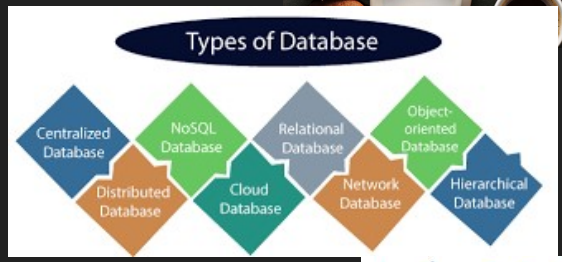
Semantic Exploration of Big Data

Maria Krommyda & Verena Kantere

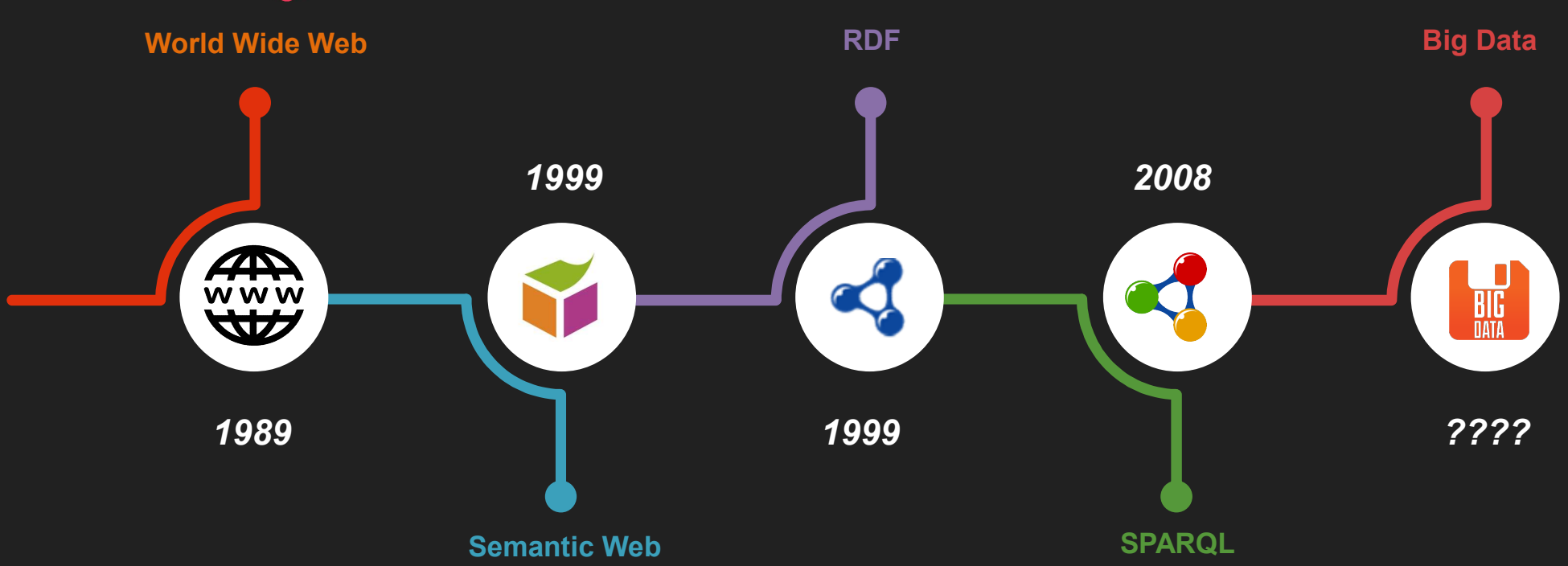
2020 IEEE International Conference on Big Data

About me!

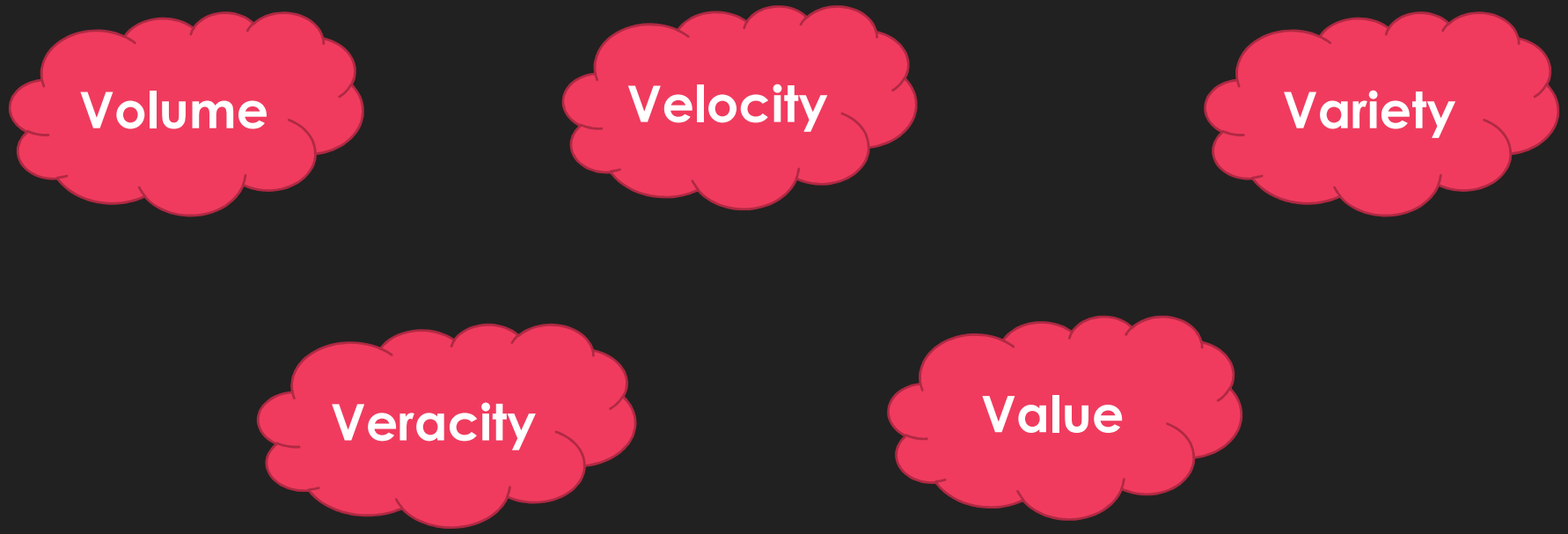
- Electrical & Computer Engineer, NTUA;
- PhD Candidate, Big Data Management & Visualization;
- Knowledge and Database Systems Laboratory;
- Software engineer, i-Sense group of ICCS.



Linked datasets



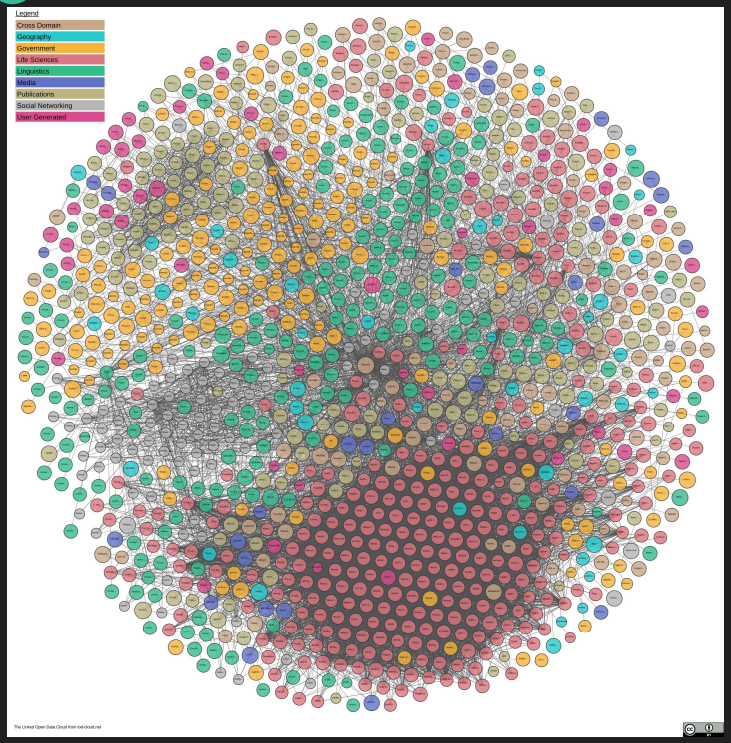
The V's of the Big Data



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

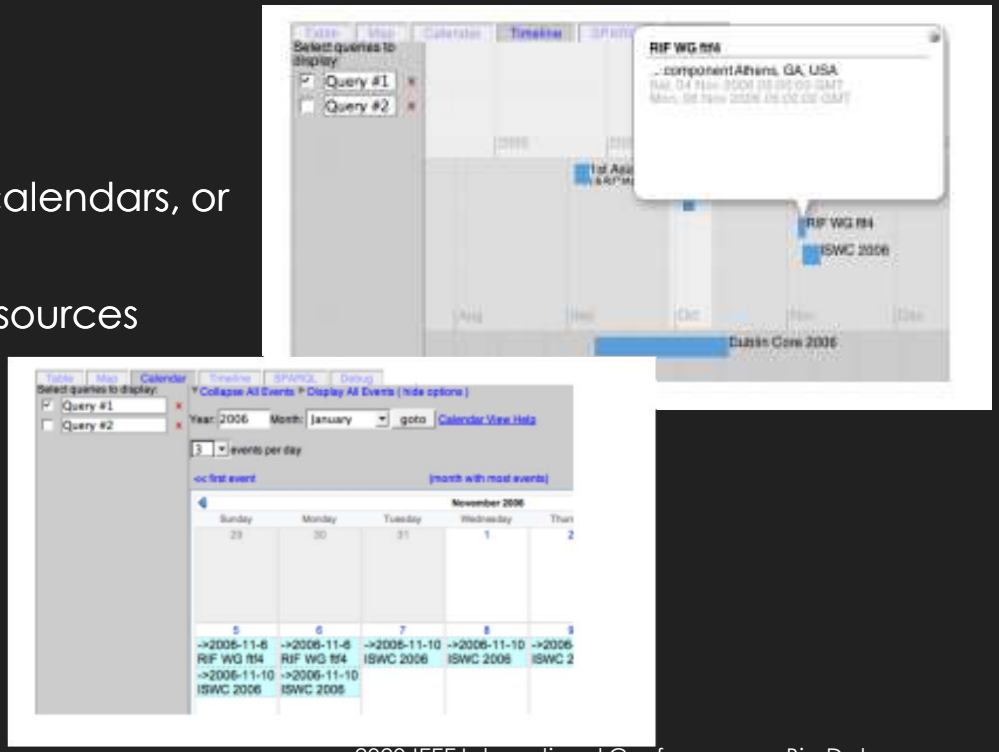
- Can link any data source;
- Provide interactive exploration;
- Most actions are intuitive, due to the experience of the user with the Web browsers;
- Semantic content is highlighted.

Cons

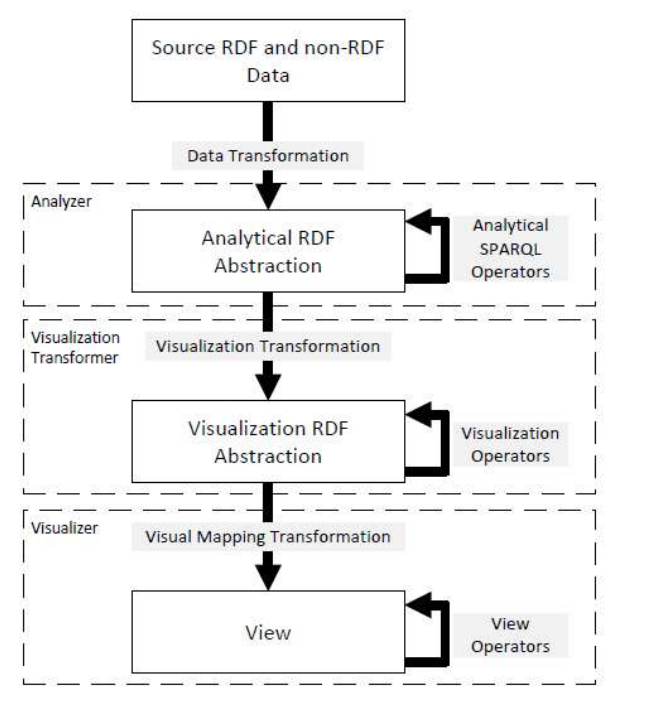
- The exploration must start from a specific node/keyword;
- Only sources that are compliant with the Semantic web can be included;
- Terms with multiple meanings may be difficult to locate.

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.



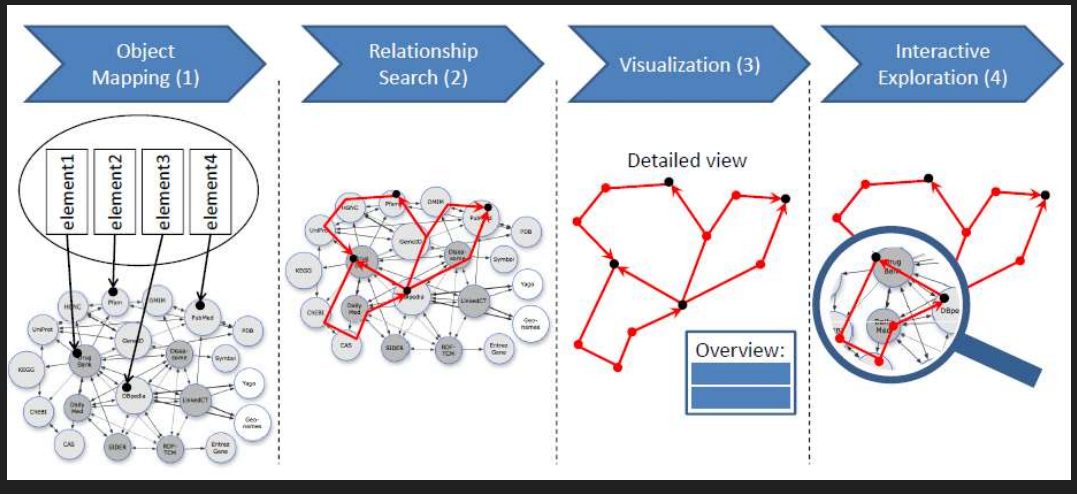
Linked Data Visualization Model



- 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.

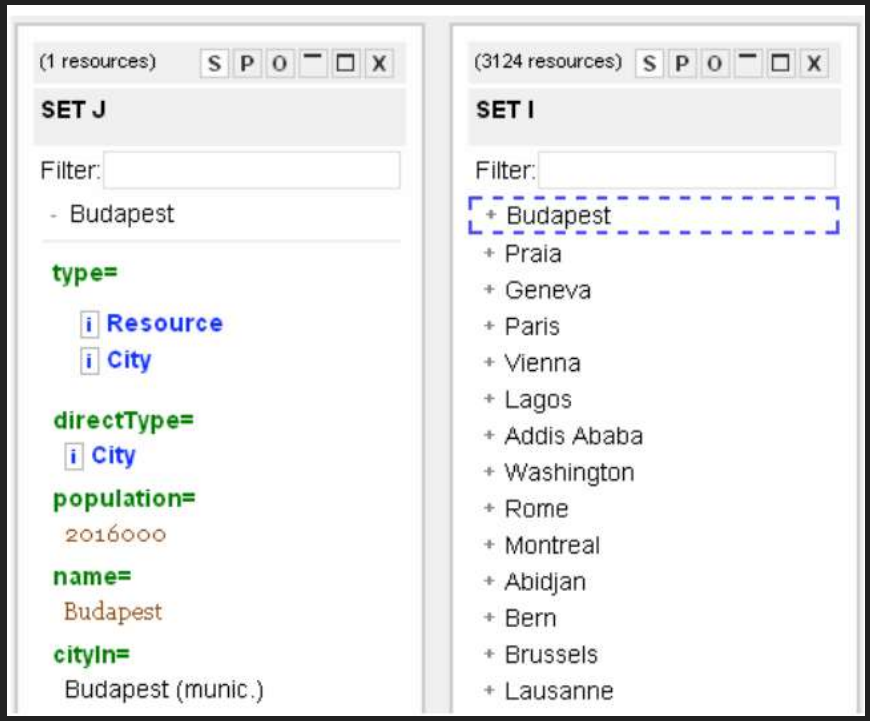
RelFinder

- 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.



Explorator

- 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.



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

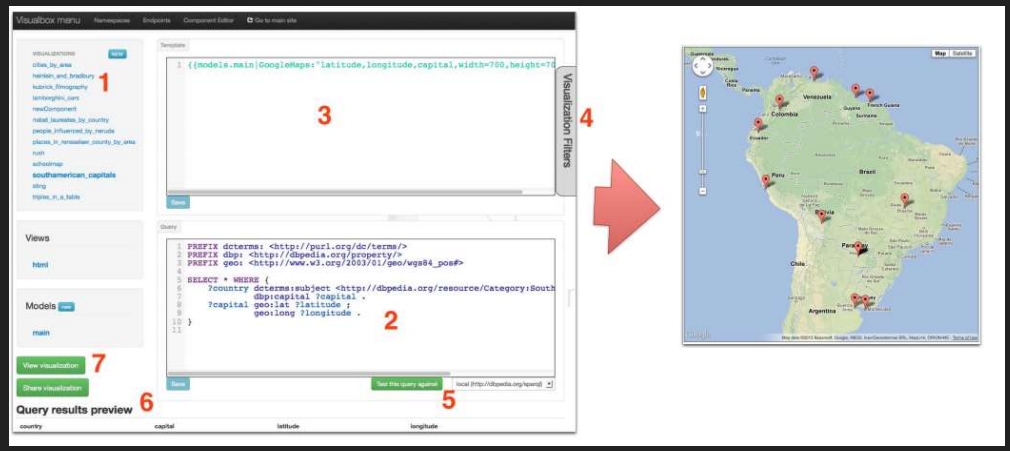
- Given that they are based on SPARQL queries, they can easily adapt to any endpoint;
- Can design their solution based on the expected results of the SPARQL.

Cons

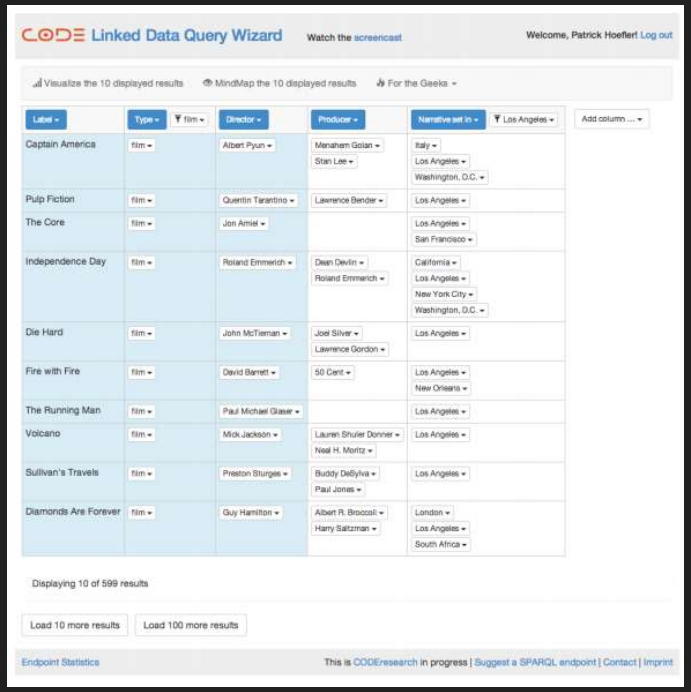
- Require at least some knowledge regarding the SPARQL language;
- Are restricted by the limitations of the endpoints;
- Are restricted by the availability of the endpoints.

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.



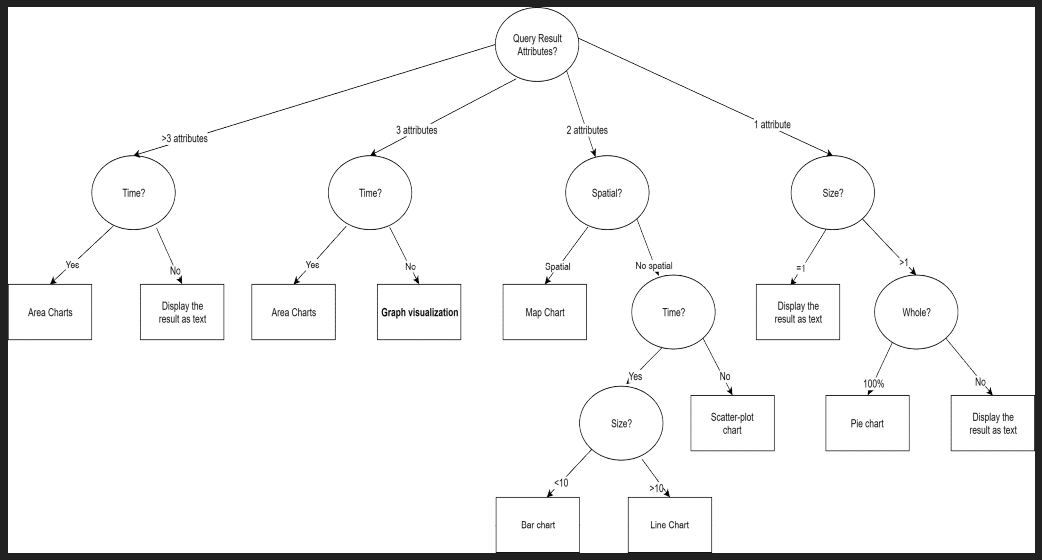
Linked Data Query Wizard



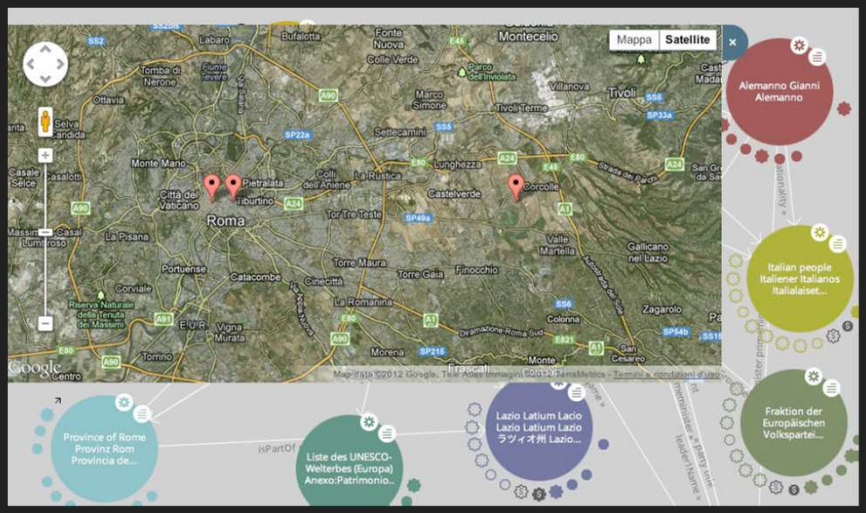
- A web-based tool for displaying, accessing, filtering, exploring, and navigating Linked Data stored in SPARQL endpoints;
- Turns the graph structure of Linked Data into a tabular interface and provides easy-to-use interaction possibilities;
- Uses metaphors and techniques from current search engines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

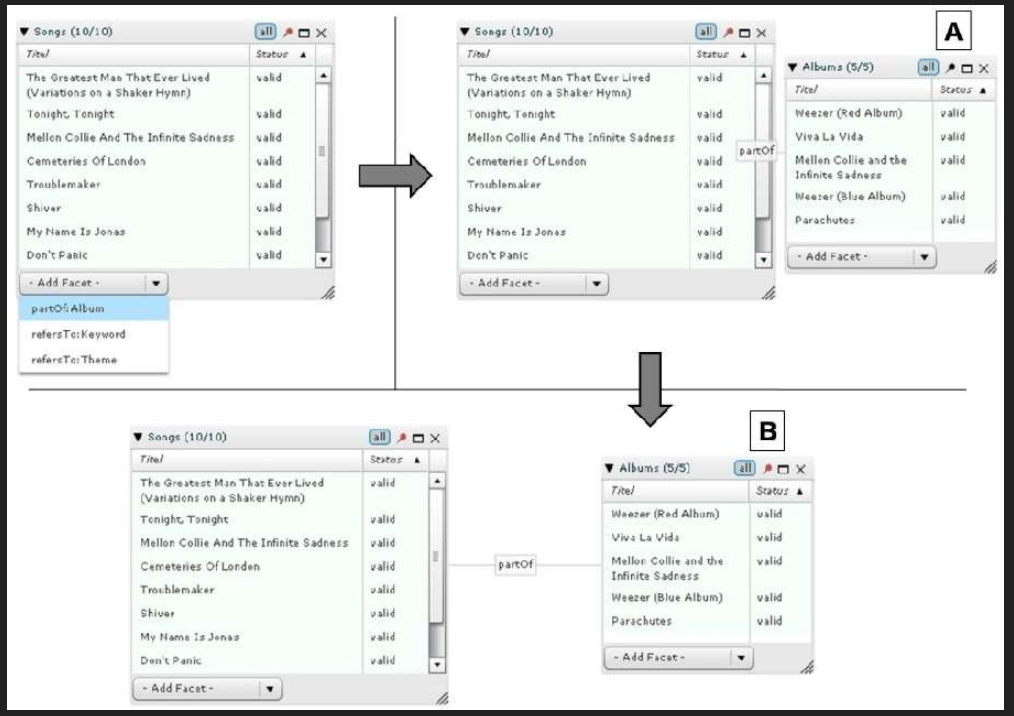
- Support the novice users in exploring the information;
- Facet-specific exploration functionalities improve the experience;
- Dynamic exploration;
- Exploration based on semantic properties and data types.

Cons

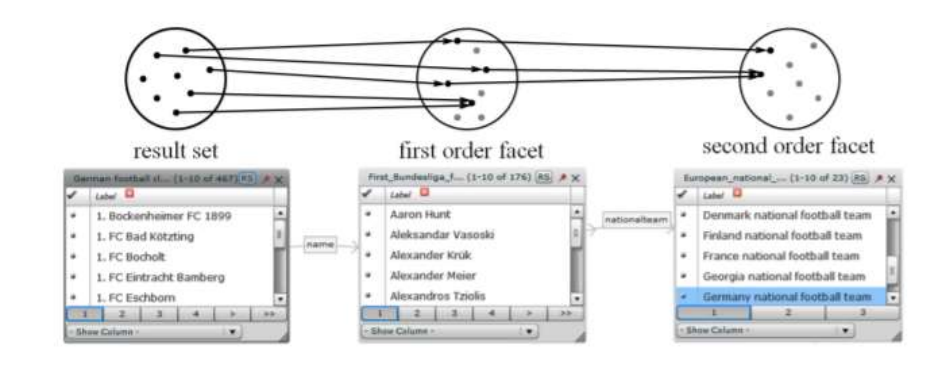
- Hierarchical exploration is not intuitive;
- Lack of dataset overview;
- Restriction on the volume of the visualized information.

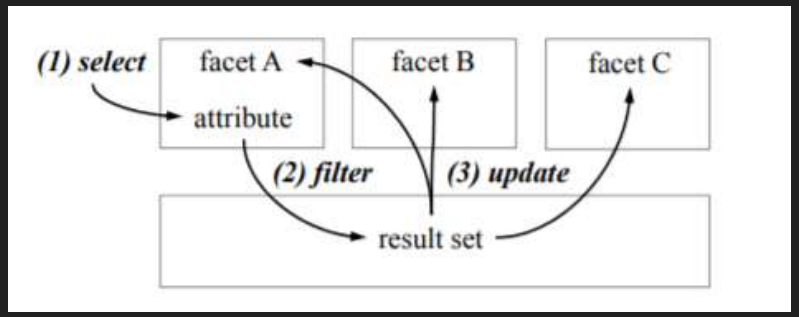
gFacet

- 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

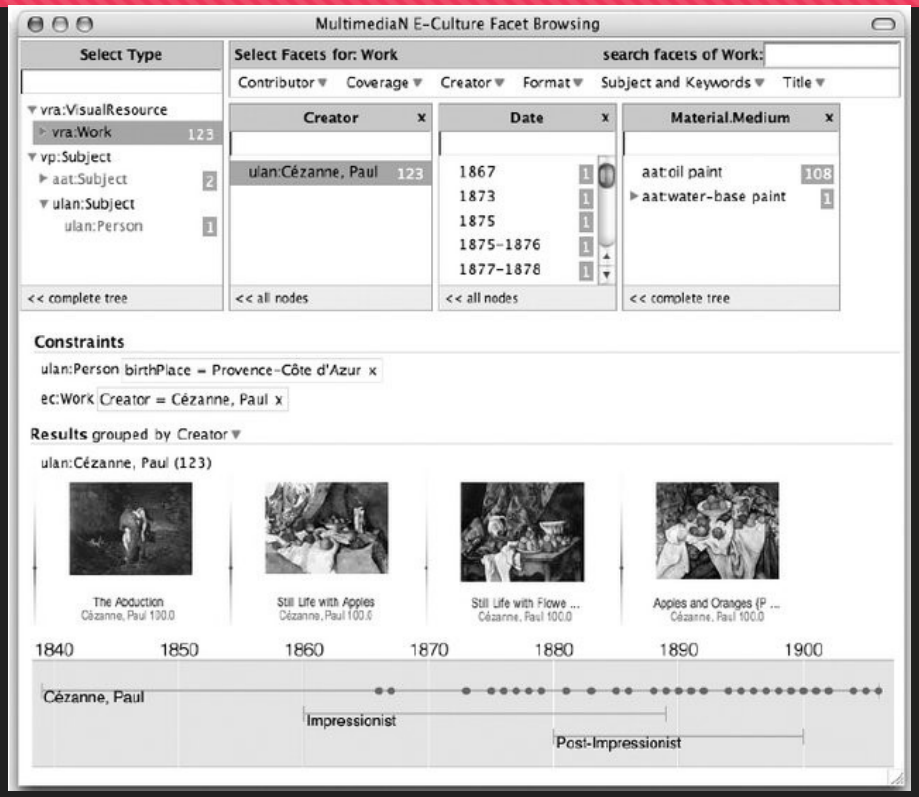




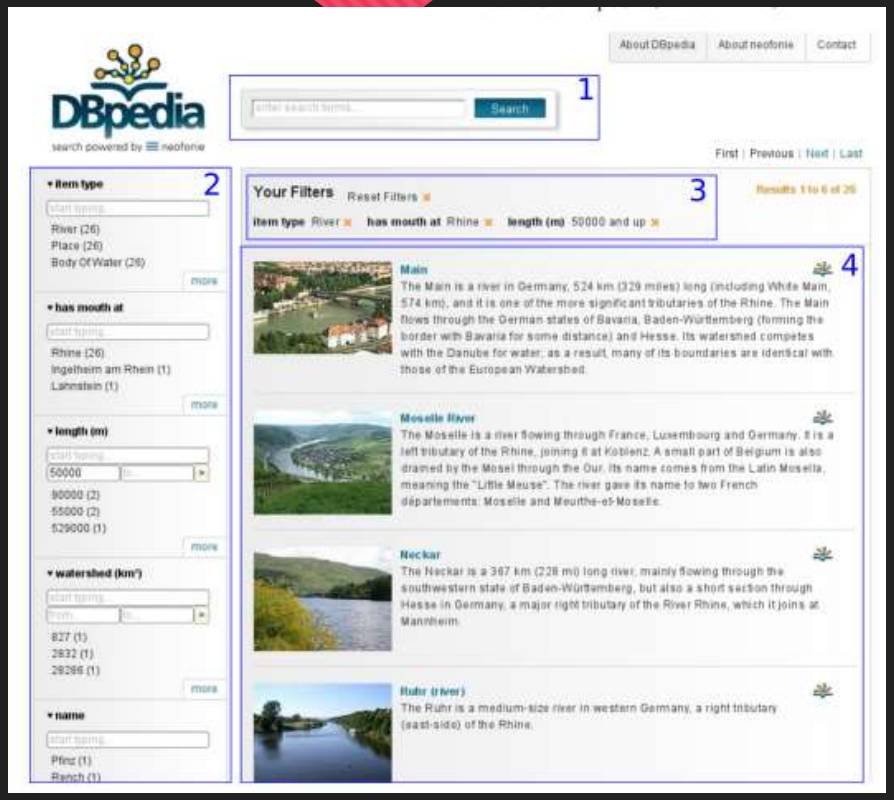
- Offers faceted search for creating semantically unique search queries;
- Search results are combined with facets and filtered to produce a personalized interface to build search queries;
- The results are presented in a graph visualization, bringing even distantly connected facets on one page, helping 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.



Faceted Wikipedia Search



- 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

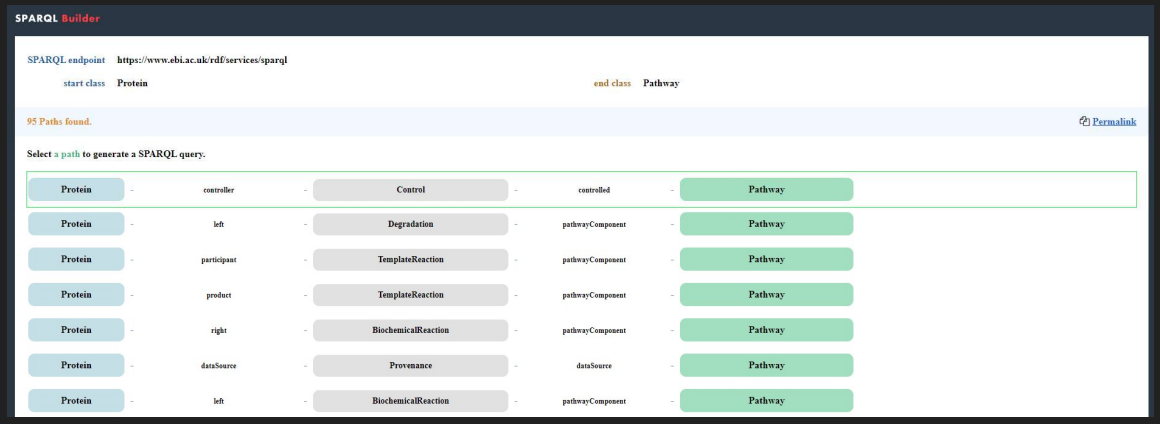
- Support the novice users in understanding the SPARQL language;
- Support the creation of meaningful queries.

Cons

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

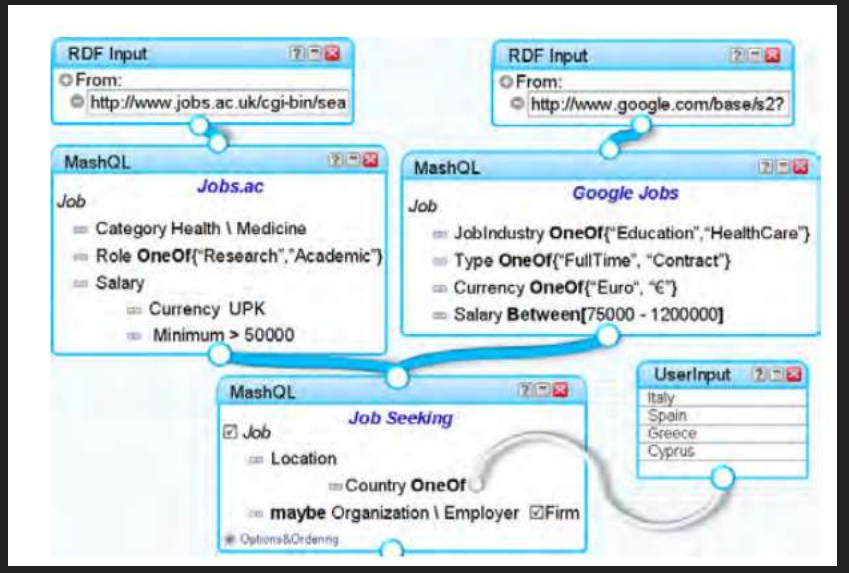
SPARQL Builder

- 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.

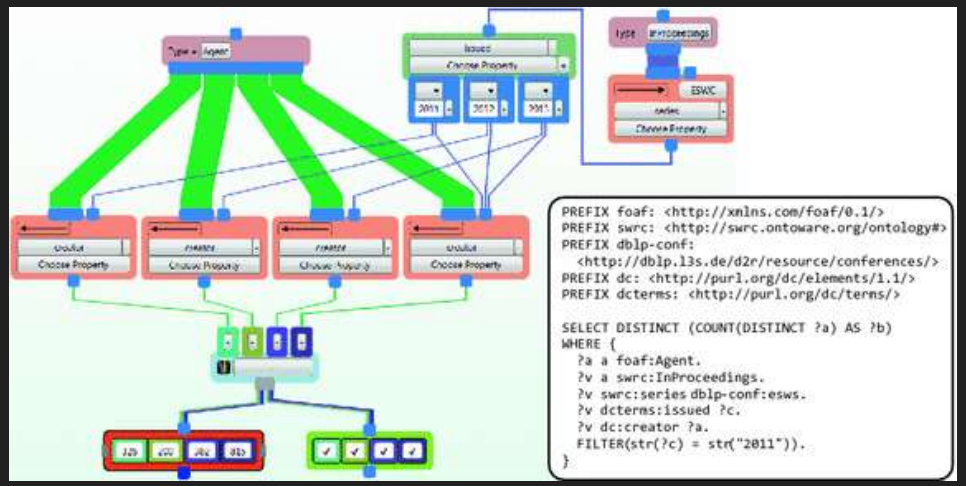


MashQL

- 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.



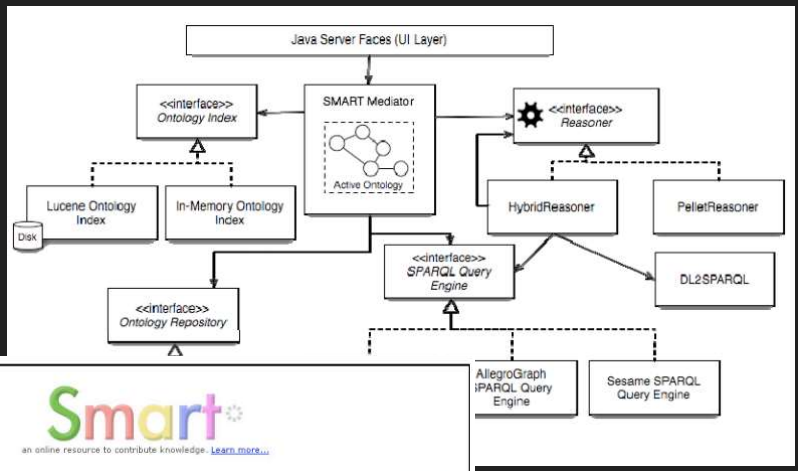
SparqlFilterFlow

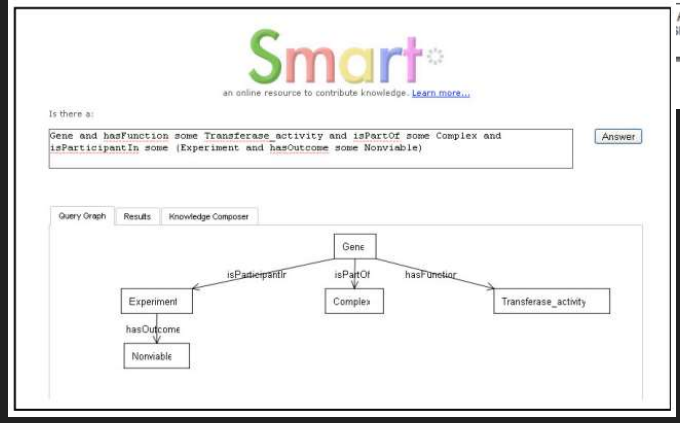


- An approach for visual SPARQL querying;
- Based on the concept of extended filter and flow graph;
- The queries can be created entirely with graphical elements.

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

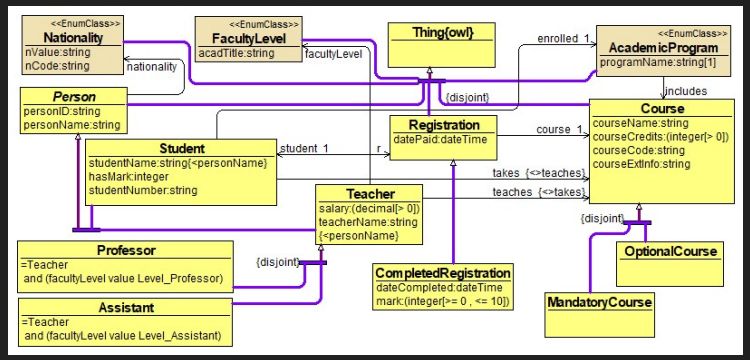
- Re-use of relational solutions;
- Presentation of the information in ways that users are familiar with;
- Support the understanding of the available information.

Cons

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

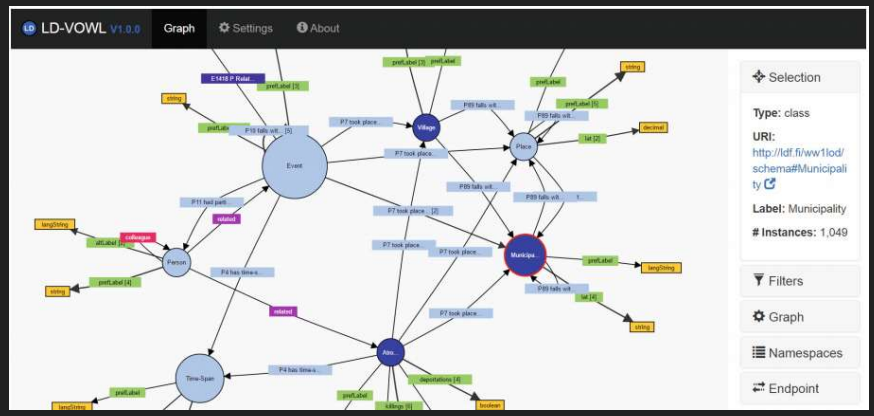
ViziQuer

- 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.

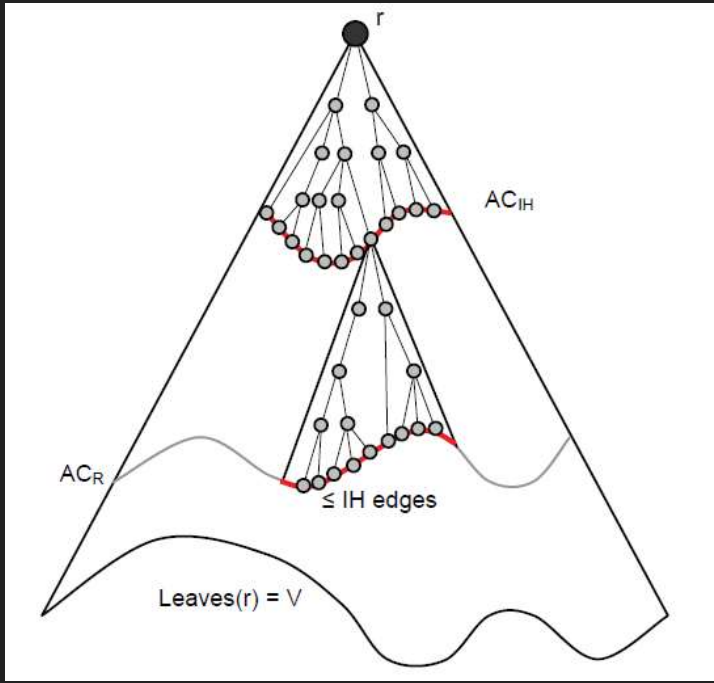
Pros

- 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.

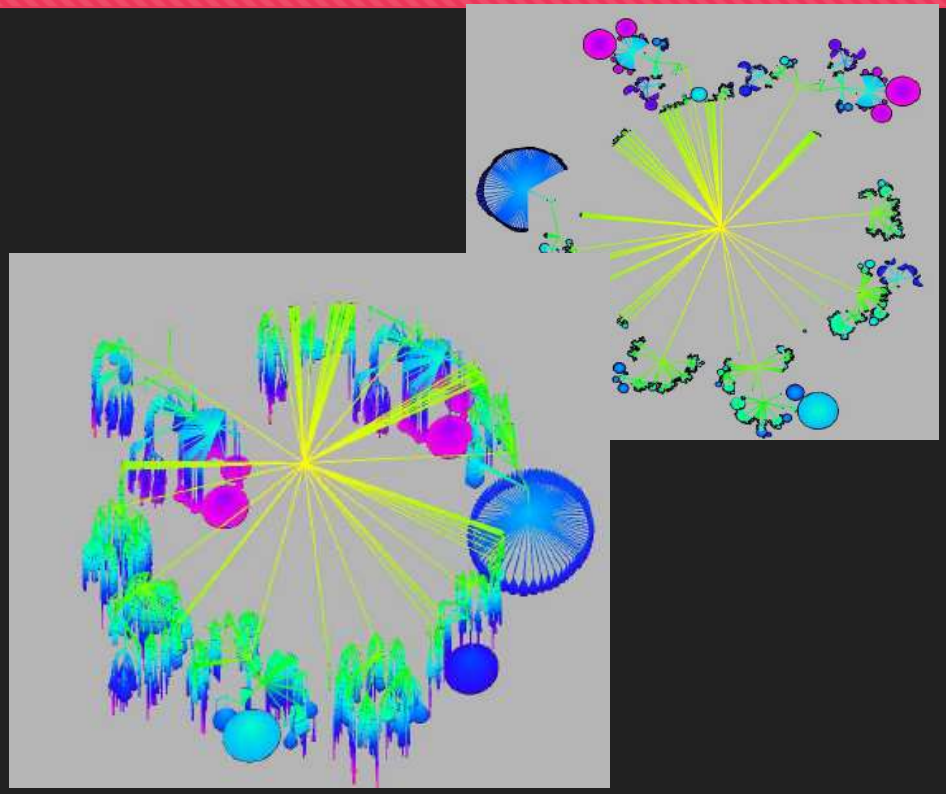
ASK- GraphView



- 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

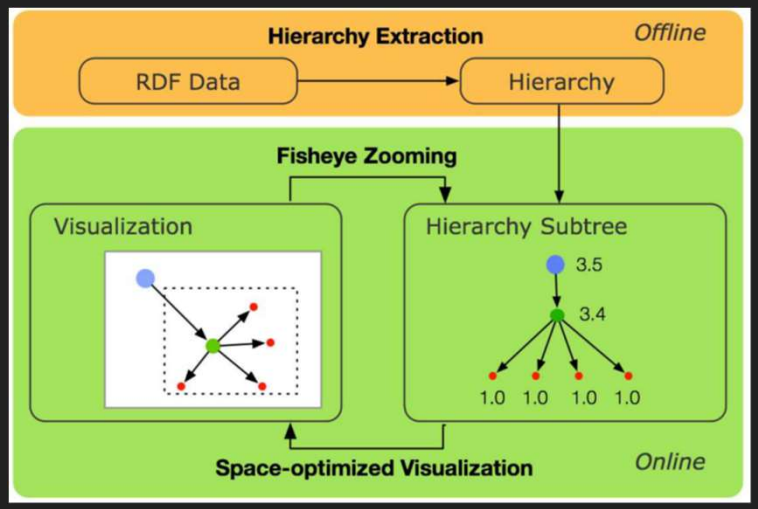
Tulip

- 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.

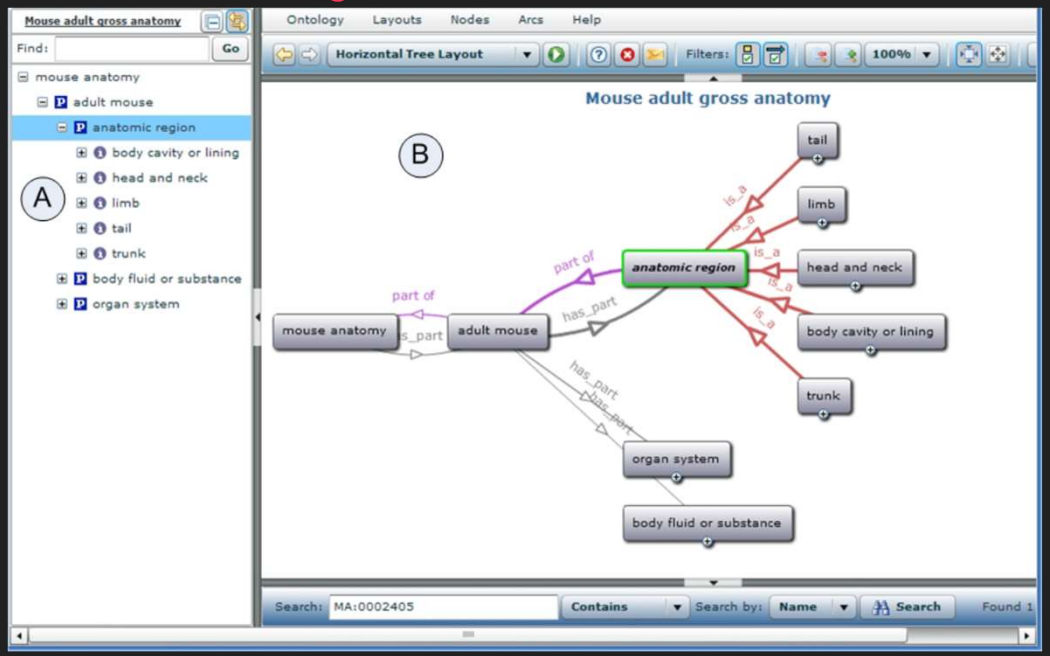


ZoomRDF

- 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.

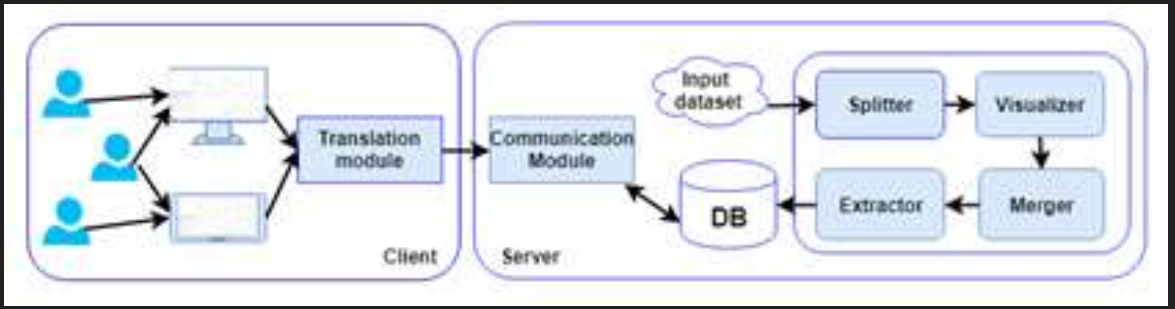


FlexViz



- 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@dblab.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.