An End-User Semantic Web Augmentation tool

Cristian Sottile[1,2], Sergio Firmenich[1,3], and Diego Torres[1,2,3]

[1] LIFIA, Facultad de Informática, UNLP, Argentina
[2] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina
[3] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
[4] Departamento de Ciencia y Tecnología, UNQ, Argentina

Abstract. Web Augmentation is usually applied to add, remove and change Web sites' functionalities, content, and presentation. Content-based Web Augmentation is commonly performed by integrating content from an external Web site into the current one. In this article, we explore the use of the Semantic Web as a source of information to be incorporated to any Web site, aiming to simplify the development of Web Augmentation based on Semantic Web data. Our approach allows end-users without any programming skills to build Web Augmentation scripts that takes some information from the current Web page, and produce new related information gathered from the Semantic Web. This article introduces a pipeline process for building SWA and an End-User Development tool called SWAX to create augmentation layers without the need for any programming or SW skills.

Keywords: Semantic Web · Web Augmentation · User Interface

1 Introduction

The Semantic Web [1, 7] (SW) provides sources of semantic information useful to be interchanged among computer systems with a standard model called RDF. Most of RDF data sources allow being queried through a SPARQL endpoint.

Web Augmentation (WA) is an approach to improve Web applications by incorporing new functionalities. A common strategy is to enhance them on the client side [2], after it is received from the server.

Semantic Web Augmentation (SWA) is a particular type of WA where the SW provides the new information. The SW is accessed via a SPARQL endpoint or RDF API to extract information pieces, which are then adapted and inserted into the DOM structure on the client side. The Website ends up including the original contents plus the new ones from the SW. In this article we tackle with a kind of SWA where the new information is related to the content of the Website. SWA developers need to be skilled in client-side and SW technologies [6].

This article introduces both a pipeline process for building SWA and an End-User Development tool called SWAX to create augmentation layers without the need for any programming or SW skills.

Section 2 describes the pipeline process for building SWA. Section 3 introduces the End-User development tool designed as a Web Browser extension.

Related work is analyzed in Section 4, and Section 5 details conclusions and further work.

2 The process

This section explains the process for building SWA. It is a pipeline process where each component receives as input the output of its predecessor, and consists of three main steps:

- **Data extration:** We stated that our kind of SWA will involve the information present in the desired Website. In this step, these data are extracted from the DOM.
- **Data fetching:** Given a generic SPARQL query and the *extracted data*, this step fills the generic query with the data, producing a specific query for each one, and then run these queries against a SPARQL endpoint, obtaining new information from the Semantic Web.
- **Insertion:** Given an HTML template and the *fetched data*, this steps generates specific HTML elements by filling the templates with the data, and inserts them in the Website, performing the Web Augmentation.

3 An End-User Development Tool for SWA

This section introduces an End-User Development Tool for building SWA, which we call SWAX[5]. It is a Web browser extension whose functionality is based upon the process described in Section 2, and allows to build a WA script based on the current Website. It can be enabled in any Web page, initiating a wizard-like UI to configure the augmentation in a sequence of intuitive steps. The result is a script that is general enough to augment all Web pages whose DOM structure and semantics are similar to the original one. One configuration, several executions. Additionally, the script can be executed to test its results.

As a guide example, we will take the IMDb[6] website, and we will aim to improve the films Websites by adding the amount of Oscars every cast member won. The *"The Godfather II"* page on IMDb[7] could be used to produce a script to augment this and every IMDb film page. In the following, we introduce the tool's sequence of steps using this example.

1. **Extraction:** First, the user selects what DOM elements from the Web page will be used as data input in the semantic augmentation, by enabling the selection mode and clicking them. Continuing with our example, these would be actor's and actress' names. Then, the user defines how to extract the information contained in the selected DOM element (e.g., the text content or the HREF) and the tool shows the *value* extracted from it.

[5] Source code and video-tutorials at https://github.com/cfsottile/swa-extension
[6] IMDb stands for Internet Movie Database. http://imdb.com
[7] https://www.imdb.com/title/tt0071562/

2. **Querying:** The semantic query that will retrieve the new information from the SW must be defined. There are different mechanisms to help the SPARQL query building with visual tools. We delegate it to Visual SPARQL Builder (VSB) [3], which provides an intuitive interface to assist users in vocabulary and properties detection. Our tool provides an augmentation layer also on VSB for communication purposes. The resulting SPARQL query will be executed against the DBpedia SPARQL endpoint, obtaining the augmentation data.
3. **Building:** This step involves the definition of the HTML template that will be used to insert the augmentation data to the Web page. The user must write the HTML code including references to the augmentation data; however, the template may consist of only references, thus not being mandatory for the user to know any HTML. References to augmentation data are presented as buttons that, when clicked, insert the reference into the template input.
4. **Insertion:** This step configures the weaving of the HTML augmentation elements built in the previous step. The user must choose the wanted places in the Web page to inject them. This is done similarly to the Selection step, by selecting HTML elements from the DOM.
5. **Saving:** The user will define a set of URLs where the augmentation should be applied. In our cast example, this would be `https://www.imdb.com/title/*`. Then, the user can *save* the augmentation script, which will be applied every time a URL within the set is opened, or *try* it in the current page for testing purposes. Figure 1 shows the cast section of the *"The Godfather II"* page before and after the augmentation.

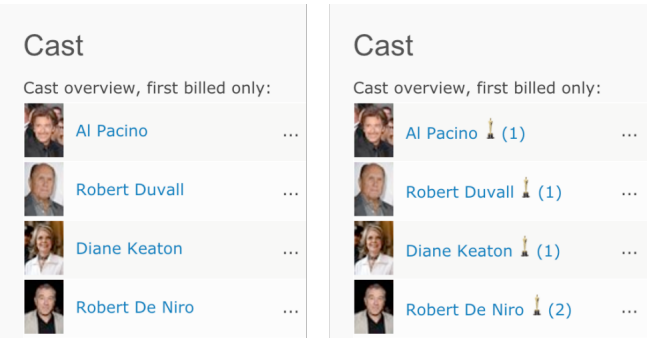

Fig. 1: Case study: Oscars

4 Related Work

Rico et al. [6] introduced a tool to allow Web developers with no SW skills to create templates capable of handling semantic data. WOA [4] presents an

approach for the creation of WA layers by firstly annotating Web contents with semantic tags. Most SW improvements of non-SW applications are based on the *semantification* of Web pages from social efforts [8, 5]. For the best of our knowledge, there exist no end-user tools for defining the weaving of information extracted from the SW in the current context of Web browsing.

5 Conclusions and Further Work

The SW is in a mature state since many years. However, it is not common to see tools for allowing Web users communities to consume all this infrastructure. In this paper, we present a pipeline processs for building SWA, and an End-User Development tool based upon it called SWAX to create augmentation layers. The approach allows end-users without any programming skills to produce generic WA scripts that allow for enriching the contents of groups of Web pages.

The first future work is the need for an exhaustive evaluation. The prominent prototype demonstrated well behavior and easy to use augmentation tasks in laboratory tests. Nonetheless, we are designing usability tests with a group of developers. Additionally, more extraction and insertion strategies are in development, as well as user experience improvements. Also, there are several SPARQL query builders like VSB that could be adapted for SWAX.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american **284**(5), 28–37 (2001)
2. Bouvin, N.O.: Unifying strategies for Web augmentation. In: Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots. pp. 91–100. ACM (1999)
3. Eipert, L.: Metadatenextraktion und vorschlagssysteme im visual sparql builder. In: Cunningham, D.W., Hofstedt, P., Meer, K., Schmitt, I. (eds.) INFORMATIK 2015. pp. 1925–1936. Gesellschaft für Informatik e.V., Bonn (2015)
4. Firmenich, S., Bosetti, G.A., Rossi, G., Winckler, M., Barbieri, T.: Abstracting and structuring web contents for supporting personal web experiences. In: Web Engineering - 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings. pp. 77–95 (2016)
5. Kahan, J., Koivunen, M.R.: Annotea: an open RDF infrastructure for shared Web annotations. In: Proceedings of the 10th international conference on World Wide Web. pp. 623–632. WWW '01, ACM, New York, NY, USA (2001)
6. Rico, M., Corcho, O., Macías, J.A., Camacho, D.: A Tool Suite to Enable Web Designers, Web Application Developers and End-users to Handle Semantic Data. Semantic-Enabled Advancements on the Web: Applications Across Industries: Applications Across Industries p. 123 (2012)
7. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE intelligent systems **21**(3), 96–101 (2006)
8. Torres, D., Diaz, A., Skaf-Molli, H., Molli, P.: Semdrops: A Social Semantic Tagging Approach for Emerging Semantic Data. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on. vol. 1, pp. 340–347. IEEE (2011)