OntoWiki 1.0

10 Years of Development – What's New in OntoWiki

Philipp Frischmuth Natanael Arndt Michael Martin

{frischmuth|arndt|martin}@informatik.uni-leipzig.de

Agile Knowledge Engineering and Semantic Web (AKSW)
University of Leipzig / Institute of Computer Science
Augustusplatz 10, 04109 Leipzig, Germany

ABSTRACT

In this demonstration (with supportive poster) we present the semantic data wiki OntoWiki, which was released in version 1.0 just recently. We focus on the changes introduced to the tool in the latest release and showcase the generic data wiki, improvements we made with regard to the documentation as well as three success stories where OntoWiki was adapted and deployed.

1. INTRODUCTION

We demonstrate the semantic data wiki OntoWiki, which was first introduced in 2006 [2]. 2016 marks the 10th anniversary of OntoWiki development and we value this occasion by finally releasing a version 1.0 of the tool. Throughout the last 10 years OntoWiki has matured and was adapted for various different use cases (cf. section 4). OntoWiki is a web based application that supports the authoring, publication and visualization of arbitrary structured information in distributed scenarios. It encodes information as quads and hence is able to handle arbitrary RDF data. Although within OntoWiki the concept of *wiki pages* is exchanged with the concept of *information resources* it still adheres to the wiki paradigm of "making it easy to correct mistakes, rather than making it hard to make them" [5]. Figure 1 shows the two main generic views of OntoWiki that are used to visualize lists of resources and the information attached to individual resources.

Besides the generic views that are provided in order to visualize information in a domain-independent manner, OntoWiki also acts as an application framework to build data-driven applications adapted to specific domains [4]. During the course of several research and industry projects it has been customized and its functionality has been employed to build prototypes as well as applications deployed in production. Since its first introduction several releases of OntoWiki were published. In this demonstration we focus on the changes introduced to the tool between the 0.9.11 and the 1.0 release. We showcase the latest version of the generic data wiki, improvements we made with regard to the documentation as well as three success stories where OntoWiki was adapted. For a more comprehensive description of OntoWiki and its features please refer to [3].

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

2. WHAT'S NEW IN ONTOWIKI

Since the 0.9.11 release, which was published at the beginning of 2014, the work on OntoWiki and the underlying Erfurt framework[1] has continued. The recent 1.0.0 release of OntoWiki includes changes and fixes from more than 153 commits. It also incorporates an updated version of Erfurt, which is now available in version 1.8.0 and includes changes and fixes from more than 48 commits.

The *title helper* is an important component employed in almost every part of OntoWiki. Taking into account a configurable list of title properties, it is responsible for fetching titles of resources that should be displayed within the tool. Since most user interface (UI) elements make heavy use of this component, its performance is critical. The title helper has been improved in this regard as it now fetches titles for multiple resources at once. Hence the number of required SPARQL queries to build up the OntoWiki UI is reduced, which results in a superior overall performance. In addition the title helper component is aware of language tags now and thus can be configured to retrieve titles for specific languages only.

The *visual SPARQL query editor* that is included in OntoWiki provides a means to query knowledge bases directly using SPARQL. Results of such queries are provided either by using the generic resource list view or one of the supported SPARQL result formats. New in the latest release is the option to export results using CSV as result format, which often is a handy format to bridge the gap to tabular data management systems.

OntoWiki is able to display RDF information in any language. On the UI level it *supports multiple languages* as well. In addition to English, German, Russian and Chinese, user interface elements have been translated to French and Hungarian with the latest release.

In order to *improve the experience for developers*, both OntoWiki and Erfurt can now benefit from the improvements throughout the last years in the PHP development ecosystem and have been updated to use Composer[2] for dependency management. It is now possible to download all

[1] https://github.com/AKSW/Erfurt
[2] Dependency Manager for PHP: https://getcomposer.org/

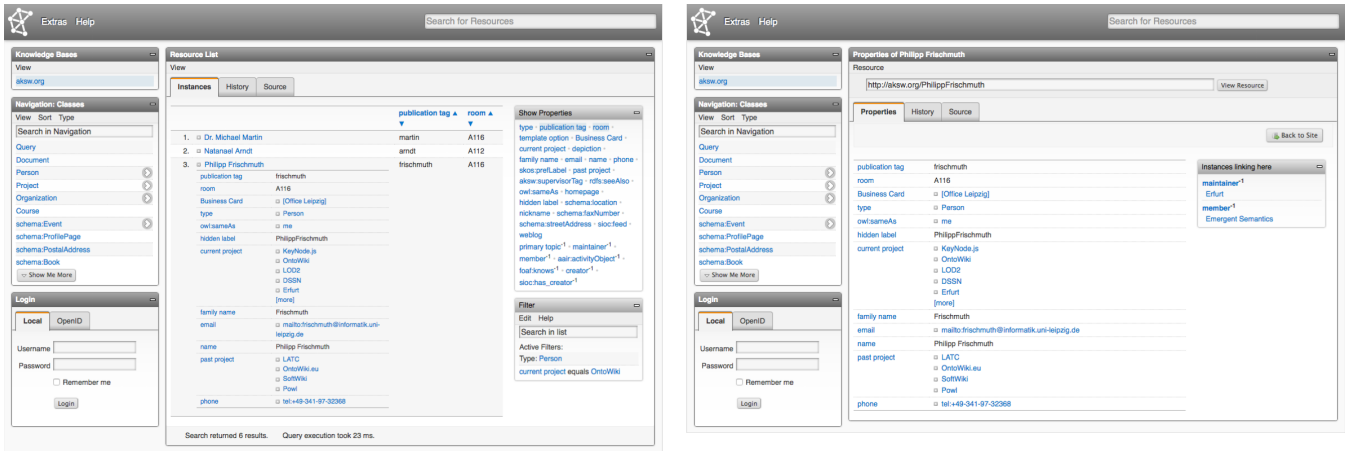

Figure 1: Generic OntoWiki views – resource list on the left (with two selected properties and an applied filter) vs. resource view on the right (with attributes table and linking instances module).

required dependencies in the appropriate versions automatically by issuing a simple `composer install` command.

Besides these new features, with the latest release we were mainly focused on stability and performance. Various bugs that have been reported over the years have been fixed in this release as well.

3. DOCUMENTATION IMPROVEMENTS

The new home of the OntoWiki user and developer documentation is now `docs.ontowiki.net`[3]. We have collected numerous documentation resources that were previously scattered across multiple places on the web, cleaned up and unified the documents and put them on the new website. The documentation is now kept in a separate repository on Github[4] and generated whenever a change is made in the repository. The new documentation website enhances navigation, since it includes a navigation tree with support for nested pages. The Github wiki (previously used for numerous documentation resources) only includes a flat list of pages for navigation. In addition the new website includes a search functionality that enables users to search inside the documentation resources.

4. ONTOWIKI SUCCESS STORIES

Since OntoWiki is not only a standalone application for curating and exploring RDF datasets but also the OntoWiki Application Framework (OAF), it was used to implement very different use cases throughout the years. One mechanism for extending OntoWiki is its extension system, which allows for a modular approach of introducing new features. In some cases it was extended mainly in its functionality, in other cases it has completely changed its appearance. In this section we present three use cases in which OntoWiki was applied to completely different domains, *amsl.technology* in the library domain, *Pfarrerbuch* and *Catalogus Profesorum Lipsiensium* as prosopographical knowledge bases in the history domain and *aksw.org*, where OntoWiki serves as a content management system (CMS) for our research group.

[3]http://docs.ontowiki.net
[4]https://github.com/AKSW/docs.ontowiki.net

4.1 amsl.technology

In the *amsl.technology*[5] project an *Electronic Resource Management System* is developed based on OntoWiki [1, 6]. amsl.technology is targeted to support libraries and other players in the library domain to manage not only physical resources, such as books, journals, CDs/DVDs, but also electronic resources (e.g. e-journals, e-books or databases). Especially challenging in this use case are new licensing and lending models which have been introduced by publishers, such as pay-per-view, patron-driven-acquisition, short term loan or big deal. Existing infrastructure is not yet prepared for managing those electronic resources, lending and licensing models. Even worse, software which is developed to meet those requirements is likely to be outdated once new media types or licensing and lending models are introduced. The amsl.technology application benefits from OntoWiki's flexible and agile data management capabilities as well as from its features for curating, exploring and annotating resources in a collaborative way. Also the development of amsl.technology brought back many improvements to OntoWiki due to its Open Source development model. A demo system, as depicted in fig. 2, is available at http://amsl.technology/demosysteme/ respective https://demo.amsl.technology/.

4.2 Pfarrerbuch and Catalogus Profesorum Lipsiensium

The projects *Sächsisches Pfarrerbuch* (engl. *Saxonian pastors book*)[6] and *Catalogus Profesorum Lipsiensium* (CPL) [7] are projects for prosopographical knowledge bases. The Pfarrerbuch is aiming to catalog all pastors serving in the Lutheran Church in Saxony since the reformation in 1517. Recently it was also extended by pastors from the Lutheran Church in Hungary. Currently the dataset is under curation and only a very small excerpt of the uncurated data is publicly available. The CPL collects all professors who have taught at Leipzig University from its foundation in 1409 to the presence. Currently it comprises more than 14,000 en-

[5]http://amsl.technology/
[6]The Pfarrerbuch project webpage: http://pfarrerbuch.de and the current curation system: http://pfarrerbuch.aksw.org

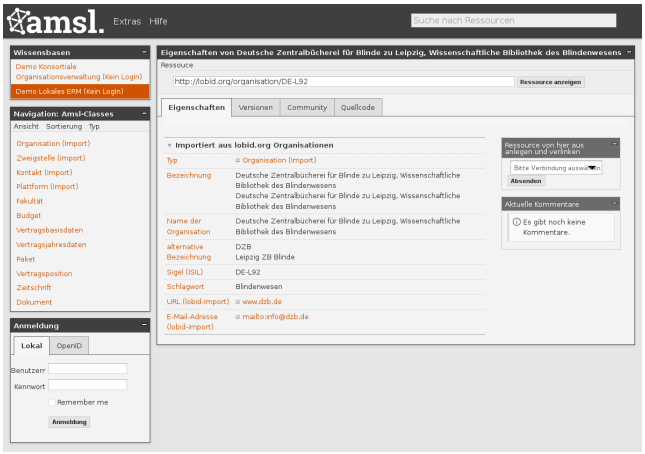

Figure 2: An entry in the amsl.technology customized OntoWiki

tities. In both systems OntoWiki is used as data curation and exploration platform especially for data scientists and domain experts in the respective historical domain. Different entities with various properties, such as persons, places (churches, schools, universities), events with personal relations, staffing and attendance relations can be easily managed wit the flexible RDF data model and editing interface in OntoWiki. Using the OntoWiki *site extension*[7] allows for a template based publication of RDF resources resulting in easy to explore HTML web pages (in addition to the default OntoWiki SPARQL endpoint and Linked Data Server). Also OntoWiki's support for multiple languages helped to implement a German and a Hungarian version of the Pfarrerbuch in one system. Further the wiki approach supports distributed groups of researchers and data scientists to flexibly collaborate on a common dataset. Figure 3 depicts an example page showing the the resource of Pfarrer Christian Friedrich Ernst Führer[8] with various properties and relations retrieved from the RDF data model. In the upper right corner one can see the search interface, which queries the OntoWiki SPARQL endpoint in real time to allow easy navigation. Below the search interface the links can be used (from left to right) to access the default OntoWiki UI, edit the currently displayed resource, create a new instance of the currently selected class and change the UI language between German and Hungarian.

4.3 aksw.org

The website of our research group[9] (AKSW) is backed by an OntoWiki with an extension for publishing websites based on templates and data from a triple store. This sophisticated extension is called the *site extension*[7] and it extends OntoWiki towards a CMS. Since the AKSW research group has become quite large recently (the website currently includes more than 50 team members and a large number of different projects) a CMS is indispensable. On the other

[7]https://github.com/AKSW/site.ontowiki
[8]Pfarrer Christian Führer was pastor at the Nikolaikirche in Leipzig and a leading figure of Monday demonstrations in East Germany, https://en.wikipedia.org/wiki/Christian_Führer
[9]http://aksw.org

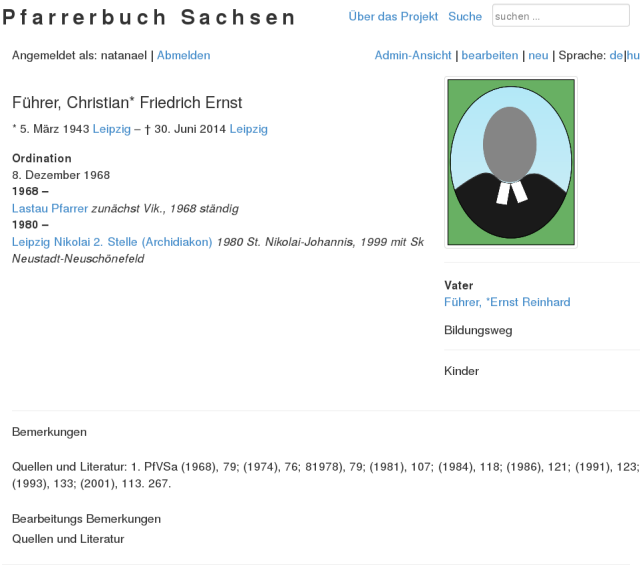

Figure 3: The entry of Pfarrer Führer from Leipzig in the pfarrerbuch database

hand a large amount of the information that needs to be presented is very structured (e.g. information about team members, project abstracts, etc.) and should be reused on different pages. In addition lists of resources of a certain type (e.g. project and team lists) should be maintained automatically. Within this use case OntoWiki (without further modifications) is used to author and maintain the data including page content in literals written in Markdown syntax. The site extension then retrieves this data and renders it according to a set of templates and SPARQL queries. Another benefit of the site extension is the ability to automatically publish the raw RDF data next to the rendered HTML content as Linked Data. Every web page created with the site extension has a related resource URI that can be used to retrieve RDF data or the HTML page using content negotiation.

Figure 4 depicts the page of the OntoWiki project on the AKSW website. This page is dynamically composed from very structured information fragments (e.g. title, abstract, team members, publications, screenshot, etc.).

5. ACKNOWLEDGEMENTS

We want to thank all contributors to the OntoWiki project, especially all reporters of issues and contributors to the OntoWiki and Erfurt codebases[10]. This work was partly supported by the following grants from the German Federal Ministry of Education and Research (BMBF) for the LEDS Project under grant agreement No 03WKCG11C and the European Union's Horizon 2020 research and innovation programme for the SlideWiki Project under grant agreement No 688095.

[10]https://github.com/AKSW/OntoWiki/graphs/contributors, https://github.com/AKSW/Erfurt/graphs/contributors

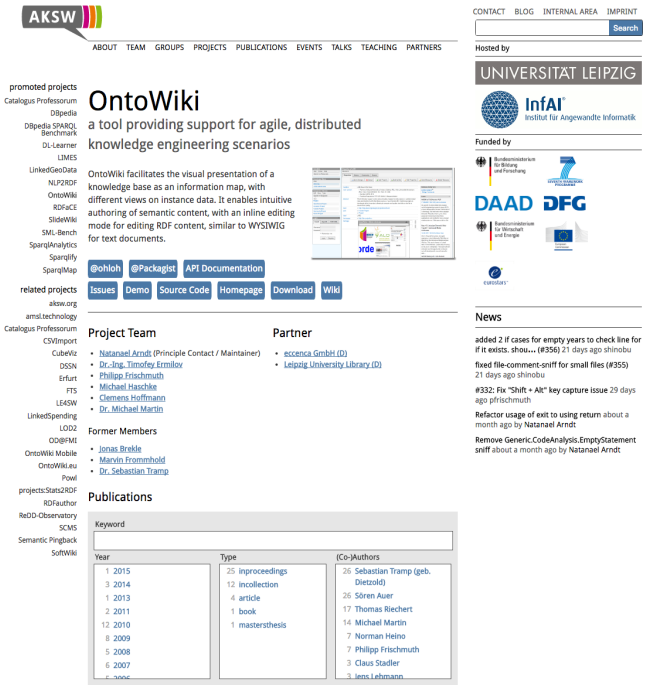

Figure 4: Page of the OntoWiki project on the AKSW website.

6. REFERENCES

[1] N. Arndt, S. Nuck, A. Nareike, N. Radtke, L. Seige, and T. Riechert. AMSL: Creating a linked data infrastructure for managing electronic resources in libraries. In M. Horridge, M. Rospocher, and J. van Ossenbruggen, editors, *Proceedings of the ISWC 2014 Posters & Demonstrations Track*, volume Vol-1272 of *CEUR Workshop Proceedings*, pages 309–312, Riva del Garda, Italy, Oct. 2014.

[2] S. Auer, S. Dietzold, and T. Riechert. OntoWiki – A Tool for Social, Semantic Collaboration. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. M. Aroyo, editors, *The Semantic Web – ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA. Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, Nov. 2006.

[3] P. Frischmuth, M. Martin, S. Tramp, T. Riechert, and S. Auer. OntoWiki – An Authoring, Publication and Visualization Interface for the Data Web. *Semantic Web, Special Issue on Semantic Web Interfaces*, 6(3):215–240, 2015.

[4] N. Heino, S. Dietzold, M. Martin, and S. Auer. Developing Semantic Web Applications with the OntoWiki Framework. In T. Pellegrini, S. Auer, K. Tochtermann, and S. Schaffert, editors, *Networked Knowledge – Networked Media, Integrating Knowledge Management, New Media Technologies and Semantic Systems*, volume 221 of *Studies in Computational Intelligence*, pages 61–77. Springer, Berlin / Heidelberg, 2009.

[5] B. Leuf and W. Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Pearson Education, May 2001.

[6] A. Nareike, N. Arndt, N. Radtke, S. Nuck, L. Seige, and T. Riechert. AMSL: Managing electronic resources for libraries based on semantic web. In E. Plödereder, L. Grunske, E. Schneider, and D. Ull, editors, *Proceedings of the INFORMATIK 2014: Big Data – Komplexität meistern*, volume P-232 of *GI-Edition—Lecture Notes in Informatics*, pages 1017–1026. Gesellschaft für Informatik e.V., Sept. 2014. © 2014 Gesellschaft für Informatik.

[7] T. Riechert, U. Morgenstern, S. Auer, S. Tramp, and M. Martin. Knowledge engineering for historians on the example of the catalogus professorum lipsiensis. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *Proceedings of the 9th International Semantic Web Conference (ISWC2010)*, volume 6497 of *Lecture Notes in Computer Science*, pages 225–240, Shanghai / China, 2010. Springer.