Browsing Linked Data with Fenfire

Tuukka Hastrup
University of Jyväskylä
Jyväskylä, Finland
Tuukka.Hastrup@iki.fi

Richard Cyganiak
Digital Enterprise Research Institute
National University of Ireland, Galway
richard.cyganiak@deri.org

Uldis Bojārs
Digital Enterprise Research Institute
National University of Ireland, Galway
uldis.bojars@deri.org

ABSTRACT
A wealth of information has recently become available as browsable RDF data on the Web, but the selection of client applications to interact with this Linked Data remains limited. We show how to browse Linked Data with Fenfire, a Free and Open Source Software RDF browser and editor that employs a graph view and focuses on an engaging and interactive browsing experience. This sets Fenfire apart from previous table- and outline-based Linked Data browsers.

1. INTRODUCTION
Linked Data [1] is Semantic Web data that emphasises the graph of relations between resources while recognising that the data comes from Semantic Web documents that need to be retrievable using standard conventions. Domain-specific applications can crawl Linked Data to pull together and display information from various sources. However, there is a need for generic Linked Data browsers as well, as they help data producers to check what they publish and data consumers to check what is available.

One generic Linked Data browser is Tabulator [2], but it does not provide a graph view. Karger and schraefel provide insights on why graph views often are not the best views of Semantic Web data [4], but they acknowledge that graph views have a high "cool factor" and a niche. We consider a case that lies in this niche: we want to show the graph data as directly as possible. We apply a Free and Open Source Software rich desktop application, Fenfire[1], and its graph view to Linked Data browsing because unlike other browsers, this is a visually appealing, engaging and interactive demonstration of the Semantic Web's capabilities. A graph view is a good way to explore a web of information, and it is close to the nature of Linked Data as a heterogeneous, web-like environment with little high-level structure.

In the following, we demonstrate how the Fenfire application provides a useful Linked Data browsing experience, go through an example, describe the implementation and conclude with some future directions.

[1]`http://fenfire.org/`

2. BROWSING EXPERIENCE
The Fenfire application is a generic RDF browser and editor with features useful for Linked Data browsing. The user interface employs the conventional graph representation of the RDF data model. To make the visualisations scalable in the number of nodes in the graph and to focus on one thing at a time, only one central node and its surroundings are displayed concurrently. It is possible to switch between two views implemented based on the concept: a simple list view of objects associated with the focused subject (say, a container), and the generic graph view from Fentwine [3].

A browsing session starts from some URI which is retrieved for a document with RDF data. This URI will be the initial focus unless the document has a `foaf:primaryTopic` defined, in which case the primary topic URI will be chosen as the initial focus.

For the surroundings, the generic graph view shows to the left of the focus all triples that have the focus as an object, and to the right all triples that have the focus as a subject. Each triple is shown as a predicate connecting the subject to the object. This view is applied recursively to each displayed node until, with distance, the graph fades away to the background. If the node has an `rdfs:label`, it is displayed instead of the URI.

Graph navigation can be done entirely via keyboard by rotating the surrounding nodes around the centre and moving focus to the node immediately to the left or right of the centre. While there is limited space for the surrounding nodes, all nodes can be navigated to via the rotation.

To enable browsing of Linked Data, Fenfire dereferences the URI of the focused node and retrieves any `rdfs:seeAlso` related to this node whenever instructed to do so. For example, in Figure 1 a user has loaded the FOAF profile of one person, followed a `foaf:knows` link to another person and retrieved the FOAF profile of this person.

As an important alleviation of incompletely linked data, Fenfire adds triples asserting that the graph retrieved contains information about all of its disconnected components.

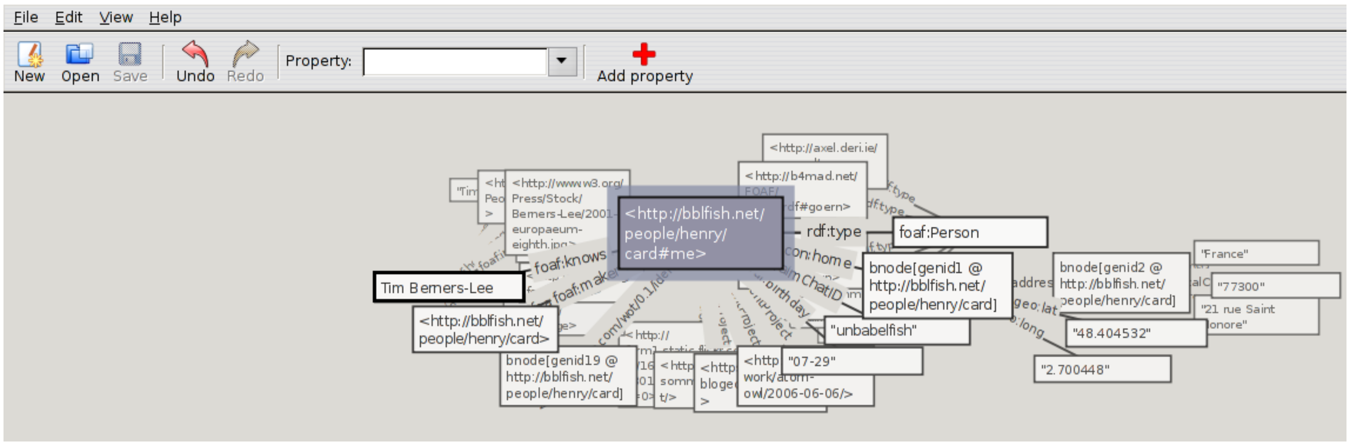
File Edit View Help
New
Open
Save
Undo
Redo
Property:
Add property
<http://bblfish.net/people/henry/card#me>
rdf:type
foaf:Person
Tim Berners-Lee
foaf:knows
foaf:maker
<http://bblfish.net/people/henry/card>
bnode[genid1 @ http://bblfish.net/people/henry/card]
bnode[genid2 @ http://bblfish.net/people/henry/card]
bnode[genid19 @ http://bblfish.net/people/henry/card]
"unbabelfish"
"48.404532"
"2.700448"
"France"
"77300"
"21 rue Saint Honore"
"07-29"
<http://axel.deri.ie/
<http://b4mad.net/

Figure 1: **Starting from Tim Berners-Lee, a user has followed FOAF data to a document about Henry Story.**

3. USE CASES

We target two audiences primarily:

- Semantic Web researchers, application developers and data producers need to explore available data on the level of individual triples. Fenfire provides a convenient alternative to manually downloading graph documents, reading the serialisation formats, and trying to match URIs to discover the links in the data.

- People who want to learn about or demonstrate the Semantic Web and what data is available benefit from a visual presentation that truthfully shows the networked nature of the data. Here it is highly advantageous that Linked Data documents that follow the guidelines include an `rdfs:label` for each node.

4. IMPLEMENTATION

Fenfire has its origins in ZigZag [6], which is a system for managing interlinked, distributed data and is completely independent of the World Wide Web standards. Fentwine [3] incorporated Semantic Web standards and became a graph-view RDF browser, as did its sibling BuoyOING [5] (Buoy-Oriented Interface, Next Generation) user interface, which adds spatial navigation and concentrates on it.

Fenfire is implemented in the Haskell programming language to achieve high programmer productivity while meeting performance requirements for real-time animation and for browsing large graphs. It uses the Raptor library of the Redland project for RDF parsing, the GTK library for the standard graphical user interface elements and the Cairo library for the animated, vector-graphics based visualisations.

A major part of the architecture is the key-frame and identity-based visualisation and animation system that provides the user continuous feedback on how the navigation and the switching between alternative views change what is visible.

5. CONCLUSIONS

We applied Fenfire to the task of browsing Linked Data with a graph view and highlighted the use cases of exploring, learning about and demonstrating Semantic Web data.

Some Linked Data browsers can edit the data as well. Fenfire can edit RDF graphs, but we need to implement remote publishing interfaces in addition to the current feature of saving to local files.

Information display and navigation can be enhanced if there is knowledge about the user interaction requirements for a specific domain. Thus, Fenfire should have some automatic ontology-awareness and more should be configurable with settings and plug-ins.

6. ACKNOWLEDGEMENTS

The implementation of Fenfire in Haskell was initially developed by Benja Fallenstein and Tuukka Hastrup.

7. REFERENCES

[1] T. Berners-Lee. Design Issues–Linked Data. Published online, May 2007. `http://www.w3.org/DesignIssues/LinkedData.html`.

[2] 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 *Proceedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06)*, Nov 2006.

[3] B. Fallenstein. Fentwine: A navigational RDF browser and editor. *Proceedings of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF Galway)*, Sep 2004.

[4] D. R. Karger and m.c. schraefel. The Pathetic Fallacy of RDF. In *Proceedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06)*, Nov 2006.

[5] J. Kujala and T. Lukka. Rendering recognizably unique textures. In *Proceedings of the 7h Internation Conference on Information Visualization, 2003. IV 2003.*, pages 396–405, Jul 2003.

[6] T. H. Nelson. A Cosmology for a Different Computer Universe: Data Model, Mechanisms, Virtual Machine and Visualization Infrastructure". *Journal of Digital Information*, 5(1), Jul 2004.