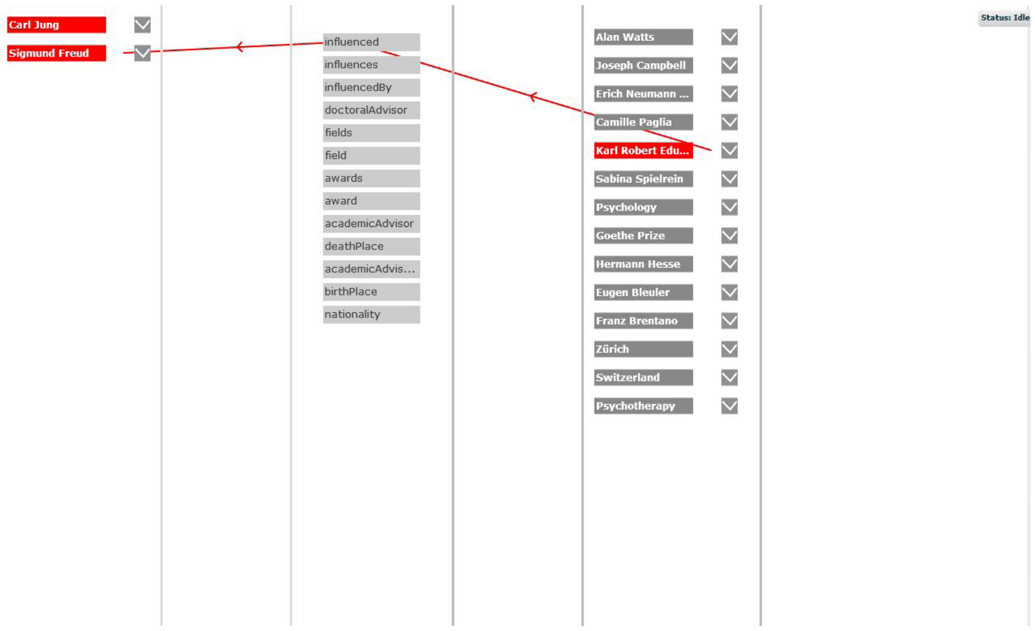
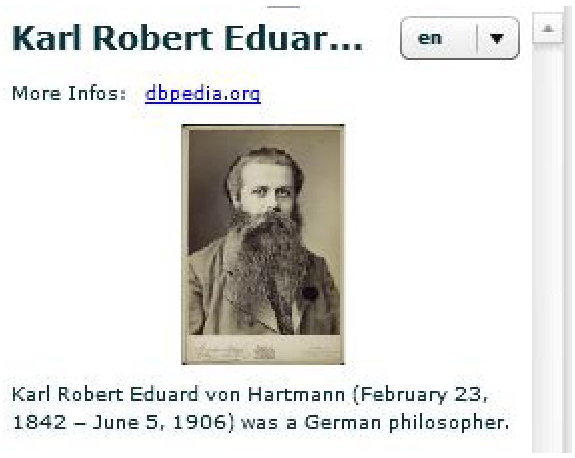


Figure 30: Linear view Karl Robert Eduard v. Hartmann

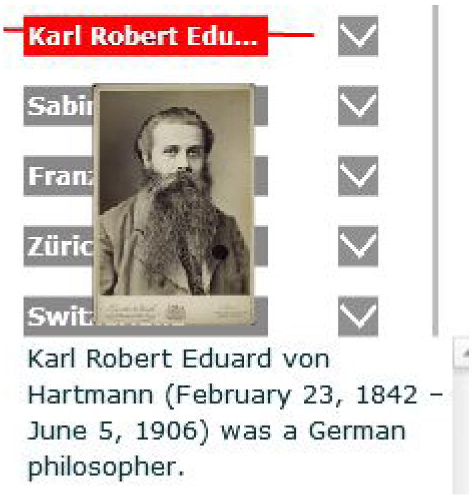

Karl Robert Eduar...

en

More Infos: dbpedia.org

Karl Robert Eduard von Hartmann (February 23, 1842 – June 5, 1906) was a German philosopher.

Figure 31: Abstract Karl Robert Eduard v. Hartmann (node-link, radial view)



Karl Robert Eduard von Hartmann (February 23, 1842 – June 5, 1906) was a German philosopher.

Figure 32: Abstract Karl Robert Eduard v. Hartmann (linear view)

Task setup 3: Friedrich Schiller and Johann Wolfgang von Goethe

Task 1: Search and click the node Weimar Classicism!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Weimar Classicism! Read in the abstract, in which country the Weimar Classicism first appeared!

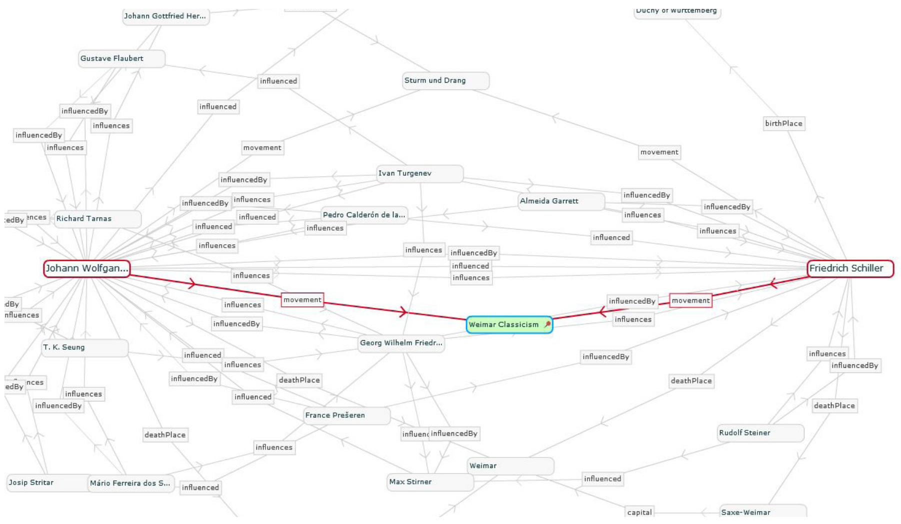


Figure 33: Node-link view Weimar Classicism

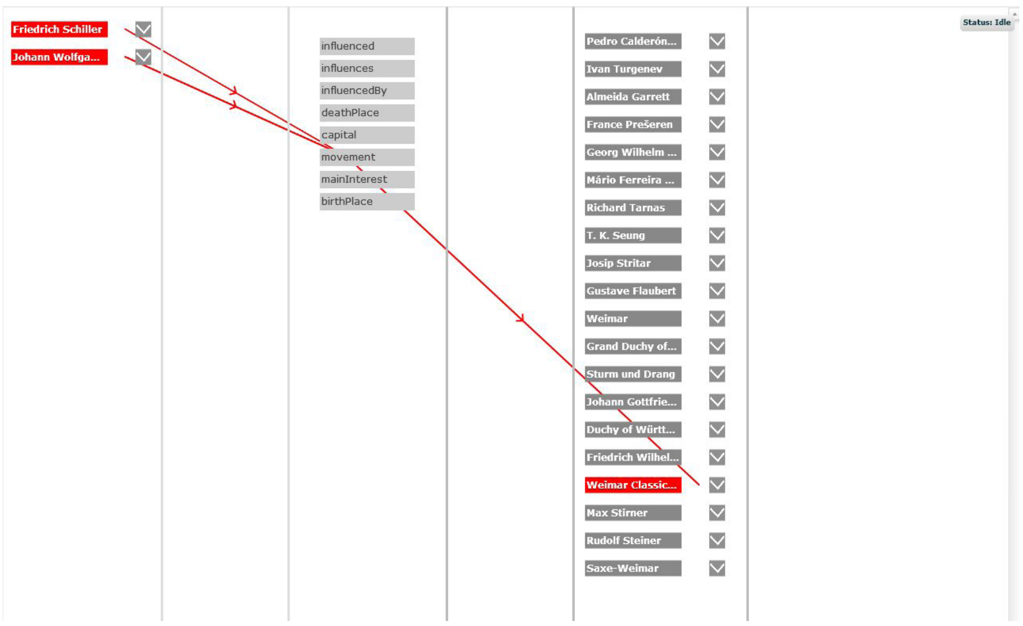


Figure 34: Linear view Weimar Classicism

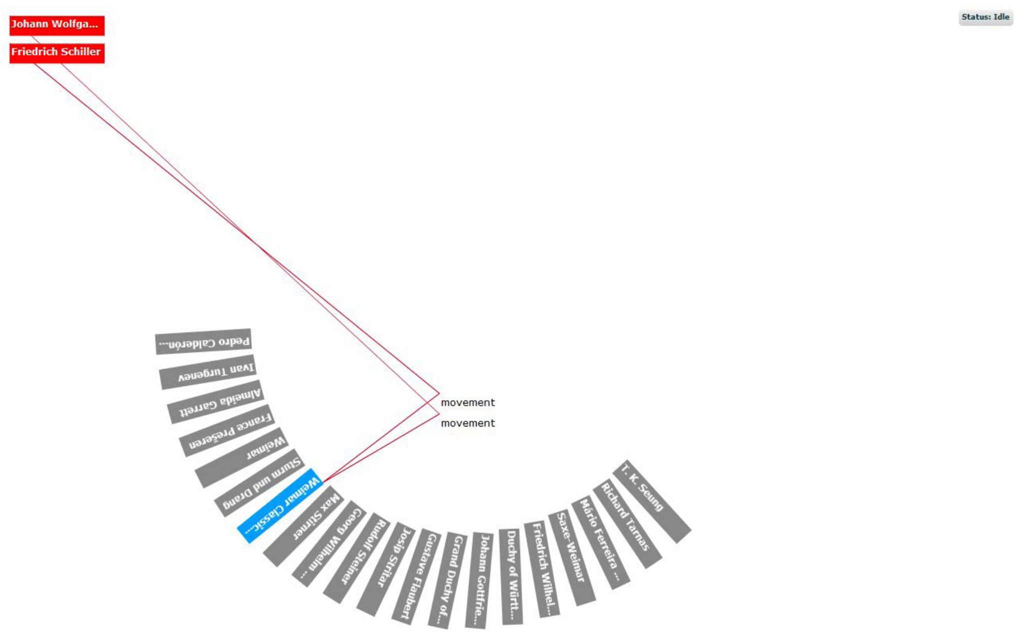
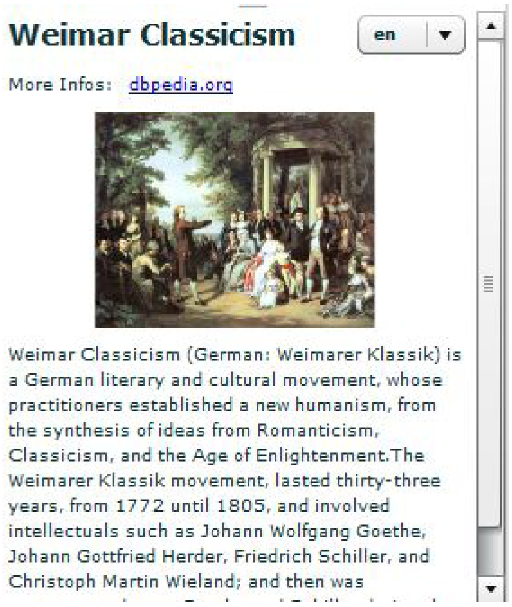


Figure 35: Radial view Weimar Classicism

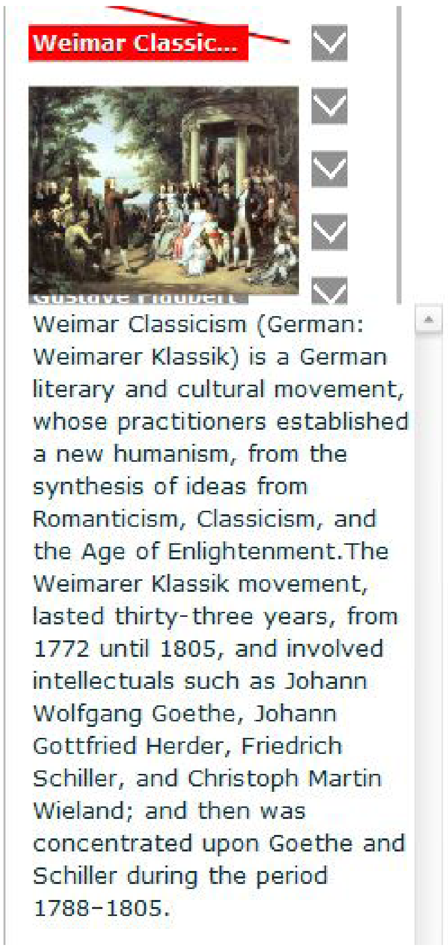

Weimar Classicism

en

More Infos: dbpedia.org

Weimar Classicism (German: Weimarer Klassik) is a German literary and cultural movement, whose practitioners established a new humanism, from the synthesis of ideas from Romanticism, Classicism, and the Age of Enlightenment.The Weimarer Klassik movement, lasted thirty-three years, from 1772 until 1805, and involved intellectuals such as Johann Wolfgang Goethe, Johann Gottfried Herder, Friedrich Schiller, and Christoph Martin Wieland; and then was

Figure 36: Abstract Weimar Classicism (node-link, radial view)



Weimar Classicism (German: Weimarer Klassik) is a German literary and cultural movement, whose practitioners established a new humanism, from the synthesis of ideas from Romanticism, Classicism, and the Age of Enlightenment.The Weimarer Klassik movement, lasted thirty-three years, from 1772 until 1805, and involved intellectuals such as Johann Wolfgang Goethe, Johann Gottfried Herder, Friedrich Schiller, and Christoph Martin Wieland; and then was concentrated upon Goethe and Schiller during the period 1788-1805.

Figure 37: Abstract Weimar Classicism (linear view)

This explained test setup was used for each test subject. At the beginning of the test design there existed a primarily test setup treated in Appendix II that could not be processed. For further investigations the reasons of the necessity to create another test setup did not concern the methods or any other relevant positions of the test design. These reasons are described in this part and could be attached without a lack of influence.

Test results

A description of the results of the done test can be given as follows. The 20 test subjects got a randomized test setup in which the node-link, the linear and the radial view was iterated against the tasks that were randomized too, so that ***none of the test subjects had the same order of views and tasks***. This prevented that the tests had any pros or cons. The training curve from view to view concerning the tasks so was balanced. Each test subject had to learn to understand the visualization given in another way and was asked textually to provide an answer to the tasks. The results concerning the correctness of the answers are shown below. **CX** ***01-060***

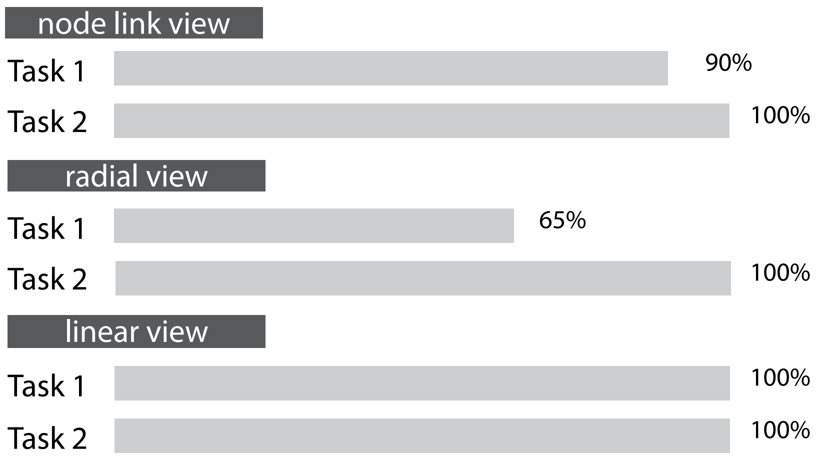


Figure 38: Correctness of the task answers

The second task depends on information out of the abstracts was answered correct by all test subjects.

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

CX ***01-061*** The established time domains the test subjects needed to ***give an answer after the nodes were clicked in 4 cases took longer concerning the list view***. The answers on the node-link view were given faster.

CX ***01-062*** To gain a more specific rating in dependence of the ***errors caused by the missing direction of the graphs visualized as arrows in the radial view*** it must be said that without the arrows the direction of the graphs was not
CX ***01-063*** always clear concerning a mentionable amount of predicates. ***The absent arrows prove the necessity of directed graphs in Linked Data triples***. Without directed graphs, Linked Data triples cannot be understood in many cases. Concerning other vocabularies an important result is mentioned in the conclusion part of the work.

The results of task 1 that figure 39 shows includes the time the test subjects needed to find and click one node. The results as quartiles show
CX ***01-064*** that the ***nodes could have been found fastest using the radial view and the linear view***.

Task 2 shows the time a person needed to find the information out of 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 easy.

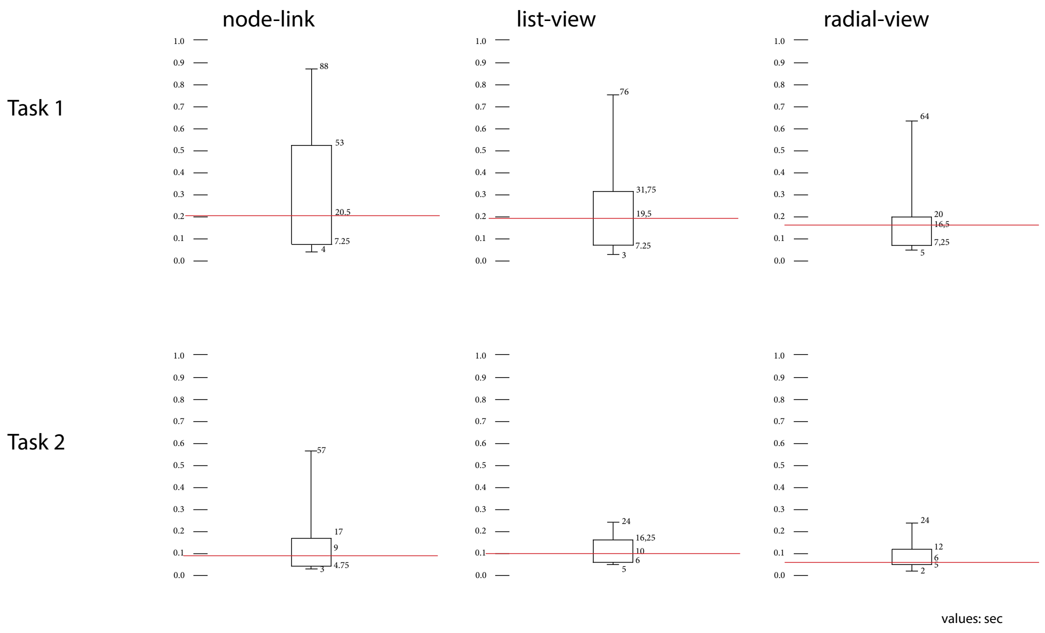


Figure 39: Times (sec) to solve task 1 (find a node) and task 2 (abstract info)

In comparison to the node-link view we can see clearly, that the time
CX 01-065 needed for solving the first task in the ***linear-view and the radial-view has nearly similar minima***. In the third quartile we see that the test subjects needed less time in both new views within the radial-view as best performed. The main differences lay in the maxima and the third quartile of task 1. In this third region we also see that test subjects perform the node-link view in longer time to find the questioned node.

CX 01-066 The ***results of task 2 are much more equal*** depending the three tested views. The procedure of task 2 is more similar concerning all the three views, so this result could have been expected.

The leading reasons of the results can only be assumed. In general by
CX 01-067 reducing the noise a standard network visualization contains ***by hiding unselected parts of the result set the searching times where less***.

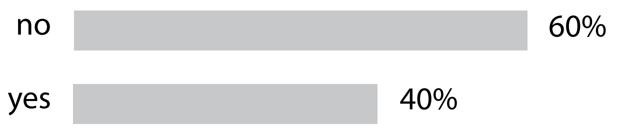
CX 01-068 This ***emphasizes the better positioning and therefore the effectiveness*** of the adapted visualizations. So research question 1 can be presumed
CX 01-069 within the result, that ***a broader range of users are able to handle the alternative views in a better way***.

CX 01-070 Secondly ***the appropriateness of using list or radial views maybe proven by the common workflow of data screening which often is managed by ordinary lists*** so that users can do something with it. For the exploration of this reason the next chapter describes the survey that had to be done in addition to the testing.

The main survey treated the three given visualizations in terms of preference, clearness and understanding and handling. Within this informati-
CX 01-071 on ***the visualization type preferred by the test subjects could be evaluated requesting its clearness and handling in addition***. To inform the test subjects about the visualization types a poster including a figure for each visualization types and a label naming these views was offered as print to simplify the test subject's choice. The charts below describe the results.

Question 1: Did you know Semantic Web and Linked Data before?

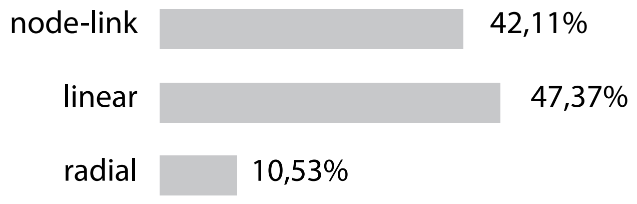
(War Ihnen Semantic Web und Linked Data vor der Testung ein Begriff?)



CX **02-035** Figure 40: ***Semantic Web publicity***

Question 2: Which version of RelFinder would you prefer?

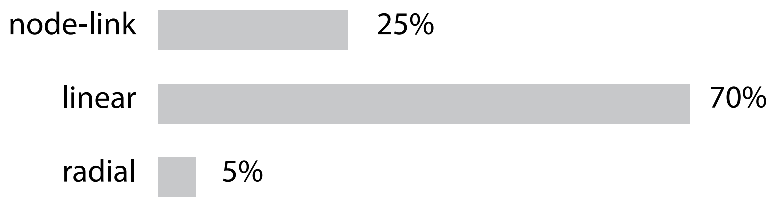
(Welche Version von RelFinder würden Sie bevorzugen?)



CX **01-072** Figure 41: ***Preference of visualization***

Question 3: Which view of RelFinder is more general in your opinion?

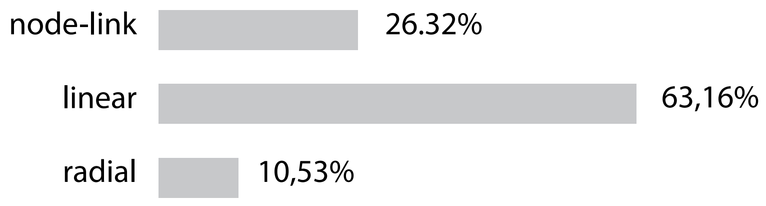
(Welche Version von RelFinder ist Ihrer Meinung nach punkto der Darstellung übersichtlicher?)



CX **01-073** Figure 42: ***Clearness of visualization***

Question 4: Which version of RelFinder is easier to handle in your opinion?

(Welche Version von RelFinder ist Ihrer Meinung nach leichter handhabbar?)



CX **01-074** Figure 43: ***Handling of visualization***

The last question was about the pros and cons of each visualization. The points describe the most important statements as sort of summary.

Node-link view

The main points in the survey mentioned pros of the node-link view where the easiness of handling, the good design and the handling while cons where mainly the bad overview and the chaos.

Linear view

The mentioned linear view's pros where the good ascertain ability, the overview, the structure while the cons where the sorting and in a few cases the difficulties of the interpretation of relations.

Radial view

In the radial view the mentioned pros where mainly the good design while the cons where the bad overview and the little ascertain ability.

To provide the anonymity of the test subjects the exact wording cannot be alleged.

The general demands of the test subjects concerning the Linked Data
CX *01-075* visualizations can be explained in ***group able found nodes*** or to sort
them which was mentioned in terms of the linear view tied together
CX *01-076* with ***shorter search procedures and clear readability***. Another demand
CX *01-077* is a better overview and less chaos. ***Specific highlighting of single found***
CX *01-078* ***nodes is favored***. ***The orientation should be simple and usual***.

Conclusion and future work 9

The criteria of good design were utilized and compared with the test and the answers of the survey. In ***terms of handling, preference and clearness the linear view's design was evaluated as best designed*** view. This may be a reason of its ***minimalistic concept reducing the noise of unfiltered paths and nodes by repositioning***. CX *01-079* CX *01-080*

CX *01-081* The ***faster handling for a broader amount of users of the radial and the linear view can be hypothetically demonstrated concerning the time the tasks took*** in comparison. The nodes could be found very much faster than in the node-link view. Concerning the crowd of test subjects it must be announced that a proof of concept would demand a larger sample.

CX *01-082* ***The placement of the abstract as a toggle able container of the main view was often realized intuitively*** and needed no further explanation. In times of responsive design there is a vast amount of screen resolutions wherein concerning larger ones an attached toggle able information box is very helpful to organize nodes.

CX *01-083* Another point is that ***no statement with reference to the normalized predicates*** of the linear view is given which could imply that this feature was not knowingly realized. One interesting point is that the users construed Linked Data in a very abstract way. Most of the time they had difficulties to express the sentence out of the visualization. In terms of the translation of the information into another language some predicates familiar with Linked Data are misunderstood. Another review was the problem of redirected predicates called WikiPage redirects. Very few information was found about it in the net. The difficulties are that the web application delivers much more edited Linked Data triples than the compiled prototype using the limit of the used endpoint so that a test could not have been managed because of the large amount of relations and these redirected nodes that give no clue about the relation because the predicate labels were bundled and overwritten.

CX 01-084 The ***unordered loading of the found nodes is also criticized*** by some test subjects. Especially in the linear and the radial view the test subjects suggested that the nodes have to be ordered alphabetically.

CX 01-085 The visualization of ***nodes in the radial view should rather be turned 180° till the middle of the view so that readability is better***. As reminder it has to be mentioned that the performance of findability is best but the visualization space for nodes positioned in a radial view is limited.

CX 01-086 Nevertheless a ***graph oriented visualization of data presented as list seems to have synergies to the presentation of normal lists*** concerning the habits and the usability of user demands.

Another point already done is the finalization of the radial view by adding graph directions represented as arrows. All the data of the developed prototypes and the most important information is hosted on a github repository and can be found easily on

https://github.com/geonb/Visualization-of-typed-links-in-Linked-Data

(10.02.2016, 10:00 am)

Therefore, concerning the Semantic Web in general, it must be depic-
CX 01-087 ted that the ***grammar of relative clauses, visualized as oriented graphs cannot be used exclusively in dependence to a certain vocabularies that***
CX 01-088 ***represent a sort of equal exchange of information***. A group of ***specific vocabularies*** like communication, meet, talked, also ***could be linked without including the direction of the graphs***.

To delineate future work on the prototypes of RelFinder it is necessary to follow the rules of visual representations respectively the results of
CX 01-089 this evaluation. Reasons for a better design where seen in the ***hiding of unnecessary information and in a better positioning of the nodes***. To illustrate in short what could be done better concerning the linear and the
CX 01-090 radial view it can be said, that the found nodes (objects) of a ***normalized list of predicates should get grouped around similar normalized predicates and could appear in one centered alignment to a clicked found node***. As main topic the distance of the node lists between the nodes can be further investigated as optimizable graph lengths.

One sentence as a developer is, that in difference, all the modern visualizations of Linked Data in connection to a model view controller concept
CX 01-091 should ***prepare the result set of a query completely in the controller*** and draw the entire visualization after this calculation.

Finally, to preserve the semantic sense of relationships of typed links I
CX 01-092 suggest a presentation where the ***composition of all relations including subjects and normalized predicates in dependence on sortable and filterable object groups represent a result set as vertical positioned linear view***.

Table of figures and listings

Literature

[01] J. H. Tim Berners-Lee and O. Lassila, *"The Semantic Web"*, Scientific American, vol. 284, no. 5, pp. 96–101, 2001.

https://www.scientificamerican.com/article/the-semantic-web/

[02] *"Joint Publication 2-0, Joint Intelligence"* (PDF). Defense Technical Information Center (DTIC). Department of Defense. 22 June 2007. pp. GL–11. February, 2013.

http://www.dtic.mil/doctrine/new_pubs/jp2_0.pdf

[03] Akash Mitra, "*Classifying data for successful modeling*", 2011

http://www.dwbiconcepts.com/data-warehousing/12-data-modelling/101-classifying-data-for-successful-modeling.html

[04] N. Bartelme, *"Einführung in Geoinformatik"*. Springer Berlin Heidelberg, 2005, pp. 1–42.

http://dx.doi.org/10.1007/3-540-27201-1_1

[05] 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

[06] 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

[07] T. Gruber, *"Toward principles for the design of ontologies used for knowledge sharing"*, International journal of human computer studies, pp. 907–928, 1993.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.5775&rep=rep1&type=pdf

[08] 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.

[09] A. Dengel, *"Das Resource Description Framework"*, Semantische Technologien, pp. 109–129, 2012.

[10] T. Berners-lee, *"Uniform Resource Locators Current practice"*, www.w3.com, Tech. Rep. July, 1994.

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

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

[13] Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak, *"Linking open data cloud diagram"*, 2014.

http://lod-cloud.net/

[14] Robin Cover, *"XML and Web Services In The News"*, xml.org. 6 October 2006.

http://www.xml.org/xml/news/archives/archive.10062006.shtml

[15] Dubost Karl, *"Web site meta data profile: favicon"*, World Wide Web Consortium. October 2005

http://www.w3.org/2005/10/profile

[16] 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

[17] 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

[18] Michael Friendly, *"Milestones in the history of thematic cartography, statistical graphics, and data visualization"*, 2008

http://www.math.yorku.ca/SCS/Gallery/milestone/milestone.pdf

[19] 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

[20] B. Preim and R. Dachselt, *"Interaktive Systeme"*, ser. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

http://www.springerlink.com/index/10.1007/978-3-642-05402-0

[21] 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

[22] C. und Bosch, S. Schiel, and T. Winder, *"Der Prozess der visuellen Wahrnehmung, in Emotionen im Marketing"*, 2006, pp. 361–429.

[23] visual-literacy.org: Business

http://demo.elearninglab.org/mod/page/view.php?id=53

[24] 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

[25] B. Shneiderman, *"The eyes have it: a task by data type taxonomy for information visualizations"*, Proceedings 1996 IEEE Symposium on Visual Languages, pp. 336–343, 1996.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=545307

http://www.cs.umd.edu/~ben/papers/Shneiderman1996eyes.pdf

[26] D. A. Keim and I. C. Society, *"Information Visualization and Visual Data Mining"*, IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 1, pp. 1–8, 2002.

http://webpages.uncc.edu/krs/courses/6010/infovis/pubs/misc/keim-vdm.pdf

[27] G. Scott Owen, *Model of Visualization*, 1999

https://www.siggraph.org/education/materials/HyperVis/abs_con1/model.htm

[28] W. L. Hibbard, *"Interactive visualization of Earth and space science computations", Computer (Volume: 27, Issue: 7), July 1994*

http://ftp.cs.wisc.edu/computer-vision/repository/PDF/hibbard.1994.computer.pdf

[29] Lloyd A. Treinish, *"A Function-Based Data Model for Visualization"*, IBM Thomas J. Watson Research Center Yorktown Heights, NY

http://www.research.ibm.com/people/l/lloydt/dm/function/dm_fn.htm

[30] D. J. Duke, K. W. Brodlie, D. A. Duce, and I. Herman, *"Do you see what I mean?"*, IEEE Computer Graphics and Applications, vol. 25, no. 3, pp. 6–9, 2005.

[31] Haber, R. B. and McNabb, *D. A., "Visualization idioms: A conceptual model for scientific visualization systems"*, In Visualization in Scientific Computing, pages 74–93. IEEE Computer Society Press. 1990.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1438250

[32] Friedrich Schmidt, *"Datenvisualisierung zur Unterstützung bei der Linux-Serveradministration in Hinblick auf Informationssicherheit"*, Institut für Informatik der Ludwig-Maximilians-Universität München, 2015

http://www.nm.ifi.lmu.de/pub/Diplomarbeiten/schm15/PDF-Version/schm15.pdf

[33] Chi, Ed H., *"A taxonomy of visualization techniques using the data state reference model".* Information Visualization, 2000. InfoVis 2000. IEEE Symposium on. IEEE, 2000

http://www-users.cs.umn.edu/~echi/papers/infovis00/Chi-TaxonomyVisualization.pdf

[34] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *"Readings in Information Visualization: Using Vision to Think"*, Academic Press, London, 1999.

http://www.luizrodrigues.com/artigos-tcc/Information%20Visualization/Readings%20in%20Information%20Visualization,%20Using%20vision%20to%20think.pdf

[35] Leland Wilkinson, *"The Grammar of Graphics Statistics and Computing"*, Springer, 1999

[36] Leland Wilkinson, *"The Grammar of Graphics".* Springer Science & Business Media, 2006

[37] Jan and Martin Voigt, *"VISO: A Shared, Formal Knowledge Base as a Foundation for Semi-automatic InfoVis Systems"*, 2013

http://www-st.inf.tu-dresden.de/semvis/papers/authors_version_polowinski_viso_chi2013_wip.pdf

[38] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, Ioannis G. Tollis, *"Algorithms for Drawing Graphs: an Annotated Bibliography"*, 1994

http://www.graphdrawing.org/literature/gdbiblio.pdf

[39] Joshua O'Madadhain, Danyel Fisher, Padhraic Smyth, Scott White, Yan-Biao Boey, *"Analysis and Visualization of Network Data using JUNG"*, 2005

http://jung.sourceforge.net/doc/JUNG_journal.pdf

[40] Chris Bennett, Jody Ryall, Leo Spalteholz and Amy Gooch, *"The Aesthetics of Graph Visualization"*, Visualization, and Imaging (2007)

[41] Kobourov, Stephen G., *"Spring Embedders and Force-Directed Graph Drawing Algorithms"*, 2012

[42] W. T. Tutte, *"How to draw a graph"*, Proceedings of the London Mathematical Society 13 (52): 743–768,), 1962

[43] 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

[44] 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.

[45] 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.

[46] 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 , pages 158{171. Springer, 2006.

[47] 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.

[48] 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

[49] 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, 1-5.

DOI=http://dx.doi.org/10.1145/1168149.1168168

Appendix I

Implementation of RelFinder <

A version of RelFinder concerned for further developing can be exported to github where it can be downloaded as .zip Archive.

http://code.google.com/p/relfinder

(12.08.2015; 01:49 pm)

Flash projects base on the actionscript3 scripting language. The usual way to implement flash projects is the Adobe Flash builder.

http://www.adobe.com/de/products/flash-builder-family.html

(12.08.2015; 01:49 pm)

For open source developing there can be used FlashDevelop

http://www.flashdevelop.org

(12.08.2015; 01:49 pm)

For the adaptation of RelFinder I chose FlashDevelop 5.0.2. and the flex SDK 3.5.

http://download.macromedia.com/pub/flex/sdk/flex_sdk_3.5.zip

(12.08.2015; 01:49 pm)

Setup FlashDevelop and Flash

FlashDevelop is written in a 32 bit version and requires java.

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

(12.08.2015; 01:49 pm)

After accepting the Oracle Binary Code License Agreement for Java SE you can download the Windows x86 - jdk-8u65-windows-i586.exe. To change the Active-X settings in the Internet Explorer. Take a look at

http://windows.microsoft.com/de-AT/windows/help/genuine/ie-activex

(12.08.2015; 01:49 pm)

In FlashDevelop it is also possible to debug the code by tracing using a debug version of flash.

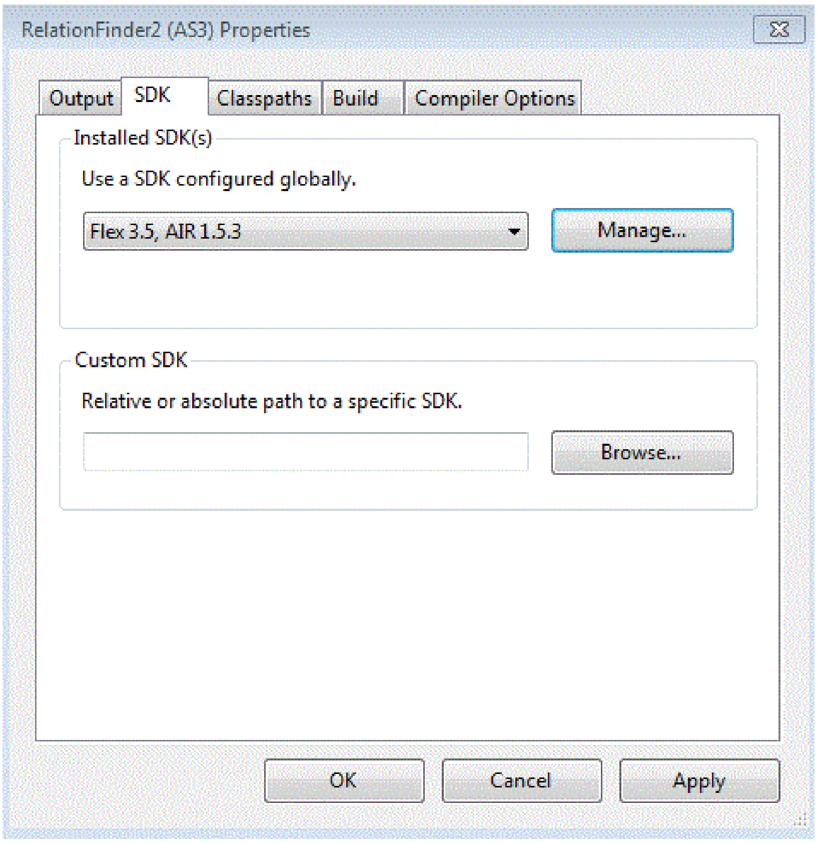
https://www.adobe.com/support/flashplayer/debug_downloads.html

(12.08.2015; 01:49 pm)

Setup Flex SDK 3.5

To setup Flex on FlashDevelop the following steps are taken.

Choose ″Properties“ in the ″Project“ menu, click the ″SDK“-tab and the Button ″Manage“ under ″Installed SDK(s)“.



RelationFinder2 (AS3) Properties

Output | SDK | Classpaths | Build | Compiler Options

Installed SDK(s)

Use a SDK configured globally.

Flex 3.5, AIR 1.5.3 — Manage...

Custom SDK

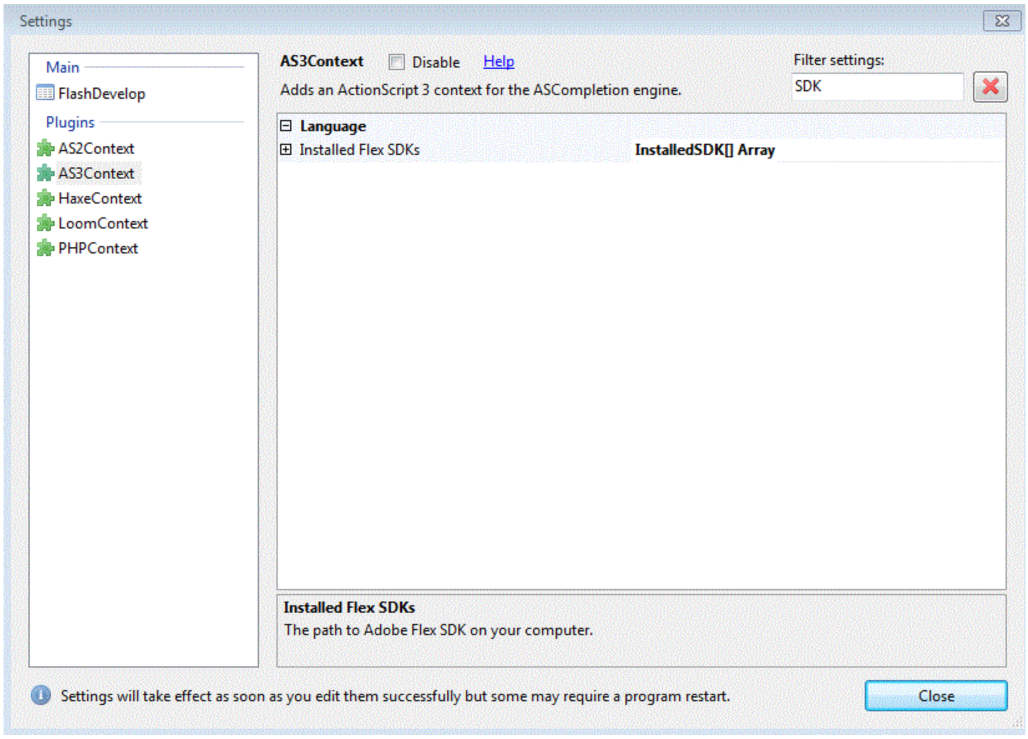
Relative or absolute path to a specific SDK.

Browse...

OK | Cancel | Apply

Figure 01: Setup Flex 01

To include the Flex SDK there is an entry named ”Installed Flex SDKs“ for AS3Context. The path to the Flex SDK can be typed manually or browsed with the explorer.



Settings

Main
- FlashDevelop

Plugins
- AS2Context
- AS3Context
- HaxeContext
- LoomContext
- PHPContext

AS3Context ☐ Disable Help

Adds an ActionScript 3 context for the ASCompletion engine.

Filter settings: SDK

⊟ Language

⊞ Installed Flex SDKs	InstalledSDK[] Array

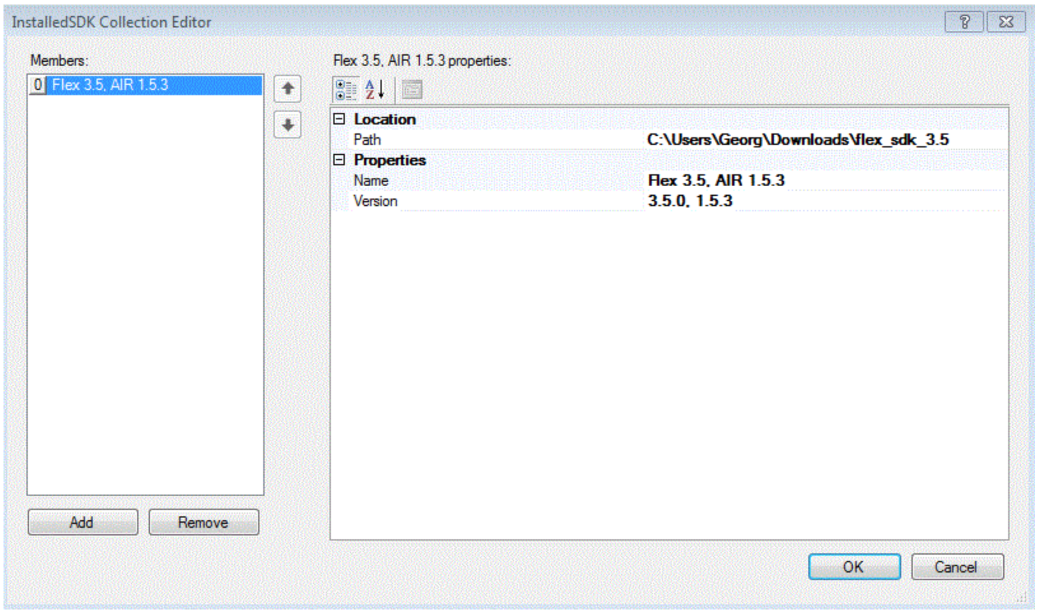
Installed Flex SDKs

The path to Adobe Flex SDK on your computer.

Settings will take effect as soon as you edit them successfully but some may require a program restart.

Close

Figure 02: Setup Flex 02


InstalledSDK Collection Editor

Members:

0 Flex 3.5, AIR 1.5.3

Flex 3.5, AIR 1.5.3 properties:

Location	
Path	C:\Users\Georg\Downloads\flex_sdk_3.5
Properties	
Name	Flex 3.5, AIR 1.5.3
Version	3.5.0, 1.5.3

Figure 03: Setup Flex 03

In the "Output"-tab it is possible to decide the Flash Player's version and optional ways of testing a project.

To test projects you can compile a ".swf"-flash file or test it in FlashDevelop.

Adaptation of the source code

A program written in actionscript 3 is very familiar to a one written in Java. To bring out the main difference at the beginning variables have following notation:

- *Accessor:* <private, protected, public>
- *Variable declaration:* <var>
- *Variable name:* <name>
- *Variable type:* <:Object, etc.>

Actionscript 3 enables object oriented scripting.

Object-oriented programing

http://help.adobe.com/en_US/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e-3d118a9b90204-7f35.html

(12.11.2015; 02:32 pm)

http://link.springer.com/chapter/10.1007/978-1-4842-0671-3_1

(12.11.2015; 02:36 pm)

Methods

Concerning methods, the return type has to be defined after the parameter list after the closing right parenthesis. The return type in actionsctipt 3 functions must be defined on the bottom of the method with the return statement. Each return statement defined in a control structure before the bottom of the method overwrites the last return value. If there is no matching case in the control structure the statement at the bottom is returned in default.

If the return type is void, a return statement must not be defined and for example an array in the namespace is manipulated or other functionalities called.

Namespace

Each variable has to be part of a namespace, which can be described as a scope in the program, the variable can be accessed or manipulated.

http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/Namespace.html

(12.11.2015; 02:32 pm)

To access variables form outside a namespace preserving the security of a program, methods can be defined encapsulating private variables as accessors to make them visible in other namespaces. This can be done for simple variables as get-or set-Method or concerning classes by inheritance of other classes or interfaces. Actionscript 3 supports single inheritance of classes and muliple inheritance of interfaces. Each variables and methods defined in a so called superclass get part of the namespace of the module of classes.

MXML

http://help.adobe.com/en_US/flex/using/flex_4.6_help.pdf

(12.08.2015; 01:49 pm)

MXML is an XML language to lay out user interface components for applications built in Adobe Flex. The visualization of RelFinder is based on MXML components which build the view of the Model View Controller (MVC) concept.

Changing visualization

<

RelFinder is a scripted object-oriented program featuring the model view controller concept.

http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1875/pdf/thesis.pdf

(12.08.2015; 01:49 pm)

The code consists of three main following parts.

- *Connection:*	The configuration for the Linked Data endpoints handling requests
- *Graph Elements:*	Definition of the components used to visualize the query results
- *Main Application:*	Main view of the application and ordered function calls of the workflow

To change the visualization the classes of the graph elements had to be adapted predominantly. A few codes had to be adjusted in the layout elements depending on the positioning of the elements. The input of the user is handled by an element called given nodes from where found relations can be integrated as class found nodes building a relation node. Each node has its own node view element written in MXML.

Normalizing predicates

The graph model is the controller class of the graph elements. To normalize the predicates two methods had to be implemented as follows.

```
private function trimHashSign(uri:String):String {
            return uri.substr(uri.lastIndexOf("#") + 1)
}
private function func_check(arr:Array, rel:Relation) : Relation
{
            var i:int;
            var temp:Relation = null;
            for (i = arr.length - 1; i >= 0; i--) {
        if(trimHashSign(unescapeMultiByte(rel.predicate.rdfLabel).split("_").join(" ")) ===
        trimHashSign(unescapeMultiByte(arr[i].predicate.rdfLabel).split("_").join(" "))) {
                rel.id = arr[i].id;}
            }
            return rel;
}
```

Listing 01: Normalization of predicates

The first method replaces the hash sign of each RDFLabel in a predicate. The RDFLabel is a variable of RelFinder that contains the string representation of the derived relation URI of a SPARQL query.

This RDFLabels were stored in an additional array of relations and could therefore be compared on equality. If the predicate of the relation equals one predicate of the array of relations the ID of the relation of the array gets mapped to the relation ID. This provides, that one unique predicate representation for all relations within the same predicate is shown in the view. The reason of that is, that similar IDs are plotted only once in RelFinder. That is already the case for object and subject labels. To map the whole relation ID then causes the normalization of the predicates in addition. Direct IDs of predicates where only readable, so they could not be mapped. Otherwise the predicates could have been exchanged by not preserving the data triple as it was written. The list of predicates include then all appearing predicates in a result set but only once.

The return type of the method is passed to the method drawRelations which is called by the method drawPath initialized in the class path.

Each time, a relation is found or the elements of the view are clicked the method drawPath is called. This caused a few adjustments in the source code for making the adapted visualization stable.

Toggle abstracts in the relation view

The elements given node view and found node view were extended with the method toggle which provides an additional view of the abstracts of the subjects and objects of a query result the Linked Data (data base) applicates.

```
<mx:Button id="arrow" styleName="arrowBtn" click="{toggle(0)}" width="20" height="20"/>
```

Listing 02: Mxml entry toggle for click event

```
public function toggle(tog:int):void {
        if (data.element.abstract != "no abstract available") {
                var limiter:String = data.element.abstract;
                limiter.split("&ndash;").join("&#8211;");
                if (mod == 0 && tog != 1) {
                        this.parent.setChildIndex(this, this.parent.numChildren - 1);
                        data.pin();
                        this.width = 320;
                        this.height = imagea.height + texta.height + 30;
                        mod = 1;
                } else {
                        this.parent.setChildIndex(this, this.parent.numChildren - 2);
                        data.pin();
                        this.width = 320;
                        this.height = 25;
                        mod = 0;
                }
        }
}
```

Listing 03: Toggle function

```
<mx:VBox id=”infoa” backgroundAlpha=”1.0” top=”32” paddingBottom=”2” padding-
Top=”2” paddingRight=”2” paddingLeft=”2” width=”100%” height=”100%” >

<local:ImageView id=”imagea” width=”150” height=”150” image_path=”{data.element.
imageURL}” />

<mx:TextArea  id=”texta” wordWrap=”true” editable=”false” borderStyle=”none” fon-
tSize=”13” top=”32” height=”460”  width=”260” visible=”{data.element.abstract != ‘no ab-
stract available’}” text=”{data.element.abstract}”  verticalScrollPolicy=”off” horizontalScroll-
Policy=”off” />

</mx:VBox>
```

Listing 04: Mxml entry

By a click on the button a switch variable for each node handles the toggling of the tex tfield including an image given and an abstract including a description of the subject or object. This is realized by scaling the canvas in two resolutions. One hiding and the other one showing the information.

To avoid interlaces of these enlarged canvas elements the ChildIndex is set on top each time it is clicked so that the top (last clicked) node is readable. While the width stays the same, the height is enlarged to a specific height by using the information in the data containing tags. If an abstract contains overlength it can be scrolled by the mouse wheel or with the vertical scrollbar.

The code reference shows the used tag elements for the information abstracts. The abstracts primarily could be merely discovered in an info box in Relfinder. The main benefit of getting the abstracts into the relation view is to be able to open each of them in the main view. The html representation in the info box either was changed to a text field representation.

Highlighting the paths

```
public function set isHighlighted(b:Boolean):void {
        if (b != _isHighlighted) {
                _isHighlighted = b;
                dispatchEvent(new Event(Path.HCHANGE));
        if (_isHighlighted) {
                _layout.settings = { alpha: 1, color: 0xD2001E /*0xFF0000*/, thickness: 1 };
                if(_isVisible) {
                        Graphmodel.getInstance().drawPath(this, true);}
                } else {
         _layout.settings = {  alpha: 0, color: 0x222222, thickness: 1 };
                if(_isVisible) {
                         Graphmodel.getInstance().drawPath(this, true);}
                }
        }
}
```

Listing 05: isHighlighted function

```
private function selectedPath():void {
        clearPaths();
        for each(var r:Relation in data.element.relations) {
                for each(var p:Path in r.paths) {
                        p.isHighlighted = true;
                }
        }
}
```

Listing 06: selectedPath function

The normalization of the predicates caused a loss of the highlighted paths of a clicked node. Therefore this had to be fixed by reimplementation. The graph model had to be accessible in the given node view to clear previously clicked node paths when changing the node's focus by clicking.

Highlighting is then set to false for each path.

The basic relation consisting of the given node inputs include the full path. The path of one relation highlighted by a click on the found nodes is always leading to one of the given nodes. So the information is provided but the readability is concerned.

In the class path the Boolean flag for highlighting is evaluated. To realize a better readability, only the paths of one relation depending on clicked nodes are visible while the alpha value of all other paths is set to 0. This has also to be applied in the initial call of the drawPath method.

Positioning of nodes

Force-directed layout is the class that is responsible for the preset of coordinate positions of each node. Initially, in RelFinder, the nodes could be moved tangible to the web of nodes or pinned to one position. For the adaptation of the view the movement of nodes was applied as restricted to the y-axis by spacing them from the center view on a specific x-value to enable a sort of list view.

```
private function addRelationToGraph(subjectNode:MyNode, predicateNode:RelationNode,
objectNode:MyNode, layout:Object):void {

            objectNode.setPosition(850, 10  - (ita*35) + itx * 35);

            subjectNode.setPosition(850, 10 - (ita*35) + itx* 35);

            predicateNode.setPosition(400, 10 - (ity*15) + itb * 28);

            var object1:Object = new Object();

            object1.startId = subjectNode.id;    //defines the direction of the link!

            if (layout != null) object1.settings = layout.settings;

            _completeGraph.link(subjectNode, predicateNode, object1);

            var object2:Object = new Object();

            object2.startId = predicateNode.id;

            if (layout != null) object2.settings = layout.settings;

            _completeGraph.link(predicateNode, objectNode, object2);

}
```

Listing 07: Position of node coordinates linear view

From the center point each node has y-spacing in the positive direction.

The main view can be scrolled by a vertical scrollbar.

To get a vertical balance d list with similar spaces, it was necessary to implement new indices for the positioning of the given nodes, relation nodes and found nodes in the graph model. The iterators are managed as global variables and calculated for each node type separately and then used in the drawRelation method to position the nodes correctly as a linear list.

Listing 08 shows the main function of positioning in the radial view. The nodes are rotated along a circle.

```
public function setup():void {

        addListeners();

        this.rotationZ = Math.atan2(this.y - Graphmodel.getInstance().centerY, this.x - Graphmodel.getInstance().centerX) * (180 / Math.PI);

        this.parent.setChildIndex(this, 1);

        elementChangeHandler(null);

        mod = 0;

}
```

Listing 08: Position of node coordinates radial view

Therefore, the Graph model had to be instantiated in the view to be able to rotate the nodes subsequently.

Appendix

First test setup

The following test setup described below, was the first setup designed. After doing several pretests with it, it appeared that related nodes to the given ones where deleted as entries from the DBPedia so that a second test setup had to be constructed to continue. Testing dynamic web data can always cause such problems during tests and often are handled by local snapshots of the web. The description shows the tasks and the solutions as screenshots.

Task setup 1: Schiller Friedrich and Johann Wolfgang von Goethe

Task 1: Search and click the node Weimar Classicism!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Johann Gottfried Herder! read in the abstract, when Johann Gottfried Herder was born

Node-Link View

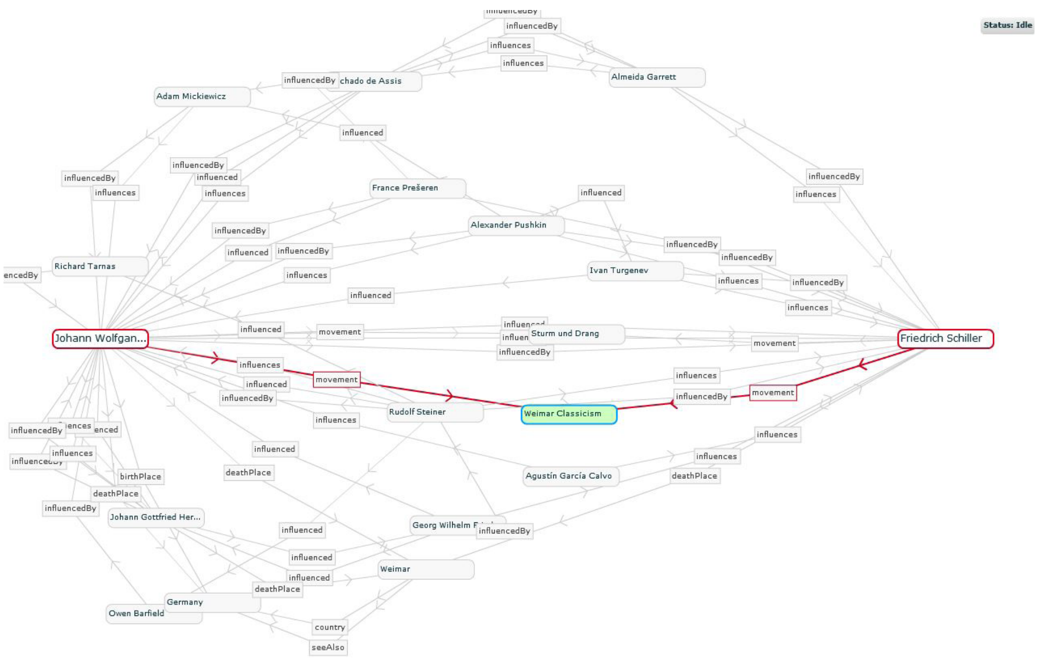
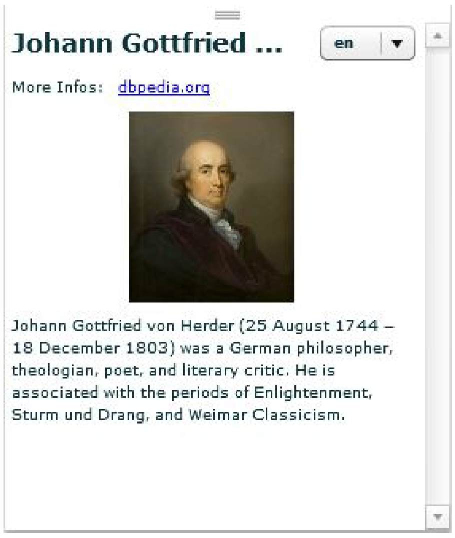


Figure 01: Node-link view Weimar Classicism



Johann Gottfried ...

More Infos: dbpedia.org

Johann Gottfried von Herder (25 August 1744 – 18 December 1803) was a German philosopher, theologian, poet, and literary critic. He is associated with the periods of Enlightenment, Sturm und Drang, and Weimar Classicism.

Figure 02: Abstract Johann Gottfried von Herder (node-link view)

Task setup 2: Sigmund Freud and Carl Jung

Task 1: Search and click the node Psychotherapy!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Friedrich Nietzsche! Read in the abstract, when Friedrich Nietzsche was born!

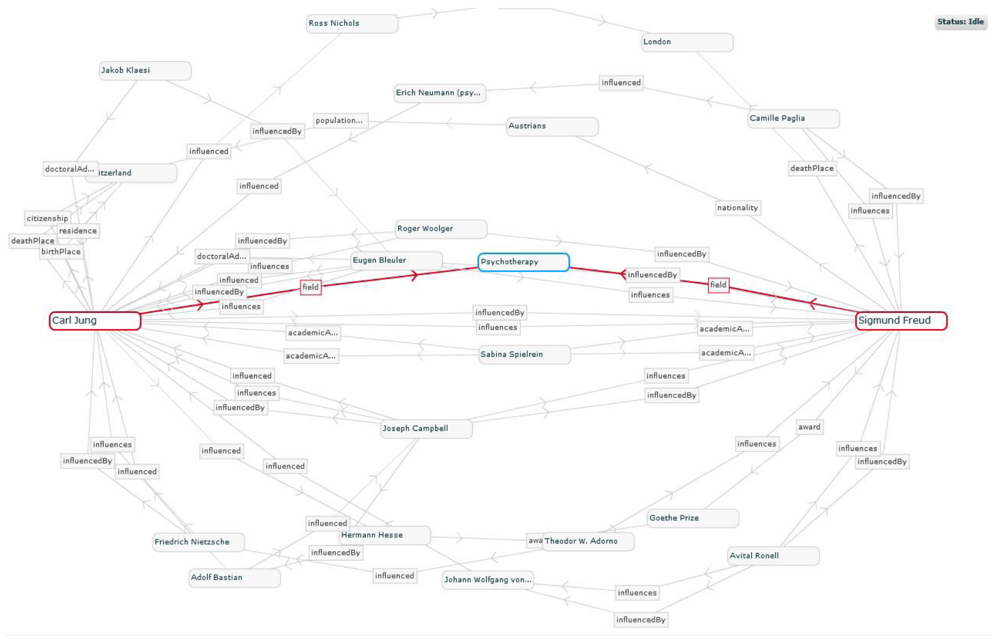
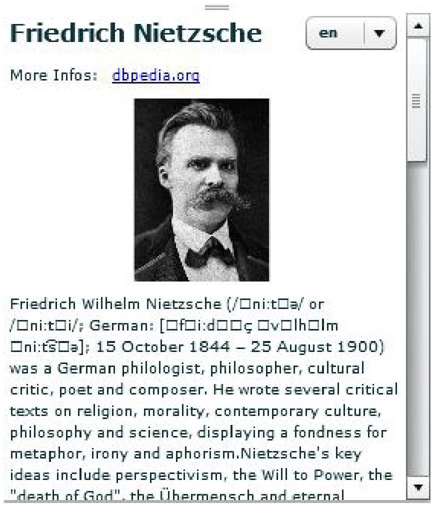


Figure 03: Node-link view Psychotherapy



Friedrich Nietzsche

en

More Infos: dbpedia.org

Friedrich Wilhelm Nietzsche (/□ni:t□ə/ or /□ni:t□i/; German: [□f□i:d□□ç □v□lh□lm □ni:ts͡□ə]; 15 October 1844 – 25 August 1900) was a German philologist, philosopher, cultural critic, poet and composer. He wrote several critical texts on religion, morality, contemporary culture, philosophy and science, displaying a fondness for metaphor, irony and aphorism.Nietzsche's key ideas include perspectivism, the Will to Power, the "death of God", the Übermensch and eternal

Figure 04: Abstract Friedrich Nietzsche (node-link view)

Task setup 3: Friedrich Nietzsche and Immanuel Kant

Task 1: Search and click the node Epicurus!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Bruno Bauer! Read in the abstract, when Bruno Bauer was born!

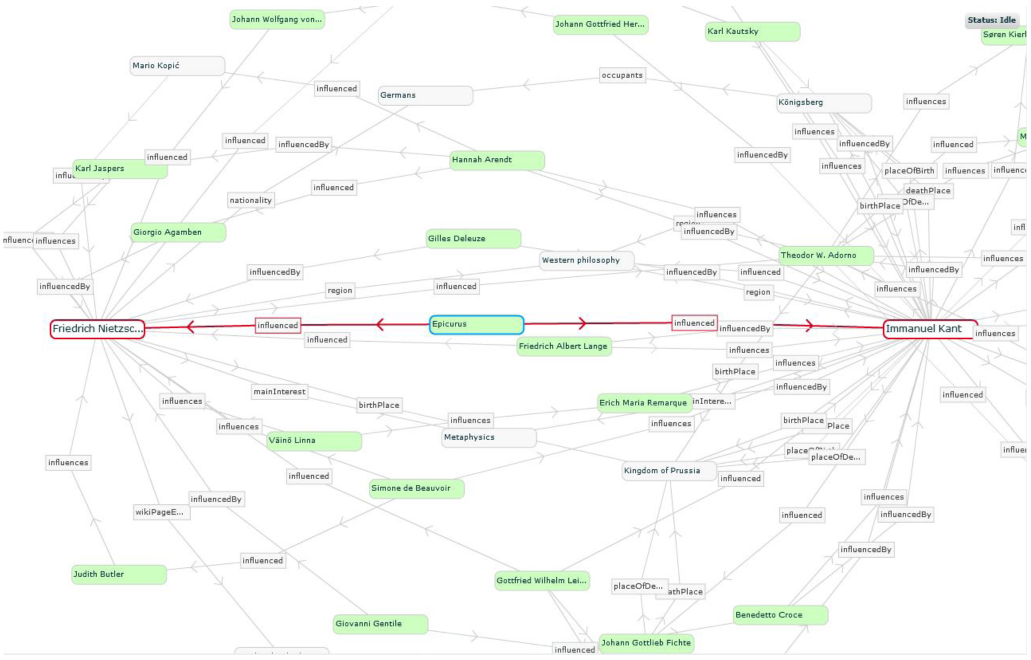
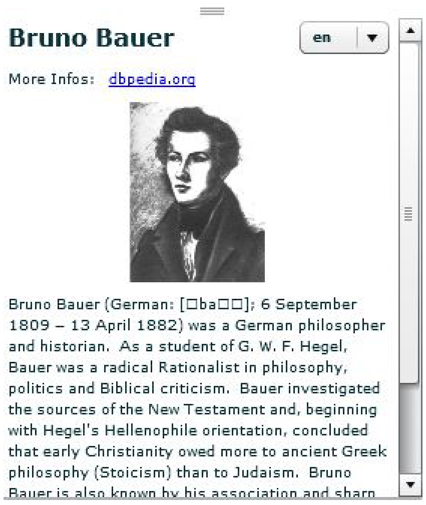


Figure 05: Node-link view Epicurus



Bruno Bauer

en

More Infos: dbpedia.org

Bruno Bauer (German: [□ba□□]; 6 September 1809 – 13 April 1882) was a German philosopher and historian. As a student of G. W. F. Hegel, Bauer was a radical Rationalist in philosophy, politics and Biblical criticism. Bauer investigated the sources of the New Testament and, beginning with Hegel's Hellenophile orientation, concluded that early Christianity owed more to ancient Greek philosophy (Stoicism) than to Judaism. Bruno Bauer is also known by his association and sharp

Figure 06: Abstract Bruno Bauer (node-link view)

Linear view

Task setup 1: Schiller Friedrich and Johann Wolfgang von Goethe

Task 1: Search and click the node Weimar Classicism!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Johann Gottfried Herder! Read in the abstract, when Johann Gottfried Herder was born!

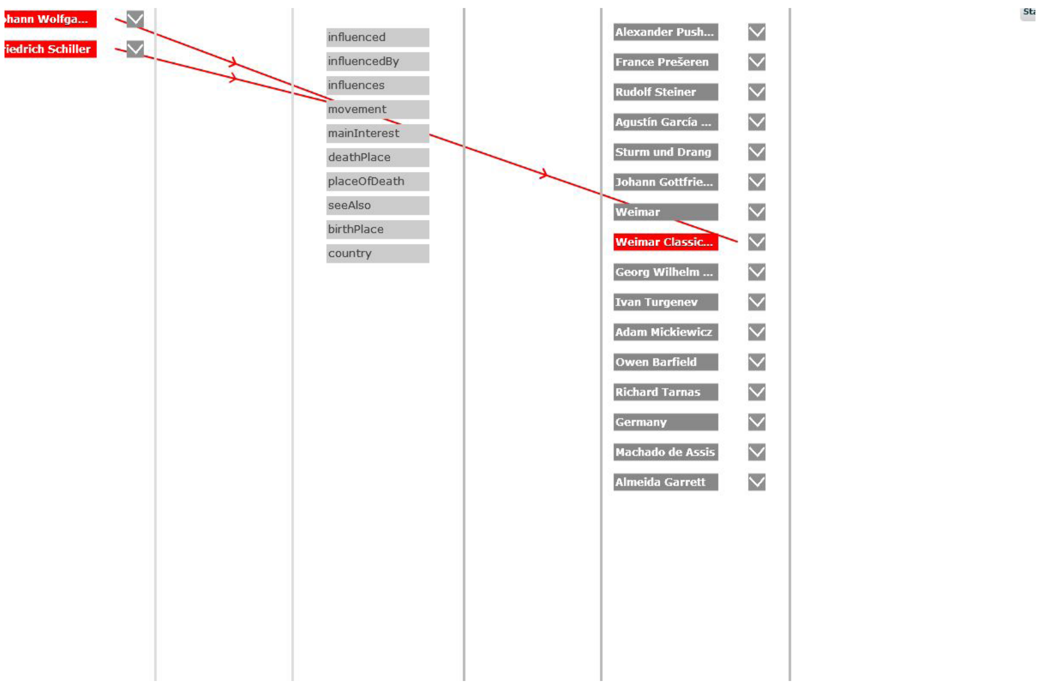
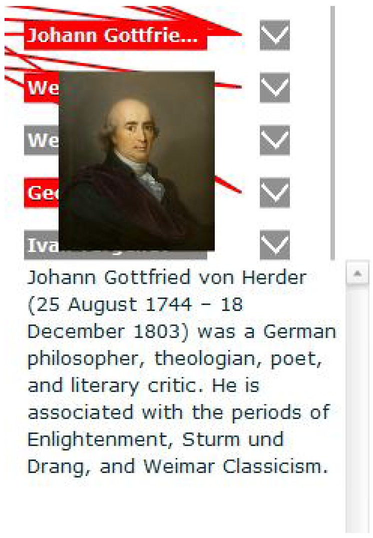


Figure 07: Linear view Weimar Classicism



Johann Gottfried von Herder (25 August 1744 – 18 December 1803) was a German philosopher, theologian, poet, and literary critic. He is associated with the periods of Enlightenment, Sturm und Drang, and Weimar Classicism.

Figure 08: Abstract Johann Gottfried von Herder (linear view)

Task setup 2: Sigmund Freud and Carl Jung

Task 1: Search and click the node Psychotherapy!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Friedrich Nietzsche! Read in the abstract, when Friedrich Nietzsche was born!

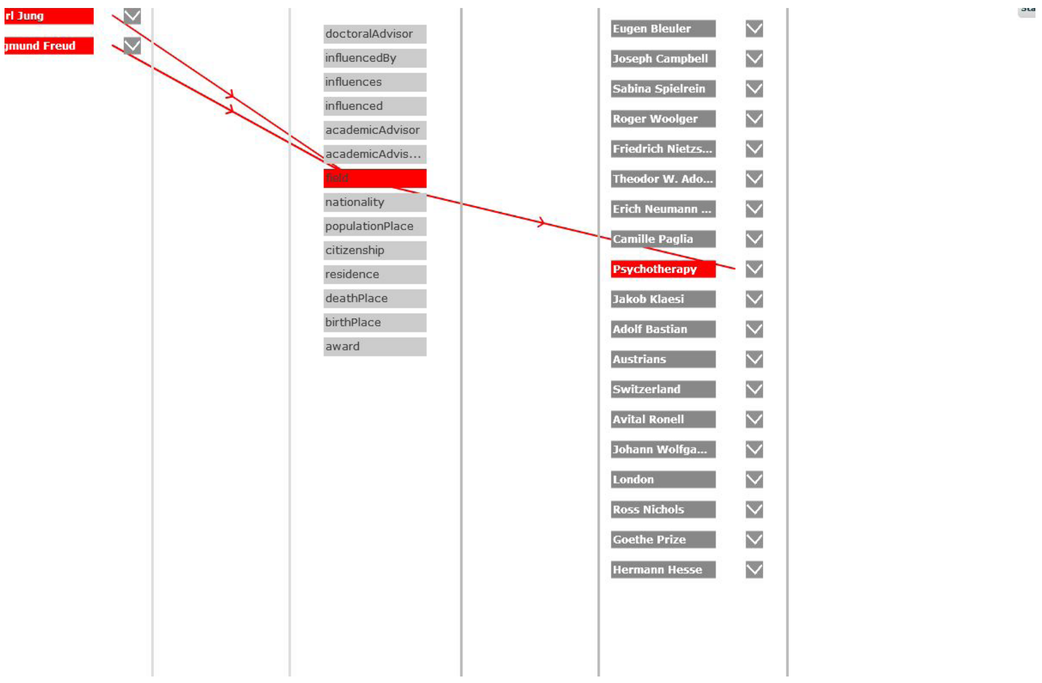
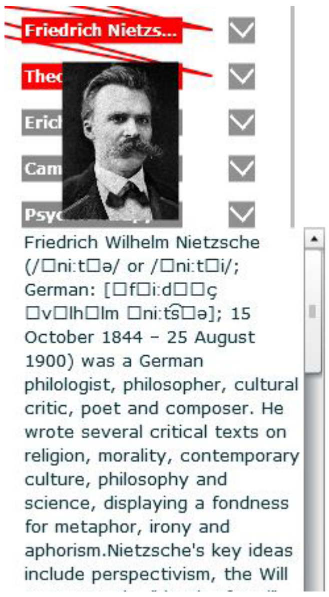


Figure 09: Linear view Psychotherapy



Friedrich Wilhelm Nietzsche (/□ni:t□ə/ or /□ni:t□i/; German: [□f□i:d□□ç □v□lh□lm □ni:ts͡□ə]; 15 October 1844 – 25 August 1900) was a German philologist, philosopher, cultural critic, poet and composer. He wrote several critical texts on religion, morality, contemporary culture, philosophy and science, displaying a fondness for metaphor, irony and aphorism.Nietzsche's key ideas include perspectivism, the Will

Figure 10: Abstract Friedrich Nietzsche (linear view)

Task setup 3: Friedrich Nietzsche and Immanuel Kant

Task 1: Search and click the node Epicurus!

Which predicate is given? Build a sentence including the sense of the information!

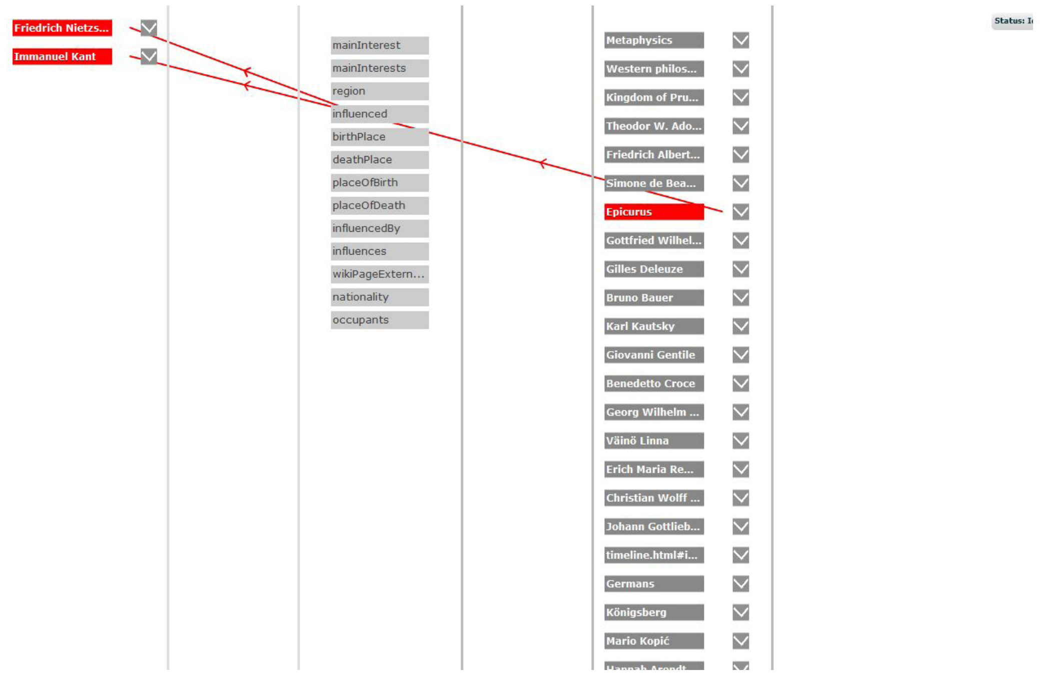
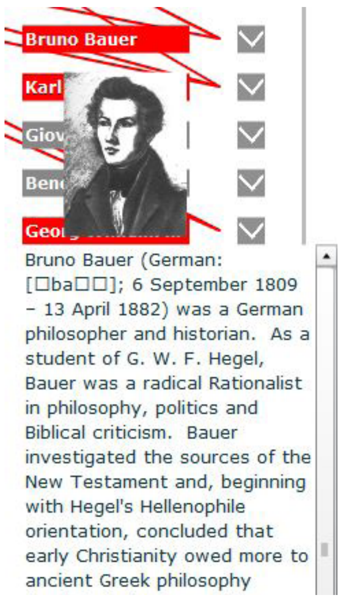


Figure 11: Linear view Epicurus

Task 2: Search and click the node Bruno Bauer! Read in the abstract, when Bruno Bauer was born!



Bruno Bauer (German: [□ba□□]; 6 September 1809 – 13 April 1882) was a German philosopher and historian. As a student of G. W. F. Hegel, Bauer was a radical Rationalist in philosophy, politics and Biblical criticism. Bauer investigated the sources of the New Testament and, beginning with Hegel's Hellenophile orientation, concluded that early Christianity owed more to ancient Greek philosophy

Figure 12: Abstract Bruno Bauer (linear view)

Radial view

Task setup 1: Schiller Friedrich and Johann Wolfgang von Goethe

Task 1: Search and click the node Weimar Classicism!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Johann Gottfried Herder! Read in the abstract, when Johann Gottfried Herder was born

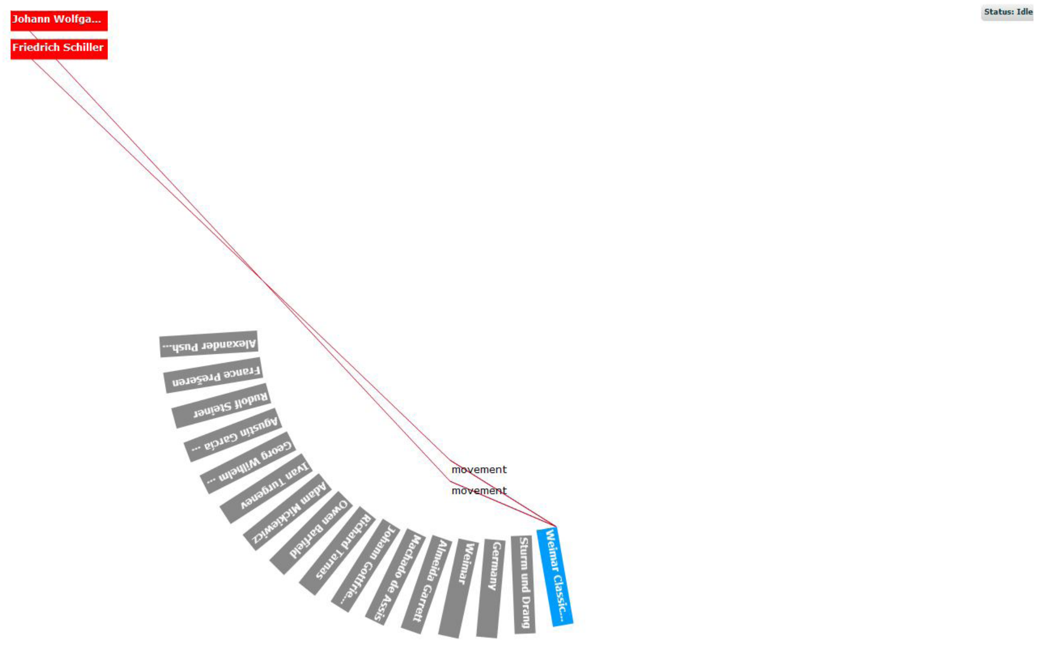
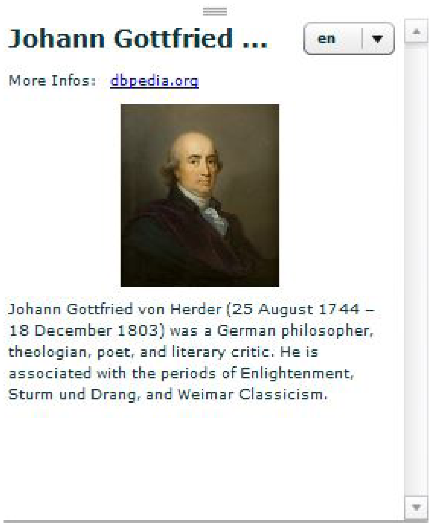


Figure 13: Radial view Weimar Classicism


Johann Gottfried ...

More Infos: dbpedia.org

Johann Gottfried von Herder (25 August 1744 – 18 December 1803) was a German philosopher, theologian, poet, and literary critic. He is associated with the periods of Enlightenment, Sturm und Drang, and Weimar Classicism.

Figure 14: Abstract Johann Gottfried von Herder (radial view)

Task setup 2: Sigmund Freud and Carl Jung

Task 1: Search and click the node Psychotherapy!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Friedrich Nietzsche! Read in the abstract, when Friedrich Nietzsche was born!

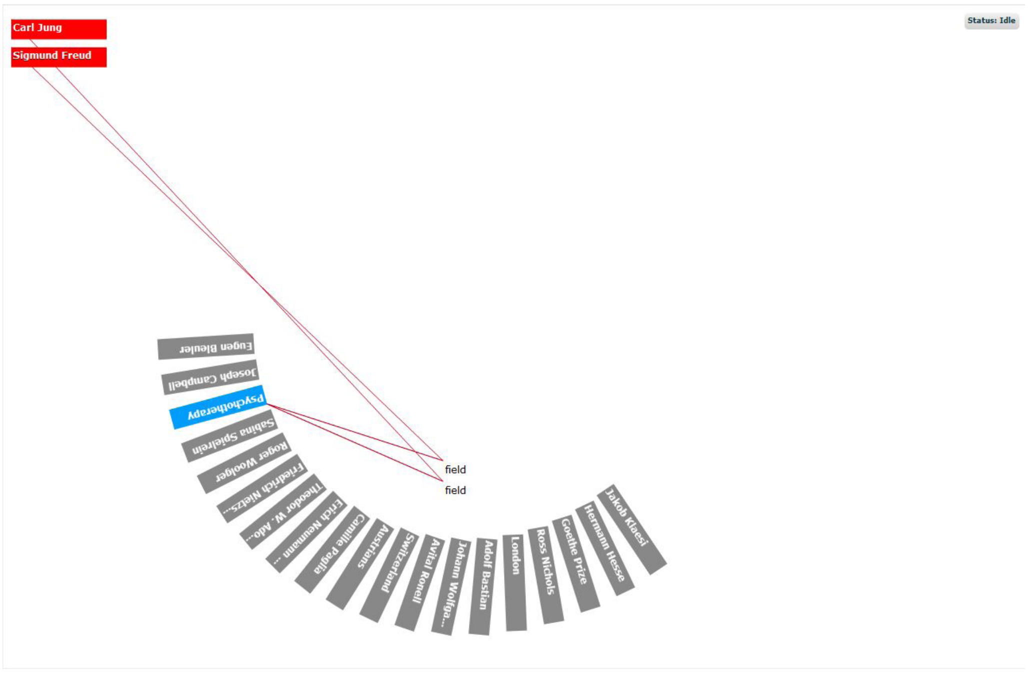
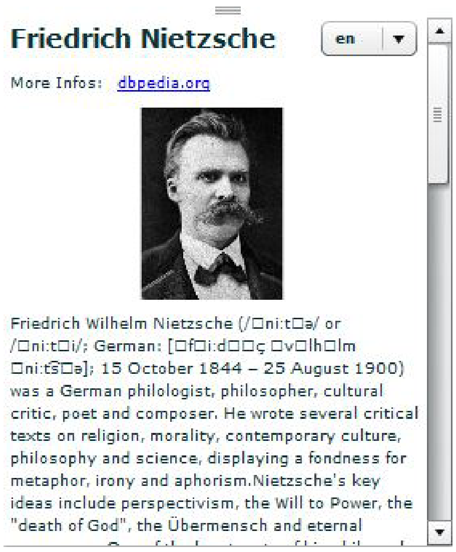


Figure 15: Radial view Psychotherapy



Friedrich Wilhelm Nietzsche (/□ni:t□ə/ or /□ni:t□i/; German: [□f□i:d□□ç □v□lh□lm □ni:ts͡□ə]; 15 October 1844 – 25 August 1900) was a German philologist, philosopher, cultural critic, poet and composer. He wrote several critical texts on religion, morality, contemporary culture, philosophy and science, displaying a fondness for metaphor, irony and aphorism.Nietzsche's key ideas include perspectivism, the Will to Power, the "death of God", the Übermensch and eternal

Figure 16: Abstract Friedrich Nietzsche (radial view)

Task setup 3: Friedrich Nietzsche and Immanuel Kant

Task 1: Search and click the node Epicurus!

Which predicate is given? Build a sentence including the sense of the information!

Task 2: Search and click the node Bruno Bauer! Read in the abstract, when Bruno Bauer was born!

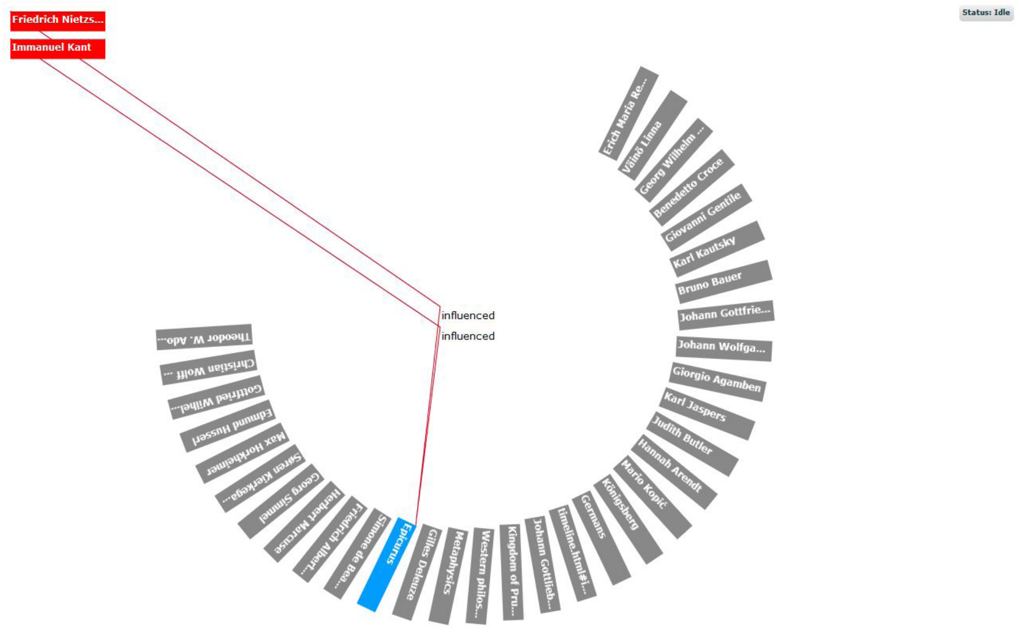
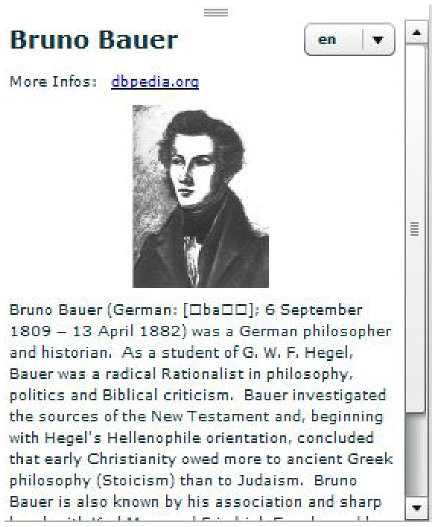


Figure 17: Radial view Epicurus



Bruno Bauer (German: [□ba□□]; 6 September 1809 – 13 April 1882) was a German philosopher and historian. As a student of G. W. F. Hegel, Bauer was a radical Rationalist in philosophy, politics and Biblical criticism. Bauer investigated the sources of the New Testament and, beginning with Hegel's Hellenophile orientation, concluded that early Christianity owed more to ancient Greek philosophy (Stoicism) than to Judaism. Bruno Bauer is also known by his association and sharp

Figure 18: Abstract Bruno Bauer (radial view)