LibSwb: Browsing the Entity Context*

Sven Buschbeck, Stefano Bortoli, and Heiko Stoermer

University of Trento,
Dept. of Information Science and Engineering,
Trento, Italy
{buschbeck, bortoli, stoermer}@disi.unitn.it

Abstract. In this paper we are presenting ongoing work on a software library for on-the-fly browsing of Semantic Web content that forms the informational *context* around an entity. This library is targeted to be integrated into user applications, to extend their range of functionalities in an easy-to-handle and unobtrusive way. We illustrate an application, give details on the implementation of a first prototype, and evaluate its feasibility.

1 Introduction to Entity-centric Semantic Web browsing

One of the most important differences between the Semantic Web and WWW is the concept of the URI as the global anchor for information about a resource in distributed information sources. Unlike the WWW, a URI for a resource can be completely arbitrary, and does not need to expose *any* reference to actual data.The paradigm we are trying to promote and explore is that at the center of attention we do not place documents, but *entities*, i.e. people, events, locations, organizations, and more.

Starting from an entity, we want to enable users to browse data in an innovative way, using the entity's URI (provided through the OKKAM Entity Name System [2]) as a pivot for semantic data (the "entity context"), which we gather from distributed sources. We call this approach "entity-centric browsing".

The idea is to define a plugin for a browser enabling the combination of World Wide Web and Semantic Web information as shown in figure 1. A sketch of a possible workflow of this application can be resumed in the following steps: (1) a user selects a set of keywords in a web page triggering the retrieval of the URI of the "identified pivot entity" through the OKKAM ENS; (2) all RDF documents containing information about the pivot entity are retrieved, merged and used to extend the context menu related to the highlighted entity; (3) optionally, the user can start browsing the Semantic Web graph following entity-centric logical connections without document boundaries; (4) if the user finds an interesting resource, it's URI is used to retrieve URLs presenting references to documents about that entity and, by selecting one of them, the user can switch back to WWW browsing.

* This work is partially supported by the FP7 EU Large-scale Integrating Project **OKKAM – Enabling a Web of Entities** (contract no. ICT-215032). For more details, visit http://www.okkam.org.

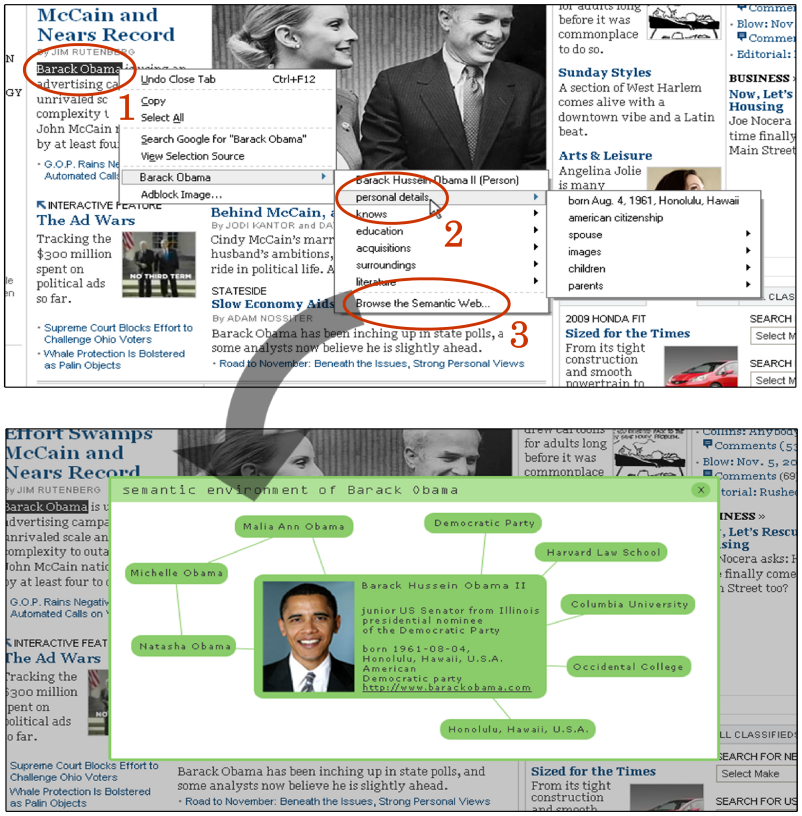

Fig. 1. Mockup of an entity-centric browsing application.

The important fact to note is that the data displayed in the entity context is automatically retrieved and joined *on-the-fly*. To a certain extent, approaches to the Semantic Web browsing topic are available. For example **Piggy Bank** [3] collects semantic data from web pages and allows users to make annotations and share them among each other. **Haystack** [6] is a standalone Semantic Web browser application collecting semantically annotated information from URLs and present it in a generic, content independent interface. **Potluck** [4] produces data mash-ups on the basis of different URLs provided by the user. **Tabulator** [8] is a well-known Semantic Web browser displaying RDF and allowing the user to follow hypertext links, if they point to another RDF document. All approaches mainly rely on URLs to information sources, thus either on a single source or a list of sources. LibSwb instead automatically retrieves information about an entity by querying for OKKAM identifiers in distributed data sources. Therefore it provides a richer set of information, and additionally is not as error-prone as relying on direct dereferencing of an URL.

2 Solution Approach

The idea behind our work is to enable Semantic Web browsing, replicating the behavior of a WWW browser hiding the complexity of the ongoing operations. Semantic Web browsing should consist in moving through the "Giant Global Graph" [1] by presenting a bounded view (a contextual subgraph) focusing on one pivot entity at each browsing step.

A brief description of the operations necessary to realize such a browsing process is presented in the following. (1) *Define starting pivot URI*; (2) *Retrieve pivot URI alternatives*; (3) *Retrieve information about the pivot URI* by inquiring data sources as Sindice [9] looking for RDF documents including the pivot URI and its alternatives; (4) *Building a partial RDF graph* by merging and pruning the retrieved RDF documents.

We implemented a prototype of LibSwb aiming to define "proof of concept", enabling the execution of some experiments. This implementation relies on the Jena framework, combined with standard graph manipulation algorithms for RDF graph management, and the Sindice index for RDF document retrieval.

In this very early state of development, we can only provide a rough approximation about runtime behavior. LibSwb has been tested by browsing the entity context of the authors' FOAF-profiles using the authors' OKKAM URIs to retrieve information. Three steps of browsing have been performed as scenario involving 11 documents with 62kB and 310 RDF statements. By running this test scenario on a Pentium 4 computer (3Ghz, 3GB RAM) an average runtime of 6847ms was determined, where the machine was occupied 6827ms for downloading the RDF files, which leaves a total of 20ms for computations, i.e. in average 6.6ms per browsing step.

The results reveal that the most time expensive activity in realizing on-the-fly entity-centric browsing is downloading of RDF files. Hence, we will focus on reducing download time on the one hand and use more sophisticated and complex algorithms for merging and pruning to improve results on the other hand.

3 Conclusion

As previously stated, the work described in this article is in an early stage of development. While improving performance, e.g. by parallelizing processes and avoiding file reloading through caching, we are also going to extend LibSwb in the direction of *contextual representation and management*[1], to lay the foundations for several context-related aspects, including the limitations of graph merging, provenance and quality of data, as well as contextual visual representation of information. The next evolution will not rely on Sindice alone, but provide a pluggable architecture for information sources like Falcon [5], Microformats[2]

[1] Context-sensitivity of RDF data has been addressed on several levels, we refer the reader to [7] and the contained Related Work section.

[2] http://microformats.org/

and emering sources. Additionally, we will *consider references to further RDF data* embedded in documents. In this way, we are moving towards the vision of the W3C Semantic Web research group about RDF browsing[3].

This highly flexible architecture will make LibSwb easily usable within many kinds of applications that require a convenient interface to Semantic Web content, such as authoring systems that need to provide up-to-date background information about entities, or extensions to web browsers that add a navigational layer on top of document hyperlink structures. Our declared goal is to ease the development of entity-centric (Semantic) Web enabled applications and extend the functional range of networked applications. We are thus planning to make the library publicly available as soon as the most important features mentioned in the previous section are implemented and sufficiently stable.

References

[1] Tim Berners-Lee. Giant global graph. *Decentralized Information Group - http://dig.csail.mit.edu/breadcrumbs/node/215*, 2007.

[2] Paolo Bouquet, Heiko Stoermer, Claudia Niederee, and Antonio Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008*, number CSS-ICSC 2008-4-28-25, pages 554–561. IEEE Computer Society, August 2008.

[3] David Huynh, Stefano Mazzocchi, and David Karger. Piggy bank: Experience the semantic web inside your web browser. In *Proceeding of 4th International Semantic Web Conference ISWC2005*, pages 413–430. Springer-Verlag, 2005.

[4] David F. Huynh, Robert C. Miller, and David R. Karger. Potluck: Data mash-up tool for casual users. *The Semantic Web*, pages 239–252, 2008.

[5] N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-ao: Aligning ontologies with falcon. In *K-CAP Workshop on Integrating Ontologies, Banff, Canada.*, 2005.

[6] Dennis Quan. How to make a semantic web browser. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20*, pages 255–265. Springer-Verlag, 2004.

[7] Heiko Stoermer, Paolo Bouquet, Ignazio Palmisano, and Domenico Redavid. A Context-based Architecture for RDF Knowledge Bases: Approach, Implementation and Preliminary Results. In Massimo Marchiori, Jeff Z. Pan, and Christian de Sainte Marie, editors, *Web Reasoning and Rule Systems, First International Conference, RR 2007, Innsbruck , Austria, June 7-8, 2007, Proceedings*, volume 4524 of *Lecture Notes in Computer Science*, pages 209–218. Springer Berlin/Heidelberg, June 2007.

[8] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceeding The 3rd International Semantic Web User Interaction Workshop November 6, 2006 Athens, Georgia, USA*, 2006.

[9] Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. In *Proceedings of the 7th International Semantic Web Conference 2007 (ISWC'07), Busan, Korea*, 2007.

[3] i.e. Tabulator RDF Browser (http://www.w3.org/2005/ajar/tab)