

Semantic MediaWiki in Operation: Experiences with Building a Semantic Portal

Daniel M. Herzig and Basil Ell

Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
{herzig, basil.ell}@kit.edu
<http://www.aifb.kit.edu>

Abstract. Wikis allow users to collaboratively create and maintain content. Semantic wikis, which provide the additional means to annotate the content semantically and thereby allow to structure it, experience an enormous increase in popularity, because structured data is more usable and thus more valuable than unstructured data. As an illustration of leveraging the advantages of semantic wikis for semantic portals, we report on the experience with building the AIFB portal based on Semantic MediaWiki. We discuss the design, in particular how free, wiki-style semantic annotations and guided input along a predefined schema can be combined to create a flexible, extensible, and structured knowledge representation. How this structured data evolved over time and its flexibility regarding changes are subsequently discussed and illustrated by statistics based on actual operational data of the portal. Further, the features exploiting the structured data and the benefits they provide are presented. Since all benefits have its costs, we conducted a performance study of the Semantic MediaWiki and compare it to MediaWiki, the non-semantic base platform. Finally we show how existing caching techniques can be applied to increase the performance.

1 Introduction

Web portals are entry points for information presentation and exchange over the Internet about a certain topic or organization, usually powered by a community. Leveraging semantic technologies for portals and exploiting semantic content has been proven useful in the past [1] and especially the aspect of providing semantic data got a lot of attention lately due to the Linked Open Data initiative. However, these former approaches of semantic portals put an emphasis on formal ontologies, which need to be build prior to the application by a knowledge engineer resulting in formal consistent and expressive background knowledge [1, 2]. This rather laborious process yields further efforts when changes and adjustments are required. Beside this disadvantage, [3] points out that versioning of the structured knowledge is missing and the community features are essential, but insufficient. Recently, [4] showed how the popular content management system *Drupal*, which will support semantic data from version 7, can be applied for

building semantic applications. We pursue an alternative approach, leveraging communities for the creation and maintenance of data.

One of the most successful techniques to power communities of interest on the web are wikis. Wikis allow users to collaboratively create and maintain mainly textual, unstructured content. The main idea behind a wiki is to encourage people to contribute by making it as easy as possible to participate. The content is developed in a community-driven way. It is the community that controls content development and maintenance processes. Semantic wikis allow to annotate the content in order to add structure. This structure allows to regard the wiki as a semi-structured database and to query its structured content in order to exploit the wiki's data and to create various views on that data. Thus wikis become even more powerful content management systems. Moreover due to the semantic annotations, structured content becomes available for mashups with semantic content residing outside the wiki, for example as Linked Open Data.

In this paper we describe how we use the semantic wiki *Semantic MediaWiki*¹ [5, 6] (SMW) for creating a portal for our institute, which can be accessed at <http://www.aifb.kit.edu>. The portal manages the web presence of the AIFB institute, an academic institution with about 150 members. The portal is a semantic web application with about 16.7k pages holding 105k semantic annotations. Table 1 gives an overview in numbers of the portal. While wikis provide free, wiki-style semantic annotations with the complete freedom regarding the adherence to any vocabulary, users can be guided to adhere to use certain vocabularies by providing form-based input. The importance of the right balance between unstructured content, which is better than no content, and structured content, which is more efficient to use, was studied already by [2]. However, this approach focussed on automatic crawling of structured data and did not regard the user as the primary provider.

This paper is structured as follows: In Section 2 we report on design and development decisions and in particular discuss the free, wiki-style editing versus guided user input. Further, we report on the development efforts and on the subsequent usage and maintenance. In Section 3 we show the advantages and features made possible by the semantics of the portal. And finally, we report in Section 4 on performance tests and compare Semantic MediaWiki to its non-semantic base platform MediaWiki, before we conclude in Section 5.

2 Designing and Developing the Portal

The most common and original application of Semantic MediaWiki and wiki systems in general is collaborative knowledge management, e.g. for communities such as semanticweb.org. In this section we present the portal we built using Semantic MediaWiki and in particular its features exploiting semantic technologies.

¹ <http://semantic-mediawiki.org/>

2.1 Free Annotations versus Guided Input

A wiki provides the users with the means of rather easy and unconstrained adding and changing of content, in the sense that they just need to know the simple wiki markup and have a web browser. Further, the users can publish content themselves without the assistance of a webmaster. One aim when designing the portal was to have low barriers for the institute's members to contribute, extend and maintain the content. Hence, we considered a wiki as an appropriate choice. In contrast to regular wiki systems, Semantic MediaWiki allows to semantically annotate the content. This free and independent annotation paradigm has the advantage of being flexible, and expandable. Moreover, it does not require the knowledge of a predefined schema. The underlying notion is that more annotations are in general better than less annotations even if they are not well organized and do not follow a predefined vocabulary or ontology. However, when using *inline queries*, see Section 3.1, one has to know the exact property names, since formal queries are strictly sensitive and minor derivations are not tolerated. The same is true for many applications building on top of the structured data. They are often build on a specific schema or vocabulary. Thus, one has to find the right balance between a predefined schema and keeping it flexible and expandable at the same time. For the case of Semantic MediaWiki, *templates* and *forms*² allow to restrict the user to a predefined set of annotations. A template defines the logic and the appearance of a part of a page. It keeps placeholder variables, which are filled by the instantiating page. Inserting annotations in the template entails the annotation of all pages using the template with the same annotations. Consequently, changing the annotation inside a template cascades this change to all pages and thereby allows a flexible modification of the structured data. Forms provide a graphical user interface for using templates correctly and do not even require the usage of wiki markup. Thereby the combination of forms and templates allows to have a set of predefined annotations.

For the portal, we created about 30 templates and corresponding forms for all major, reoccurring resources, like *people*, *lectures*, *publications*, and so on. Figure 1 shows an example form for editing a page about a project. Forms can consist of different types of fields, e.g. for text, dates, choices, etc. Behind each field is an annotation, i.e. a property. By entering a value in a field, the value is assigned to the corresponding property. It is good practice to import these properties from already existing vocabularies³, e.g. FOAF⁴ for persons, if applicable. In order to keep the possibility for free, unconstraint annotations, the forms can contain text areas, which can contain text with arbitrary annotations. Thereby, we tried to find a balance between guided input with predefined annotations and the possibility to have free annotations.

The advantage of this mixture of guided input and open annotations is that the structure of the data can evolve dynamically, which we report on in Section 2.3.

² http://www.mediawiki.org/wiki/Extension:Semantic_Forms

³ http://semantic-mediawiki.org/wiki/Help:Import_vocabulary

⁴ <http://xmlns.com/foaf/spec/>

The figure consists of two side-by-side screenshots. The left screenshot shows a web form titled 'Projekt' with several input fields: 'Kurzname:', 'Name:', 'Name EN:', 'Beschreibung DE:', 'Beschreibung EN:', 'Kontaktperson:', 'URL:', and 'Start:'. Each field has a text input area and a small 'Hilfe (öffnet neue Seite)' link below it. The 'Beschreibung DE' and 'Beschreibung EN' fields contain pre-filled text. The 'Start:' field has a date picker set to April 2009. The right screenshot shows a browser window with a rich text editor. The text is in German and includes a title, a date, and a paragraph of text. The text is formatted with bold tags and a center-align tag. At the bottom, there are several small icons and a 'Zusammenfassung:' field.

Fig. 1. The left side shows guided input via a form. The form consists of several different input fields. The content of each field is assigned to a predefined annotation. The form holds also free text areas, which can contain text with arbitrary annotations. The right hand side shows entirely free, wiki-style editing without any constraints regarding predefined annotations.

2.2 User Roles

When we proposed a wiki system for the portal, the first concern from colleagues was the fear that *everybody* can edit it, even *anonymously*. This is of course not the case. We used MediaWiki's internal user right management⁵ to create four different groups: The anonymous web surfers can only read regular pages, i.e. those in the main namespace. The authenticated users may also read pages in other namespaces and in addition are allowed to edit pages, except for pages in the template and form namespace. The latter can only be manipulated by admins. The fourth group are bureaucrats, which have the same rights as admins, but in addition they can appoint and withdraw the admin right.

Since having an extra user account might impose a barrier for people to participate, we used the *Lightweight Directory Access Protocol* (LDAP) extension⁶ for MediaWiki and an SSL encrypted connection between the portal and the LDAP server. This allows to use already existing user accounts for the authentication at the portal.

⁵ http://www.mediawiki.org/wiki/Manual:User_rights

⁶ http://www.mediawiki.org/wiki/Extension:LDAP_Authentication

2.3 Development and Dynamics of the Structured Knowledge Representation

The development efforts of the portal can be broadly separated into four different areas: system setup, visual design and custom function development, data import, and finally template development, which comprises modeling the structured data, i.e. the properties and classes.

Setting up a SemanticMediaWiki takes less than one hour⁷ and developing a so-called *skin*, i.e. the look and feel of the platform, depends on the given specifications. In our case, it took a student developer about 80 hours to meet our organization's 148 pages in length design guideline.

In order to measure the efforts of the development, we counted the edits, i.e. the revisions, of artifacts and the newly created artifacts over time. Figures 2 and 3 show these counts per month, where each point represent the count accumulated over the month marked on the horizontal axis. The life cycle phases, i.e. development, internal release for testing, and release into production are illustrated in the figures as well.

Dynamics of the Structured Knowledge Representation As discussed in Section 2.1, Semantic MediaWiki provides the means to keep a flexible, structured data schema consisting of properties and classes. Figure 2 shows how these elements changed over time and how often new elements were added. Categories group pages and correspond to classes in the structured representation. Regarding the manipulation of classes and properties, one can see that most of the structured data layout was done at the very beginning of the project in April 2009. In particular, the classes involved were known right from the beginning and relatively few changes were needed during the subsequent phases. The same holds for the properties in an alleviated form. Still, one can see that a small, but steady number of properties and classes were added or changed over the course of the project with the exception of the peaks in March 2010. In this month, the annual institute report about events, publications, and people was prepared. The data for the report was exported from the portal. Since the editors requested changes and additions to the data, e.g. splitting names into first and last name or adding the location of publication to some publication types, we needed to change the structure in the portal accordingly. In particular, the class structure underwent refactoring, e.g. splitting the class *employee* into former and active members.

All these adjustments were done in an agile way driven only by requirements and demands. In particular, one has to keep in mind that all changes happen on the application level. Touching the underlying database was never necessary nor taking the system offline for modifications. Furthermore, the wiki provides a versioning system, which tracks all changes, also those of properties and classes, a crucial capability for semantic portals [3].

⁷ http://www.mediawiki.org/wiki/Manual:FAQ#How_do_I_install_MediaWiki.3F

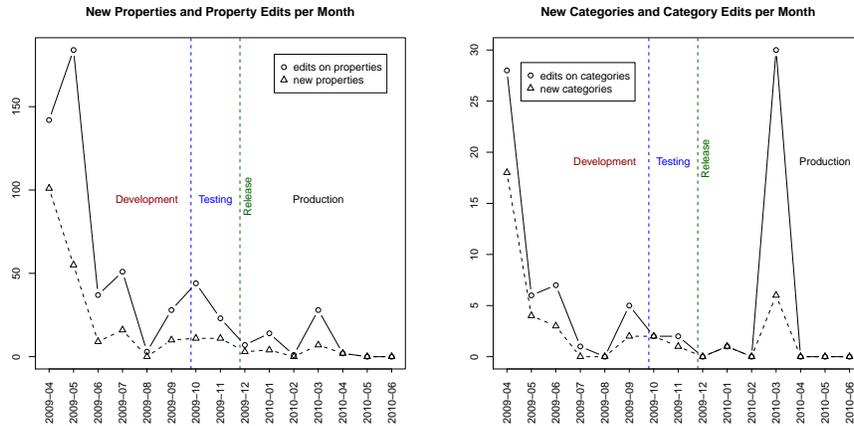


Fig. 2. The plot on the left side shows the number of new property types and edits on property types per month. The plot on the right side shows the number of new categories and edits on categories. Categories correspond to classes of the structured data. Since properties and classes are the elements of the structured data, these plots show the evolution of the structured data over time.

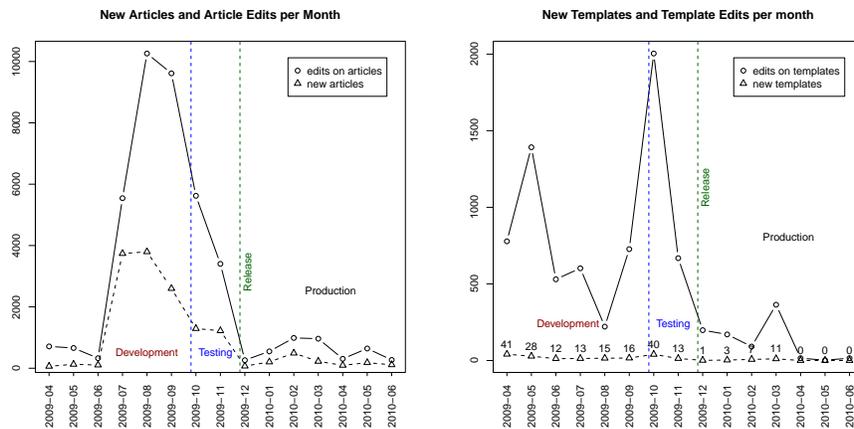


Fig. 3. The plot on the left side shows the number of new articles and edits on articles per month for the different periods from development to production. The high numbers during the development phase are due to automatic batch jobs populating the portal with content. The plot on the right side shows new templates and edits on templates. Since these can only be edited by admins, this plot allows to estimate the development effort as well as the maintenance effort after the release. The peak in March 2010 was the result of implementing the annual reporting, see Section 2.3.

pages	16.716
templates	219
forms	30
uploaded files	1.773 (1.2 GB)
users (total)	142
active users (last 91 days)	83
annotations (property instances)	104.182
property types	191
categories (classes)	40
OWL/RDF	238k triples
code base	132 MB
database	99.5 MB

Table 1. The portal in numbers as of June 2010.

Data Migration and Template Development In order to populate the platform, we used the Pywikipediabot framework⁸, a Python application for manipulating wiki pages via scripts. The loading of the existing data into the platform explains the high peak on the left plot of Figure 3. Creating the templates was the most time consuming task. However, the peak at the start of the testing period is solely due to the tidy visual requirements.

Usage and Maintenance Since the release, we observe a steady user participation with an average of 550 edits/month on articles and about 195 new pages/month, as shown in the production period in left plot of Figure 3. About 83 users or 66% of the full time employees of our institute contributed within the last 3 months, i.e. April 18th to June 18th 2010. At the same time, manipulation of templates declined constantly, from 200 to less than 10 edits/month, which suggests that the maintenance by the admins is within reasonable bounds, which can be seen in the right plot of Figure 3.

2.4 Multilingual Content

MediaWiki per se is monolingual and uses *interwiki-links* to point to another wiki holding an article on the same topic in a different language. Since we wanted users to have one single point of data entry, we abstained from setting up a wiki in each language. However, we needed an English and German view on our web presence. Therefore, we chose to create subpages for the English version of a German page by adding */en* to the page name. The users add the German and the English content via one form for predefined resources.

⁸ <http://meta.wikimedia.org/wiki/Pywikipediabot>

2.5 Challenges during Design and Development

The biggest challenge during the development was the creation of templates and trimming them to the strict design guidelines. Whereas it is acceptable for most wiki applications to have little blemish and accept the free and sometimes untidy appearance, an official web presence should avoid it, e.g. all empty variables in templates needed to be hidden. Moreover, due to the tidy appearance requirement, some annotations contain markup, e.g. for italic font style or font size, which is desirable in the structured data representation. Furthermore, the templates combine the description of the appearance of a page and the logic at the same time, which makes them complex and overcharged and require advanced knowledge of the wiki markup for further development and maintenance. Therefore, template manipulations are restricted to admins in our portal.

3 Where Semantics Help - Features of Semantic MediaWiki

In the previous section we reported on the development process and the dynamics of the structured data. In this section, we show the features taking advantage of the structured data.

3.1 Inline Queries

The biggest advantage of SMW, beside its flexible annotation paradigm, is the possibility to reuse data across the platform by querying it from other pages. These *inline queries* allow to request sets of data or just single property values and display them on a page in various result formats, such as tables, list, charts, maps, etc. This reuse of data avoids data redundancy, e.g. the information about a person, like name, email, or phone number, is entered once on the page about this person and then later this information is queried and displayed on pages about projects, publications, etc., where this person is involved in. If the data changes on the source page, the data on the requesting page changes accordingly when the inline query is executed again. Inline queries create dynamic pages. Figure 4 illustrates an example of an inline query and its results as it appears on the requesting page.

3.2 Querying Linked Open Data Sources

We created an extension that allows querying external sources using the simple syntax of inline queries [7]. This mediation-based approach allows for either displaying or importing externally retrieved data from the Linked Open Data source Freebase, other SMWs, or from CSV files, in order to enrich the wiki's content with external data. In the first two cases a mediator translates an inline query into a query in the query language supported by the remote source, which is MQL in the case of Freebase. Figure 5 illustrates an example. Translation is

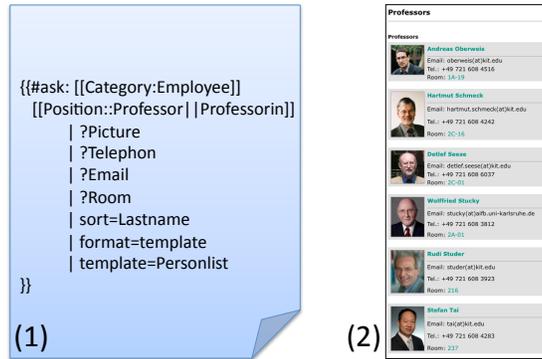


Fig. 4. An inline query requesting all employees, which are professors, and information about them (1) and its result representation (2).

not a task of solely syntactical transformation but also involves ontology mapping. The mappings are stored in the wiki as annotations. Thus they can be contributed and maintained by users.

In our portal we query the SMW of semanticweb.org in order to retrieve events, such as conferences or workshops and present them on a timeline in order to offer visitors of our page an interactive conference radar with up-to-date information. Moreover, we are using Freebase to retrieve location information about the institute's industrial and academic partners, in order to be able to sort them by region.

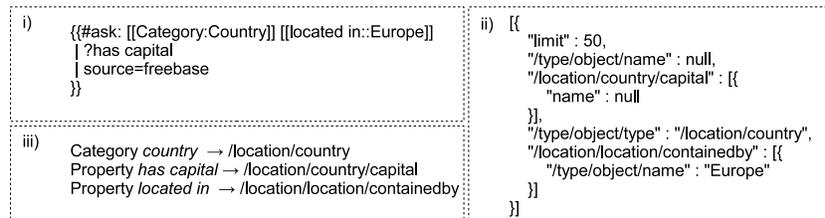


Fig. 5. Using the Freebase mediator an inline query such as in i) is translated into an MQL query such as in ii) by using the mapping information such as in iii).

3.3 Exploiting the Semantics for Search

One certain advantage of having the content of the portal available in a structured form is the ability to exploit it for search. [8] presents an approach for

semantic search in wikis, which we apply for the portal⁹. This approach allows to use keywords as the means to express an information need, because most users are used to this common search paradigm. These keywords are then transformed into interpretations using the structured data of the wiki as the search space. The interpretations are shown to the user, who can select the interpretation fitting best to his information need and further refine it in the next step. Figure 6 shows an example search over the structured data for employees, their email addresses, and office location. In contrast to the inline queries, which use a simple, but formal query syntax and are therefore inadequate for ad-hoc search, *Ask The Wiki* is suitable for end users and exploits the semantic annotations.

The screenshot shows the 'Ask The Wiki' interface. At the top, there are three steps: 'Step 1: Enter keywords', 'Step 2: Choose interpretation', and 'Step 3: View and refine results'. Below the steps, there are instructions: 'There are 108 results matching your interpretation.' and 'Use the Facets to the right to expand or narrow the results.' A legend indicates 'Concepts', 'Relations', 'Labels', and 'Literals'. The main content area shows a diagram with four concepts: 'Mitarbeiter', 'mbox', 'Raum', and 'Gebäude'. Arrows indicate relationships: 'Mitarbeiter' is linked to 'mbox' and 'Raum'; 'mbox' is linked to 'Raum'; 'Raum' is linked to 'Gebäude'. Below the diagram is a table with three rows of data:

Mitarbeiter	mbox	Raum	Gebäude
Thomas Karle	thomas.karle(at)aifb.uni-karlsruhe.de	1A-13	05.20
Alaa Ismaeel	als(at)aifb.uni-karlsruhe.de	2B-09	05.20
Duc Thanh Tran	ducthanh.tran@kit.edu	260	11.40

On the right side, there is a 'Facets' menu for 'Mitarbeiter' with various options like 'workplaceHomepage', 'sameAs', 'depiction', 'Land', 'Public Key', 'Stellung', 'Info', 'mbox', 'Bundesland', 'lastName', 'Mobile', and 'Telefax'.

Fig. 6. This figure shows the result of a search for all employees, their emails, room numbers, and corresponding building numbers. The facets menu on the right hand side allows to refine the result based on the structured data.

4 Performance

MediaWiki, the platform powering Wikipedia, runs on many sites and is well known for being scalable and fast. Although the usefulness of the features provided by Semantic MediaWiki get the interest of many potential users, often skepticism about SMW's resource requirements, its stability and scalability are brought forward. In this section, we report on stress tests conducted on both Semantic MediaWiki and MediaWiki with the data from the portal in order to allow for their comparison.

⁹ <http://www.aifb.kit.edu/web/Spezial:ATWSpecialSearch>

Test Environment We performed the tests with a common desktop computer, which has one CPU with 2GHz, 2 GB memory and runs on Debian 5.0¹⁰. The wiki runs on an Apache web server with PHP 5 and uses a MySQL database¹¹. The load tests are conducted with Apache JMeter and the server was monitored by sysstat¹². The client sending the requests was connected through the same 100 MBit backbone to the server.

This system configuration is a common single machine setting and by no means laid out for high-performance. However, it allows to compare the two systems, MediaWiki 1.15.3 and Semantic MediaWiki 1.5. The wiki holds the data of the AIFB portal, see Table 1 for an overview. The data contains annotations in the SMW syntax. If SMW is not enabled, MediaWiki interprets these statements just as text. SMW allows to restrict the usage of inline queries, e.g. the maximal number of conditions, query depth, and maximal retrieved results, in order keep reasonable bounds. However, these settings were all set to unlimited for the tests. The response time at the client side, i.e. from sending the request until receiving the response, is taken as the performance metric. For the measurements, 310 pages ($\sim 2\%$ of all pages) accessible from the main page within 2 clicks were chosen. The pages are a representative subset of all pages, ranging from pages with little semantic annotations and queries to pages that make heavy use of these features. On average a page holds 10 inline queries and 12 semantic annotations.

4.1 MediaWiki vs. Semantic MediaWiki

A test consisted of N parallel users requesting the 310 pages in random order. Figure 7 shows a box-plot illustrating the response times in milliseconds for MediaWiki (MW) and Semantic MediaWiki (SMW), and Semantic MediaWiki with Caching (SMW+C) for $N = \{1, 10, 25, 50\}$ parallel users. It shows that the response times are linear with respect to the number of parallel users. This linear behavior becomes apparent in Table 4.1, which shows the throughput, i.e. the number of served requests per second. The throughput is constant at about 4.7 requests/sec for MW and 4.1 requests/sec for SMW, which means that using SMW costs about 13% in performance compared to MW during operation.

However, it is unexpected that the spread of the response times over the 310 pages is so little, as illustrated in the box-plot of Figure 7. Especially in the case of SMW, the pages contain semantic annotations and in particular inline queries, which need to be parsed and processed. The response time should depend on the number and complexity of these queries. This low spread is due to the implicit caching. The web server has a build-in PHP code cache (APC), which MW and therefore also SMW exploits. Also, a page is not rendered for each request, but only when necessary. In addition, the database caches requests (InnoDB). All these build-in caches absorb most of the additional overhead by SMW during regular operation and make it possible to run SMW at a constant cost of about

¹⁰ AMD Athlon 64 3200+, 2.6.26-2-amd64 kernel

¹¹ Apache 2.2.9, PHP 5.2.6-1+lenny3 with APC 3.0.19, MySQL 5.0.32

¹² Apache JMeter 2.3.4, sysstat 7.0

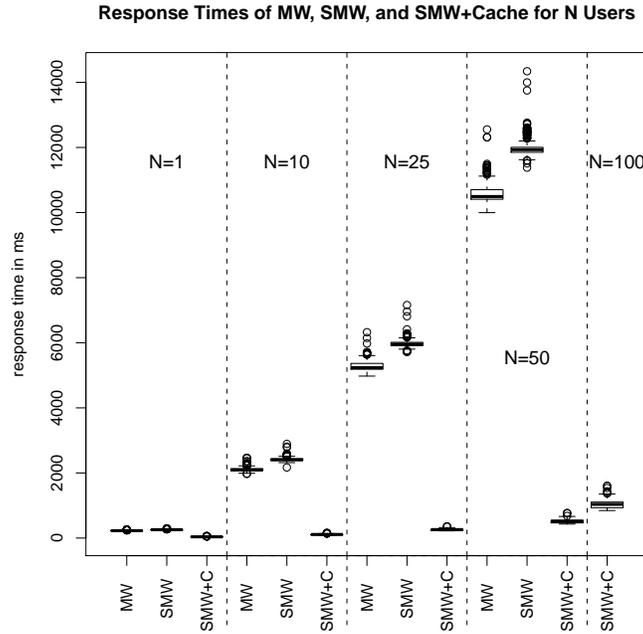


Fig. 7. Box-plot illustrating the response time per page for MW, SMW, SMW+Cache with N parallel users requesting 310 pages in random order.

13% in performance decrease compared to the non-semantic MediaWiki, see Table 4.1. The bottle neck resource during these tests was the CPU for MW and SMW for all runs with more than one user. The CPU was consumed for about 95% by the web server and 5% by the database.

N	1	10	25	50	100
MW	4.36	4.75	4.75	4.73	n/a
SMW	3.83 (-12 %)	4.10 (-14%)	4.13 (-13 %)	4.13 (-13%)	n/a
SMW+C	> 25.68 (+489%)	> 90.80 (+1810%)	> 96.78 (+1930%)	> 96.31 (+1930%)	> 95.01

Table 2. Throughput (requests/sec) for N parallel users. The percentages are compared to the MediaWiki (MW) baseline. When applying the cache (SMW+C) the server’s limits were not met.

In order to avoid the implicit caching behavior and to assess the actual resource requirements, we performed a *cold* test run, i.e. we restarted the machine after each page was requested once, and repeated this for 10 times. Figure 8 shows the average response time over the 10 runs for each page. The pages are sorted by number of inline queries in ascending order, which is displayed on the

sources outside the wiki when querying via mediator. The page holding the inline query is dynamic in the sense that its appearance and displayed content changes although the source code of the page does not. Therefore, we encountered the problem that the `Last-Modified` entry in the HTTP header remained the same, because the web server did not recognize a change. Since this entry is used by the cache to determine whether a page is still fresh, we needed to modify the caching mechanism. We chose an aggressive caching strategy by suppressing the `Last-Modified` entry and set a hard maximal expiration time of 3 hours for pages. Thereby, implicit changes of a dynamic page will be updated within this period at the latest. When a page is edited directly, it is immediately purged from the cache. Images and other static content is cached for longer periods. Applying the cache yields a huge performance increase to about 90 requests/sec as shown in Table 4.1 and Figure 7. However this is by far not the limit, since the CPU was used to about 30%, even for 100 parallel users. One needs to setup multiple physical clients sending requests to asses the actual limit when using a cache, which was beyond our scope.

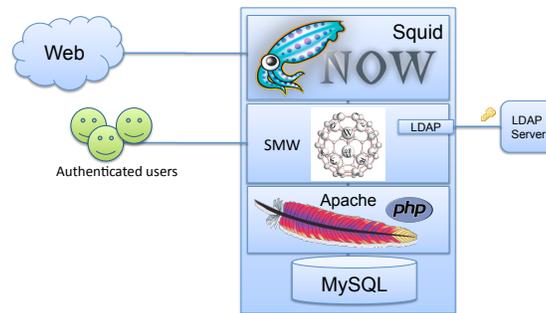


Fig. 9. The infrastructure stack of the portal. Anonymous readers get the content served from the Squid cache, if available. Authenticated users are directly connected to the web server.

4.3 Performance in Operation

The portal is online since more than six month now with only one interruption due to an DoS attack, which we addressed by restricting the number of connections to the web server. Therefore, we regard the solution as stable and quite robust. There are between 60k and 120k hits per day, which results in an average CPU usage of 6% and an average load of 0.1 on our production web server, which has 2 CPUs¹⁴ and 2GB memory. The median response time, in this case the time between arrival of a request at the server and the sending of

¹⁴ Intel(R) Xeon(R) CPU E5450 @ 3.00GHz

the response, is about 2 ms on average, where cache hits take slightly less than 2ms and cache misses take about 200ms to serve on average. The cache has a request hit ratio of about 80% and a size of about 550MB on disk.

5 Conclusion

In this paper we have shown how to apply the wiki paradigm of collaborative editing to a web portal using semantic technologies. We discussed how free, unconstrained annotations can be combined with predefined annotations in order to allow flexible and expendable structured data. Further, we reported on how the structured data made available by semantic annotations evolved over time and that it was possible to extend and change it during operation without touching the underlying database. How the semantic data is used and taken advantage of by Semantic MediaWiki's features is illustrated by several examples. Finally, we evaluated the performance and compared it to its non-semantic alternative and showed how caching can be applied to boost the performance. Taking everything in consideration, one can see how Semantic MediaWiki can be used as a successful portal platform providing the advantages of semantic technologies.

6 Acknowledgments

Special thanks go foremost to Martin Zang, whose dedication to this project contributed a big part to its success. Also Nicole Arlt and Fabio Garzotto are thanked for their commitment as well as Philipp Sorg and the IT team at AIFB for their valuable feedback and technical support. Work presented in this paper has been funded by the EU IST FP7 project ACTIVE under grant 215040.

References

1. Maedche, A., Staab, S., Stojanovic, N., Studer, R., Sure, Y.: Semantic portal - the seal approach. In Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., eds.: *Spinning the Semantic Web*. MIT Press, Cambridge, MA. (2003) 317–359
2. Hotho, A., Maedche, A., Staab, S., Studer, R.: Seal-II the soft spot between richly structured and unstructured knowledge. *Journal of Universal Computer Science* **7**(7) (2001) 566–590
3. Lara, R., Han, S.K., Lausen, H., Stollberg, M., Ding, Y., Fensel, D.: An evaluation of semantic web portals. In: *IADIS Applied Computing International Conference*. (2004) 23 – 26
4. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Decker, S.: Produce and consume linked data with drupal! In: *8th International Semantic Web Conference (ISWC2009)*. Volume 5823 of LNCS., Springer (2009)
5. Krötzsch, M., Vrandečić, D., Völkel, M., Haller, H., Studer, R.: Semantic wikipedia. *Journal of Web Semantics* **5**(4) (2007) 251–261
6. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic mediawiki. In: *Proceedings of the 5th International Semantic Web Conference (ISWC2006)*. Volume 4273 of LNCS., Springer (2006) 935–942

7. Ell, B.: Integration of external data in semantic wikis. Master's thesis, Hochschule Mannheim (2009)
8. Haase, P., Herzig, D.M., Musen, M., Tran, D.T.: Semantic wiki search. In: 6th European Semantic Web Conference (ESWC2009). Volume 5554 of LNCS., Springer Verlag (2009) 445–460