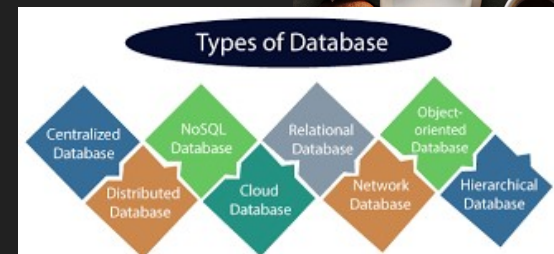


Semantic Exploration of Big Data

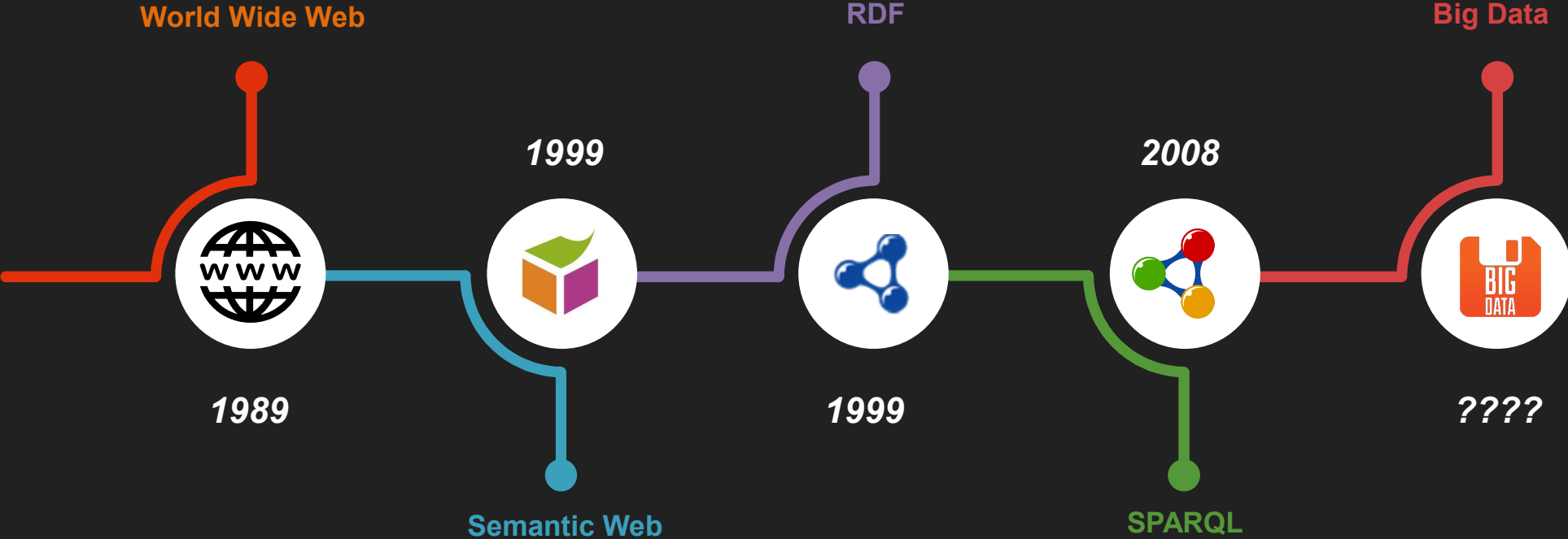
Maria Krommyda & Verena Kantere
2020 IEEE International Conference on Big Data

About me!

- Electrical & Computer Engineer, NTUA;
- PhD Candidate, Big Data Management & Visualization;
- Knowledge and Database Systems Laboratory;
- Software engineer, i-Sense group of ICCS.



Linked datasets



The V's of the Big Data

Volume

Velocity

Variety

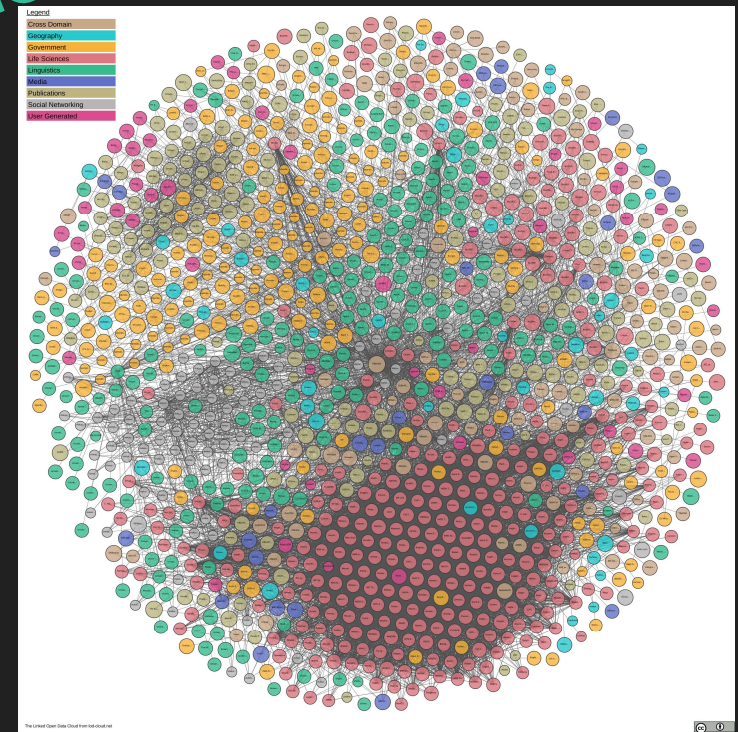
Veracity

Value

Semantic Exploration

- Thousands of datasets from many domains
- Users with limited or no experience are now interested in them
- What is available?
- How can I discover the information that answers my question?
- Why semantic exploration?

All the questions that we ask are based on semantics



Challenges

- Volume -> Too much information -> Dynamic;
- Velocity -> Datasets grow too fast -> Scalable;
- Variety -> Datasets with different characteristics -> Adaptable;
- Veracity -> Datasets with mistakes -> Resilient;
- Value -> Different use cases ->Flexible;

Semantic Exploration Techniques

- Semantic browsers;
- SPARQL endpoint visualization tools;
- Facet browsers;
- Query Writers;
- Schema Identifiers;
- Filtering-based exploration systems.

Semantic Exploration Techniques

- **Semantic browsers;**
- SPARQL endpoint visualization tools;
- Facet browsers;
- Query Writers;
- Schema Identifiers;
- Filtering-based exploration systems.

Semantic browsers

- Semantic browsers are adaptations of the Web Browsers to the Semantic Web;
- Can link different data as well as different data sources;
- Navigate the interconnected web of data;
- Follow the Semantic Web standard.

Semantic browsers

Pros

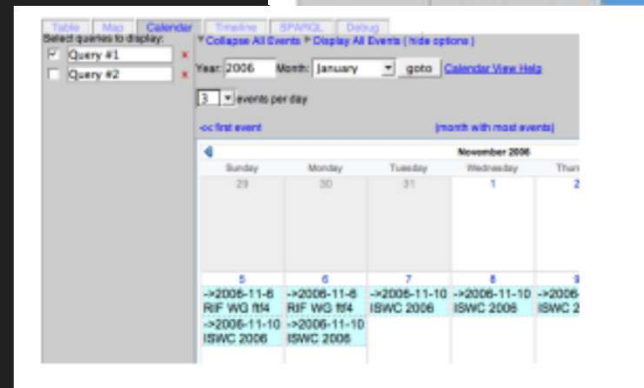
- Can link any data source;
- Provide interactive exploration;
- Most actions are intuitive, due to the experience of the user with the Web browsers;
- Semantic content is highlighted.

Cons

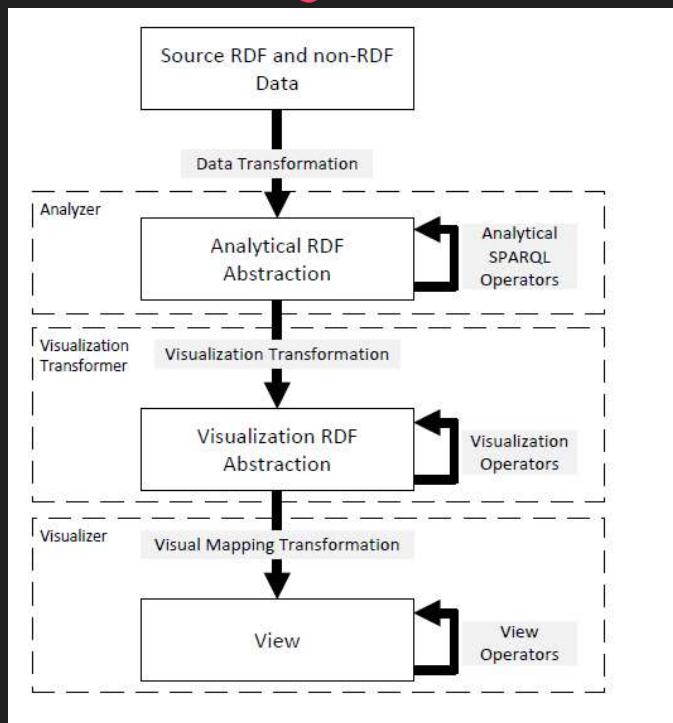
- The exploration must start from a specific node/keyword;
- Only sources that are compliant with the Semantic web can be included;
- Terms with multiple meanings may be difficult to locate.

Tabulator

- A generic browser for linked data on the web;
- Avoid domain-specific visualizations such as calendars, or address books;
- Recognize and follow RDF links to other RDF resources based on the user's exploration and analysis;
- Allow the combination of views, visualizations and data sources.



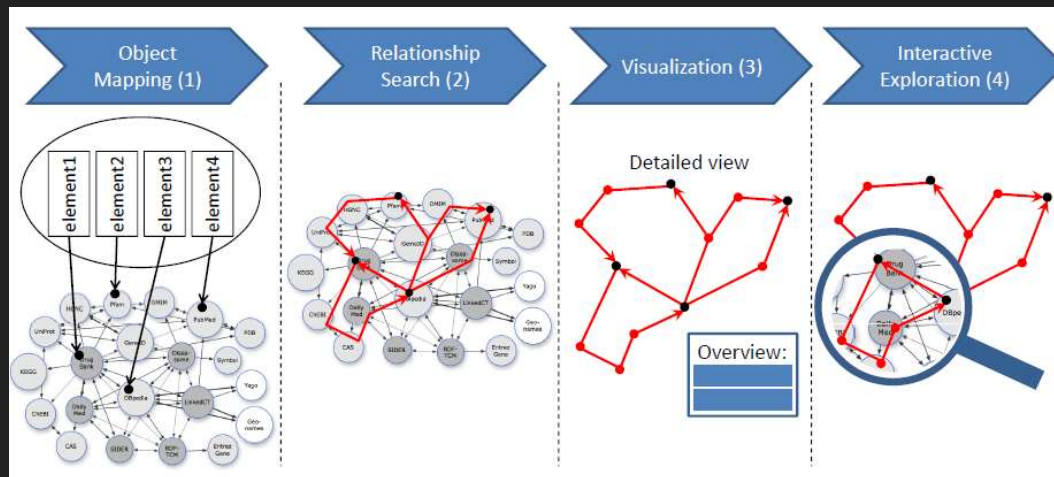
Linked Data Visualization Model



- Applies information visualization techniques to semantic data;
- Dynamic data extraction and visualization;
- Data presented as tree maps or maps if applicable;
- Details on demand, focus is on the overview;
- The user must select the SPARQL endpoint, the analyzer and the visualizer that will be used.

RelFinder

- Interactive discovery of semantic relationships between selected elements;
- Object selection must be unique, manual disambiguation is used;
- All relationships are identified but only few can be presented to the user, an overview is available to show the rest;
- Path length and relationships included/excluded can be filtered at a second iteration.



Explorer

- An open-source exploratory search tool for RDF graphs;
- Implemented as a direct manipulation interface metaphor;
- Implements a custom model of operations;
- Provides a Query-by-example interface.

The screenshot displays two panels from the Explorer interface. The left panel, labeled 'SET J', shows a search filter for 'Budapest' and a list of properties: 'type=' (with 'Resource' and 'City' as options), 'directType=' (with 'City' as an option), 'population=' (with the value '2016000'), 'name=' (with the value 'Budapest'), and 'cityIn=' (with the value 'Budapest (munic.)'). The right panel, labeled 'SET I', shows a search filter and a list of cities: '+ Budapest', '+ Praia', '+ Geneva', '+ Paris', '+ Vienna', '+ Lagos', '+ Addis Ababa', '+ Washington', '+ Rome', '+ Montreal', '+ Abidjan', '+ Bern', '+ Brussels', and '+ Lausanne'. The '+ Budapest' entry in the right panel is highlighted with a dashed blue box.

Semantic Exploration Techniques

- Semantic browsers;
- **SPARQL endpoint visualization tools;**
- Facet browsers;
- Query Writers;
- Schema Identifiers;
- Filtering-based exploration systems.

SPARQL endpoint visualization tools

- SPARQL endpoints are designed for machines;
- They offer no information interpretation, visualization or exploration support for human users;
- They do offer a gateway to many resources and datasets.

SPARQL endpoint visualization tools

Pros

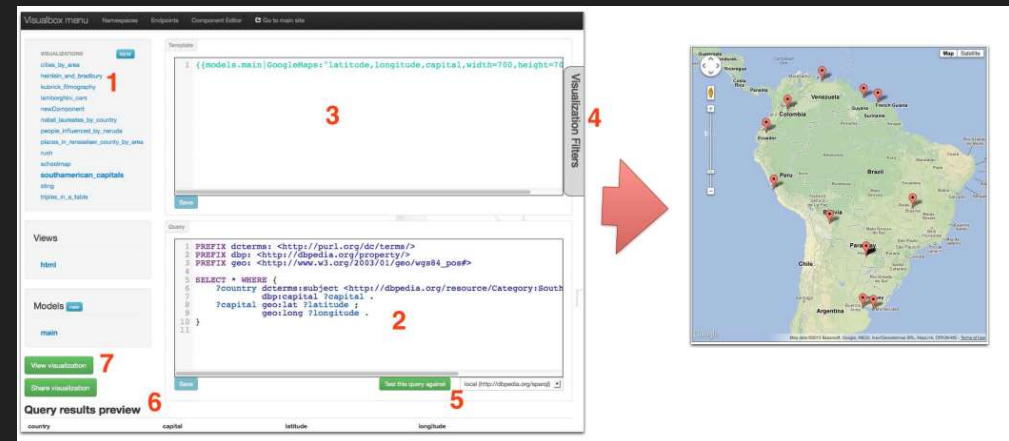
- Given that they are based on SPARQL queries, they can easily adapt to any endpoint;
- Can design their solution based on the expected results of the SPARQL.

Cons

- Require at least some knowledge regarding the SPARQL language;
- Are restricted by the limitations of the endpoints;
- Are restricted by the availability of the endpoints.

Visualbox

- Makes it easier for non-programmers to create web visualizations based on Linked Data;
- Provides a unified environment that supports the whole process of creating a visualization based on a SPARQL query;
- It runs a query on the server and provides a useful caching mechanism that allow users to visualize the data even if an endpoint is down or unresponsive.



Linked Data Query Wizard

CODE Linked Data Query Wizard Watch the screencast Welcome, Patrick Hoefler Log out

Visualize the 10 displayed results MindMap the 10 displayed results For the Geeks

Label	Type	Director	Producer	Narrative set in
Captain America	film	Albert Pyun	Menahem Golan Stan Lee	Italy Los Angeles Washington, D.C.
Pulp Fiction	film	Quentin Tarantino	Lawrence Bender	Los Angeles
The Core	film	Jon Amiel		Los Angeles San Francisco
Independence Day	film	Roland Emmerich	Dean Devlin Roland Emmerich	California Los Angeles New York City Washington, D.C.
Die Hard	film	John McTiernan	Joel Silver Lawrence Gordon	Los Angeles
Fire with Fire	film	David Barrett	50 Cent	Los Angeles New Orleans
The Running Man	film	Paul Michael Glaser		Los Angeles
Volcano	film	Mick Jackson	Lauren Shuler Donner Neal H. Moritz	Los Angeles
Sullivan's Travels	film	Preston Sturges	Buddy DeSylva Paul Jones	Los Angeles
Diamonds Are Forever	film	Guy Hamilton	Albert R. Broccoli Harry Saltzman	London Los Angeles South Africa

Displaying 10 of 599 results

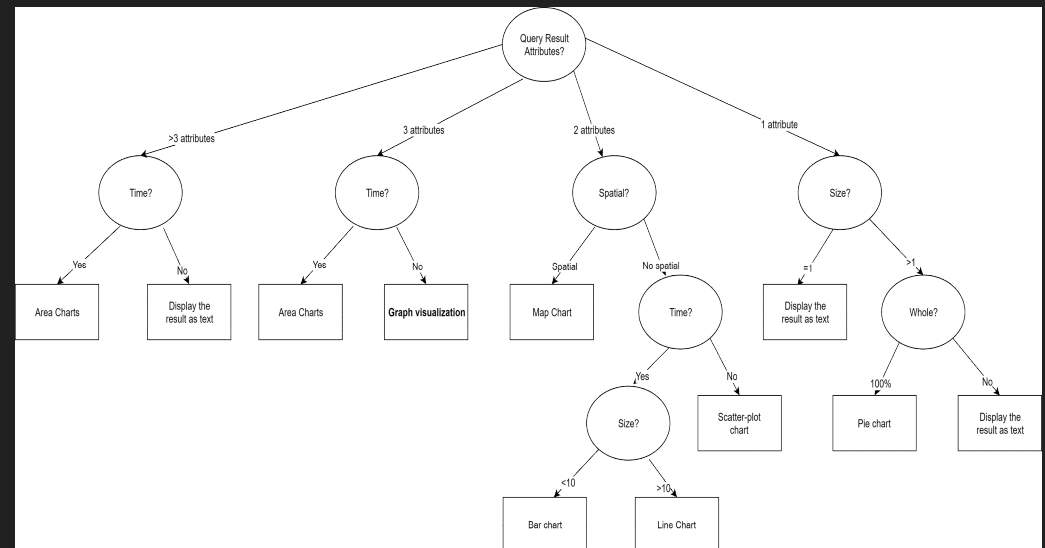
Load 10 more results Load 100 more results

Endpoint Statistics This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

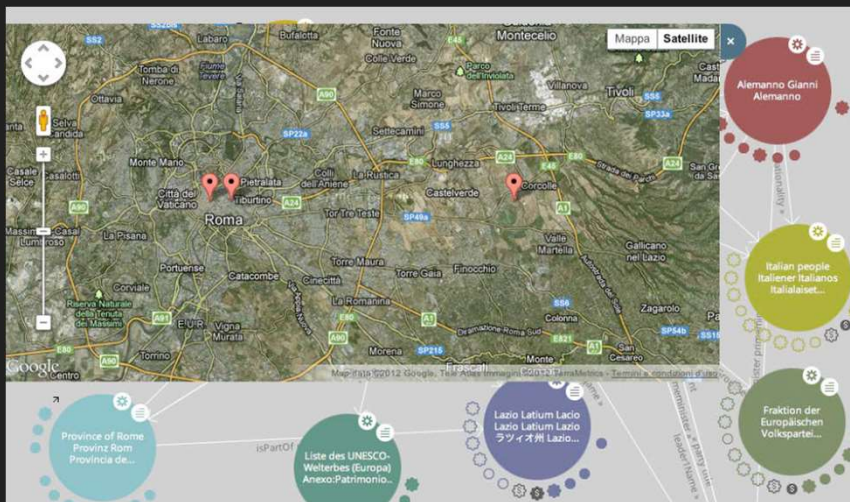
- A web-based tool for displaying, accessing, filtering, exploring, and navigating Linked Data stored in SPARQL endpoints;
- Turns the graph structure of Linked Data into a tabular interface and provides easy-to-use interaction possibilities;
- Uses metaphors and techniques from current search engines and spreadsheet applications that regular web users are already familiar with.

SPARQL-vision

- Supports the visualization of multiple types of SPARQL queries;
- Offers a Decision Support System that identifies the right visualization type for the right data;
- Can visualize the information as graph or chart as needed;
- Provides interactive filtering.



Lodlive



- Exploratory tool that build upon SPARQL endpoints;
- Allow users to browse linked data using interactive graph navigation;
- Starting from a given URI, the user can explore linked data by following the links;
- Spatial data and images are extracted by the endpoint and show in a map and gallery accordingly.

Semantic Exploration Techniques

- Semantic browsers;
- SPARQL endpoint visualization tools;
- **Facet browsers;**
- Query Writers;
- Schema Identifiers;
- Filtering-based exploration systems.

Facet browsers

- A user friendly interface to data repositories;
- Dynamic navigation through facets of resources, property and data types used for exploration;
- Specific techniques are employed to support the exploration;
- Facet-specific display options are available.

Facet browsers

Pros

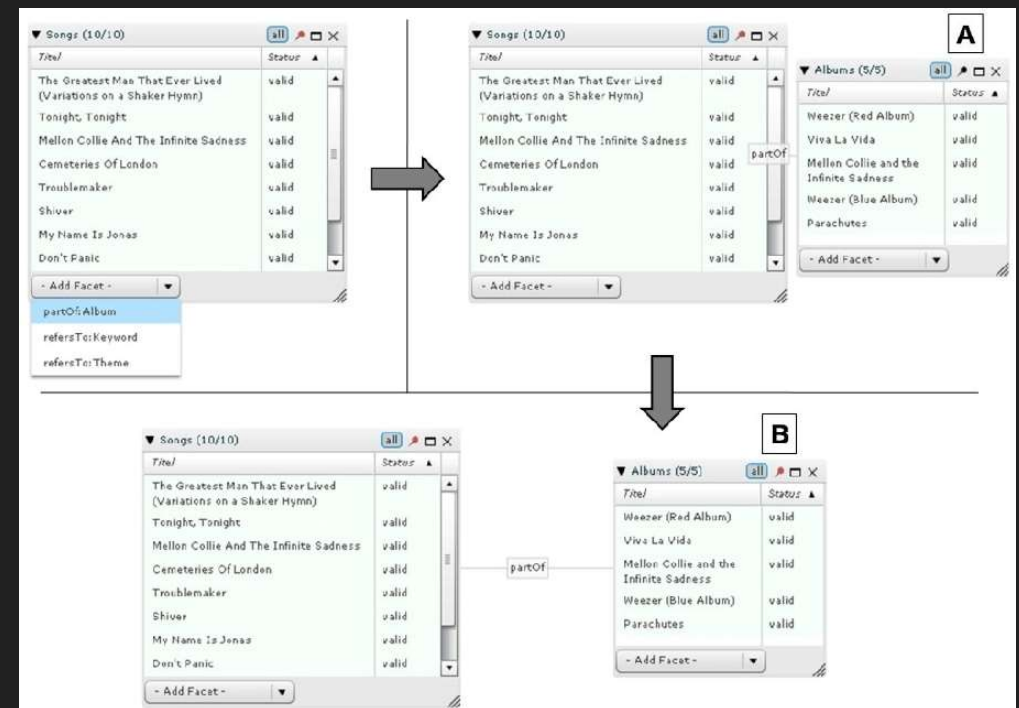
- Support the novice users in exploring the information;
- Facet-specific exploration functionalities improve the experience;
- Dynamic exploration;
- Exploration based on semantic properties and data types.

Cons

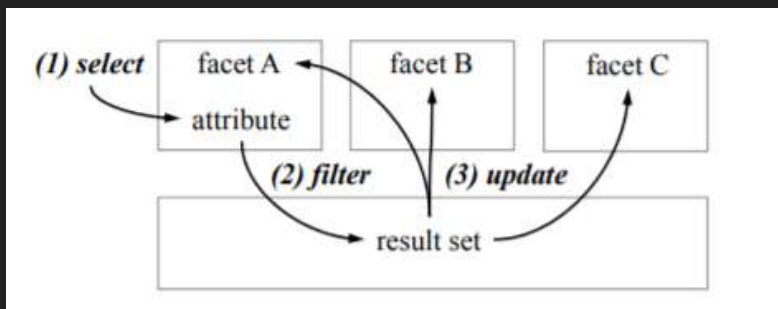
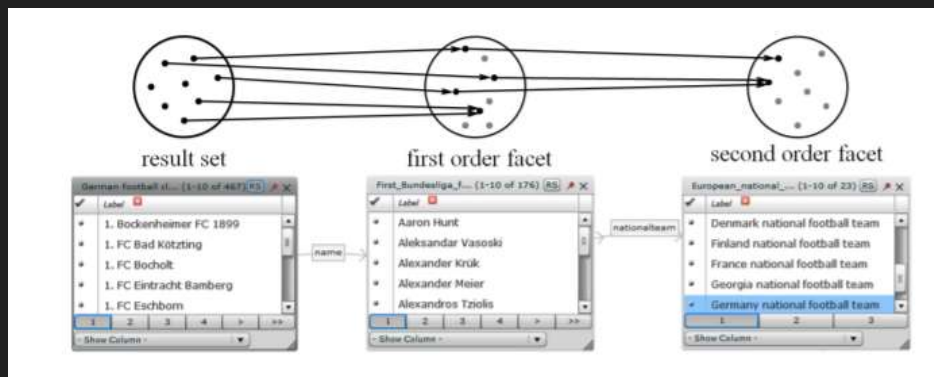
- Hierarchical exploration is not intuitive;
- Lack of dataset overview;
- Restriction on the volume of the visualized information.

gFacet

- Combines graph-based visualization with faceted filtering techniques;
- Supports the integration of different domains;
- Offers efficient exploration of highly structured and interrelated datasets;
- Allows to access information from distant user-defined perspectives.



Facet Graphs



- Offers faceted search for creating semantically unique search queries;
- Search results are combined with facets and filtered to produce a personalized interface to build search queries;
- The results are presented in a graph visualization, bringing even distantly connected facets on one page, helping the users with the exploration'
- Each node contains a list that provides sorting, paging and scrolling functionalities and thereby enables the easy handling of even large amounts of objects.

/facet

- Users are able to select and navigate through facets of resources of any type and to make selections based on properties of other, semantically related, types;
- Offers a keyword search interface that dynamically makes semantically relevant suggestions.
- Allows the inclusion of facet-specific display options that go beyond the hierarchical navigation ;
- Works on any RDFS dataset without any additional configuration;
- Offers exploration of the complete dataset;
- The automatic facet configuration generated by the system can then be further refined to configure it as a tool for end users.

The screenshot displays the 'MultimediaN E-Culture Facet Browsing' application. The interface is divided into several sections:

- Select Type:** A tree view showing the hierarchy of types: `vra:VisualResource` (123), `vra:Work` (123), `vp:Subject`, `aat:Subject` (2), `ulan:Subject`, and `ulan:Person` (1).
- Select Facets for: Work:** A search bar and a list of facets: `Contributor`, `Coverage`, `Creator`, `Format`, `Subject and Keywords`, and `Title`.
- Facet Selection:** Three active facets are shown:
 - Creator:** `ulan:Cézanne, Paul` (123)
 - Date:** 1867, 1873, 1875, 1875-1876, 1877-1878
 - Material/Medium:** `aat:oil paint` (108), `aat:water-base paint` (1)
- Constraints:** `ulan:Person birthPlace = Provence-Côte d'Azur` and `ec:Work Creator = Cézanne, Paul`.
- Results grouped by Creator:** A list of works by Paul Cézanne (123):
 - `The Abduction` (Cézanne, Paul 130.0)
 - `Still Life with Apples` (Cézanne, Paul 130.0)
 - `Still Life with Flowe ...` (Cézanne, Paul 100.0)
 - `Apples and Oranges (P ...` (Cézanne, Paul 100.0)
- Timeline:** A horizontal axis from 1840 to 1900, showing the artist's period (Cézanne, Paul) and art movements (Impressionist, Post-Impressionist).

Faceted Wikipedia Search

The screenshot shows the DBpedia search interface. At the top left is the DBpedia logo and navigation links. A search bar (1) contains the text "enter search terms" and a "Search" button. Below the search bar are navigation links: "First | Previous | Next | Last". On the left side, there are several faceted filters (2):

- item type**: A dropdown menu showing "River (26)", "Place (26)", and "Body Of Water (26)".
- has mouth at**: A dropdown menu showing "Rhine (26)", "Ingelheim am Rhein (1)", and "Lahnstein (1)".
- length (m)**: A range selector with "50000" selected, showing "90000 (2)", "55000 (2)", and "529000 (1)".
- watershed (km²)**: A range selector with "827 (1)", "2632 (1)", and "28286 (1)".
- name**: A dropdown menu showing "Pfinz (1)" and "Ranch (1)".

On the right side, there is a "Your Filters" section (3) showing the active filters: "item type: River", "has mouth at: Rhine", and "length (m): 50000 and up". Below this, there are three search results (4):

- Main**: A river in Germany, 524 km (328 miles) long (including White Main, 574 km), and it is one of the more significant tributaries of the Rhine. The Main flows through the German states of Bavaria, Baden-Württemberg (forming the border with Bavaria for some distance) and Hesse. Its watershed competes with the Danube for water, as a result, many of its boundaries are identical with those of the European Watershed.
- Moselle River**: The Moselle is a river flowing through France, Luxembourg and Germany. It is a left tributary of the Rhine, joining it at Koblenz. A small part of Belgium is also drained by the Mosel through the Our. Its name comes from the Latin *Mosella*, meaning the "Little Mouse". The river gave its name to two French départements: Moselle and Meurthe-et-Moselle.
- Neckar**: The Neckar is a 347 km (228 mi) long river, mainly flowing through the southwestern state of Baden-Württemberg, but also a short section through Hesse in Germany, a major right tributary of the River Rhine, which it joins at Mannheim.
- Ruhr (river)**: The Ruhr is a medium-size river in western Germany, a right tributary (east-side) of the Rhine.

- Query complexity, "Which Rivers flow into the Rhine and are longer than 50 kilometers?";
- No key word matching is performed;
- Queries are answered based on structured information that has been extracted from many different Wikipedia articles;
- Queries Wikipedia like a structured database.

Semantic Exploration Techniques

- Semantic browsers;
- SPARQL endpoint visualization tools;
- Facet browsers;
- **Query Writers;**
- Schema Identifiers;
- Filtering-based exploration systems.

Query writers

- SPARQL language is complicated;
- Novice users often struggle to form meaningful queries;
- Behavior and results differ a lot from the relational models that most users are familiar with.

Query writers

Pros

- Support the novice users in understanding the SPARQL language;
- Support the creation of meaningful queries.

Cons

- Limit the querying capabilities of the SPARQL;
- Hide the actual SPARQL code from the user.

SPARQL Builder

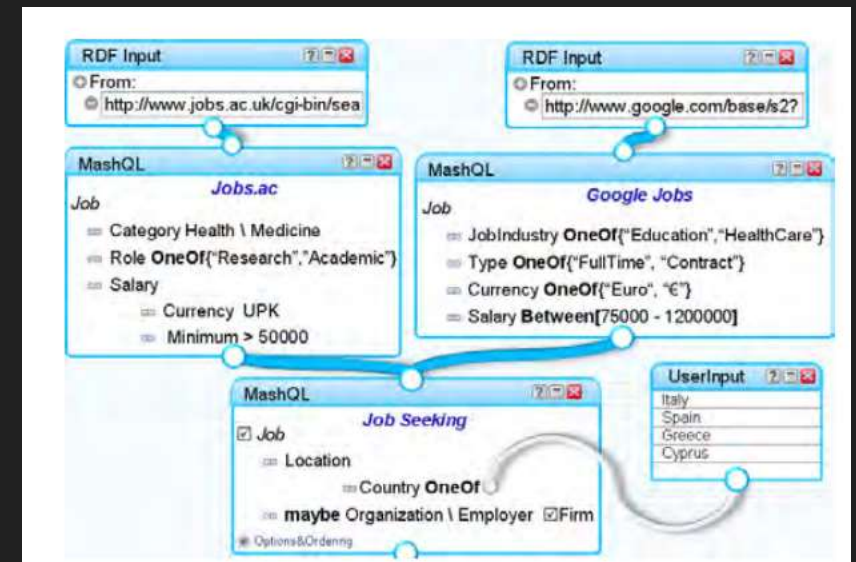
- An intelligent tool by which users with no knowledge of SPARQL can generate SPARQL queries;
- Can support them in creating the right query to retrieve results satisfying their requirements;
- SPARQL Builder collaborates with TogoTable, a web application enabling biological researchers to upload their data in a table form and add annotations obtained from SPARQL endpoints.

The screenshot shows the SPARQL Builder interface. At the top, it displays the SPARQL endpoint URL: `https://www.ebi.ac.uk/rd/services/sparql`. Below this, the start class is set to `Protein` and the end class is `Pathway`. A message indicates that 95 paths were found, with a permalink icon. The main section is titled "Select a path to generate a SPARQL query." and shows a sequence of selection steps:

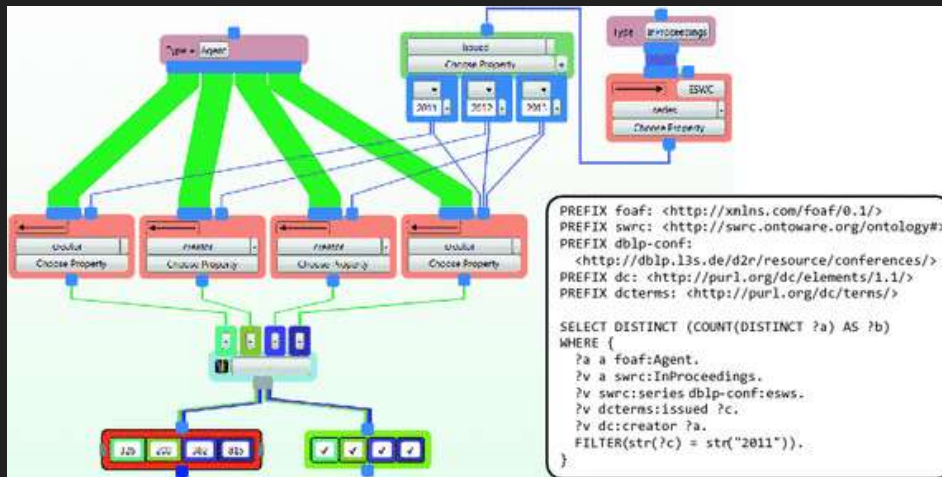
Protein	-	controller	-	Control	-	controlled	-	Pathway
Protein	-	left	-	Degradation	-	pathwayComponent	-	Pathway
Protein	-	participant	-	TemplateReaction	-	pathwayComponent	-	Pathway
Protein	-	product	-	TemplateReaction	-	pathwayComponent	-	Pathway
Protein	-	right	-	BiochemicalReaction	-	pathwayComponent	-	Pathway
Protein	-	dataSource	-	Provenance	-	dataSource	-	Pathway
Protein	-	left	-	BiochemicalReaction	-	pathwayComponent	-	Pathway

MashQL

- A query-by-diagram language that regards the Internet as a database and generalizes the idea of mashups;
- People are allowed to build data mashups diagrammatically;
- MashQL queries are translated into and executed as SPARQL queries;
- It allows querying a data source without any prior understanding of the schema or the structure of this source.



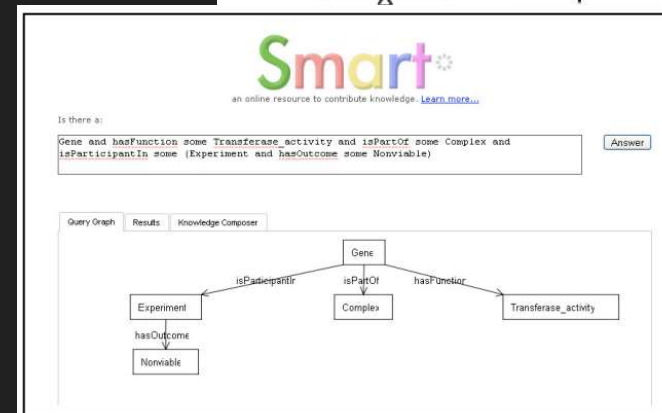
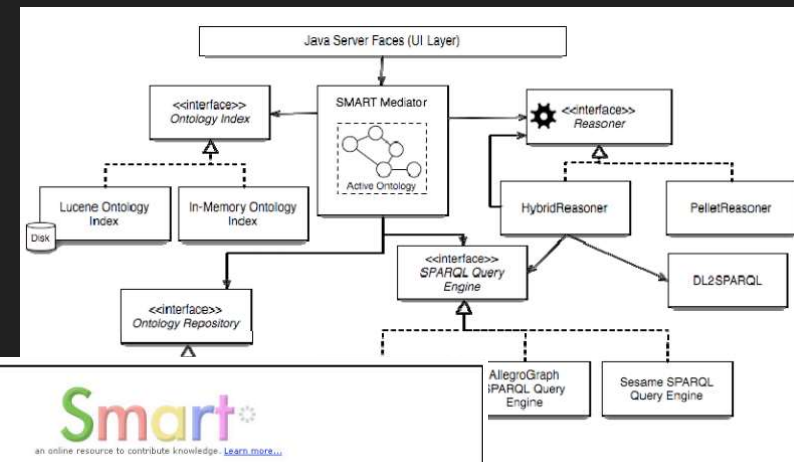
SparqlFilterFlow



- An approach for visual SPARQL querying;
- Based on the concept of extended filter and flow graph;
- The queries can be created entirely with graphical elements.

SMART

- Semantic web information Management with automated Reasoning Tool;
- Aims to provide intuitive tools for life scientists to represent, integrate, manage and query heterogeneous and distributed biological knowledge;
- Features include semantic query composition and validation, a graphical representation of the query, and the retrieval of pre-computed inferences from an RDF triple store.



Semantic Exploration Techniques

- Semantic browsers;
- SPARQL endpoint visualization tools;
- Facet browsers;
- Query Writers;
- **Schema Identifiers;**
- Filtering-based exploration systems.

Schema Identifiers

- Try to extract the SPARQL endpoint schema;
- Focus on re-using available relational techniques;
- Support the understanding of information from user familiar with relational models.

Schema Identifiers

Pros

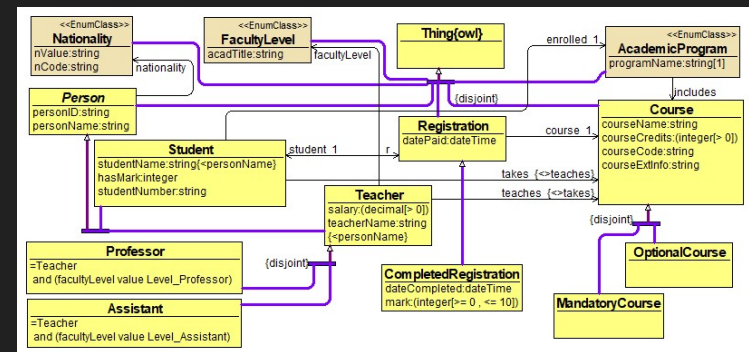
- Re-use of relational solutions;
- Presentation of the information in ways that users are familiar with;
- Support the understanding of the available information.

Cons

- Demanding on the endpoint;
- A schema may not be available;
- The extracted schema is an approximation; may conceal information of interest.

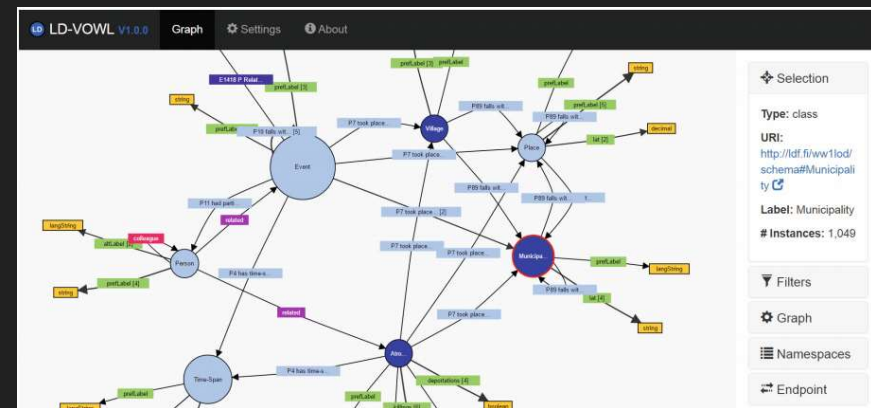
ViziQuer

- Starts from the address of a SPARQL endpoint provided by the user;
- It extracts and visualizes graphically the data schema of the endpoint;
- The user is able to overview the data schema and use it to construct a SPARQL query;
- The schema is extracted using a predefined sequence of SPARQL queries at the SPARQL endpoint;
- This is a time consuming process dependent on the ontology size and speed of the SPARQL endpoint while only typed data are supported.



Tbox-based visualization

- Aims to extract and visualize the information on the used schema, also called TBox from SPARQL endpoints;
- The tool infers the schema based on several SPARQL queries;
- This information is incrementally added to an interactive graph visualization based upon the Visual Notation for OWL Ontologies;
- A node-link-based graph visualization is chosen, as it allows users to grasp certain structural criteria at a single glance, such as the presence of highly linked central classes or largely disjoint clusters of classes, before proceeding to a deeper analysis.



Semantic Exploration Techniques

- Semantic browsers;
- SPARQL endpoint visualization tools;
- Facet browsers;
- Query Writers;
- Schema Identifiers;
- **Filtering-based exploration systems.**

Filtering-based exploration systems.

- Many systems offer semantic exploration through filters;
- The filters can be dynamic or predefined;
- They are usually following the data type or semantic properties of the dataset;
- The exploration can be either in raw format or through visualization.

Filtering-based exploration systems.

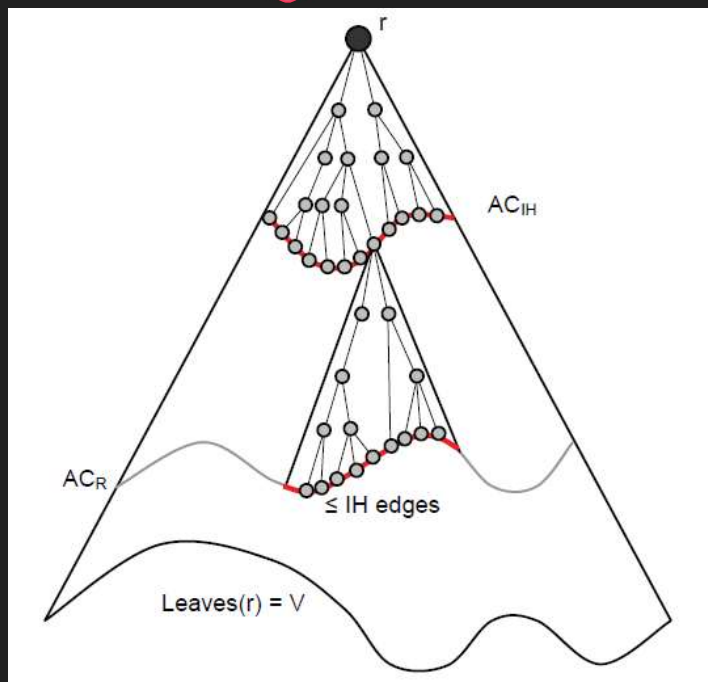
Pros

- Exploration over multiple detail levels;
- Filters support the exploration of large datasets by allowing users to focus on the information of interest;
- Scalable for large and complex datasets;
- Few if any requirements regarding data types.

Cons

- The dataset overview may not be available;
- Not all objects will fit in a screen;
- Hiding some information may result in loss of value.

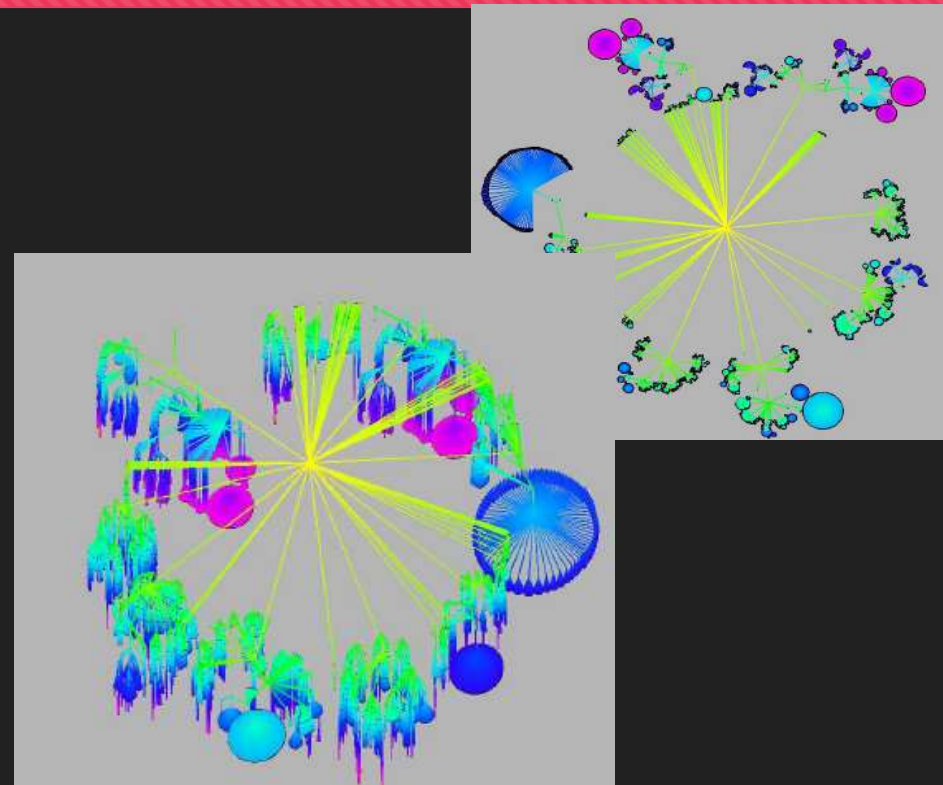
ASK- GraphView



- Node-link-based graph visualization system;
- Uses graph topology clustering methods – without requiring extra information;
- Supports interactive navigation of large graphs;
- Client- server system that takes advantage of the permanent storage

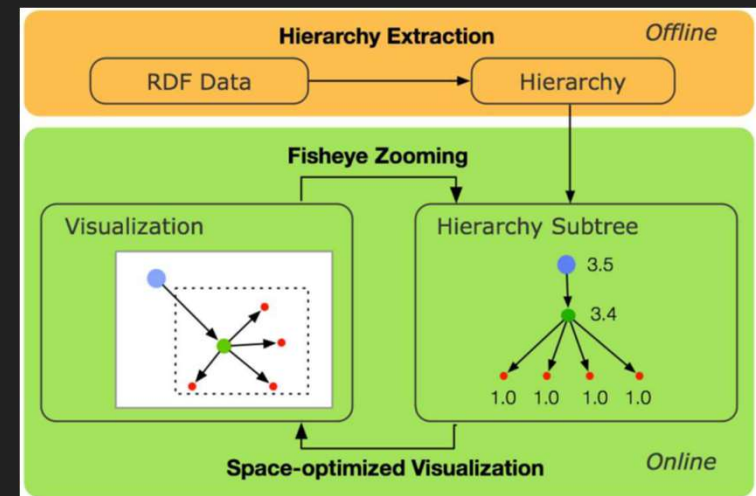
Tulip

- Designed around the principal of overview first, then filtering and details only on demand;
- Introduces a data type that can support the visualization of a graph hierarchy;
- Creates clusters giving emphasis on maintaining the semantic coherence;
- Ensures that the graph will maintain the semantic information.

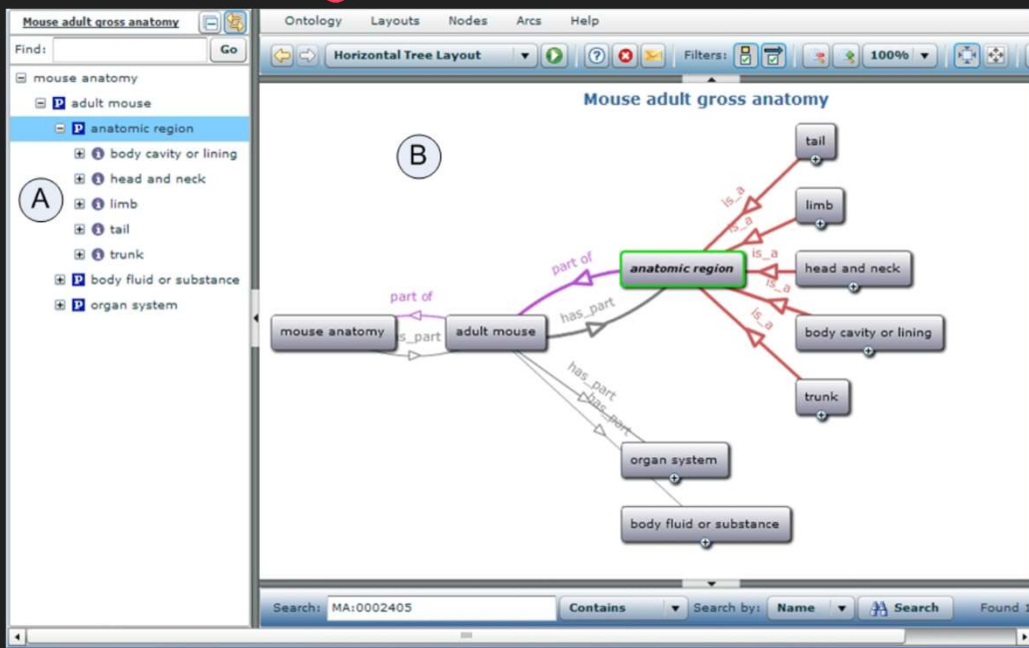


ZoomRDF

- Employs a space-optimized visualization algorithm for RDF;
- Displays more resources at the available display space;
- Introduces a fisheye zooming concept, which assigns more space to some individual nodes while still preserving the overview structure;
- Evaluates the importance of the nodes based on the user's choices, giving more space to important to the user elements.

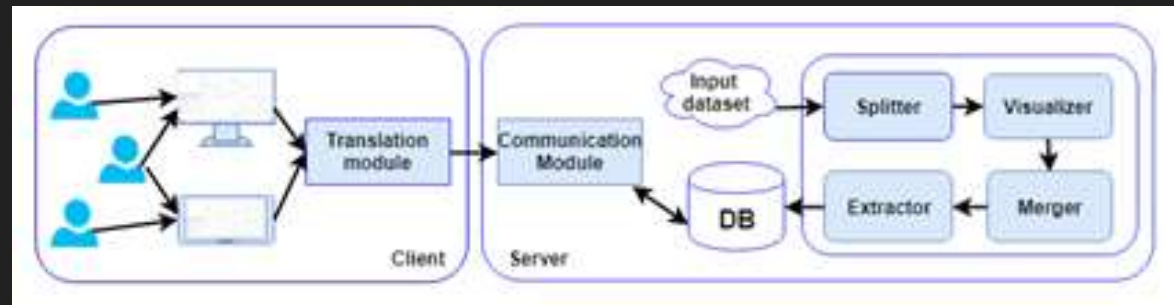


FlexViz



- Offers node and edge specific filters that are based on search and navigation criteria;
- Reduces the amount of handled data;
- Provides meaningful subsets to the user;
- Provides step by step node expansion.

Interactive Visualization of Large Graphs



- Pre-processes the input dataset;
- Creates one continuous graph in the two-dimensional space ;
- Store the information in a graph database;
- Offers filtering and abstraction functions that can further support the navigation.

Questions



**Thank you for your
time and attention!!!**

Maria Krommyda

PhD Candidate

mariakr@dbl-lab.ece.ntua.gr

References

[Slide 5] <https://lod-cloud.net/>

[Slide 11] Berners-Lee, Tim, et al. "Tabulator: Exploring and analyzing linked data on the semantic web." *Proceedings of the 3rd international semantic web user interaction workshop*. Vol. 2006. 2006.

[Slide 12] Brunetti, Josep Maria, et al. "Formal linked data visualization model." *Proceedings of International Conference on Information Integration and Web-based Applications & Services*. 2013.

[Slide 13] Heim, Philipp, Steffen Lohmann, and Timo Stegemann. "Interactive relationship discovery via the semantic web." *Extended Semantic Web Conference*. Springer, Berlin, Heidelberg, 2010.

[Slide 14] S. F. De Araujo and D. Schwabe, "Explorator: a tool for exploring rdf data through direct manipulation," in *DOW2009*, 2009.

References

[Slide 18] A. Graves, "Creation of visualizations based on linked data," in International Conference on WIMS. ACM, 2013.

[Slide 19] P. Hoefler, M. Granitzer, E. E. Veas, and C. Seifert, "Linked data query wizard: A novel interface for accessing sparql endpoints." in LDOW, 2014.

[Slide 20] M. Krommyda and V. Kantere, "Understanding SPARQL endpoints through targeted exploration and visualization," in IEEE Graph Computing, 2019.

[Slide 21] D. V. Camarda, S. Mazzini, and A. Antonuccio, "LodLive, exploring the web of data," in I-SEMANTICS, 2012.

[Slide 25] Heim, P. et al. "gFacet: A Browser for the Web of Data." *IMC-SSW@SAMT* (2008).

References

[Slide 26] Heim, Philipp & Ertl, Thomas & Ziegler, Jürgen. (2010). Facet Graphs: Complex Semantic Querying Made Easy. 6088. 288-302. 10.1007/978-3-642-13486-9_20.

[Slide 27] Hildebrand, Michiel, Jacco Van Ossenbruggen, and Lynda Hardman. "/facet: A browser for heterogeneous semantic web repositories." *International Semantic Web Conference*. Springer, Berlin, Heidelberg, 2006.

[Slide 28] Rasmus Hahn, Christian Bizer, Christopher Sahnwaldt, Christian Herta, Scott Robinson, Michaela Bürgle, Holger Düwiger, Ulrich Scheel: [Faceted Wikipedia Search](#). 13th International Conference on Business Information Systems (BIS 2010), Berlin, Germany, May 2010.

[Slide 32] A. Yamaguchi, K. Kozaki, K. Lenz, H. Wu, and N. Kobayashi, "An intelligent sparql query builder for exploration of various life-science databases." in IESD@ ISWC, 2014

References

[Slide 33] J. Mustafa and M. D. Dikaiakos, "Mashql: A query-by-diagram topping sparql towards semantic data mashups," University of Cyprus.

[Slide 34] F. Haag, S. Lohmann, and T. Ertl, "Sparqlfilterflow: Sparql query composition for everyone," in ESWC. Springer, 2014.

[Slide 35] A. D. L. Battista, N. Villanueva-Rosales, M. Palenychka, and M. Dumontier, "Smart: A web-based, ontology-driven, semantic web query answering application." Semantic Web Challenge, vol. 295, 2007.

[Slide 39] M. Zviedris and G. Barzdins, "Viziquer: a tool to explore and query sparql endpoints," in ESWC. Springer, 2011.

[Slide 40] M. Weise, S. Lohmann, and F. Haag, "Extraction and visualization of tbox information from sparql endpoints," in EKAW. Springer, 2016.

References

[Slide 44] J. Abello, F. van Ham, and N. Krishnan, "ASK-GraphView: A Large Scale Graph Visualization System," *TVCG*, vol. 12, no. 5, 2006.

[Slide 45] D. Auber, "Tulip - A Huge Graph Visualization Framework," in *Graph Drawing Software*, 2004.

[Slide 46] K. Zhang, H. Wang, D. T. Tran, and Y. Yu, "ZoomRDF: semantic fisheye zooming on RDF data," in *WWW*, 2010.

[Slide 47] S. Falconer, C. Callendar, and M.-A. Storey, "A Visualization Service for the Semantic Web," in *Knowledge Engineering and Management by the Masses*, 2010.

[Slide 48] Krommyda, Maria, Verena Kantere, and Yannis Vassiliou. "IVLG: Interactive Visualization of Large Graphs." *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019.