SEMANTIZATION OF RESEARCH GROUP'S WIKI

Jakub ŠIMKO, Martin MARKECH

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
jsimko@fiit.stuba.sk, matomarkech@gmail.com

Abstract. Research groups often organize their activities through wikis. Their content is created by multiple group members and can grow into considerable size. It is therefore viable to have it semantically described in order for it to be easily accessible for machine processing (e.g. search engines). Groups also want to connect and cooperate with another research groups similar to their research domain. With semantic wikis they can find each-other easier thanks to the Linked Data. In this chapter we present extension of current faculty wiki called Semantic Bonsai. We propose an approach and tool for semantic annotation of wiki content. The tool is tailored for creation of semantics typically needed for describing research content: publications, problem domain concepts, scientific events, etc. We evaluate the tool in qualitative experiments.

1. Introduction

Web has changed dramatically during past few years. The content published on Web is constantly growing and Web is difficult to process. This naturally applies to all of its sub-domains, such as the research resources (e.g. publications and people) that we are interested in. We are still unable (in general) to get answers to complicated queries like *Find all researchers from Eastern Europe, who are interested in Linked Data.* It is because of many reasons, like [1] :

- Data and resources published on Web are in raw format (CSV, XML), without structure and semantics.
- Data are published in proprietary formats, which are hard to process.
- The HTML language defines how a document should look like instead of describing its semantic content.
- Hypertext links do not contain information about role in interconnected documents.
- Documents are readable for humans, but not for machines.

- The web connects documents instead of real-world entities.

Semantic Web aims to solve these problems. Thanks to the Linked Data and well-defined semantics for documents, researchers and research groups may much more easily find and connect to each-other.

Commonly, researchers use wikis to present their results, interests and information about projects they work on. The question then is how to acquire semantics in wiki context. Although there are many semantic wiki systems, they support only general semantics creation and none has yet been dedicated to specific needs of research information publishing. Also, many of research groups use customized or even their own wiki systems. Therefore, extending these wikis with semantic extensions raises a question of heterogeneity needed to be overcome. In our work, our aim is to bring in an extension as less dependent on wiki's implementation as possible.

By semantic extension, we mean a *tool enabling effective manual creation of semantics attached to wiki content*. We follow the idea of manual semantics creation (instead of automated one) because of higher possible reliability of acquired semantics. At the same time, we aim to exploit the awareness about semantics of the content that the user is creating within the wiki. In that course, our primary focus lies on ad hoc semantics creation, rather than post hoc one.

Our specific motivation is to improve existing wiki at our university taking into account the specific needs of an academic research group for personalized web (PeWe). PeWe uses its wiki for presentation and also to self-organize its members. There are published regular events, member's awards and various types of member's publications. Along with the inclusion of semantics, we also aim to improve the user interface of our wiki by creating templates to enable easier creation of its content. In particular, we aim reduce the time to write semantic bibliography links by implementing a web service which parses digital library and creates semantic bibliography links in required format. This chapter focuses on several issues we are facing:

- How the semantics should be attached to the content, regarding the required independence on wiki's implementation.
- Each wiki page may have many revisions and the triplets must be attached to correct revisions.
- Wiki also allows creating multiple page parts, so our triplet editor needs to support multiple triplet contexts.

Moreover, we wanted the semantics of our wiki stored in a semantic store, so we can connect to store's SPARQL endpoint to get answers to complex queries. In that way, semantics could be re-used in other interesting projects created at faculty. From a technical standpoint, we also wanted the extension to be browser independent.

2. Related Work

The first wiki software was created by Howard G. Cunningham in 1994. It was called WikiWikiWeb[1]. It was easy-to-use system, which allowed creating, modifying and interconnecting pages. Instead of HTML, Cunningham used markup language WikiML [2].

The first semantic wiki system was created ten years later, in 2004 [3]. It was called Platypus and was an extension of Cunningham's wiki WikiWikiWeb. Many semantic wikis has been created since then, but most of them died and are no longer under active development. The semantic wikis can be divided into several categories [3]:

- Wikis with typed links, including wikis with metadata specified inline content source, and wikis with metadata specified externally.
- Advanced wikis, which use ontologies.
- Atypical wikis

Platypus[2] is wiki of first type. Metadata can be edited in separate area, not directly in text. Subjects, predicates and objects of ontologies can be modified through select boxes. User is able to choose only valid select box value – valid subject/property/object or literal. The small disadvantage of this approach is that the content must be created at first, because metadata can be created only when the text is written.

OntoWiki[3] was created in 2006 by AKSW group. It has different approach from Platypus and other wikis. It is representative of the second type of wikis. It was built with intuitive authoring of semantic content in mind. OntoWiki facilitates the visual presentation of a knowledge base as an information map, with different views on instance data [4]. It enables inline editing mode for editing RDF content, which allows very fast editing of the content. OntoWiki is also social-oriented and allows discussion on changes in every part of knowledge base. Currently in 2013, it is still a very active project.

RDFaCE[4] is an extension of popular WYSIWYG content editor called TinyMCE. RDFaCE supports different views for semantic content authoring. It supports automatic content annotation and connects to different semantic APIs to get relevant triplets for words in text. The triplet and namespace editor allows to use different ontologies.

Many wikis, like ours, use wiki markup language like Markdown instead of a WYSIVIG editor. It is not possible to implement RDFaCE to these wikis. This is one of the reasons, why we are implementing our own extension.

3. Approach description

Our approach for manual semantization of the researcher wiki content combines direct editing of inline semantic definitions with "side" graphical formularies mirroring the actual state of the semantics in the document. The method and the application we propose feature several key elements:

---

[1] http://c2.com/cgi/wiki?WikiWikiWeb
[2] http://platypuswiki.sourceforge.net/
[3] http://ontowiki.net/Projects/OntoWiki
[4] http://aksw.org/Projects/RDFaCE.html

- Templates to generate semantics for events, member list, news.
- Web service for auto-generating of the semantic bibliography links.
- Triplet editor for creation and modification of semantics attached to text.
- Namespace editor for changing of used ontologies.
- Dereferenced URIs to browse each triplet in semantic store.
- SPARQL endpoint allowing the consumption of the semantics by external services.

There are many blocks of text with similar structure on wiki. Part of our method are templates for this blocks with easy-to-use UI separated to simple and advanced mode. Templates helps people to keep the same text structure and to auto generate semantics.

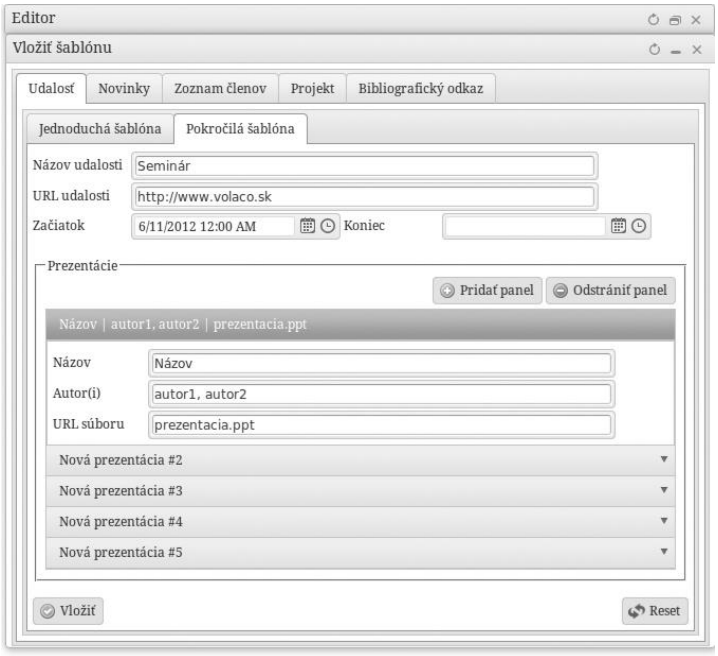

Figure 1. Advanced template for event editation.

When using a template for creating a new content entry, for example information about an upcoming event, the typical user scenario is as follows (using interface in the Figure 1):

1. User places the cursor somewhere within the Markdown (wiki content source code).
2. User chooses simple or advanced template mode.
3. User fills in the required fields.
4. User submits the form.
5. Semantic triplets are generated in background and inserted to triplet editor.
6. Markdown code is generated in the background and the result is inserted to the code area depending on the cursor location. This code contains not only the fragments required for content rendering, but also non-renderable tags that carry the semantics.

Using the template this way ensures that the created text has properly defined semantics. Values filled into the fields are inserted to preselected triplets combinations with predefined ontologies. User can later modify the triplet's values at any time, using the triplet editor. User can also modify (or even create) the semantics directly within the Markdown code by manipulating with generated semantic tags. It is also possible to copy-paste and modify the "semantized" code portions. These are recognized by the extension as new blocks of content and semantics. In order to keep the markdown readable, we also employ a highlighting extension that slightly suppress the visibility of semantics tags by lighter colors, so the user can focus on the renderable content itself.

We have also included a simple method for automated identification and generation of semantic bibliography references for the Markdown code. Its use scenario is as follows:

1. User fill in some information about publication – DOI, authors, or title
2. Web service sends query to a web search engine in format *site:<digital library site name> <searched string>.* We use ACM digital library portal, as it also contains many references to other digital libraries (and thus matches the identities with the original publisher sites).
3. Several top results are taken. Those that contain a publication entry are parsed for publication metadata (e.g. titles) and are again matched against the information provided by the user. If the match is high enough (e.g. majority of words match with the title, the author names are matched), the references are considered identical.
4. Webservice load the page in background and download BibTex data.
5. User confirms the correctness of the found match.
6. Then it converts BibTex format to ISO 690 reference string added to the Markdown. Publication semantics entries are created in our semantics repository and a link to the digital library is added to the Markdown as well.

Because these queries might be repeated, our method caches the results, which were confirmed as correct by user. If someone do query with equal or highly similar values, the result will be fetched from cache.

Wikis are complex systems, which supports revisions and many page parts. Due to this, we need to save triplets in multiple contexts. Semantic store Sesame, which our method is using, has a support for special field called context. It's proposed for special situation like this. In our method we use value of page id + revision number + page part id as a hash for context field. Therefore, we can successfully attach triplets to correct page versions.

4. Evaluation and conclusion

We have devised and evaluated our extension in a qualitative manner based on interviewing users and on controlled use of the application. We chose a qualitative evaluation mainly because of the character of our approach. Our hypotheses were formulated as follows:

1. The usability of the original wiki content creation interface has not been decreased after introducing the extension.
2. The readability of the Markdown enriched by semantic tags has not been decreased after introducing the semantic tags.

In a two-stage experiment, we interviewed two users in the first and one user in the second stage. All users are members of the research group and are often engaged in creation of group's wiki content. Between the two stages, we used the gained feedback to improve the application. Therefore the final closures are mainly, sourcing from the last interview.

We assigned the users with several tasks, e.g. adding students, events and publication entries as content to the wiki. We have observed their behavior and interviewed them to assess, whether they are confused or lost. We have evaluated whether they finished their assignments and to what degree we needed to help them.

After the experiments, we can conclude, that the time needed for creation of the semantized content was in average slightly longer than in the case without semantics. Though, as we found out, users differed in the preference of semantics creation paradigm. Two of them stick to the direct editing of the content source code, while the last user preferred more the form interface (which was a quicker alternative for him).

As for the second hypothesis, the markup also proved more complicated (less readable), however, after the introduction of syntax highlighting, the readability turned back to normal, as reported by users.

5. Conclusion and future work

In this chapter we presented semantic extension to current faculty wiki called Bonsai. This extension helps with semantics creation process at the time of the content creation. Thanks to templates, appropriate triplets can be created with minimal additional effort.

Yet, the future of this work lies heavily in improving the user interface aspects of the solution. The users are apparently manageable, when it comes to explanation of what is the semantics describing their research content good for, but they are not very tolerant when it comes to additional work burdening them (although the additional effort may bring benefits if the semantics is exploited properly). A big challenge for the future is also development of the automated support tools for the semantics detection at the time of the content creation (such as the tool for publication reference detection).

Acknowledgement: This work was partially supported by grants No. VG1/0675/11 and APVV 0208-10.

References

[1] Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. In *Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS)*, (2009), vol. 5, no. 3, pp. 1-26.

[2] Buffa, M., Gandon, F., Ereteo, G., Sander, P., Faron, C.: SweetWiki: A semantic wiki. In *Web Semantics: Science, Services and Agents on the World Wide Web*, (2008), vol. 6, no. 1, pp. 84–97.

[3] Bry, F., Schaffert, S., Vrandečić, Weiand, K.: Semantic Wikis: Approaches, Applications, and Perspectives. In *Reasoning Web. Semantic*, (2012), vol. 7487, pp. 329–369.

[4] Decker, S., Kashyap, V.: The Semantic Web: Semantics for Data and Services on the Web. Springer, (2008), ISBN 3540764518.