

# Ozone Browser: Augmenting the Web with Semantic Overlays

Grégoire Burel<sup>1</sup>, Amparo E. Cano<sup>1</sup> and Vitaveska Lanfranchi<sup>1</sup>

<sup>1</sup> OAK Group, Department of Computer Science, University of Sheffield,  
Sheffield, United Kingdom  
{G.Burel, A.Cano, V.Lanfranchi}@dcs.shef.ac.uk

This paper presents Ozone Browser<sup>1</sup> (OB), a JavaScript tool that uses the semantics embedded in Web documents for improving the user interaction experience.

Recently, the introduction of Microformats<sup>2</sup> and then RDFa<sup>3</sup> created a bridge between the Semantic Web and the Syntactic Web by embedding semantics into Web documents. This bridge facilitates the consumption of semantic data by third party applications such as screen scrappers, or search engines. However, until now the human-readable presentation of these semantic meta-information has been set aside. From the reader's point of view, the actual interpretation of a rendered document relies on: 1) The contextual knowledge provided by the hyperlinks contained in a document; 2) The background knowledge of the reader that helps him to bind unlinked information to known concepts. However, when following hyperlinks the reader may be disturbed on its cognitive process as it loses the information given by the previous document. Moreover, the association process with the user knowledge rather than the publisher knowledge may conduct to misinterpretation or partial understanding of the document. As computers, a reader should use the document's internal and external semantics for leading him to a better understanding of it. For making it possible, it is required to augment the document's representation by making explicit this underlying and external knowledge in a human readable way. OB is a graphical overlay that uses RDFa enabled documents for fulfilling these requirements. It uses a progressive enhancement strategy based on RDFa markups for adding different presentation layers to an existing document.

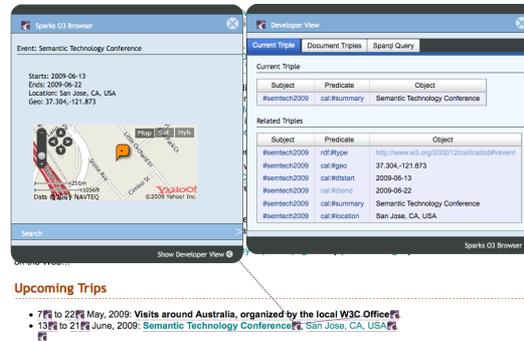
First, OB extracts the semantic content from a document by creating a lightweight client-side triple store from which inferences from the document's RDFa statements can be derived. Then, it binds a graphical overlay on the top of the current web document, highlighting all the RDFa statements containing the *property* tag in the document via icons. The reader can request more information about these statements by pointing on them. When doing so, inferences from inside and outside the page are executed for that statement, a contextualized view of the obtained knowledge is presented and the related information within the document (statements with the same predicate) is visualised as lines superimposed on the document itself. In order to provide these visualisations, OB dynamically generates views accordingly to the name space of the RDFa statement using a plug-in approach (Fig. 1).

---

<sup>1</sup> Sparks Ozone Browser, <http://www.dcs.shef.ac.uk/~gregoire/sparks/>

<sup>2</sup> Microformats, <http://microformats.org/>

<sup>3</sup> RDFa, <http://www.w3.org/TR/xhtml-rdfa-primer/>



**Fig. 1.** Ozone Browser in action: a) the summary of an event; b) the related developer's view.

Currently two experimental views are implemented. The first one provides a view of a Twitter<sup>4</sup> account (last Tweets of the considered account and the last public Tweets). The second one, an Event Visualiser, displays a summary of geotagged events using a map interface. This summary is generated from the inferences derived from the statements contained in the triple store. When no suitable views are available, a default visualisation is provided. It displays the RDFa triple object alongside with the possibility of finding the related information using search engines. OB also provides a *developer view* targeting people aware of Semantic Web technologies. This view shows the triples of the document in a raw format and offers an experimental interface for performing simple SPARQL<sup>5</sup> queries on the triple store.

OB is implemented as a bookmarklet<sup>6</sup> and can be applied to any web document with RDFa embedded metadata. It is built on top of two toolkits and uses a JSONP<sup>7</sup> bridge from *Yahoo Pipes*<sup>8</sup>. It uses the *rdfquery library*<sup>9</sup>, which is a lightweight client-side toolkit for RDFa, and the *Sparks framework*<sup>1</sup>, both implemented in JavaScript.

OB has been designed within the WeKnowIt<sup>10</sup> project as a solution for interacting implicitly with knowledge outside and inside the document boundaries, while preserving the knowledge context. Future work includes: 1) Refining the context-based views and introducing more sophisticated visualisation plug-ins; 2) Providing means to manipulate the knowledge objects in the page, thus offering easy ways to enrich a document with the user knowledge.

### Acknowledgements

The research leading to these results has received funding from the EU project WeKnowIt<sup>10</sup> (ICT-215453).

<sup>4</sup> Twitter, <http://twitter.com/>

<sup>5</sup> SPARQL, <http://www.w3.org/TR/rdf-sparql-query/>

<sup>6</sup> A bookmarklet consists of a snippet of JavaScript code that can be dragged/dropped into the browser's tool bar and can be applied to any web page in order to process a Web document.

<sup>7</sup> JSONP, <http://bob.pythonmac.org/archives/2005/12/05/remote-json-jsonp/>

<sup>8</sup> Yahoo Pipes, <http://pipes.yahoo.com/pipes/>

<sup>9</sup> Rdfquery Library, <http://code.google.com/p/rdfquery/>

<sup>10</sup> WeKnowIt, <http://www.weknowit.eu/>