Balloon Synopsis: A jQuery plugin to easily integrate the Semantic Web in a website*

Kai Schlegel, Thomas Weißgerber, Florian Stegmaier,
Michael Granitzer and Harald Kosch

University of Passau, Germany
Innstrasse 41, 94032 Passau
{forename.surname}@uni-passau.de

Abstract. The Semantic Web grows constantly and promises a huge amount of machine-interpretable information. Unfortunately, the integration and usage of semantic information is not feasible by everyone. Hence, a large number of Semantic Web applications are lacking and the potential of the semantic knowledge remains unexploited. We propose *balloon Synopsis*, an easy-to-use jQuery plugin to integrate Semantic Web information in a website. It provides a modern visualisation and browser for RDF information including automatic remote information enhancing, similarity analysis and ontology templates. *Balloon Synopsis* enables web developers to rely on known tools and programming language to benefit of the global knowledge graph.

1 Easily integrate semantic information in a website

The Semantic Web is evolving fast. Many different knowledge bases produce a vast amount of RDF data, leading to the initial idea of a global knowledge graph. Semantics are even an important topic for modern web developers. RDFa [1] and microformats[1] allow developers to specify structured metadata and semantic content within their webpages. Usually, search engines and web crawlers extract this information . But web developers could exploit the fact, that RDF data is interlinked with other online resources and provide a richer browsing experience for users by integrating related and context information. While there are many tools available to present RDF data to a user [2–4], most of them are not available in web-browsers and use common node-link layouts, which are very space consuming and hinder an easy integration in a website. This paper demonstrates a modern visualization and browser for RDF data, which is intended to be easily integrated in webpages. *Balloon Synopsis* was developed to allow web developers to rely on known tools and programming language to enable a simple access and benefit of the global knowledge graph.

The key-highlights of this approach include *(i)* a human friendly presentation of RDF utilizing a *node-centric layout* with *(ii) ontology templating* and *(iii) semantic colors* for similar entities. The remaining paper discusses these in more detail.

* The presented work was developed within MICO and EEXCESS projects partially funded by the EU Seventh Framework Programme, grant agreement number 610480 and 600601.

[1] http://microformats.org/

2 Overview

Balloon Synopsis features a tile-based visualization for RDF data and is inspired by modern operating systems and design trends. Instead of presenting the graph based nature of RDF by viszualizing common node-link layouts, *Balloon Synopsis* pursues an approach of node-centric visualization. As a result, the content and direct related information of an entity are focused rather than showing a global and complex context. Figure 1 shows a screenshot of *balloon Synopsis*, visualizing information and relations about the DBpedia entity `http://dbpedia.org/resource/University_of_Passau`.

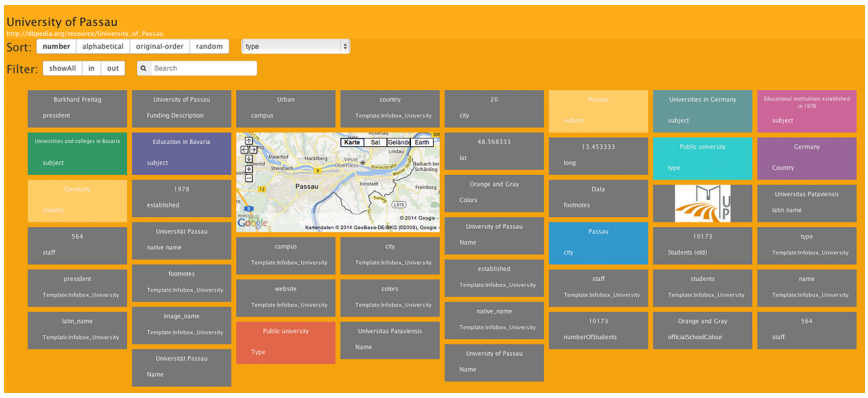

Fig. 1. Viewing an example entity with enriched information and ontology templating

This view allows browsing and discovering further and detailed information about all related entities by clicking on the according tile. At any time, the user has the possibility to sort, filter and search (using keywords and regular expressions) the displayed tiles to gather information according to the need of a user. To cope with an overloaded visualization, *balloon Synopsis* offers the possibility to process multiple pieces of information and aggregates them into a single tile to generate human-friendly representation of the data. As an example, Figure 1 shows an instance of Google Maps[2] instead of longitude and latitude information. These customizable ontology templates are discussed in detail in section 3. Based on its configuration, *balloon Synopsis* can either work with local RDF data or can automatically query a configured remote SPARQL endpoint to enhance the current view. Therefore, *balloon Synopsis* can be used as a local RDF viewer or as a Linked Data browser.

3 Ontology Templates

Typically a tile represents low-end information like resources, blank nodes or literals. This could result in overloaded visualizations, because a specific entity can be included in many different triples. To overcome this issue, *balloon Synopsis*

[2] https://www.google.com/maps/

features customizable filters, referred to as ontology templates. A developer can easily integrate custom ontology templates by means of a JavaScript function and a HTML snippet using handlebars syntax.[3] The custom filter also has access to the local store to load additional information or alter existing tiles. As an example, listing 1.1 shows the implementation of showing Google Maps instead of longitude and latitude coordinates. The developer has to provide a JavaScript function, which iterates through all current available nodes. A node represents the information, which are in a tile - in general it's the predicate and object of a triple. In the example, only the predicate `http://www.georss.org/georss/point` is focused for the Google Maps ontology template. The coordinates for the map can be extracted easily, because the GeoRSS point[4] contains a single latitude-longitude pair separated by a whitespace (e.g. "48.573 13.456"). As last step, the developer has to provide a HTML template to generate the desired visualization. The just created JavaScript variables `node.lat` and `node.long` are accessible using the handlebars syntax.

```
maps: {
  fn: function(plugin, nodes, config) {
    $.each(nodes, function(i, node) {
      if(node.predicates[0].value === 'http://www.georss.org/georss/point') {
        node.lat = node.value.split(" ")[0];
        node.long = node.value.split(" ")[1];
      }
    });},
  template: '<iframe src="maps.google.com/?ll={{node.lat}},{{node.long}}"/>'
}
```

Listing 1.1. Google Maps Ontology Template

Balloon Synopsis comes with some preconfigured ontology templates. For example there is a *Merging* template which combines equal predicates and nodes in a common tile or a *Blacklisting* template which excludes specific predicates or entities. Besides the aforementioned *Google Maps* template there is also a *City Weather Chart* template. The *City Weather Chart* template is shown in figure 2 and combines DBpedia month-based weather information like temperature, humidity or precipitation of cities in a Google Chart[5].

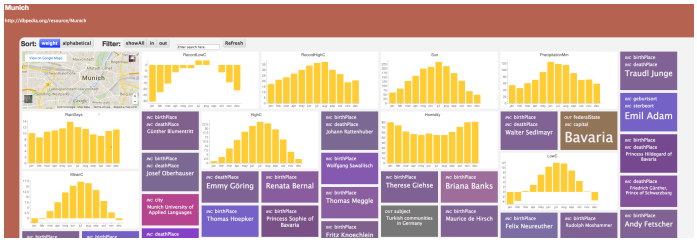

Fig. 2. City Weather Chart template combines weather information of cities in a Google Chart

[3] http://handlebarsjs.com/
[4] http://www.georss.org/
[5] https://developers.google.com/chart/

4 Semantic Colors

The next step to counteract the complexity of the large amount of information is to create uniform colors for similar pieces of information. The user should be able to recognize basic characteristics and similarity between the tiles at one glance. Motivated by the idea of calculating color for a resource in a deterministic way by "Color the Linked Data Web"[6], *balloon Synopsis* implemented the functionality that similar entities get similar background colors in the according tiles. Figure 3 illustrates the basic idea of semantic colors. Human entities like athletes, politicians or writers have a similar purple background, whereas geographic points have reddish colors.

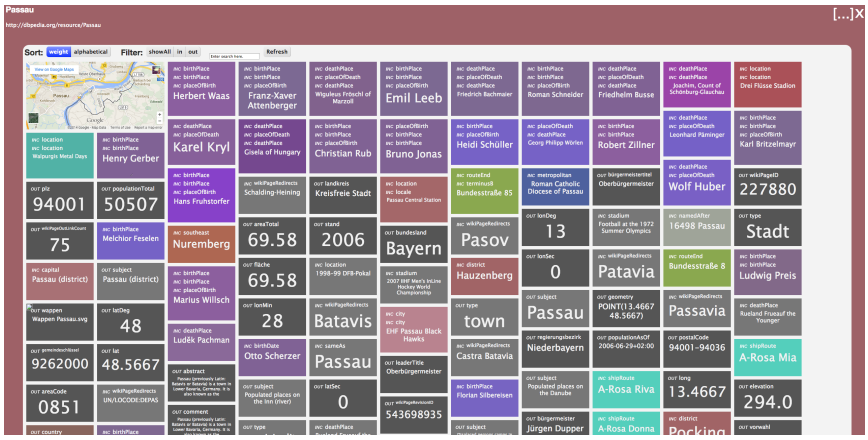

Fig. 3. Semantic colors: Similar entities get similar background color

For this purpose, the `rdf:type` relationships of each entity are fetched automatically, either from a configured remote endpoint or by using the *balloon Commonalities* type index API.[7] A color for each `rdf:type` is basically calculated by hashing the URI and using the first 6 characters of the resulting hash as a RGB HEX color. Afterwards, the resulting colors of the different types are blended to get a single color to represent the entity. In this process, scheme information in the `owl` or `rdf(s)` namespace are not inlcuded, because these basic information would distort the resulting color. Early results showed up, that merging colors in the well-known RGB color-space didn't result in vibrant colors and mostly produced gray and muddy colors. In the case of *balloon Synopsis* the generated color should maintain color saturation and correspond to the human perception of similarity. As a consequence, the perceptual-based CIE-Lab color

[6] http://cold.aksw.org/
[7] http://schlegel.github.io/balloon/balloon-commonalities.html

space appears to be the most suitable for this task. Building upon the chroma.js library[8], the applied color-space can be configured easily.

5 Internals

Balloon Synopsis features a HTML and JavaScript implementation and is available as jQuery-plugin. At its core, balloon Synopsis uses a local SPARQL capable RDF store.[9] Besides common RDF serializations like Turtle, N3 or JSON-LD, *balloon Synopsis* offers the possibility to specify a SPARQL endpoint and SPARQL query to load remote RDF data portions. In the whole project, cross domain problems are circumvented by using CORS [5] or YQL[10] as backup, to enable remote data querying on the client. To give more detailed information, Figure 4 highlights essential internal components.

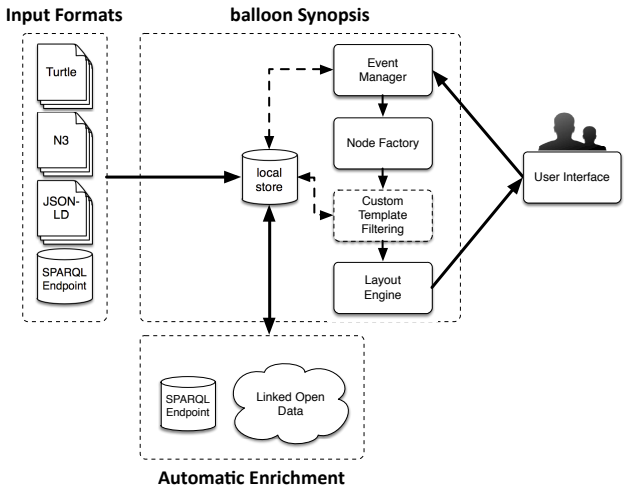

Fig. 4. Conceptual overview of internal processing steps

By clicking on a tile or programmatically calling a entity view, the **user interface** invokes the **event manager** to show a specific detail view. The event manager is supposed to query the local store for information about the desired entity. *Balloon Synopsis* offers an **automatic enrichment** of the local RDF data by querying *(i)* remote SPARQL endpoints or *(ii)* performing an query federation over Linked Data endpoints utilizing the recently introduced *balloon*

[8] http://driven-by-data.net/about/chromajs
[9] https://github.com/antoniogarrote/rdfstore-js
[10] http://developer.yahoo.com/yql/

Fusion service [6]. The event manager transmits all results to the **node factory**, which transforms the RDF data to corresponding JavaScript components to represent the content of a basic tile. These components are then forwarded to a **ontology template filtering**. Besides aforementioned pre-packed templates to simplify the view, a developer can easily integrate custom ontology templates. The integration of Web Workers [7] is currently planned to speed up the processing of several parallel templates. The final layout is then computed by the **layout engine**, which can influence the ranking, scale and color of tiles based on importance or similarity (e.g. scheme information have a low priority and similar entities can be clustered by their semantic color). In addition, the user interface itself can affect the layout due to searching, sorting or responsive design events (e.g. resizing or panning). Integrated layouting mechanisms, building upon isotope[11] and Shuffle[12], enable flexible and responsive reordering of the view with animated transitions.

6 Conclusion

Balloon Synopsis features an open source jQuery plugin to integrate Semantic Web information in a website. Web developers can easily embed RDF information to offer a richer browsing experience for users and benefit of the global knowledge graph. A demonstration as well as the sources are available at GitHub.[13]

References

1. Birbeck, M., Adida, B., McCarron, S., Herman, I.: RDFa core 1.1 - second edition. W3C recommendation, W3C (August 2013) http://www.w3.org/TR/2013/REC-rdfa-core-20130822/.
2. Camarda, D.V., Mazzini, S., Antonuccio, A.: Lodlive, exploring the web of data. In: Proceedings of the 8th International Conference on Semantic Systems. I-SEMANTICS '12, New York, NY, USA, ACM (2012) 197–200
3. Sayers, C.: Node-centric rdf graph visualization. Mobile and Media Systems Laboratory, HP Labs (2004)
4. Pietriga, E.: Isaviz: a visual environment for browsing and authoring rdf models. In: Eleventh International World Wide Web Conference Developers Day. (2002)
5. van Kesteren, A.: Cross-origin resource sharing. W3C recommendation, W3C (January 2014) http://www.w3.org/TR/2014/REC-cors-20140116/.
6. Schlegel, K., Stegmaier, F., Bayerl, S., Granitzer, M., Kosch, H.: Balloon Fusion: SPARQL Rewriting Based on Unified Co-Reference Information. In: 5th International Workshop on Data Engineering Meets the Semantic Web, co-located with the 30th IEEE International Conference on Data Engineering. (2014)
7. Hickson, I.: Web workers. Candidate recommendation, W3C (May 2012) http://www.w3.org/TR/2012/CR-workers-20120501/.

[11] http://isotope.metafizzy.co/
[12] http://vestride.github.io/Shuffle/
[13] http://schlegel.github.io/balloon/balloon-synopsis.html