BRAMBLE: A Web-based Framework for Interactive RDF-Graph Visualisation

Nikolas Schmitt, Mathias Niepert, and Heiner Stuckenschmidt

KR & KM Research Group
University of Mannheim

Abstract. Most graph visualisation tools for RDF data are desktop applications focused on loading complete ontologies and metadata from a file and allowing users to filter out information if needed. Recently both scientific and commercial frameworks have started to shift their focus to the web, however they still rely on plugins such as Java and rarely handle larger collections of RDF statements efficiently.
In this abstract we present a framework which visualises RDF graphs in a native browser environment, leveraging both the SVG standard and JavaScript technology to provide a responsive user interface. Graphs can be directly expanded, modified and explored. Users select nodes and edges from a central data repository containing millions of statements. The resulting graph can be shared with other users retaining full interactivity for collaborative work or presentation purposes.

1 Introduction

With the growth of datasets such as the Linked Open Data project[1], finding and presenting relevant information becomes an increasingly complex process. Graphs are well suited to browse and visualise linked data and a wide range of tools exist that provide graph drawing functionality on various platforms and in a myriad of formats. In particular there are commercial products[2] integrating graphs with web applications, allowing users to view and interact with a graph using their web-browser. So far these solutions rely on either a heavy server back-end generating the graph visualisation as images and sending them to the client, which imposes heavy restrictions with regard to interactivity of the graph, or they use a Java Applet or similar plugin which has to be downloaded first and does not run natively in the browser. The *WiGis* framework [1] shifts the entire workload onto the server and sends images to the browser at a constant frame rate after every user interaction. While this does solve some performance and scalability problems with large graphs, it makes the user interface feel slow and unresponsive since any actions taken by the user have to be evaluated by the server before there can be a response. All of the graph visualisation frameworks

[1] http://linkeddata.org/

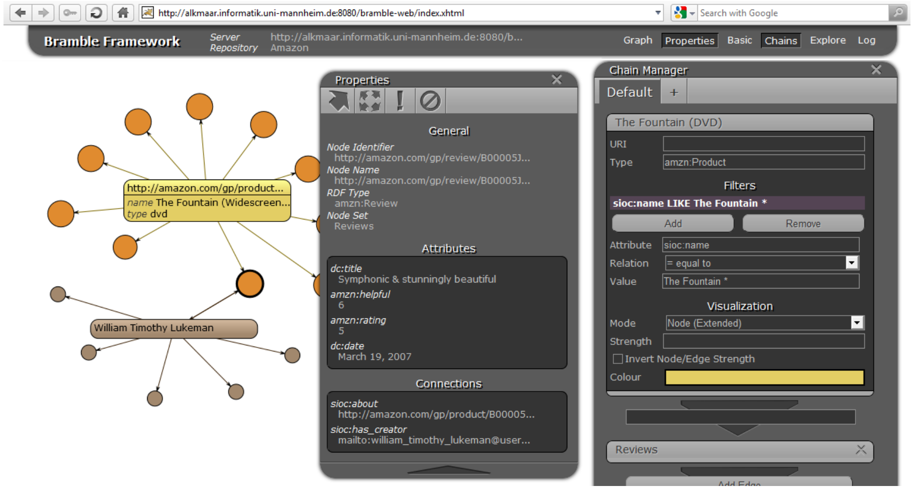
[2] Tom Sawyer Software: Tom Sawyer Visualization (http://www.tomsawyer.com)
TouchGraph: TouchGraph Navigator (http://www.touchgraph.com)

rely on the ”Overview - Filter” approach where the user is presented a graph over the entire collection of nodes and edges and is then expected to filter out the information he needs. This method has some inherent disadvantages: The whole graph has to be loaded and drawn up front. Thus frameworks employing this approach can only process a few hundred nodes and even those developed for large graphs have difficulties handling more than 10,000 nodes at an acceptable speed.

In this abstract we demonstrate how the *Bramble Framework*[3] implements web-based graph visualisation aiming for fully interactive graphs displayed in a native, browser-based environment. After the user has selected graph nodes and edges through a browser-based graphical user interface, the data is retrieved from a server-based RDF repository. Graphs are presented directly in a browser using Scalable Vector Graphics (SVG). The user may expand the graph by adding new nodes that are connected to existing nodes or explore the graph dynamically by moving from node to node. Attributes are displayed for nodes and edges and the user may influence their visualisation (colour, text) or delete them from the graph.

2 User Interface

After connecting to the data repository of choice, the user can utilise the *Chain Manager* dialog to retrieve specific information. The process is started by defining a *Node Set*: one or more entities from the datastore that share common features.



http://alkmaar.informatik.uni-mannheim.de:8080/bramble-web/index.xhtml

Server http://alkmaar.informatik.uni-mannheim.de:8080/b...
Repository Amazon

http://amazon.com/gp/product...
name The Fountain (Widescreen...
type dvd

William Timothy Lukeman

Node Identifier
http://amazon.com/gp/review/B00005J...
Node Name
http://amazon.com/gp/review/B00005J...
RDF Type
amzn:Review
Node Set
Reviews

dc:title
Symphonic & stunningly beautiful
amzn:helpful
6
amzn:rating
5
dc:date
March 19, 2007

sioc:about
http://amazon.com/gp/product/B00005...
sioc:has_creator
mailto:william_timothy_lukeman@user...

The Fountain (DVD)
URI
Type amzn:Product
sioc:name LIKE The Fountain *
Attribute sioc:name
Relation = equal to
Value The Fountain *
Mode Node (Extended)
Strength
□ Invert Node/Edge Strength
Colour

Fig. 1. The framework interface with Chain Manager and Properties dialog open.

[3] http://alkmaar.informatik.uni-mannheim.de:8080/bramble

Entities will be represented in the graph as nodes. Figure 1 shows a node set which contains entities whose `rdf:type` conforms to a product from the Amazon.com database and whose `sioc:name` matches the pattern ”The Fountain*”. Relevant classes and relations are suggested by the interface based on the RDF schema definitions present in the datastore. A set of nodes can be chained to another set either by specifying a relation that connects the two or leaving the field blank, in which case the framework will retrieve any relations between both sets. In Fig. 1 reviews (orange) are added to the graph by attaching a node set to the chain which automatically selects entities linked to the product (yellow).

To enhance the visualisation of the graph the user may customise the style of each individual node set and the edges that connect them. Customisation includes changing the colour and size of nodes, the strength of edges and adding textual descriptions to entities. The graph layout is determined by choosing one of the available algorithms such as the Kamada-Kawai [2] algorithm and optionally tweaking their parameters.

Once the graph is generated the user may select any graph element to retrieve additional information. Clicking nodes opens a property dialog displaying all links to other resources. These links can be added to the graph with their predicate becoming an edge and their object becoming a new node. Any graph element can be transformed into a more detailed version of itself, displaying information directly in the graph, or be removed from the graph.

3 Architecture and Implementation

The server-client architecture depicted in fig. 2 offers many advantages for a graph visualisation framework: Users upload data they want to analyse to a central repository. Any users with access to the repository can now use their browser to construct and customise graphs. Saving their work to the server allows them to continue editing from another computer or platform and sharing fully interactive graphs with other users.

A Java web service is used as the interface between the OpenRDF Sesame repository and the client web application. It connects to the repository, retrieves data and relays them to the web application using JSON. The web service takes *Chains* as input and translates them into SeRQL queries. The query results are parsed and transformed into node and edge objects which the web application can use to draw the graph.

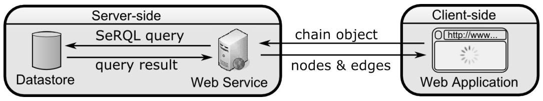


Fig. 2. Server-Client architecture with data query.

The web application is served as XHTML markup with a large collection of JavaScript aided by popular libraries such as jQuery. No server page technology is used, the communication to the server relies sololy on AJAX requests to the web service. The entire process of evaluating the data, applying custom style rules, calculating the graph layout using the algorithm of choice and drawing the graph is done on the client-side. The graph rendering engine generates the SVG markup (which uses XML syntax) and embeds scripts and triggers to enable interactivity as well as RDF metadata.

4 Use Case

In order to test and evaluate the framework an example repository has been created using the data gathered from the Amazon.com database by the Multi-Domain Sentiment Data Analysis project [3]. The repository contains millions of product reviews which provide information on how highly certain products are regarded by different groups of users.

The data on products from the category DVD make up for almost 100,000 reviews written by 50,000 users about 13,000 products. Visualising all the information in one graph is not only technically infeasible but also useless as a medium for research. The *Bramble Framework* enables analysts to search for information about one certain product and visualise how well it is received by users. Different styles can be used to identify quickly which customers are in favour of a product and which are not. Analysts are able to display additional information about a customer who dislikes a certain product and find out if and why another product is preferred. This leads to identifying strong market contenders and fields to improve the product in.

5 Conclusion

The *Bramble Framework* demonstrates how graphs can be visualised over the web while retaining full interactivity and a responsive user interface. Using SVG to render nodes and edges has several advantages over plugins such as Java or Adobe Flash: graphs can be exported as SVG to be modified on a per-element basis by external programs and scaled without loss of quality. In addition, metadata can be embedded into the XML markup and the technology is supported by all major browsers and platforms. The client-server approach with a web-based frontend makes data in existing web repositories more accessible. Furthermore, complex relationships in datasets consisting of millions of statements can be visualised using the frameworks flexible selection method.

References

1. B. Gretarsson, S. Bostandjiev, J. O'Donovan and T. Höllerer: WiGis: A Framework for Scalable Web-based Interactive Graph Visualizations. (2009)
2. T. Kamada and S. Kawai: An algorithm for drawing general undirected graphs. Inf. Process. Lett., 31, 7–15 (1989)
3. J. Blitzer, M. Dredze and F. Pereira: Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In ACL, 187–205 (2007)