

SlideWiki – A Platform for Authoring FAIR Educational Content

Ali Khalili and Klaas Andries de Graaf

Department of Computer Science, Vrije Universiteit Amsterdam, NL
{a.khalili, ka.de.graaf}@vu.nl

Abstract. SlideWiki.org is a Web-based OpenCourseWare (OCW) authoring system that enables educators and learners to collaborate on creating, sharing, re-using and re-purposing multi-lingual open educational content. The SlideWiki platform allows people to author FAIR (Findable, Accessible, Interoperable, and Reusable) educational content. SlideWiki supports many features to semantically-enrich educational content to support FAIR authoring. In this paper we will present those features of the platform such as Linked Data interface, manual and automatic content annotation as well as content linking and metadata.

1 SlideWiki Platform

A major obstacle to increase the efficiency, effectiveness and quality of education is the lack of widely available, accessible, multilingual, timely, engaging and high-quality educational material (i.e. OpenCourseWare). The creation of comprehensive OpenCourseWare (OCW) is tedious, time-consuming, and expensive. Courseware employed by educators is therefore often incomplete, outdated, dull, and inaccessible to learners with disabilities. With the open-source SlideWiki platform (available at SlideWiki.org) the effort in creating, translating, and evolving FAIR (Findable, Accessible, Interoperable, and Reusable) OCW can be widely shared (i.e. crowdsourced). Similarly to Wikipedia for encyclopaedic content, SlideWiki [1] allows its users (1) to collaboratively create comprehensive OCW (curricula, slide presentations, self-assessment tests, illustrations etc.) online in a crowdsourcing manner, (2) to semi-automatically translate this content into more than 50 different languages and to improve the translations in a collaborative manner and (3) to support engagement and social networking of educators and learners around that content. SlideWiki.org (funded by an EU H2020 grant¹) is already used by thousands of educators and learners.

Figure 1 depicts the 3-tier technical architecture of the SlideWiki platform where *data*, *service* and *user interaction* concerns are decoupled as individual stand-alone components. Our contribution to semantically enrich educational content in SlideWiki, touches upon all those 3 layers: RDF and Linked Data version of content are provided in the data layer, NLP (Natural Language Processing) services are exposed for automatic content annotation, and user interfaces for manual content annotation together with inline metadata to increase

¹ see <http://slidewiki.eu> for more information

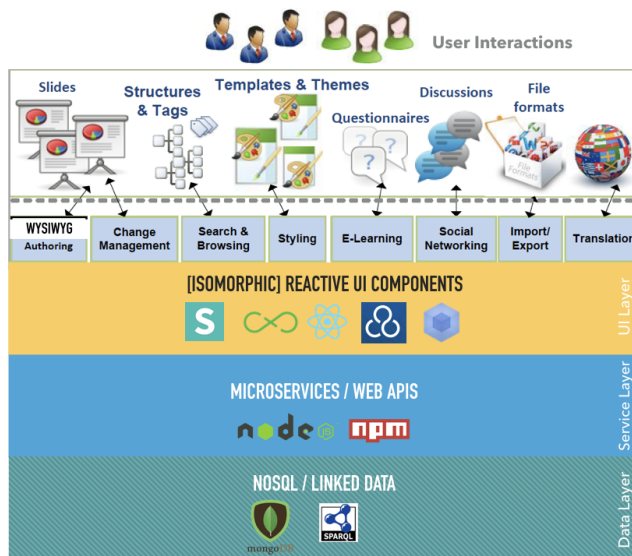


Fig. 1. SlideWiki Platform Architecture.

the findability of content are provided in the interaction layer. In the following subsections we briefly introduce the SlideWiki semantics-related features:

1.1 RDF Generation and Linked Data Interface

An RDF data model allows easy exposure and integration of educational data and the interlinking of data across system boundaries. In order to foster the generation of RDF from legacy systems based in relational databases, there is a stack of technologies and standards such as a) R2RML (W3C recommendation²) to generate RDF from relational databases, b) RML (an extension to R2RML³) to generate RDF from JSON, XML and CSV datasets, c) XR2RML [2] (an extension to RML) to generate RDF from non-SQL databases, such as MongoDB.

To generate the RDF version of SlideWiki, we employed Morph (Morph-XR2RML⁴) that implements the standard XR2RML. Morph is a java program that reads two files: a config file and a mapping file. Although both files can be modified by the developers, in practical terms the config file is fixed (stable in time) and the real effort is focused on the mapping file. A mapping file establishes how the elements in the SlideWiki MongoDB instance are converted. The result of SlideWiki RDF conversion is exposed as a Linked Data interface on top of an open SPARQL endpoint⁵.

² <https://www.w3.org/TR/r2rml/>

³ http://rml.io/RML_R2RML.html

⁴ <https://github.com/frmichel/morph-xr2rml/tree/morph-xr2rml-1.0>

⁵ available at <http://slidewiki.oeg-upm.net/sparql>

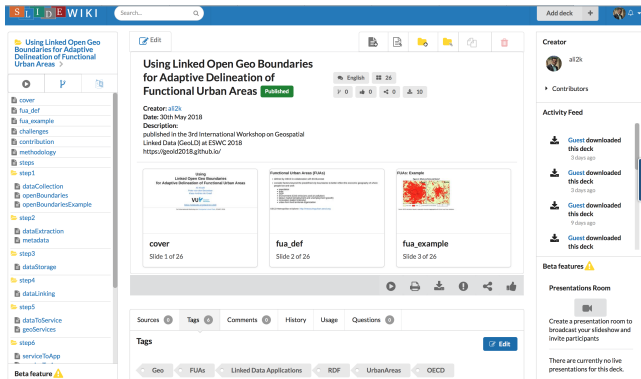


Fig. 2. A sample deck on SlideWiki with tags: <https://slidewiki.org/deck/111329>

1.2 Manual Content Annotation

Presentations get more visibility when they are tagged with words or entities about their topic and contents. Annotations make the contents of a deck more meaningful to users as tags tell a user what the deck is about. An author can manually set tags for his/her presentation. To support the user in this process, tag recommendations are calculated via a Natural Language Processing (NLP) service (based on calculation of important words and named entities in the presentation) and presented to the user. The user can select/approve recommended tags to improve the semantic enrichment of his/her presentation (cf. Figure 2).

Another content annotation method is via *in-slide annotation*. Users select phrases in a slide and then annotate these phrases to be instances of an ontology class. We will provide both our own SlideWiki specific ontology (partially based on existing OCW and OER ontologies) a user-created or uploaded ontology, and the existing ontologies or vocabularies used in knowledge bases accessible on the internet via a SPARQL endpoint. We also call DBpedia spotlight to give suggestions to annotate phrases based on the DBpedia knowledge base (linking to a DBpedia instance or instantiating as DBpedia concept). Presentations are considered semantically enriched when they are linked to external knowledge bases. The in-slide annotation (cf. Figure 3) is still under development.

1.3 Automatic Content Annotation

There are also automatic processes in SlideWiki for content annotation: In each slide DBpedia Spotlight entities are identified by the NLP Analysis from the NLP service (See <https://nlpstore.experimental.slidewiki.org/documentation>) including a DBpedia URI. Named Entity Recognition is used to recommend tags via the NLP service. The calculated results of the NLP service are stored in the NLPstore in order to provide annotation and tag suggestions to the user in a fast manner by querying pre-processed recommendations in the future.

The main NLP API called *nlpForDeck* performs several NLP steps encapsulated in one service. It can be divided in 3 main parts: 1) Preprocessing (like html

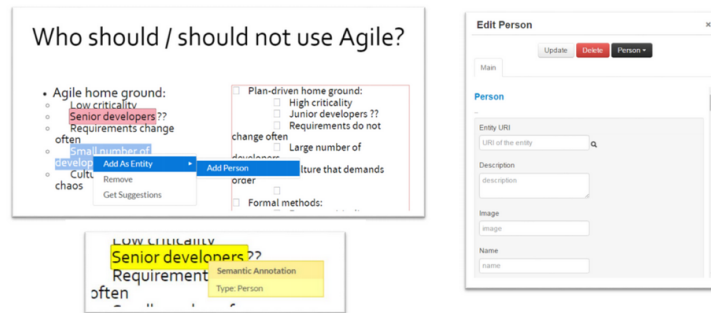


Fig. 3. Screenshot of current prototype for selecting text and adding in-slide semantic annotations.

to text, automatic language detection, tokenization) 2) Automatically retrieve entities used in the slides via DBpedia spotlight (use for in-slide annotation and semantic enrichment) 3) Identification of important words and entities useful for tagging the presentation in the platform⁶.

1.4 Content Interlinking

Presentations are considered semantically enriched when linked to presentations with similar content. The user can dive deeper into the given topic and get a better understanding of the topic by viewing related presentations. This can be either done by showing presentations with the same tags or by linking to presentations with similar content (even if they do not share the same tags). The latter is performed by the deck recommendation based on the content of the presentation.

1.5 Metadata

As most web search engines incorporate semantic data in their search, semantic enrichment of the educational resources will result in a higher ranking and better visibility. This is particularly important for improving search engine performance for content within SlideWiki. SlideWiki will make use of metadata HTML tags as well as embedded Microdata and JSONLD description of educational resources to support SEO (Search Engine Optimization).

References

1. A. Khalili, S. Auer, D. Tarasowa, and I. Ermilov. Slidewiki: elicitation and sharing of corporate knowledge using presentations. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 302–316. Springer, 2012.
2. F. Michel, L. Djimenou, C. Faron-Zucker, and J. Montagnat. Translation of relational and non-relational databases into rdf with xr2rml. In *WEBIST2015*, pages 443–454, 2015.

⁶ The importance of the words and entities is determined via TFIDF (term frequency-inverse document frequency) ranking.