Semantic Wikipedia

Markus Krötzsch [a] Denny Vrandečić [a]
Max Völkel [b] Heiko Haller [b] Rudi Studer [a,b]

[a] *Institut AIFB, Universität Karlsruhe (TH), Germany, {mak|dvr|rst}@aifb.uka.de*
[b] *FZI, Karlsruhe, Germany, {voelkel|haller}@fzi.de*

Abstract

Wikipedia is the world's largest collaboratively edited source of encyclopaedic knowledge. But in spite of its utility, its content is barely machine-interpretable and only weakly structured. With *Semantic MediaWiki* we provide an extension that enables wiki-users to semantically annotate wiki pages, based on which the wiki contents can be browsed, searched, and reused in novel ways. In this paper, we give an extended overview of Semantic MediaWiki and discuss experiences regarding performance and current applications.

Key words: Semantic Web, Wikis, Wikipedia, Collaborative content editing

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. Wikis are usually viewed as tools to manage online content in a quick and easy way, by editing some simple syntax known as wikitext. This is mainly plain text with some occasional markup elements.

Wikipedia is the best known example for a wiki. It aims at creating a multilingual, free encyclopaedia that everyone can edit. The information contained in Wikipedia however is hardly usable by external tools: *using* Wikipedia currently means *reading* articles – there is no way to gather information scattered across multiple articles, like to request a list of all movies from the 1960s with Italian directors. Although the data is quite structured (each movie has its own article, there are links to actors and directors), its meaning is unclear to the computer, because it is not represented in a machine-processable, i. e. formalised way.

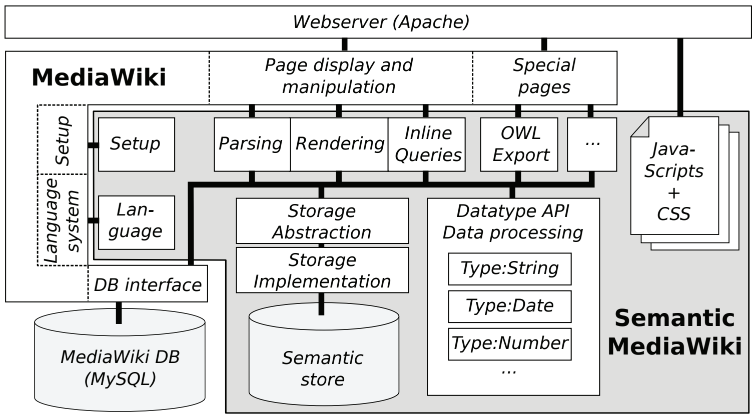


Fig. 1. Architecture of SMW's main components in relation to MediaWiki.

Semantic MediaWiki (SMW) 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 challenging and time-consuming.
- *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.

SMW is free software, available as an extension of the popular wiki engine MediaWiki. Figure 1 provides an overview of SMW's core components which we will discuss in more detail throughout this paper. The integration between MediaWiki and SMW is based on MediaWiki's extension mechanism: SMW registers for certain events or requests, and MediaWiki calls SMW functions when needed. SMW thus does not overwrite any part of MediaWiki, and can be added to existing wikis without much migration cost. Usage information about SMW, installation instructions, and the complete documentation are found at SMW's homepage. [1]

Next, Section 2 explains how structural information is collected in SMW, and how this data relates to the OWL DL ontology language. Section 3 surveys SMW's main features for wiki users: semantic browsing, semantic queries, and data exchange on the Semantic Web. Queries are the most powerful way of retrieving data from SMW, and their syntax and semantics is presented in detail. The practical use of SMW is the topic of Section 4, where we consider existing usage patterns in (non-semantic) Wikipedia, usage statistics from a medium-sized SMW site, and typical current uses of SMW. Section 5 focusses on performance, first by discussing mea-

[1] `http://ontoworld.org/wiki/SMW`

sured processing times on a real SMW installation, and next by comparing SMW's query performance with various RDF stores. We conclude by reviewing some related systems (Section 6), and give a short summary and outlook (Section 7). All descriptions refer to SMW 1.0, the most recent version at the time of this writing.

2 Annotation of Wiki Pages

The main prerequisite of exploiting semantic technologies is the availability of suitably structured ("semantic") data. For this purpose, SMW introduces ways of adding further structure to MediaWiki by means of *annotating* textual content of the wiki. In this section, we recall some of MediaWiki's current means of structuring data, and introduce SMW's annotations with properties. Finally, a formal semantic interpretation of the wiki's structure in terms of OWL DL is presented.

The primary structural mechanism of most wikis is the organisation of content within wiki pages. In MediaWiki, these pages are further classified into *namespaces*, which distinguish different kinds of pages according to their function. Namespaces cannot be defined by wiki users, but are part of the configuration settings of a site. A page's namespace is signified by a specific prefix, such as "User:" for user homepages, "Help:" for documentation pages, or "... talk:" for various kinds of discussion pages. Page titles without a known namespace prefix simply belong to the main namespace. Most pages are subject to the same kind of technical processing for reading and editing, denoted *Page display and manipulation* in Fig. 1. The major exception are so-called *special pages* – built-in query forms without user-edited content – that use "Special:" as a namespace prefix (compare Fig. 1).

Adhering to MediaWiki's basic principles, semantic data in SMW is also structured by pages, such that all semantic content explicitly belongs to a page. Semantically speaking, every page corresponds to an ontological element (including classes and properties) that might be further described by annotations on that very page. This locality is crucial for maintenance: if knowledge is reused in many places, users must still be able to understand where the information originated. Different namespaces are used to distinguish the semantic rôles that wiki pages may play: they can be *individual elements* (the majority of the pages, describing elements of the domain of interest), *categories* (used to classify individual elements, and also to create subcategories), *properties* (relationships between two pages or a page and a data value), and *types* (used to distinguish different kinds of properties, as described in the Section 2.2). Categories have been introduced into MediaWiki in 2002, whereas properties and types were introduced by SMW.

The following section describes existing structuring features in MediaWiki, followed by the newly introduced semantic annotations enabled by SMW. Section 2.3 defines a mapping of the structural features within SMW to the OWL standard used for exporting the knowledge within a wiki.

2.1 Content Structuring in MediaWiki

The primary method for entering information into a wiki currently is *wiki text*, a simplified markup language that is transformed into XHTML pages for reading. Accordingly, wiki text already provides many facilities for describing formatting, and even some for structuring content. For defining the interrelation of pages within a wiki, hyperlinks are arguably the most important feature. They are vital for navigation, and sometimes even used to classify articles informally. In Wikipedia, for example, articles may contain links to pages of the form "as of 2005" to state that the given information might need revalidation or updates after that year.

Many wiki engines generally use links for classifying pages. For instance, searching for all pages with a link to the page "France" is a good way to find information about that country. In MediaWiki, however, this use has been replaced by a more elaborate category system. Every page can be assigned to one or many categories, and each category is associated with a page in the "Category:" namespace. Category pages in turn can be used to browse the classified pages, and also to organise categories hierarchically. Page categories and their hierarchy can be edited by all users via special markup within wiki texts. Overall, the category system probably is the one function of MediaWiki that is closest in spirit to the extensions of SMW.

Another structuring problem of large wikis are synonymous and homonymous titles. In case of synonyms, several different pages for the same subject may emerge in a decentralised editing process. MediaWiki therefore has a *redirect* mechanism by which a page can be caused to forward all requests directly to another page. This is useful to resolve synonyms but also for some other tasks that suggest such forwarding (e.g. the mentioned articles "as of 2005" are redirects to the page about the year 2005). Homonyms in turn occur whenever a page title is ambiguous, and may refer to many different subjects depending on context. This problem is addressed by so-called *disambiguation pages* that briefly list the different possible meanings of a title. Actual pages about a single sense then are augmented with parentheses to distinguish them, e.g. in the case of "1984 (book)".

A final formatting feature of significance to the structure of the wiki is MediaWiki's *template system*. The wiki parser replaces templates with the text given on the template's own page. The template text in turn may be parametrized. This can be used to achieve a higher consistency, since, for example, a table is then defined only once, and so all pages using this table will look similar. The idea of capturing semantic data in templates has been explored inside Wikipedia[2] and in external projects [2].

[2] See, e.g., `http://de.wikipedia.org/wiki/Hilfe:Personendaten`.

In addition to the above, MediaWiki knows many ways of structuring the textual content of pages, e.g. by sections or tables. SMW, however, aims at collecting information about the (abstract) concept represented by a page, not about the associated text. The layout and structure of article texts do hardly carry such data in a machine-accessible way, since they must follow didactic considerations. The possibility of assigning semantic information to single sections also seems less relevant in the context of Wikipedia, the main target application of SMW, since Wikipedia normally contains independent pages for significant subtopics anyway.

2.2 Semantic annotations in SMW

SMW collects semantic data by letting users add annotations to the wiki text of pages via a special markup. The processing of this markup is performed by the components for *Parsing* and *Rendering* in Fig. 1. While annotation syntax is most relevant (and most visible) to wiki editors, it is but a small part of the overall SMW system. The underlying conceptual framework, based on *properties* and *types* is rather more relevant.

Properties in SMW are used to express binary relationships between one semantic entity (as represented by a wiki page) and some other such entity or data value. Each wiki-community is interested in different relationships depending on its topic area, and therefore SMW lets wiki users control the set of available properties. SMW's property mechanism follows standard Semantic Web formalisms where binary properties also are a central expressive mechanism. But unlike RDF-based languages, SMW does not view property statements (subject-predicate-object triples) as primary information units. SMW rather adopts a page-centric perspective where properties are a means of augmenting a page's contents in a structured way.

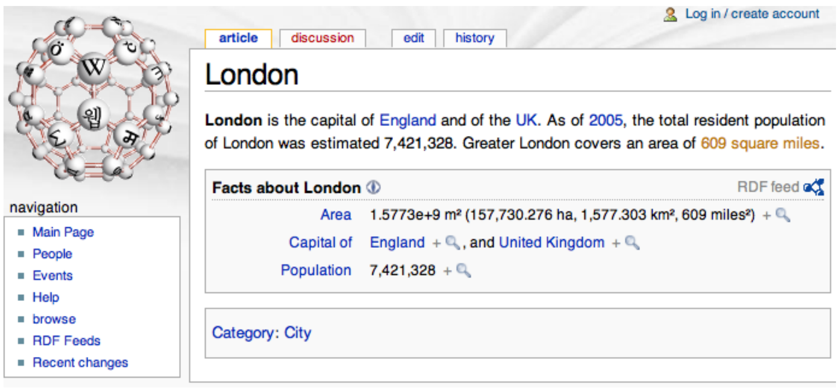
MediaWiki offers no general mechanism for assigning property values to pages, and a surprising amount of additional data becomes available by making binary relationships in existing wikis explicit. The most obvious kind of binary relations in current wikis are hyperlinks. Each link establishes *some* relationship between two pages, without specifying what kind of relationship this is, or whether it is significant for a given purpose. SMW allows links to be characterised by properties, such that the link's target becomes the value of a user-provided property. But not all properties take other wiki pages as values: numeric quantities, calendar dates, or geographic coordinates are examples of other available types of properties.

For a concrete example, consider the article shown in Fig. 2 (top). The markup elements are easy to read: `'''...'''` is used for text that should appear bold-faced, and text within square brackets `[[...]]` is transformed into links to the wiki page of that name. The given links to `England`, `United Kingdom`, and `2005` do not carry any semantics yet. To state that London is the capital of England, one just extends

```
'''London''' is the capital city of [[England]] and of the [[United Kingdom]]. As of
[[2005]], the population of London was estimated 7,421,328. Greater London covers an
area of 609 square miles.                                            [[Category:City]]
```

```
'''London''' is the capital city of [[capital of::England]] and of the [[capital of::United
Kingdom]]. As of [[2005]], the population of London was estimated [[population::7,421,328]].
Greater London covers an area of [[area::609 square miles]].         [[Category:City]]
```

Fig. 2. Source of a page about London in MediaWiki (top) and in SMW (bottom).



London

London is the capital of England and of the UK. As of 2005, the total resident population of London was estimated 7,421,328. Greater London covers an area of 609 square miles.

Facts about London

Area	1.5773e+9 m² (157,730.276 ha, 1,577.303 km², 609 miles²) +
Capital of	England + , and United Kingdom +
Population	7,421,328 +

Category: City

Fig. 3. A semantic view of London.

the link to `[[England]]` by writing `[[capital of::England]]`. This asserts that "London" has a property called "capital of" with the value "England." This is even possible if the property "capital of" has not been introduced to the wiki before.

Figure 2 (top) shows further interesting datavalues that are not corresponding to hyperlinks, e.g. the given population number. A syntax for annotating such values is not as straightforward as for hyperlinks, but we eventually decided for using the same markup in both cases. An annotation for the population number therefore could be added by writing `[[population::7,421,328]]`. In this case, "7,421,328" is not referring to another page and we do not want our statement to be displayed as a hyperlink. To accomplish this, users must first *declare* the property "population" and specify that it is of a numerical type. This mechanism is described below. If a property is not declared yet, then SMW assumes that its values denote wiki pages such that annotations will become hyperlinks. An annotated version of the wikitext for "London" is shown in Fig. 2 (bottom), and the resulting page is displayed in Fig. 3. The box at the bottom of this page sums up the discovered annotations, and possibly provides some related information.

Properties are introduced to the wiki by just using them on some page, but it is often desirable to specify additional information *about* properties. SMW supports this by introducing wiki pages for properties. For example, a wiki might contain a page "Property:Population" where "Property:" is the namespace prefix. A property page can contain a textual description of the property that helps users to employ it consistently throughout the wiki, but it also can specify semantic features of a property. One such feature is the aforementioned *(data)type* of the property. In the case of "Property:Population" one would add the annotation `[[has type::Number]]` to describe that the property expects numerical values. The property "has type" is a built-in property of SMW with the explicated special interpretation. It can also be described on its property page but it cannot be modified or deleted.

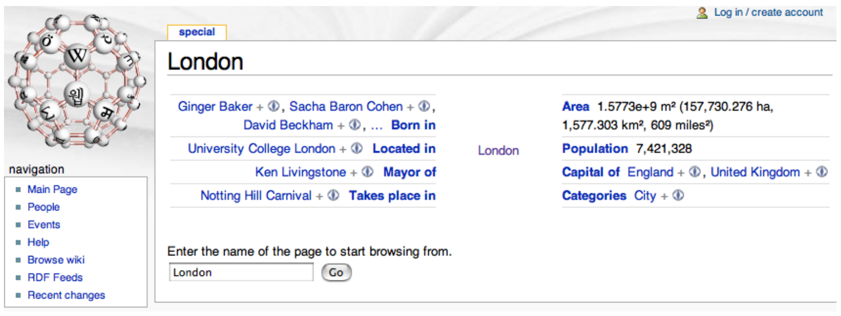
SMW provides a number of datatypes that can be used with properties. Among those are "String" (character sequences), "Date" (calendar dates), "Geographic coordinate" (locations on earth), and the default type "Page" that creates links to other pages. Each type provides own methods to process user input, and to display data values. SMW supplies a modular *Datatype API* as shown in Fig. 1 that can also be extended by application-specific datatypes. Just like properties, types also have dedicated pages within the wiki, and every type declaration creates a link to the according page. To some extent, it is also possible to create new customised datatypes by creating new type pages. These pages of course cannot define the whole computational processing of a data value, but they can create parametrised versions of existing types. The main application of this is to endow numerical types with conversion support for specific units of measurement. For example, the property "Area" in Fig. 2 (bottom) might use a custom type that supports the conversion between km^2 and square miles. Unit conversion is of great value for consolidating annotations that use different units, which can hardly be avoided in a larger wiki.

2.3 Mapping to OWL

The formal semantics of annotations in SMW is given via a mapping to the OWL DL ontology language. Most annotations can easily be exported in terms of OWL DL, using the obvious mapping from wiki pages to OWL entities: normal pages correspond to abstract individuals, properties correspond to OWL properties, categories correspond to OWL classes, and property values can be abstract individuals or typed literals. Most annotations thus are directly mapped to simple OWL statements, similar to RDF triples. It must be noted that this semantic data is not meant to describe the page's HTML-document but rather its (intended) subject.

OWL further distinguishes object properties, datatype properties, and annotation properties, and SMW properties may represent any of those depending on their type. Types themselves do not have OWL semantics, but may decide upon the XML Schema type used for literal values of a datatype property. Finally, containment of pages in MediaWiki's categories is interpreted as class membership in OWL.

SMW offers a number of built-in properties that may also have a special semantic interpretation. The above property "has type," for instance, has no equivalent in OWL and is interpreted as an annotation property. Many properties that provide SMW-specific meta-information (e.g. for unit conversion) are treated similarly.



London

Ginger Baker + , Sacha Baron Cohen + , David Beckham + , ... **Born in**	London	**Area**	1.5773e+9 m² (157,730.276 ha, 1,577.303 km², 609 miles²)
University College London + **Located in**		**Population**	7,421,328
Ken Livingstone + **Mayor of**		**Capital of**	England + , United Kingdom +
Notting Hill Carnival + **Takes place in**		**Categories**	City +

Enter the name of the page to start browsing from.

London

Fig. 4. Browsing through the wiki knowledge base.

MediaWiki supports the hierarchical organisation of categories, and SMW can be configured to interpret this as an OWL class hierarchy (this may not be desirable for all wikis [15]). Moreover, SMW introduces a special property “subproperty of” that can be used for property hierarchies. No type checking is done when declaring subproperties, but subproperty statements between incompatible types are neglected for semantic processing. Overall, the schematic information representable in SMW is intentionally shallow, since the wiki is not intended as a general purpose ontology editor that requires users to have specific knowledge about semantic technologies. Yet SMW has also been used in conjunction with more expressive background ontologies, evaluated by external OWL inference engines [16].

OWL DL constraints subject and objects of statements to individuals, whereas SMW does not have such constraints, thus allowing a kind of meta-modelling. Semantically this kind of meta-modelling is similar to “punning” as introduced by OWL 1.1 [8], which does not make reasoning substantially more complex.

3 Exploiting Semantics

However simple the process of semantic annotation may become, the majority of users will justifiably neglect it as long as it does not bear immediate benefits. In the following, we introduce several features of SMW that show contributors the usefulness of semantic markup.

3.1 Browsing

As shown Fig. 3, the rendered page includes a so called *Factbox* which is placed at the bottom of the page to avoid disturbing normal reading. The Factbox summarises the given annotations, provides feedback on possible errors, e.g. if a given data value does not fit a property’s type, and offers links to related functions.

These links can be used to browse the wiki based on its semantic content. The page

```
QUERY ::= CONJ ('||' CONJ)*          PROP ::= '[[' TITLE '::' VALUE ('||' VALUE)* ']]'
 CONJ ::= ATOM (' ' ATOM)*          VALUE ::= '+' | SUB | (('>'|'<'|'!')? STR )
 ATOM ::= SUB | PROP | CAT | PAGE     CAT ::= '[[Category:' TITLE ('||' TITLE)* ']]'
  SUB ::= '<q>' QUERY '</q>'         PAGE ::= '[[:' FULLTITLE ('||' FULLTITLE)* ']]'
```

```
QUERY ::= '(' CONJ ('⊔' CONJ)* ')'    PROP ::= '∃' TITLE '.(' VALUE ('⊔' VALUE)* ')'
 CONJ ::= '(' ATOM ('⊓' ATOM)* ')'   VALUE ::= '⊤' | SUB | ('ge('|'le('|'ne('|'eq(') STR ')'
 ATOM ::= SUB | PROP | CAT | PAGE     CAT ::= '(' TITLE ('⊔' TITLE)* ')'
  SUB ::= '(' QUERY ')'              PAGE ::= '({' FULLTITLE '}' ('⊔{' FULLTITLE '}')* ')'
```

Fig. 5. Production rules for SMW queries (top) and according DL queries (bottom).

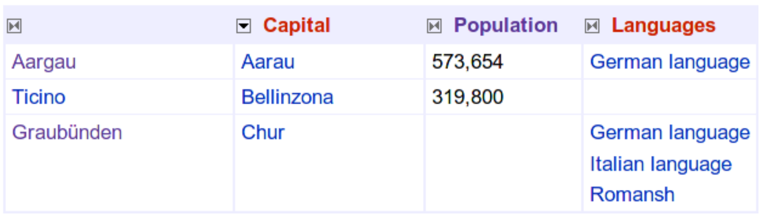
title in the Factbox heading leads to a *semantic browsing* interface that shows not only the annotations within the given page, but also all annotations where the given page is used as a value (Fig. 4). The magnifier icon behind each value leads to an *inverse search* for all pages with similar annotations. Both of those user interfaces are realised as *special pages*, architecturally similar to the special page *OWL Export* in Fig. 1. In addition, the Factbox shows links to property pages, which in turn list all annotations for a given property. All those browsing features are interconnected by appropriate links, so that users can easily navigate within the semantic knowledge base.

3.2 Querying

SMW includes a query language that allows access to the wiki's knowledge. The query language can be used in two ways: either to directly query the wiki, or to add the answer to a page by creating an *inline query* (cf. Fig. 1). The latter enables editors to add dynamically created lists or tables to a page, thus making up-to-date query results available to readers who are not even aware of semantic queries. Figure 6 shows a query result as it might appear within an article about Switzerland. Compared to manually edited listings, inline queries are more accurate, easier to create, and easier to maintain.

The syntax of SMW's query language is closely related to wiki text, whereas its semantics corresponds to certain class expressions in OWL DL. Each query is a disjunction of conjunctions. Fundamental conditions are encoded as *query atoms* whose syntax are similar to that of SMW's annotations. For instance, `[[located in::England]]` is the atomic query for all pages with this annotation. Queries with other types of properties and category memberships are constructed after the same principle. Instead of single fixed values one can also specify ranges of values, and even specify nested query expressions.

A simplified form of SMW's query language is defined in Fig. 5 (top). The main control symbols used to structure queries are: `||` as the disjunction operator, <q> and </q> as (sub)query delimiters, + as empty condition that matches everything,



	Capital	Population	Languages
Aargau	Aarau	573,654	German language
Ticino	Bellinzona	319,800	
Graubünden	Chur		German language Italian language Romansh

Fig. 6. A semantic query for all cantons of Switzerland, together with their capital, population, and languages. The data stems from an automatically annotated version of Wikipedia.

and <, >, ! to express comparison operators ≤, ≥, and ≠. Some nonterminals in Fig. 5 are not defined for reasons of space: `TITLE` is for page titles, `FULLTITLE` is for page titles with namespace prefix, and `STR` is for Unicode strings. In those, we do not permit symbols that could be confused with other parts of the query, e.g. page titles must not start with <. SMW provides means to escape such characters.

The following is an example query for all cities that are located in an EU-country or that have more than 500,000 inhabitants: `[[Category:City]] <q>[[located in::<q>[[Category:Country]] [[member of::EU]]</q>]] || [[population:: >500,000]] </q>`. The formal semantics of such queries is given by a mapping to class expressions in OWL DL, i.e. a query retrieves all inferred members of the according OWL class. It is not hard to see that every SMW query emerges from a unique sequence of production steps, and we can follow the same steps in the grammar in the lower part of Fig. 5. The result is a description logic (DL) concept which could be translated to OWL DL as usual (which space does not permit here). Formally, this also requires to define the underlying DL-language, to introduce punning to allow the same names for DL roles, concepts, and individuals, to give concrete domains for SMW's datatypes, and to fix the interpretation of *ge*, *le*, *ne*, *eq* and ⊤ (when used for datatype properties). We omit these easy but tedious definitions for reasons of space, and only note that our above example corresponds to the following DL concept: $\mathtt{City} \sqcap (\exists\mathtt{located\_in}.(\mathtt{Country} \sqcap \exists\mathtt{member\_of}.\{EU\}) \sqcup \exists\mathtt{population}.ge(500,000))$.

Just like OWL DL, SMW's query language does not support explicit variables, which essentially disallows cross-references between parts of the query. For instance, it is not possible to ask for all people who died in the city they were born in. This restriction makes query answering tractable, which is essential for SMW's usage in large wikis. In contrast, when variables are allowed querying is at least NP-hard, and it becomes harder still even for tractable fragments of OWL 1.1 [10].

SMW-queries as introduced above merely define a result set of pages. In order to retrieve more information about those results, SMW allows so-called *print requests* as parts of queries. For instance, writing `[[has capital::*]]` within a query will cause all values of the property "Has capital" to be displayed for each result. Fig. 6 shows a typical output for a query with multiple print requests. Using further parameters in query invocation, result formatting can be controlled to a large degree.

In addition to tabular output, SMW also supports various types of lists and enumerations, interactive timelines [3] , and many further custom formats.

3.3 Giving Back to the Web

The Semantic Web is all about exchanging and reusing knowledge, facilitated by standard formats that enable the interchange of structural information between producers and consumers. Section 2.3 explained how SMW's content is grounded in OWL DL, and this data can also be retrieved via SMW's web interface for *OWL export*. As shown in Fig. 1, this service is implemented as a special page, which can be queried for information about certain elements. The link "RDF feed" within each Factbox also leads to this service (see Fig. 3).

Exported data is provided in OWL/RDF encoding, using appropriate URIs as identifiers to prevent confusion with URLs of the wiki's HTML documents. The generated OWL/RDF is "browsable" in the sense that URIs can be used to locate further resources: all URIs point to a web service of the wiki that uses content negotiation to redirect callers either to the OWL export service or to the according wiki page. Together with the compatibility to both OWL and RDF this enables a maximal reuse of SMW's data. Even tools like Tabulator [3] that incrementally retrieve RDF resources during browsing can easily retrieve additional semantic data on user request. SMW furthermore provides scripts for generating the complete export of all data within the wiki, which is useful for tools that are not tailored towards online operation such as the faceted browser Longwell. [4] Sample files of such export are found at `http://ontoworld.org/RDF/`.

4 Practical Experiences

In this section we describe the use of SMW in practical applications, and discuss some basic usage data, observed problems and successes. It is clearly important to ask how annotations are typically used within a wiki, and in which way existing usage schemes need to be augmented or modified when introducing semantic features. However, this question is much more complex than it might seem at first, since each wiki has its own usage and editing culture and workflows, which can hardly be generalised to other wikis. Without conclusive studies on the usage of wikis in general, any prediction on the effect of introducing semantics in these environments lacks justification. Moreover, raw data for conducting such studies is generally not available. Of course public wikis provide access to all content and

[3] `http://simile.mit.edu/timeline/`
[4] `http://simile.mit.edu/wiki/Longwell`

editing logs, but actual edits are just a small part of wiki usage that hardly captures the actual editing workflow. The usage of the wiki for reading and searching is even harder to estimate, given that page access counts are not available on larger wikis.

We therefore try to derive usage data from different sources. Firstly, we look at comparable workflows in Wikipedia, as the most well studied and maybe least typical wiki. Secondly, we discuss usage data from *ontoworld.org*, as a much smaller semantic wiki site with the advantage of being completely accessible to us for such studies. Thirdly, we review some existing uses of SMW in other wikis, focussing on general successes and problems that occurred in those settings.

4.1 Structuring of Wikipedia Content

One way of estimating the use of SMW annotations in Wikipedia is to compare it to existing features that are likely to have similarities with respect to editing or usage. For this purpose we focus on Wikipedia's category system. Just like SMW's properties, categories are conceived as an aid for structuring the wiki, and users manage not only page classification but also the available categories in general, including their (hierarchical) organisation. From a user perspective, the purpose of categories is to find information, and the according browsing features are comparable to SMW's functions for properties. Complex querying as in SMW has no counterpart in Wikipedia, and must be excluded from our considerations. Yet we believe that the below observations on categories are good indicators of the presumed use of properties.

Reading access in Wikipedia is monitored only very coarsely, and we are in fact more interested in the process of adding (category or SMW) annotations to the wiki. The overall editing processes in Wikipedia are still not understood properly. A first analysis was conducted in [13], where it was found that wiki editors can roughly be classified into two groups: most edits are performed by a small group of core contributors, whereas most content is contributed by a much larger group of content authors. The former group has more expertise and interest in the wiki as a whole, whereas the latter group takes interest in a small set of articles, usually of their specific expertise. As [13] puts it: "an outsider makes one edit to add a chunk of information, then insiders make several edits tweaking and reformatting it. In addition, insiders rack up thousands of edits doing things like changing the name of a category across the entire site – the kind of thing only insiders deeply care about." More extensive studies are required to refine our picture of the editing in Wikipedia, but these first observations indicate that structural enhancements like categorisation are mostly handled by just a small portion of contributors.

Another important aspect of categorisation in Wikipedia are the associated semi-official workflows, guidelines, and rules. Such policies are an important part of structuring Wikipedia in general [14], and for categories we find detailed descriptions on how to decide whether a category is appropriate, and if it should be used in a given article. [5] Categorisation policies and their application are subject of ongoing discussions of interested community members – the main decision process in Wikipedia. We also note that, according to the guidelines, categories are conceived as annotations in the sense that the reason of their use must be evident from the text of the article. Categories should not add completely new information, but make some aspect of a page's content explicit. This use of categories is not inherent to the technical category system, but reflects the usage in (English) Wikipedia.

Finally, we note that the use of categories in Wikpedia is governed mostly by practical considerations. The purpose of categorisation is not to build a taxonomy of the world's knowledge, but to create helpful structures for readers to discover information. Likewise, SMW's properties are not an attempt at the utopian task of building a "world ontology" but a practical tool to aid the everyday usage of Wikipedia.

4.2 Usage on ontoworld.org

Large wikis like Wikipedia cannot afford the performance overhead of detailed usage logging, and creating such logs requires server-side access which is not available in general. Luckily, the Semantic Web community wiki *ontoworld.org*, maintained by the authors, is available for more extensive experiments. Its main use is to collect information about Semantic Web events, researchers and projects, and to serve as a conference wiki for the international Semanitc Web conferences. However, ontoworld.org also is still the home of SMW and is used by many as a testbed for semantic features. The advantage is the existence of a real non-centralised community and a rather heterogeneous page structure, but of course ontoworld.org is also more biased towards the use of SMW than other wikis might be.

As of today, ontoworld.org has 6,319 content pages (including short pages a.k.a. "stubs"). The site has 930 registered users, most of which have contributed little to the total of recorded 37,880 edits. The semantic knowledge base of ontoworld.org counts 17,562 property annotations for 808 properties. Most of those have a page in the wiki, and about 50% are of type *Page* (i.e. used to annotate hyperlinks).

Further considering the usage of properties, it is characteristic that 5% of the properties account for over 74% of the annotations, whereas the least used 80% of the properties make up less than 9% of all semantic statements. Very similar power-law distributions occur for Wikipedia's categories [15] and indeed in many data collections that are created in a distributed fashion by many users. This is an important characteristic of community-driven sites, whereas small wikis often have a more contrived structure.

[5] `http://en.wikipedia.org/wiki/Wikipedia:Categorization`

Views	u#	r#	p#	u#/p#	r#/p#
All	16,525	52,192	6,319	2.62	8.26
Category:	598	4,166	484	1.24	8.61
Property:	781	3,150	756	1.03	4.17
Type:	263	336	43	6.12	7.81
Help:	1,766	773	41	43.07	18.85

Special	u#
Browse	637
PropSearch	1,326
Querying	222
Properties	83
Types	59

Special	u#
Changes	479
Backlinks	284
Search	419
Categories	65
OWL export	575

Fig. 7. Access statistics for ontoworld.org for a period of 7 days. The columns u#, r#, and p# denote numbers of user requests, robot requests, and existing pages, respectively.

We have monitored the use of ontoworld.org over a period of seven days, during which a total of 68,000 normal page views (excluding edits and special pages) occurred. As expected, the majority of requests is caused by identified web crawlers (including 75% of all page views) but also automated spammers (causing about 16,000 blocked edit attempts). The latter cannot be fully separated from real users, but functions for searching and browsing are normally not triggered by spammers, and thus give good indications of actual usage.

Access statistics for the observed timespan are given in Fig. 7. The leftmost table shows page views by namespaces, where we use the ratio between requests and existing pages as a slightly more normalised measure for comparing results. Properties appear to be of slightly less interest, especially to robots: this is due to a very recent relabelling of this namespace which caused all URLs to change. Otherwise, we note that categories and properties are of comparable popularity among users. The Help- and Type-namespaces mostly contain SMW's user documentation, which explains the increased interest in those pages.

The middle and right tables of Fig. 7 show usage counts for special functions: "Browse" and "PropSearch" are SMW's special browsing functions (Section 3.1); "Querying" is a dedicated interface for using SMW-queries (Section 3.2); "Properties", "Types", and "Categories" display all elements of the respective kinds; "Changes" lists recent edits; "Backlinks" lists all pages with links to a given page; "Search" is MediaWiki's text search; "OWL export" is SMW's OWL export. We can observe that SMW's browsing features (property pages, Browse, PropSearch) are frequently used, also when compared to MediaWiki's browsing interfaces (category pages and Backlinks). SMW's stand-alone query interface is also employed by many users even though requiring much more interaction. This is evident when mentioning that robots issued about 20,000 browsing requests, but called the query interface only 28 times. Also, SMW's maintenance pages for properties and types are of similar popularity as MediaWiki's category list.

In conclusion, we find that SMW's features are already used at least as much as comparable MediaWiki functions. We also noted that query and browsing requests typically related to the wikis main topics, and were used for navigating to further content pages. We conjecture that those requests reflect real usage, as opposed to mere testing of SMW functions. Yet, of course, access statistics do not allow us to conclude whether or not the requested functions were actually considered useful for a given purpose, and further research – starting with the evaluation of MediaWiki and Wikipedia – is needed to obtain definite results.

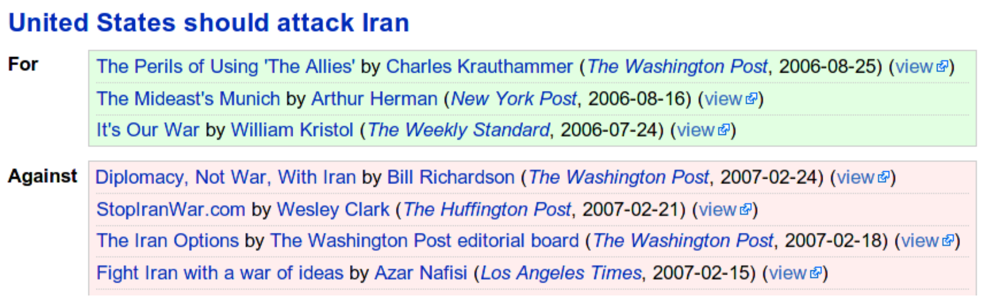
Finally, the usage of SMW's OWL export is also remarkable. A surprisingly high number of 575 users accessed export data, but 190 of those requests were caused by user agents automatically following links in the head of SMW's HTML pages. Few of the remaining 385 requests are due to deliberate user requests from the wiki, while others refer from search engines, or were caused by unidentified user agents (possibly involving RDF-browsers). The vast majority of export requests, however, is caused by robots. While classical search engines are less active on those pages, an additional amount of Semantic Web applications such as Swoogle (`http://swoogle.umbc.edu`) show exclusive interest in the OWL data. This usage of SMW as a source for reusable semantic data is in the true spirit of the Semantic Web, and has potential to attract additional users to a wiki.

4.3 SMW in other Semantic Wikis

Finally, we have a brief look at practical uses of SMW in other wikis. It is hardly possible to track down novel installations of SMW on the web, but we are aware of numerous sites that currently employ the extension (`http://ontoworld.org/wiki/Sites_using_Semantic_MediaWiki`). Many of those early adopters are very specific wikis with a small editing community, and usage patterns might be different from large wikis. Yet we can observe some successful uses and problems.

Protégé Wiki (`http://protegewiki.stanford.edu`) is a good example of a carefully structured wiki site. Being recently started, it counts about 160 pages, 16 registered users, and 350 annotations. Naturally, we do not encounter the large amounts of hardly used properties and categories either. Many of today's wikis (semantic or not) are built similarly by a very small number of people. It is important to understand that the usage patterns in such a small wiki will be very different from one like ontoworld.org. Protégé Wiki still is a good example of using SMW: it contains a good balance of textual and semantic content, makes suitable use of inline queries, and provides editing help that is tailored to its own workflows.

SMW is conceived as an extension for text-based wikis, but it also can be used in cases where almost all content is strongly structured. SMW thus, somewhat unintentionally, takes the place of a database application with a web frontend for collaborative editing. A good example for this type of semantic wiki is *DiscourseDB* (`http://discoursedb.org`), which collects information about ongoing political debates. Its actual content pages are short, strongly structured articles about single publications, but it offers many automatically generated query pages that provide novel views on this content as shown in Fig. 8.


United States should attack Iran

For	The Perils of Using 'The Allies' by Charles Krauthammer (*The Washington Post*, 2006-08-25) (view)
	The Mideast's Munich by Arthur Herman (*New York Post*, 2006-08-16) (view)
	It's Our War by William Kristol (*The Weekly Standard*, 2006-07-24) (view)
Against	Diplomacy, Not War, With Iran by Bill Richardson (*The Washington Post*, 2007-02-24) (view)
	StopIranWar.com by Wesley Clark (*The Huffington Post*, 2007-02-21) (view)
	The Iran Options by The Washington Post editorial board (*The Washington Post*, 2007-02-18) (view)
	Fight Iran with a war of ideas by Azar Nafisi (*Los Angeles Times*, 2007-02-15) (view)

Fig. 8. DiscourseDB uses SMW to generate overviews of public political debate. The above inline queries display positions on the Iran nuclear crisis.

The above examples are typical for a number of SMW installations, but many other uses of SMW do not fit into a given scheme. We have rarely encountered cases where users found it hard to come up with initial ideas on how to add semantic structure, but some topic areas, such as theology (see e.g. BibleWiki `http://bible.tmtm.com`), are not easily cast into OWL. Other exceptions are cases where most of the content is not plain text but some other form of multimedia. This challenges MediaWiki as well as SMW, but first interesting prototypes for managing e.g. video in SMW have been presented recently [7].

5 Performance Evaluation

Performance is a key factor in web applications, since today's high numbers of requests can produce significant (and expensive) server loads. This is especially true for Wikipedia, which often receives over 30,000 requests per second. Hence, while collecting the usage data discussed in Section 4.2, we also measured processing times of many central operations of the software. Raw results of ongoing measurements can be accessed at `http://ontoworld.org/profileinfo.php`.

Our main goal was to find out how much additional processing time SMW requires in a typical medium-sized wiki. Results are influenced by many complex factors (e.g. database caching effects), and are useful for estimating overall performance impacts rather than exact runtimes of particular software modules. During the experiment, MediaWiki was called 214,356 times, with a total runtime of about $4.54 \times 10^5 s$. The aggregated total runtime of all SMW functions amounted to only 8.6% of this time. The major components of this runtime were the evaluation of semantic queries (40.5%), various browsing functions (22.5%), OWL export (22.2%), and parsing of annotations (12.3%). These functions depend on the amount of semantic content of the wiki, i.e. performance impair relates to actual usage. Loading SMW takes only 2.3% of SMW's time (less than 0.2% of total time).

As we saw in Section 4.2 ontoworld.org clearly uses semantic features rather frequently, and the small overall performance impact is encouraging. Yet, especially queries can have very high costs. We recorded 51,569 query evaluations overall (including many inline queries), but the 0.4% of requests to "Querying" in Fig. 7 account for 11% of processing time. Various features of SMW help to manage query load. First, all inline query results are cached with their pages: improved caching mechanisms as on Wikipedia also reduce query load. Second, query complexity and processing can be restricted in SMW by many parameters. Individual reasoning features can be disabled, and overall size of query constructs can be limited up to allowing only most trivial queries.

Query computation in principle is a polynomial time problem [10], but the cost of processing still depends heavily on the actual implementation. In SMW, the *Storage Implementation* shown in Fig. 1 processes all queries. This component is wrapped by a *Storage Abstraction* that hides all implementation details. The current default store used by SMW is based on the wiki's existing relational database (MySQL), but this component could be replaced by faster stores if available. Hence we analysed SMW's query performance in isolation by directly measuring computation times for the database queries issued by SMW. The resulting times are not influenced by network delays, preprocessing times, or caching effects, and thus can be compared to other semantic storage implementations.

To obtain a large yet realistic dataset, we uploaded an article dump from English Wikipedia to a local installation of SMW. To obtain initial annotations, we used a dataset from *dbpedia.org*, created by interpreting template parameters as RDF triples [2]. Since this data relies on Wikipedia's internal labels for template parameters, not all statements are correct or even meaningful. But the data is based on manually created content, and shows a realistic structure that is more important for a performance test than factual accuracy. Finally, we exported all semantic data as OWL/RDF, now including exactly the statements considered by SMW for querying. The dataset now contained 13,802,002 triples, modelling 3,524,650 elements.

We considered six different test queries with different result sets. For reference, we label them as follows (number of results in parentheses): q_1 (1), q_2 (8677), q_3 (56), q_4 (13), q_5 (3612), q_6 (4). Queries q_1 to q_3 consist of nested query conditions and thus describe chains of properties. Query q_4 contains no subqueries but many conjunctive conditions, q_5 emphasises range restrictions on data values, and q_6 combines conditions of all those kinds. Equivalent versions of each query were formulated in SMW and in SPARQL, and SMW queries then were translated to SQL queries by SMW's storage implementation. The queries and the complete dataset are available at `http://ontoworld.org/RDF/`.

We compared the runtime of the SQL queries on SMW's MySQL database with the time needed to answer the corresponding SPARQL queries on four RDF storage implementations: Jena in-memory, Jena on an embedded Apache Derby DB [6], Sesame2 in-memory, and Sesame2 native b-tree [5]. For this experiment, we concentrated on runtime, and disregarded memory usage and loading times although both are strictly restricted in practical cases. Sesame2 in-memory clearly turned out to be the fastest store, usually 10 to 50 times faster than MySQL. The only exception is q_3, where Sesame2 in-memory is slower than MySQL by a factor of 5. Surprisingly, Sesame2 native is faster than its in-memory sibling on q_3 but still 4 times slower than MySQL. With the exception of q_3 and q_1, MySQL tends to be up to two times slower than Sesame2 native, but its much simpler implementation is still competitive. All tests on Sesame2 native used all available indexing modes, as the defaults were up to 40 times slower for q_1, q_2, and q_4. Jena, finally, could hardly compete with the other stores. Jena in-memory was hardly able to load the dataset within 6 GB memory, and crashed when trying to answer queries. Jena Derby was slightly faster than MySQL on q_6 but up to 40 times slower on all other queries.

Summing up, SMW's MySQL store compares surprisingly well to other storage solutions, and shows the most continuous performance overall. Sesame2 is generally faster, but collapses unexpectedly for specific queries. As expected, running stores in-memory is usually much faster, but may not be feasible for very large datasets.

6 Related Systems

Besides SMW a number of other semantic wiki implementations have been created. The most notable (and stable) systems currently are *IkeWiki* [12] and *MaknaWiki* [11]. Both of these are similar to SMW with respect to the supported kinds of easy-to-use inline wiki annotations, and various search and export functions. In contrast to SMW, both systems introduce the concept of *ontologies* and (to some extent) *URIs* into the wiki, which emphasises use-cases of collaborative ontology editing that are not the main focus of SMW.

MaknaWiki is tailored for closed-domain settings where a suitable schema for annotation can be anticipated. It supports the usage of expressive ontologies and integrates according inferencing features, but allows only administrators to change the ontological schema. IkeWiki in turn uses URIs to identify concepts, but provides interfaces for simplifying annotation, e.g. by suggesting properties. In contrast to MaknaWiki and SMW, IkeWiki is not based on any existing wiki implementation, which has advantages and disadvantages.

Besides text-centered semantic wikis, various collaborative database systems have appeared recently. Examples of such systems include *OntoWiki* [1], *OpenRecord* (`http://www.openrecord.org`), *Metaweb* (`http://www.metaweb.com/`), and *OmegaWiki* (`http://www.omegawiki.org`). Such systems typically use form-based editing, and are used to maintain data records, whereas SMW concentrates on text, the main type of content in wikis. OntoWiki draws from concepts of semantic technologies and provides a built-in faceted (RDF) browser. The other systems have their background in relational databases.

7 Outlook

Semantic MediaWiki is under continuous development, and numerous additions are planned for the near future. Upcoming features involve improved annotation support (including suggestions and auto-completion), additional functions for maintaining the emerging vocabulary, and storing (additional) semantic data not given within article texts, which is useful for database-like applications.

In parallel, Wikipedia-related showcases are planned to further advertise the utility and feasibility of the system in this particular use-case. Initially, this may be based on our current test data [2], but more reliable data from elaborate off-line extraction methods might be preferred [9,4]. Extracted information like this can be used to bootstrap a Semantic Wikipedia, allowing to augment and improve the learned results within what would be the hitherto largest collaboratively edited semantic knowledge repository.

As of today, SMW is one of the most popular semantically enhanced collaborative knowledge management systems, mostly because it strives to make semantic technologies accessible to non-expert users. Though SMW has found many unexpected uses in a wide range of applications, the original goal of enabling a Semantic Wikipedia is still the main driving use case that we are aiming at, and is becoming more concrete as the system matures.

Acknowledgements. The presented work was supported by the European Union under the projects SEKT, NEON and Nepomuk, and by Vulcan Inc. under the Halo project. We thank Holger Lewen, Uta Hellinger and the anonymous reviewers for helpful comments, and all contributors of Semantic MediaWiki for their work.

References

[1] Sören Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki – A tool for social, semantic collaboration. In Yolanda Gil, Enrico Motta, Richard V. Benjamins, and Mark Musen, editors, *Proc. 5th Int. Semantic Web Conference (ISWC'05)*, number 4273 in LNCS, pages 736–749. Springer, 2006.

[2] Sören Auer and Jens Lehmann. What have Innsbruck and Leipzig in common? Extracting semantics from wiki content. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *Proc. 4th European Semantic Web Conference (ESWC)*, 2007.

[3] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the Semantic Web. In Lloyd Rutledge, m.c. schraefel, Abraham Bernstein, and Duane Degler, editors, *Proc. 3rd Int. Semantic Web User Interaction Workshop*, 2006.

[4] Stephan Bloehdorn and Sebastian Blohm. A self organizing map for relation extraction from wikipedia using structured data representations. In Wray Buntine and Henry Tirri, editors, *Proc. Int. Workshop on Intelligent Information Access (IIIA-2006)*, 2006.

[5] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF Schema. In Ian Horrocks and James Hendler, editors, *Proc. 1st Int. Semantic Web Conference (ISWC'02)*, number 2342 in LNCS, pages 54–68, 2002.

[6] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the Semantic Web recommendations. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proc. 13th Int. World Wide Web Conference (WWW'04 alternate track papers & posters)*, pages 74–83, 2004.

[7] Michael Dale. Metavid wiki - real time collaborative semantic audio/video metadata in mediawiki, 2007. Talk presented at WikiMania2007.

[8] Bernardo Cuenca Grau et al. OWL 1.1 Web Ontology Language Guide. Available at `http://owl1_1.cs.manchester.ac.uk/`, March 2007.

[9] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Manuela Veloso, editor, *Proc. 20th Int. Joint Conference on Artificial Intelligence (IJCAI'07)*, 2007.

[10] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In Karl Aberer, Key-Sun Choi, and Natasha Noy, editors, *Proc. 6th Int. Semantic Web Conf. (ISWC'07)*. Springer, 2007.

[11] Lyndon J. B. Nixon and Elena Paslaru Bontas Simperl. Makna and MultiMakna: towards semantic and multimedia capability in wikis for the emerging web. In Sebastian Schaffert and York Sure, editors, *Proc. Semantics 2006*. Österreichische Computer Gesellschaft, 2006.

[12] Sebastian Schaffert. IkeWiki: A semantic wiki for collaborative knowledge management. In Robert Tolksdorf, Elena Paslaru Bontas, and Klaus Schild, editors, *1st Int. Workshop on Semantic Technologies in Collaborative Applications (STICA'06)*, Manchester, UK, June 2006.

[13] Aaron Swartz. Who writes Wikipedia?, 2006. Available at `http://www.aaronsw.com/weblog/whowriteswikipedia`.

[14] Fernanda Viégas, Martin Wattenberg, Jesse Kriss, and Frank van Ham. Talk before you type: Coordination in Wikipedia. In *Proc. 40th Hawaii International Conference of System Sciences*. Computer Society Press, 2007.

[15] Jakob Voss. Collaborative thesaurus tagging the Wikipedia way, 2006. Available at `http://arxiv.org/abs/cs/0604036v2`.

[16] Denny Vrandečić and Markus Krötzsch. Reusing ontological background knowledge in semantic wikis. In Max Völkel, Sebastian Schaffert, and Stefan Decker, editors, *Proc. of the 1st Workshop on Semantic Wikis – From Wikis to Semantics*, 2006.