Semantic MediaWiki

Markus Krötzsch[1], Denny Vrandečić[1], and Max Völkel[2]

[1] AIFB, Universität Karlsruhe, Germany
[2] FZI Karlsruhe, Germany

Abstract. Semantic MediaWiki is an extension of *MediaWiki* – a widely used wiki-engine that also powers *Wikipedia*. Its aim is to make semantic technologies available to a broad community by smoothly integrating them with the established usage of MediaWiki. The software is already used on a number of productive installations world-wide, but the main target remains to establish “Semantic Wikipedia” as an early adopter of semantic technologies on the web. Thus usability and scalability are as important as powerful semantic features.

1 Introduction

Wikis have become popular tools for collaboration on the web, and many vibrant online communities employ wikis to exchange knowledge. For a majority of wikis – public or not – primary goals are to organise the collected knowledge and to share this information. We present the novel wiki-engine Semantic MediaWiki [**?**] that leverages semantic technologies to address those challenges.

Wikis are usually viewed as tools to manage online content in a quick and easy way, by editing some simple syntax known as wiki-text. This is mainly plain text with some occasional markup elements. For example, a link to another page is created by enclosing the page’s name in brackets, e.g. by writing [[Danny Ayers]]. To enhance usability, we introduce new features by gently extending such known syntactical elements.

2 System overview

Semantic MediaWiki (SMW)[3] is a semantically enhanced wiki engine that enables users to annotate the wiki’s contents with explicit, machine-readable information. Using this semantic data, SMW addresses core problems of today’s wikis:

- *Consistency of content:* The same information often occurs on many pages. How can one ensure that information in different parts of the system is consistent, especially as it can be changed in a distributed way?
- *Accessing knowledge:* Large wikis have thousands of pages. Finding and *comparing* information from different pages is a challenging and time-consuming task.
- *Reusing knowledge:* Many wikis are driven by the wish to make information accessible to many people. But the rigid, text-based content of classical wikis can only be used by reading pages in a browser or similar application.

[3] SMW is free software, and can be downloaded at `http://sourceforge.net/projects/semediawiki/` (current version 0.5). Sites on which SMW is already running are given below.

But for a wiki it does not suffice to provide some technologies to solve these problems – the key is to make those technologies *accessible* to a broad community of non-expert users. The primary objective for SMW therefore is the seamless integration of semantic technologies into the established usage patterns of the existing MediaWiki system. For this reason, SMW also is available in multiple languages and was designed to easily support further localisation.

Semantic wikis are technologically interesting due to their similarity with certain characteristics of the Web in general. Most importantly, information is dynamic and changes in a decentralised way, and there is no central control of the wiki's content. In our case, this even extends to the available annotations: there is no central control for the annotation schema. Decentralisation leads to heterogeneity, but wikis still have had tremendous success in integrating heterogeneous views. SMW ensures that existing processes of consensus finding can also be applied to the novel semantic parts. While usually not its primary use, wikis contain not only text but also uploaded files, especially pictures and similar multimedia content. All functions described below are also available for such extended content.

Details of practical usage are discussed in the next Sect. **??**. SMW is based on a simple and unobtrusive mechanism for semantic annotation (Sect. **??**). Users provide special markup within a page's wiki-text, and SMW unambiguously maps those annotations into a formal description using the OWL DL ontology language. To make immediate use of the semantic data, the wiki supports a simple yet powerful query language (Sect. **??**). By embedding queries into wiki-text, users can create dynamic pages that incorporate current query results (Sect. **??**).

As we will see Sect. **??**, SMW also provides various interfaces to data and tools on the Semantic Web. To enable external reuse, formal descriptions for one or more articles can be obtained via a web interface in OWL/RDF format (Sect. **??**). As reviewed in Sect. **??**, it is also possible to import data from OWL ontologies and to map wiki-annotations to existing vocabularies such as FOAF. Since SMW strictly adheres to the OWL DL standard, the exported information can be reused in a variety of tools (Sect. **??**). As a demonstration, we provide an external SPARQL query service that is synchronised with the wiki's semantic content.

Semantic wikis have many possible applications, but the main goal of SMW is to provide the basis for creating a *Semantic Wikipedia*. Consequently, the software has a particular focus on scalability and performance. Basic operations such as saving and displaying of articles require only little resources. Even evaluation of queries can usually be achieved in linear time wrt. the number of annotations [**?**]. In Sect. **??**, we give examples that illustrate the high practical relevance of the problems we claimed above, and we sketch the fascinating opportunities that semantic information in Wikipedia would bring. Finally, we present some further current practical uses of SMW in Sect. **??**, and give a brief summary and outlook on upcoming developments in Sect. **??**.

3 Practical usage: ontoworld.org

We illustrate the practical use of SMW via the example of `http://ontoworld.org`, which is a community wiki for the Semantic Web and related research areas. It contains information about community members, upcoming events, tools and developments. Recently, SMW was also used as a social wiki for various conferences[4], and the evaluation and user feedback made it clear that a single long-term wiki like ontoworld.org would be more advantageous than many short-living conference wikis.

In the following, we introduce the main novelties that a user encounters when using SMW instead of a simple MediaWiki.

3.1 Annotating pages

The necessary collection of semantic data in SMW is achieved by letting users add annotations to the wiki-text of articles via a special markup. Every article corresponds to exactly one ontological element (including classes and properties), and every annotation in an article makes statements about this single element. This locality is crucial for maintenance: if knowledge is reused in many places, users must still be able to understand where the information originally came from. Furthermore, all annotations refer to the (abstract) concept represented by a page, not to the HTML document. Formally, this is implemented by choosing appropriate URIs for articles.

Most of the annotations that occur in SMW correspond to simple *ABox statements* in OWL DL, i.e. they describe certain individuals by asserting relations between them, annotating them with data values, or classifying them. The schematic information (*TBox*) representable in SMW is intentionally shallow. The wiki is not intended as a general purpose ontology editor, since distributed ontology engineering and large-scale reasoning are currently problematic.[5]

Categories are a simple form of annotation that allows users to classify pages. Categories are already available in MediaWiki, and SMW merely endows them with a formal interpretation as OWL classes. To state that the article `ESWC2006` belongs to the category `Conference`, one just writes `[[Category:Conference]]` within the article `ESWC2006`.

Relations describe relationships between two articles by assigning annotations to existing links. For example, there is a relation `program chair` between `ESWC2006` and `York Sure`. To express this, users just edit the page `ESWC2006` to change the normal link `[[York Sure]]` into `[[program chair::York Sure]]`.

Attributes allow users to specify relationships of articles to things that are not articles. For example, one can state that `ESWC2006` started at June 11 2006 by writing `[[start date:=June 11 2006]]`. In most cases, a relation to a new page `June 11 2006` would not be desired. Also, the system should understand the meaning of the given date, and recognise equivalent values such as `2006-06-11`.

Annotations are usually not shown at the place where they are inserted. Category links appear only at the bottom of a page, relations are displayed like normal links, and attributes just show the given value. A *factbox* at the bottom of each page enables users to view all extracted annotations, but the main text remains undisturbed.

[4] WWW 2006, Edinburgh, Scotland and ESWC 2006, Budva, Montenegro

[5] However, SMW has been used in conjunction with more expressive background ontologies, which are then evaluated by external OWL inference engines [**?**].

It is obvious that the processing of Attributes requires some further information about the *Type* of the annotations. Integer numbers, strings, and dates all require different handling, and one needs to state that an attribute has a certain type. As explained above, every ontological element is represented as an article, and the same is true for categories, relations, and attributes. This also has the advantage that a *user documentation* can be written for each element of the vocabulary, which is crucial to enable consistent use of annotations.

The types that are available for attributes also have dedicated articles. In order to assign a type in the above example, we just need to state a relationship between `Attribute:start date` and `Type:Date`. This relation is called `has type` (in English SMW) and has a special built-in meaning.[6] SMW has a number of similar *special properties* that are used to specify certain technical aspects of the system, but most users can reuse existing annotations and do not have to worry about underlying definitions.

3.2 Querying and searching

Nobody will spend time on annotating a wiki if there is no immediate benefit for usage. We already mentioned that a factbox is displayed in each article, and this factbox also features quicklinks for browsing and searching. For example, attributes that denote geographic coordinates will produce links to online map services. However, the main advantage for users *within* the wiki is SMW's querying capability.

Users can search for articles using a simple query language that was developed based on the known syntax of the wiki. Indeed, the query for retrieving all articles that have York Sure as a program chair, one simply writes `[[program chair::York Sure]]`. In other words, the syntax for specifying an annotation is identical with the syntax for searching it. Multiple such query statements are interpreted conjunctively.

The query language becomes more powerful by allowing searches that include wildcards, ranges, and subqueries. For example, the query

```
[[Category:Conference]] [[start date:=>May 14 2006]]
    [[program chair::<q>[[member of::AIFB]]</q>]]
```

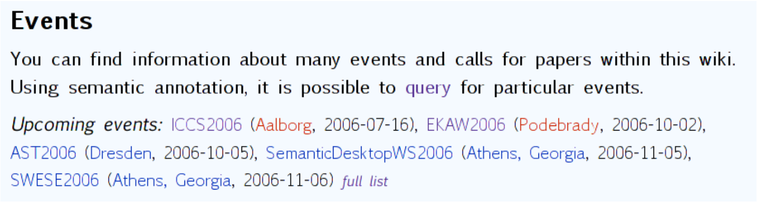
displays all conferences that started *after* May 14 2006 and which had a program chair from AIFB. We also remark that queries for category membership do a limited (sound but not complete) form of reasoning to take subclass relationships into account. Further information on the query language is found in the online documentation.[7]

3.3 Dynamic pages

The query functionality of SMW can be used to embed dynamic content into pages, which is a major advantage over traditional wikis. To do so, an *inline query* is written in wiki-text and enclosed in `<ask>` and `</ask>`. The article then shows the results of the given query at this position. Moreover, the query syntax involves statements for displaying further properties of the retrieved results, and for modifying the appearance within the page.

[6] Also, it is treated as an owl:AnnotationProperty in order to stay in OWL DL.

[7] `http://ontoworld.org/wiki/Help:Inline_queries`



Events

You can find information about many events and calls for papers within this wiki. Using semantic annotation, it is possible to query for particular events.

Upcoming events: ICCS2006 (Aalborg, 2006-07-16), EKAW2006 (Podebrady, 2006-10-02), AST2006 (Dresden, 2006-10-05), SemanticDesktopWS2006 (Athens, Georgia, 2006-11-05), SWESE2006 (Athens, Georgia, 2006-11-06) *full list*

Fig. 1. Dynamic content on the main page of ontoworld.org.

For example, the *Main Page* of ontoworld.org displays upcoming events, their dates and locations (see Fig. **??**). Those results are generated dynamically by looking for conferences starting after the *current* date, ordering them according to their dates, and printing the first five results.

Many other forms of inline queries can be found in the online documentation.

4 Reuse in the Semantic Web

4.1 Mapping to OWL DL

In this section, we describe how annotations in SMW are formally grounded in OWL. As explained above, every article represents an ontological element, i.e. an element of one of the RDF classes `Thing`, `Class`, `ObjectProperty`, `DatatypeProperty`, and `AnnotationProperty`. Moreover, every article needs a URI which is different from its URL in order to prevent confusion of concepts and HTML pages. We map each URL injectively to a URI which, when requested in a browser, will still be redirected to the original URL.[8]

The type of the elements is fixed for most kind of annotations. Normal articles are just OWL individuals, categories become classes, and relations become object properties between articles. Attributes might be datatype, annotation, or object properties, depending on their type within the wiki.

Based on this mapping, SMW generates OWL/RDF for any page on request. The simplest way to access this RDF is to use the link “View as RDF” at the bottom of each annotated page. More elaborate export settings, which also allow bulk export, backlink inclusion, and recursive export, are found on a dedicated special page.[9] This page also serves as an endpoint for external services that want to access the wiki’s semantic data.

4.2 Reusing existing ontologies

Since SMW is compatible with the OWL DL knowledge model, it is also feasible to use existing ontologies within the wiki. This is possible in two ways: *ontology import* is a feature that allows to create and modify pages in the wiki to represent the relationships

[8] This does not work in all cases, since OWL/RDF requires the use of URIs as XML ids, which in turn cannot use all characters allowed in URLs.

[9] See `http://ontoworld.org/wiki/Special:ExportRDF`.

that are given in some existing OWL DL document; *vocabulary reuse* allows users to map wiki pages to elements of existing ontologies.

The ontology import feature employs the RAP[10] toolkit for reading RDF documents, and extracts statements that can be represented in the wiki. Article names for imported elements are derived from their labels, or, if no labels are available, from the section identifier of their URI. The main purpose of the import is to bootstrap a skeleton for filling the wiki. Also, ontology import inserts special annotations that generate equivalence statements in the OWL export (i.e. owl:sameAs, owl:equivalentClass, or owl:equivalentProperty). Importing ontologies is only allowed for site administrators, since it could otherwise be used to spam the wiki with thousands of new articles.

Importing vocabulary allows users to identify elements of the wiki with elements of existing ontologies. For example, the `Category:Person` in our online example is directly exported as the class `foaf:Person` of the Friend-Of-A-Friend vocabulary. Wiki users can decide which pages of the wiki should have an external semantics, but the set of available external elements is explicitly provided by administrator users. By making some vocabulary element known to the wiki, they ensure that vocabulary reuse respects the type constraints of OWL DL. For example external classes such as `foaf:Person` cannot be imported as Relations.

4.3 External reuse in practice

OWL/RDF export is a means of allowing external reuse of wiki data, but only practical application of this feature can show the quality of the generated RDF. To this end, we have employed a number of Semantic Web tools to the RDF output. SMW cooperates nicely with most tested applications, such as *FOAF Explorer*, the *Tabulator* RDF browser, or the *Piggy Bank* RDF browser extension. Details on the tested tools, including their basic functionality and URL, are given at ontoworld.org.[11]

Moreover, we provide an externally hosted SPARQL querying service.[12] The system is based on the stand-alone RDF server *Joseki*[13] that is synchronised with the semantic content of the wiki. Synchronisation employs the wiki's RSS feed for reporting recent changes to reload changed articles quickly. The SPARQL endpoint thus demonstrates that it is feasible to mirror the wiki's RDF content via small incremental updates, and offers another access point for semantic technologies to reuse the data.

5 Wikipedia

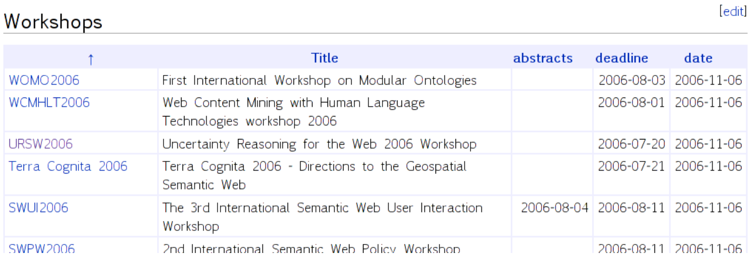
The most important usage scenario for SMW are the various project sites of the *Wikimedia foundation*, especially the different language versions of *Wikipedia*. We already mentioned that ease of use and scalability are central for this application, and SMW has been built to fulfil these requirements. On the other hand, Wikipedia has no reason to use a novel extension at all if it does not bring immediate advantages. However,

[10] RDF API for PHP, `http://www.wiwiss.fu-berlin.de/suhl/bizer/rdfapi/`

[11] See `http://ontoworld.org/wiki/SMW_reuse`

[12] See `http://ontoworld.org/wiki/SPARQL_endpoint`

[13] See `http://www.joseki.org/`



Workshops

[edit]

↑	Title	abstracts	deadline	date
WOMO2006	First International Workshop on Modular Ontologies		2006-08-03	2006-11-06
WCMHLT2006	Web Content Mining with Human Language Technologies workshop 2006		2006-08-01	2006-11-06
URSW2006	Uncertainty Reasoning for the Web 2006 Workshop		2006-07-20	2006-11-06
Terra Cognita 2006	Terra Cognita 2006 - Directions to the Geospatial Semantic Web		2006-07-21	2006-11-06
SWUI2006	The 3rd International Semantic Web User Interaction Workshop	2006-08-04	2006-08-11	2006-11-06
SWPW2006	2nd International Semantic Web Policy Workshop		2006-08-11	2006-11-06

Fig. 2. Excerpt of an automatically generated list on ontoworld.org.

the core wiki problems presented in Sect. **??** all have very concrete consequences for today's Wikipedia.

Consistency of content is a major aspect of the overall quality of Wikipedia. Currently, articles like the "List of cities in Russia by population" are edited manually. If any of the given population numbers changes, a number of articles have to be updated. The dynamic creation of pages described in Sect. **??** solves many instances of this problem, e.g. by generating tables as in Fig. **??**.

Accessing knowledge in Wikipedia can be extremely difficult due to its sheer size. For example, the reader may want to try to find a list of all *female physicists* in Wikipedia. Keyword searches fail miserably at this task: *Marie Curie* does not appear at all when the term "female" is used, but the majority of the returned articles do not even describe people at all. A simple query as described in Sect. **??** would immediately solve the problem.

Reusing knowledge contained in Wikipedia is very desirable as it is the largest human-edited source of information. Currently, tools such as the media player *Amarok* (`amarok.kde.org`) embed whole articles into their GUI to allow users to access the information. While those implementations show the need for external reuse, they are currently limited to the mere reproduction of Wikipedia's page content.

We thus believe that we address core problems of Wikipedia as a whole, and hope that the extension can be introduced to Wikipedia soon. SMW was discussed among Wikipedians and developers at the recent 2nd International Wikimedia Conference in Boston (Wikimania2006), and large-scale evaluations with a copy of Wikipedia are now being made.

The opportunities of a semantic Wikipedia are huge. Wikipedia covers a large number of knowledge areas, and even the use of Wikipedia as a resource for thoroughly documented URIs would be helpful. With the added semantic information, Wikipedia's knowledge could be exploited in numerous applications. The scientific relevance of large annotated text corpora, and of real-life knowledge models that really represent "shared conceptualisations" would be enormous. Formalised knowledge in different language Wikipedias can be a basis for investigating cultural differences, but independent semantic wikis also provide intersting sources of data for ontology alignment.

While SMW currently uses only simple annotations, users start to articulate the need for more expressive features. However, integration is not trivial at all, and tractability, explanation components, local closed world assumption, paraconsistent reasoning, and other issues might play an important role in extending the expressive power of the wiki.

Beyond such scientific challenges, Wikipedia can also enable a new range of tools based on semantic technologies. As mentioned in Sect. **??** we employed a number of tools, but we were also disappointed by the immature state of many of them. We assume that this is due to the lack of interesting semantic data on the web, and believe that Wikipedia can stimulate much more intensive development of semantic tools.

6 Further use cases

SMW is currently used in a number of real-life applications and has an active user community. This can also be seen from the many requests to our public mailing lists and the more than 200 registered users of ontoworld.org (not counting the spammers). Moreover, a number of external sites have already introduced SMW into their wikis. Examples are provided online [**?**], but not all SMW installations are open to public access (e.g. AIFB and FZI use SMW for internal knowledge management). Finally, SMW has been used as a local tool for personal information management on the desktop.

7 Outlook

In this short note, we could only sketch the core features and uses of the Semantic MediaWiki system. Other features, such as the use of *semantic templates* or the support for *units of measurement*, could not be introduced at all. We therefore suggest the reader to refer to the more detailed online documentation [**?**]. Semantic MediaWiki currently is developed very actively. At the time of this writing, improved browsing capabilities, easy generation of RDF dumps, and support for Timeline[14] output are planned for the next version. Yet, the overall focus remains on producing a stable, scalable, and usable implementation of the core features that will be employed in Wikipedia.

Acknowledgements. We wish to thank all past and present contributors and users of SMW, in particular the volunteer developers S Page and Kai Hüner, and *doccheck* GmbH, Germany. Research reported in this work has been partially supported by the European Union in projects SEKT (IST-2003-506826) and NEPOMUK (IST 027705).

[14] See `http://simile.mit.edu/timeline/`