Semantic Wikipedia

Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller
Institute AIFB
University of Karlsruhe (TH)
D-76128 Karlsruhe, Germany
{voelkel,kroetzsch,vrandecic,haller}@aifb.uni-karlsruhe.de

ABSTRACT

Wikipedia is the world's largest collaboratively edited source of encyclopaedic knowledge. But in spite of its utility, its contents are barely machine-interpretable. Structural knowledge, e. g. about how concepts are interrelated, can neither be formally stated nor automatically processed. Also the wealth of numerical data is only available as plain text and thus can not be processed by its actual meaning.

We provide an extension to be integrated in Wikipedia, that allows to type links between articles and to specify typed data inside the articles in an easy-to-use manner.

Enabling even casual users to participate in the creation of an open semantic knowledge base, Wikipedia has the chance to become a hitherto unknown resource of semantic statements regarding size, scope, openness, and internationalisation. These semantic enhancements bring to Wikipedia the benefits of today's semantic technologies like more complex or more specific ways of searching and browsing. Also, the RDF export, that gives direct access to the formalised knowledge, opens Wikipedia up to a wide range of external applications, that will be able to use it as a background knowledge base.

In this paper, we present the design, implementation, and possible uses of this extension.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Systems; H.5.3 [**Information Interfaces**]: Group and Organization Interfaces—*Web-based interactions*; I.2.4 [**Artifical Intelligence**]: Knowledge Representation; K.4.3 [**Computers and Society**]: Organizational Impacts—*Computer-supported collaborative work*

General Terms

Human Factors, Documentation, Languages

Keywords

Semantic Web, Wikipedia, RDF, Wiki

1. INTRODUCTION

This paper describes an extension to be integrated in Wikipedia, that enhances it with Semantic Web [6] technologies. Wikipedia, the free encyclopaedia, is well-established as the world's largest online collection of encyclopaedic knowledge, and it is also an example of global collaboration within an open community of volunteers.

The information contained in Wikipedia is still unusable for many fields of application, because *using* Wikipedia currently means *reading* it: Although the data is highly structured, its meaning is unclear to the computer, because it is not stored in a formalised way.

The extension described in this article enables the huge and highly motivated community of Wikipedians to render the shared factual knowledge of Wikipedia's machine-processable. In addition to technical aspects of this endeavour, the main challenge is to introduce semantic technologies into the established usage patterns of Wikipedia. We propose small extensions to the wiki syntax and an enhanced article view to show the interpreted semantic data to the user.

We expose Wikipedia's fine-grained human edited information in a standardised and machine-readable way by using the W3C standards on RDF [14], XSD [10], RDFS [7], and OWL [20]. This opens new ways for improving Wikipedia's capabilities for querying, aggregating, or exporting knowledge, based on well-established Semantic Web technologies. We hope that Semantic Wikipedia can help to demonstrate the promised value of semantic technologies to the general public, e. g. serving as a base for powerful question answering interfaces.

The primary goal of this project is to supply an implemented extension to be actually introduced into Wikipedia in the near future. The implementation is rapidly developing, and the software can be tested online at `http://wiki.ontoworld.org`.

In this article, we review major achievements and shortcomings of today's Wikipedia (Section 2), and discuss our basic ideas and their effect on practical usage (Section 3). We describe the underlying architecture of our system (Section 4) and give an overview of the concrete implementation (Section 5). In Section 6, we point out various potential knowledge-based applications (both local and web-based), that could be realised based on our semantic extension of Wikipedia. After a brief review of related approaches to semantic wikis (Section 7), we conclude with a summary and point to open research issues in Section 8.

2. TODAY'S WIKIPEDIA

Wikipedia is a collaboratively edited encyclopaedia, available under a free licence on the web.[1] It was created by

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

[1] `http://www.wikipedia.org`

Jimbo Wales and Larry Sanger in January 2001, and has attracted some ten thousand editors from all over the world. As of 2005, Wikipedia consists of more than 2,5 million articles in over two hundred languages, with the English, German, French and Japanese editions being the biggest ones [24].

It is based on a wiki software. The idea of wikis was first introduced by Ward Cunningham [9] within the programming language patterns group. A wiki is a simple content management system, that is especially geared towards enabling the reader to change and enhance the content of the website easily. Wikipedia is based on the MediaWiki[2] software, which was developed by the Wikipedia community especially for the Wikipedia, but is now used in several other websites as well. The idea of Wikipedia is to allow everyone to edit and extend the encyclopaedic content (or simply correct typos).

Besides the encyclopaedic articles on many subjects, Wikipedia also holds numerous articles that are meant to enhance the browsing of Wikipedia: rock'n'roll albums in the sixties,lists of the countries of the world, sorted by area, population, or the index of free speech, the list of popes sorted by length of papacy, their name or the year of inauguration[3]. As it is now, all these lists are created manually, introducing several sources of inconsistency, only maintainable through the sheer size of the community. Smaller Wikipedia communities, like the Latin Wikipedia or the Asturian Wikipedia will hardly be able to afford the luxury of maintaining several redundant lists.

These lists may be regarded as queries, the answers of which are created manually. Whereas queries about the biggest countries may be anticipated, rather seldom asked (but still highly relevant) queries like the search for "all the movies from the 1960s with Italian directors" will hardly be created, or else badly maintained, often being dependant on a single editor. Changes in the articles do not reflect in all the appropriate lists, but have to be updated manually.

Besides those hand-crafted lists, Wikipedia offers a full-text search of its content and a categorisation of articles (where categories can be organised hierarchically). There is no other way to access the huge data included in Wikipedia right now. In particular, Wikipedia's content is only accessible for human reading. The automatic gathering of information for agents and other programs is hardly possible right now: only complete articles may be read as blobs of text, which is hard to process, understand and put to further usage by computers.

3. GENERAL IDEA

Our primary goal is to provide an extension to MediaWiki which allows to render important parts of Wikipedia's knowledge machine-processable with as little effort as possible. The prospect of making the world largest collaboratively edited source of factual knowledge accessible in a fully automatic fashion is certainly appealing, but the specific setting also creates a number of challenges that one has to be aware of.

First and foremost, any extension of Wikipedia must satisfy highest requirements on usability, since the large community of volunteers is the primary strength of any wiki. Users must be able to use the extended features without any technical background knowledge or prior training. Furthermore, it should be possible to simply ignore the additional possibilities without major restrictions on the normal usage and editing of Wikipedia. Another important aspect of using a wiki is that the community must always have full control, and that users can freely modify and extend the content.

A second important challenge in the context of Wikipedia is the sheer size of the system, and the fact that the knowledge base is growing continuously. Performance and scalability are thus highly relevant. Other technical challenges concern the interfaces towards applications: making Wikipedia accessible to machines also involves concrete interfaces and export functions.

The latter point also involves the task of finding appropriate semantic description languages for exchanging information. A range of formalisms has been proposed for this task, but for usage in Wikipedia, one must be careful to stay open to future developments as well. Furthermore, various semantic difficulties, like the local creation of global inconsistencies, must be avoided at all cost.

In the rest of this section, we focus on usability and review the main features of the system from a Wikipedia editor's viewpoint. We start with an overview of the kind of semantic information that is supported, and proceed by discussing *typed links* and *attributes* individually. Technical aspects considering system architecture, scalability, and export format are detailed later on in Section 4.

3.1 The Big Picture

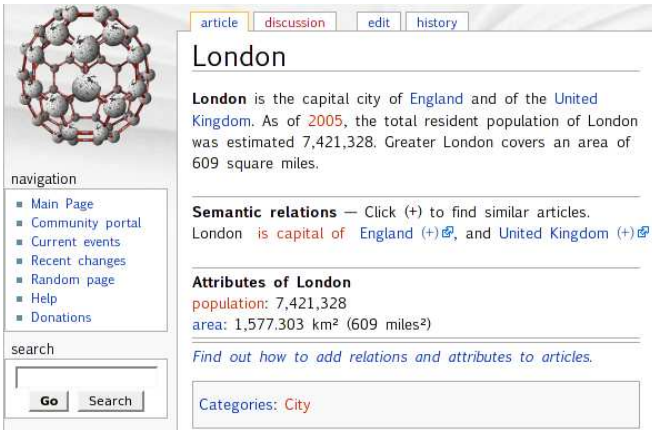
To ensure seamless integration with the common usage experience in a wiki, we approach our goal by slightly extending the way of creating a hyperlink between articles. As for the Web in general, links are arguably the most basic and also most relevant markup within a wiki, and their syntactical representation is ubiquitous in the source of any Wikipedia article. Through a minor, optional syntax extension, we allow wiki users to create *typed links*, which express a *relation* between two pages (or rather between their respective subjects). Below, we describe how this additional information can be added in an unobtrusive and user-friendly way.

Just like a normal link, a typed link is a link from one wiki page to another wiki page. A typed link simply additionally has a *type* which states the kind of relation between concepts. As an example, one could type a link from the article "London" to "England" with the relation "capital of". Even very simple search algorithms would then suffice to provide a precise answer to the question "What is the *capital of England*?" In contrast, the current text-driven search returns only a list of articles for the user to read through.

In order for such extensions to be used by editors, there must be new features that provide some form of *instant gratification*. Semantically enhanced search functions improve the possibilities of finding information within Wikipedia and would be of high utility to the community. But one also has to assure that changes made by editors are immediately reflected when conducting such searches. Additionally, Wikipedia's machine-readable knowledge is made available for external use by providing RDF dumps. This allows for the creation of additional tools to leverage Wikipedia contents and re-use it in other contexts. Thus, in addition to the traditional usage of Wikipedia, a new range of services is

[2] `http://www.mediawiki.org`

[3] There is even a list of persons with asteroids named after them



London

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

Semantic relations — Click (+) to find similar articles.
London is capital of England (+), and United Kingdom (+)

Attributes of London
population: 7,421,328
area: 1,577.303 km² (609 miles²)

Find out how to add relations and attributes to articles.

Categories: City

Figure 1: A semantic view of London.

enabled inside and outside the encyclopaedia. Experience with earlier extensions, such as Wikipedia's category system, assures us that the benefits of said services will lead to a rapid introduction of typed links into Wikipedia.

To further improve the utility of our approach, we provide means to make other information explicit as well. First of all, Wikipedia's category system suggests itself for machine processing as well. At the moment, categories are used solely for browsing. Advanced searching, possibly incorporating the given hierarchy, or even simple Boolean operations are not supported for categories.

For another interesting source of machine readable data, we also incorporate the great number of data values in the encyclopedia. Typically, such values are provided in the form of numbers, dates, coordinates, and the like. For example, one would like to obtain access to the population number of London. It should be clear that it is not desirable to solve this problem by creating a semantic link to an article entitled "7421328" because this would create a unbearable amount of mostly useless number-pages whereas the textual title does not even capture the intended numeric meaning faithfully (e.g. the natural lexicographic order of titles does not agree with the natural order of numbers). Therefore, we introduce an alternative markup for describing attribute values in various datatypes. The Wikipedia project "WikiData" [4] needed pursues a similar objective, but is targeted at different use-cases where fixed forms can be used to input data. Moreover, we address the problem of handling units of measurement that are often given for data values.

To summarise these features, consider the example depicted in Figure 1. This screenshot shows a short article on London. The markup contained in the article source allows to extract key facts from this text, which are displayed below the article. In the following sections, we explain in detail how this is achieved by an editor.

3.2 Usage of Typed Links

A regular hyperlink already states that there is some relation between two pages—an information that search engines like Google successfully use to rank pages in a keyword search scenario. Typed links can even go beyond that, since they are interpreted as *semantic relations* between two concepts described within articles. Here, instead of discussing the "URI crisis" again, we can assume that a wiki page URI unambiguously represents the concept discussed on that page and not the page itself. We remark that not every link should be turned into a typed link: many links do not have an explicit meaning and serve merely navigational purposes. The utility of typed links stems from their ability to distinguish links that are "meaningful" in a given context.

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

Figure 2: Source of an article on London using Wikipedia's current markup.

The suggested typing mechanism can be understood as a kind of categorisation of links. Like in the case of categories and articles, wiki authors are free to employ arbitrary descriptive labels for typing links. To do so, one has to extend the syntactical description of a hyperlink. Without semantic extensions, the source of the article in Figure 1 looks as in Figure 2. In Wikipedia, links to other articles are created by simply enclosing the article names in double brackets. Classification in categories is achieved in a similar fashion by providing a link to the category. However, category links are always displayed in a special format at the article bottom, without regard to the place of the link inside the text.

Now in order to explicitly state that London is the capital of England, one just extends the link to `[[England]]` by writing `[[capital of::England]]`. This states that a relation "capital of" holds between "London" and "England." Typed links therefore stay true to the wiki-nature of Wikipedia: every user can add a type to a link or change it. It should be clear that the textual labels of a link type can be chosen arbitrarily by the user. Of course, in order to make improved searching and similar features most efficient, the community will settle down to re-use existing link types. As in the case of categories, we allow for the creation of descriptive articles on link types to aid this process—for details see Section 4.

In the rare cases where links should have multiple types, users can write [[type$_1$::type$_2$::...::type$_n$::target article]]. Sometimes it is desirable that the displayed text of a hyperlink is different from the title of the article it links to. In Wikipedia, this can by achieved by writing `[[target article|link text in article]]`, and this option is not affected by any of the syntactical extensions we allow for specifying the target article.

Note how typed links integrate seamlessly into current wiki usage. In contrast to all other semantic wikis we are aware of (see Section 7), Semantic MediaWiki places semantic markup directly within the text and thus ensures that machine-readable data agrees with the human-readable data within the article. The order of writing relation and linked article in our notation makes the extended syntax largely self-explicatory (provided that labels are chosen carefully).

Typed links already enable some practical applications. Besides the immediate question for the capital of England, the above information can also be used in more advanced ways. If we know that "capital of" is a special case of being

[4] http://meta.wikimedia.org/wiki/Wikidata

"located in" (ways for stating this are discussed in Section 4), we can *infer* that London is located in England—a fact that has never been entered into Wikipedia. We can also allow for *aggregated queries* that combine several search criteria to return a list of articles. For example, using our knowledge that England is indeed located in Europe and the membership in the category "City," we are able to find "London" when searching for all European cities. In contrast to simple keyword queries, we will not find rivers, routes, or mountains in that list. These examples illustrate various levels of reasoning with semantic information, and it is clear that there is a trade-off between added strength and required effort for computation and implementation.

3.3 Attributes and Types

Data values play a crucial role within an encyclopaedia, and machine access to this data yields numerous additional applications. In contrast to the case of links, attribute values are usually given as plain text, so that there is no such straightforward syntactical extension for marking this information. Yet we settled down to introduce a syntax that is very similar to the above markup for typed links. Namely, in order to make, e.g., the value for the population of London explicit, one writes `[[population:=7,421,328]]`. Using `:=` instead of `::` allows for an easy distinction between attributes and typed links, while also reflecting the propinquity of these concepts.

In spite of these advantages, we are aware that introducing link syntax for parts of text that do not create hyperlinks might be a possible source of confusion for users. But the fact that most characters are allowed inside MediaWiki articles severely restricts the syntactical options. Therefore, MediaWiki also uses link syntax for denoting category membership, for relating articles across languages, and for including images, each of which does not create a normal hyperlink at the place of the markup. We thus believe that our choice is tenable, but we also allow to syntactically encapsulate annotations into Wikipedia's *template* mechanism, as described in Section 3.4.

Besides this, attributes behave largely similar to typed links from a user perspective. In particular, we enable users to freely choose names for new attributes and to provide alternative texts with the markup. The latter feature is often helpful, e.g. in order to write "`London has a population of [[population:=7,421,328|around 7.5 million]]`."

Combining this information with the semantic data discussed above, the system should be able build a list of all European cities ordered by their population on the fly. To enable this, we also need a powerful query language and tools that support it. Fortunately, such languages have been developed for various explicit representations of knowledge, SPARQL[5] being the most recent outcome of these developments. Disclosing these achievements to our system requires us to describe semantic data in RDF and to provide an appropriate storage architecture, as discussed in Section 4. Another challenge is to develop user interfaces that grant access to such powerful query mechanisms without requiring knowledge of SPARQL syntax. Luckily, we can address this problem gradually, starting with intuitive special-purpose interfaces that restrict expressivity. Section 5 sketches how our system supports the creation of such tools in a way that is mostly independent from our implementation.

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

Figure 3: Source of an article on London with semantic extensions.

So far, attributes appear to be quite similar to typed links, but there are a number of further issues to be taken into account. For example, a statement like "Greater London has a total area of 609" does not mean anything. What is missing here is the *unit of measurement.* To solve this problem, our current implementation provides generic support for such units in a fully intuitive way. In the present example, the user would just write "`[[total area:=609 square miles]]`," though the unit could also be denoted as "mi^2," "sq mile," and much more. Observe that there are often multiple identifiers to denote the same unit, but that there are also different units to measure the same quantity. Therefore, the system offers support for multiple unit identifiers. In a growing number of cases, the system provides automatic conversion of a value to various other units as well, such that searches can be conducted over many values irrespective of their unit. This allows users to state their values either in square miles or square kilometers.

For the user, these features are completely transparent: she just enters the unit of her choice. Of course, the degree of support varies among units. Users also receive the immediate benefit of having converted values displayed within the article. Thus the markup given in Figure 3 does indeed produce the output of Figure 1. Finally, if a unit is unknown, the (numerical) value is just processed together with its (textual) unit identifier. Note that the rendered value within the text of a wiki page is never affected by any automatism and always displayed as entered.

Due to the fact that units can have ambiguous identifiers (e.g. "ml" for "miles" and for "millilitres"), users must be able to state which kind of units are supported for an attribute. Many features, such as sorting results according to some attribute, also require knowledge about its basic datatype (integer, float, string, calendar date, . . .). Details on how unit support is implemented, and on how users can supply the required type information are given in Section 4 below. Here, we only remark that users will usually find a sufficient amount of existing attributes, so that they do not have to bother about their declaration. Furthermore, many different attributes behaves completely similar with respect to the type of data and the supported units. For example, the declaration of `total area` can be reused for `water area`, `surface area`, `acreage`, and many other kinds of areas.

3.4 Semantic Templates

With the above features, the system is also able to implement a technology sometimes referred to as *semantic templates.* Wikipedia already offers a template mechanism that allows users to include predefined parts of text into articles. Templates can also include placeholders which are instantiated with user-supplied text when the template is included

[5] `http://www.w3.org/TR/rdf-sparql-query/`

into an article. This feature allows to put varying content into a fixed format, and was mainly introduced to ensure a consistent layout among articles. However, by simply adding typed links or attributes to the template text, our implementation allows to employ templates for encapsulating semantic annotation. In some cases, one could even modify existing templates to obtain a large amount of semantic data without changing any article. Yet, many existing uses of templates disallow such semi-automatic annotation. Indeed, placeholders in templates are often instantiated with short descriptive texts, possibly containing multiple links to other articles, such that no single entity can be annotated. Yet, semantic templates are a valuable addition to our approach, since they can simplify annotation in many cases.

3.5 User Experience

We already saw that editors experience only small changes in form of some easy to grasp syntax, appearing at various places within articles. Here, we discuss some other changes that users will encounter in a Semantic Wikipedia.

Most prominently, users now find an infobox for semantic information at the bottom of each page. It helps editors to understand the markup, since it immediately shows how the system interprets the input. But the infobox also provides extended features for normal users. As shown in Figure 1, typed links are augmented with quicklinks to "inverse searches." For example, if the article about *Hyde Park* states that it is located in London, a single click in the infobox suffices to trigger a search for further things with this property.

Additionally, data values can be connected with special features based on their datatype. For instance, articles that specify geographic coordinates can be furnished with links to external map services[6]. In the case of calendar dates, links could refer to specialised searches that visualise a timeline of other events around the given date.

Of course it is also possible to conduct searches directly. New special pages for "*Semantic Search*" will allow users to pose advanced queries. Providing user interfaces for this task is not at the core of our current work, but the system already includes a simple search for articles that have a certain relation to a given (set of) other articles. We provide means for developers to add their own search interfaces in an easy way, and we expect that many customised searches to appear (both experimental and stable, both within Wikipedia and on external sites).

4. DESIGN

In this section, we device an architecture for a concrete implementation. We start by introducing the overall format and architecture for data storage, and continue with discussing the workflow for evaluating typed links and attributes given in articles. For attributes, this includes declaration, error handling, and unit conversion.

Seeking a suitable formal data model, we notice the close resemblance of our given input data to RDF and RDFS. Typed links basically describe RDF properties between RDF resources that denote articles, attributes correspond to properties between articles and RDF data literals, and Wikipedia's current classification scheme can be modelled with RDFS-classes. Note that Wikipedia already restricts the use of categories to the classification of articles—there are no categories of categories. In other words, we are dealing with a fragment of RDFS that is largely compatible with the semantics of OWL DL, and we might choose representation in either formalism without major semantic obstacles.[7]

These considerations enable us to view our system as a convenient tool to author RDF specifications, removing technical barriers to allow cooperative editing by a large number of users. Though this viewpoint neglects the specific scenario of a semantic Wikipedia, it is helpful for the more technical parts of the architecture. In particular, the consistent and well-understood data model of RDF simplifies design choices and allows us to reuse existing software.

The availability of mature *free*[8] software for processing and storing RDF is indeed very important. First of all, it helps us to tackle the huge scalability challenges linked to our usage scenario: Wikipedia has to cope with a massive amount of simultaneous read and write request over a steadily growing database. Specialised RDF databases ("triplestores") available today should be able to deal with the expected loads[9], and do even provide basic reasoning support for RDFS constructs. Furthermore, recent RDF-tools provide simplified data access via query languages like SPARQL which can be used to implement search interfaces. Free RDF software libraries like Redland[10] facilitate the task for external developers who want to reuse semantic data exported from Wikipedia in an XML-based RDF serialisation. The overall architecture we envision is pictured in Figure 4, which we will consider in more detail in Section 5.

4.1 Typed Links

In the following, we explicate how the above ideas can be substantiated in a concrete architecture that allows input, processing, and export of typed links in MediaWiki. This encompasses the treatment of categories, attributes, and schema information as well, though some additional measures are required for some of these cases.

Section 3 explained how users can include type information into existing links. According to our above discussion on the relationship to RDF, the type that a user gives to a link is just a string that serves as a label for an RDF property. Thus, proper encoding in RDF can be achieved without requiring users to declare types of links in advance, and arbitrary names can be introduced by users. Yet, it is clear that the complete lack of such schema information complicates the task of annotation which requires consistent usage of type identifiers. Many names might be introduced for the same relationship, and, possibly more problematic, the same name might be used with different meanings.

These problems are already encountered in the current usage of Wikipedia's categories, and experience shows that

[6] Experimental versions of such services are already provided in English Wikipedia, but the focus there is not on further usage of the data.

[7] One must be aware that Wikipedia's categories describe collections of articles, as opposed to categories of article subjects. E.g., the category "Europe" denotes a class of articles *related* to Europe, not the class of all "europes".

[8] Being free (as in speech) is indispensable in our setting, since only free software is used to run Wikipedia.

[9] English Wikipedia *currently* counts 800,000 articles, but it is hard to estimate how many RDF triples one has to expect. High numbers of parallel requests yield another invariant often ignored in benchmarks of current stores.

[10] `http://librdf.org/`

they are not critical. However, to ameliorate the situation, one allows for the creation of special articles (distinguished by their Wikipedia *namespace* “Category:”) that provide a human readable description for most categories. Similarly, we introduce a namespace “Relation:” that allows to add descriptions of possible relations between articles.[11]

In spite of these additions, the processing of typed links entered by a user does not involve articles on relations. Hence only local data sources need to be accessed for transforming typed links into RDF-triples that are sent to the triplestore which serves as the storage backend. To assure that the triplestore contains no information that is not contained in the text of the article, we must delete outdated triples prior to saving new information. Luckily, every article affects exactly those triples that involve this article as the subject, so that we just have to delete these triples on update. Note how crucial this *locality* of our approach is for obtaining a scalable system. Summing up, any user-triggered update of an article leads to the following three steps:

1. Extraction of link types from the article source (parsing).
2. Transformation of this semantic information into RDF triples.
3. Updating the triplestore by deleting old triples and storing new ones.

Each of these operations can be performed with a minimum of additional resources (computation time, bandwidth), and the update of the triple store could even be achieved asynchronously after the article text was saved. Edit conflicts by concurring changes are detected and resolved by the MediaWiki software.

Our architecture also ensures that the current semantic information is always accessible by querying the triple store. This is very convenient, since applications that work on the semantic data, including any search and export functions, can be implemented with reference to the storage interface independently from the rest of the system. In Section 5, we give some more details on how this is achieved. Here we just note the single exception to this general scheme of accessing semantic data: the infobox at the bottom of each article is generated during parsing and does not require reading access to the triplestore. This is necessary since infoboxes must be generated even when previewing articles *before* they are actually saved, but of course it also reduces overhead.

4.2 Attributes

Attributes are similar to typed links, but describe relationships between an article and a (possibly complex) data value, instead of between two articles. With respect to the general architecture of storage and retrieval, they thus behave completely similar to typed links. But additional challenges for processing attribute values have already been encountered in Section 3. In this section, we formally relate the lexical representation of attribute data to the data representation of XML Schema. This leads us to consider the declaration of datatypes for semantic attributes, before focussing on error handling and the complex problem of unit support.

4.2.1 Value Processing and Datatypes

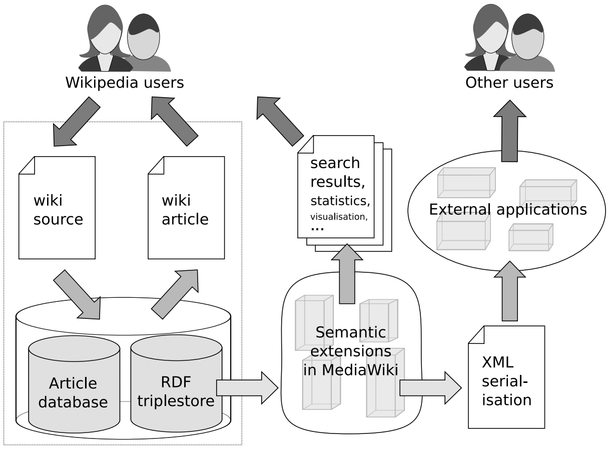


Figure 4: Basic architecture of the semantic extensions to MediaWiki. On the left, Wikipedia users edit articles to enter semantic information (see Section 4). Further extensions to MediaWiki use this data and export it to external applications (see Section 5).

A primary difficulty with attributes is the proper recognition of the given value. Since attribute values currently are just given as plain text, they do not adhere to any fixed format. On the other hand, proper formatting can be much more complicated for a data value than for a typed link. A link is valid as soon as its target exists, but a data value (e.g. a number) must be *parsed* to find out its meaning. Formally, we distinguish the *value space* (e.g. the set of integer numbers) from the *lexical space* (e.g. the set of all acceptable string representations of integers). A similar conceptualisation is encountered for the representation of datatypes in RDF, which is based on XML Schema (XSD).

Now it is tempting to build support for attributes in Wikipedia based on the original XSD datatypes. We certainly want to use XSD representations for storing data values in RDF triples. Unfortunately, we cannot adopt lexical space from XSD, since it does usually not allow for the data representation that is most common to users, especially if their language is not English. Thus we allow users to provide input values from a defined lexical space of Wikipedia, which is usually not equal to a lexical space of XSD. Formally, we combine two conversions between data representations: (1) the conversion between Wikipedia’s lexical space and the (XSD) value space, and (2) the conversion between the value space and the according lexical space of XSD.

Thus the lexical space within Wikipedia must be localised to adapt to writing customs of other languages, while the XML Schema representation is universal, allowing for machine processing of arbitrary language Wikipedias. Parsing user provided data values ensures that the stored RDF triples conform to the XML specification, and allows us to perform further operations on the data, as described below.

Clearly, we must have prior knowledge of the lexical and value spaces involved in the conversion, since these can in general not be derived from the input. In other words, we need a predefined datatype for any attribute for which values

[11] We deviate from the label “type” here, since its meaning is ambiguous unless additional technical terms are juxtaposed.

are given. Note how this differs from the case of typed links, where arbitrary identifiers could be introduced without further declaration. In Section 4.2.2 below, we discuss how the system still can minimise the formal constraints that might obstruct intuitive usage.

We arrive at a system that consists of three major components: *data values* (e.g. “3,391,407”) are assigned to *attributes* (e.g. “population”) which are associated with a given *datatype* (e.g. “integer”). To allow users to declare the datatype of an attribute, we introduce a new Wikipedia namespace “Attribute:” that contains articles on attributes. Within these articles, one can provide human-readable descriptions as in the case of relations and categories, but one can also add semantic information that specifies the datatype. To minimise the amount of new syntactic constructs, we propose to use the concept of typed links as introduced in this paper. Therefore we reserve a third (and last) namespace for types: “`Type:`”. Then, using a relation with built-in semantic we can simply write

```
[[hasType::Type:integer]]
```

to denote that an attribute has this type. The declaration of datatypes can also be facilitated by templates as introduced in Section 3.4. For instance, one can easily create a template that provides the semantic information that an attribute is of type integer, but that also includes human-readable hints and descriptions.

The types themselves are built-in, and their articles only serve to give a description of proper usage and admissible input values. If need arises, one could also conceive a system to customise types by giving additional descriptions inside the articles in the “Type:”-namespace. It is important to understand that a type in Wikipedia always implies some XML Schema datatype, but is generally more specific than this. For example, one can have many types that support XSD “date” while mapping its values to different (language specific) lexical spaces in Wikipedia.

Note that the general workflow of processing attribute input becomes slightly more complex than in the case of typed links. Before extracting the attribute values from the article source, we need to find out about the datatype of an attribute. This requires an additional reading request to the storage backend and thus has some impact on performance. Since other features, such as template inclusion, have similar requirements, we are optimistic that this impact is tenable.

4.2.2 Error Handling

In contrast to typed links, the handling of attributes involves cases where the system is not able to process an input properly, although the article is syntactically correct. Such a case occurs whenever users refer to attributes for which no datatype was declared, but also if the provided input values do not belong to the lexical space of the datatype.

A usual way to deal with this is to output an error that specifies the problem to the user. However, in a wiki-environment, it is vital to tolerate as much errors as possible. Reasons are that technical error messages might repel contributors, that users may lack the knowledge for understanding and handling error messages, and, last not least, that tolerating (temporal) errors and inconsistencies is crucial for the success of collaborative editing. Thus our system aims at catching errors and merely issuing warning messages wherever possible.

For the case of missing datatype declarations, this can be achieved by presuming some feasible datatype based on the structure of the input. Basically, when encountering a numerical value, the input is treated as floating point number. Otherwise it is processed as a string. A warning inside the infobox at the article bottom informs the user about the possible problem. Here we exploit that RDF includes a datatype declaration for each value of a property. A property can thus have values of multiple datatypes within one knowledge base. This is also required if an existing datatype declaration is changed later on (recall that anyone can edit declarations). The new type only affects future edits of articles while existing data is still valid.

If the datatype is specified but the input does not belong to the supported lexical space, we do not store any semantic information and restrict to issuing a warning message within the infobox.

4.2.3 Units of Measurement

The above framework provides a feasible architecture for treating plain values, like the number of inhabitants of some city. However, many realistic quantities are characterised not only by a numerical value, but also by an associated unit. This is a major concern in an open community where users may have individual preferences regarding physical units. Units might be given in different systems (kilometres vs. miles) or on different scales (nanometres vs. kilometres). In any case, reducing such input to a mere number severely restricts comparability and thwarts the intended universal exchange of data between communities and languages. Using different attributes for different units is formally correct, but part of the problem remains (values of “length (miles)” and “length (kilometres)” remain incomparable to RDF tools).

Our solution to this problem is twofold. On the one hand, we provide automatic unit conversion to ensure that large parts of the data are stored in the same unit. On the other hand, we recognise even those units for which no conversion support is implemented yet, so that we can export them in a way that prevents confusion between values of different units. To this end, note that it is fairly easy to separate a (postfixed) unit string from a numerical value. Given both the numerical value and the unit string, one can unambiguously store the information by including unit information into attribute (property) names. For example, the input `[[length:=3km]]` is exported as value 3 of a property identified as “length#km”. Users can freely use new units, and exported data remains machine-processable.

Since the power of semantic annotation stems from comparing attributes across the database, it is desirable to employ only a small number of different units. Users can achieve this manually by converting values to some standard unit and giving other units as optional alternative texts only. We automate this process by providing built-in unit conversion for common units. From the RDF output it is not possible to tell whether a unit conversion has taken place automatically, or whether the user has provided the value in the given unit right away. This has the further advantage that one can safely add unit support gradually, without affecting applications that already work with the exported data. Built-in unit support also allows us to provide automatic conversions inside the article text or infoboxes, which provides immediate gratification for using attributes.

Formally, the added unit support extends the lexical space of Wikipedia's datatypes, requiring more complex parsing functions. Consequently, unit information is also provided through the datatype. We do not restrict to primitive types like "integer" or "decimal," but may also have more complex types like "temperature" or "astronomic distance." Note that unit-enabled types also have to account for differences in scale: it is not feasible to convert any length, be it in nanometres or in light years, to metres, unless one intends to support arbitrary precision numbers.

5. CURRENT IMPLEMENTATION

Important parts of the architecture from Section 4 have already been implemented. Readers who want to touch the running system are pointed to our online demo at `wiki.ontoworld.org` and to the freely available source code.[12] We would like to encourage researchers and developers to make use of the fascinating amount of real world data that can be gathered through Semantic MediaWiki and to combine it with their own tools.

At the moment, Semantic MediaWiki is still under heavy development, and many features are just about to be implemented. Like MediaWiki itself, the system is written in PHP and uses a MySQL database. Instead of directly modifying the source code of the wiki, we make use of MediaWiki's *extension* mechanism that allows developers to register functions for certain internal events. For this reason, the Semantic MediaWiki extension is largely independent form the rest of the code and can be easily introduced into a running system.

Moreover, interested readers can easily implement their own extensions to the semantic extension. The general architecture for adding extensions is depicted in Figure 4. The box on the left represents the core functions of editing and reading, implemented according to the description in Section 4. As shown in the Centre of Figure 4, the obtained (semantic) information can be exploited by other extensions of MediaWiki by simply accessing the triplestore. Functions for conveniently doing so are provided with our implementation. Our current effort comprises some such extensions, specifically a basic semantic search interface and a module for exporting RDF in an XML serialisation. Additional extensions, such as improved search engines, can be added easily and in a way that is largely independent from our source code. MediaWiki provides means of adding "Special:" pages to the running system in a modular way, so that a broad range of semantic tools and interfaces could be registered and evaluated without problems.

Finally, the data export (as well as possible dumps of the database) can be utilised by independent external applications. Programmers thus obtain convenient access to the worlds largest encyclopaedic knowledge base, and to the results of any other MediaWiki-based cooperative online project. The possibilities this technology offers to enhance desktop applications and online services are immense – we discuss some immediate application scenarios in the next section.

6. APPLICATIONS

A wide range of applications becomes possible on the basis of a semantically enhanced Wikipedia. Here, we briefly outline the diversity of different usage areas, ranging from the integration of Wikipedia's knowledge into desktop applications, over enhanced *folksonomies*, to the creation of multilingual dictionaries. Moreover, automated access to Wikipedia's knowledge yields new research opportunities, such as the investigation of consensus finding processes.

Many desktop applications could be enhanced by providing users with relevant information from Wikipedia, and it should not come as a surprise that corresponding efforts are already underway.[13] For instance, the *amaroK* media player[14] seamlessly integrates Wikipedia articles on artists, albums, and titles into its user interface, whenever these are available. With additional semantic knowledge from Wikipedia, many further services could be provided, e.g. by retrieving the complete discography of some artist, or by searching the personal collection for Gold and Platinum albums. Similarly, media management systems could answer domain-specific queries, e.g. for "music influenced by the Beatles" or "movies that got an Academy Award and have a James Bond actor in a main role." But the latter kind of question answering is not restricted to media players: educational applications can gather factual data on any subject, desktop calendars can provide information on the current date, scientific programs can visualise Wikipedia content (genealogical trees, historical timelines, topic maps, . . .), imaging tools can search for pictures on certain topics—just to name a view.

These usage scenarios are not restricted to the desktop. A web-based interface does also make sense for any of the above services. Since Wikipedia data can be accessed freely and without major legal restrictions, it can be included in many web pages. Cooperations with search engines immediately come to mind, but also special-purpose services can be envisaged, that may augment their own content with Wikipedia data. A movie reviewer could, instead of adding the whole data about the movie herself, just use a template and integrates it on her website, pulling the data live from Wikipedia. Finally, portals that aggregate data from various data sources (newsfeeds, blogs, online services) clearly benefit from the availability of encyclopaedic information too.

On the other hand, Wikipedia can also contribute to the exchange of other information on the web. Folksonomies [15] are collaborative tagging or classification systems that freely use on-the-fly labels to tag certain web resources like websites (del.icio.us[15]), pictures (flickr[16]), or even blog entries [11]. In these applications, tagging simply means assigning a keyword to a resource, in order to allow browsing by those labels. These keywords distinguish neither language nor context. For example, the tag "reading" could either mean the city, *Reading, MA*, or the act of reading a book; the tag "gift" could either mean the German word for poison or the English synonym for present. On the other hand, different tags may mean mostly the same thing, as in the case of "country," "state," and "nation." Searching for either one of them does not yield results tagged with the other, although they would be as relevant. A semantic tagging system, however, could offer labels based on meaning rather than on potentially ambiguous text strings. This task is simplified when using concept names and synonyms retrieved from a seman-

[12] See `http://sourceforge.net/projects/semediawiki`.

[13] `http://meta.wikimedia.org/wiki/KDE_and_Wikipedia`.

[14] `http://www.amarok.org`

[15] `http://del.icio.us`

[16] `http://www.flickr.com`

tically enhanced Wikipedia in the user's chosen language. Similarly, Wikipedia's URIs can serve as a universal namespace for concepts in other knowledge bases and ontologies.

Semantic Wikipedia could also act as an incubator for creating domain ontologies, by suggesting domains and ranges for roles, or applicable roles for a certain concept (if the majority of countries seem to have a head of state, a system could suggest to add such a role to the ontology). Semantic Wikipedia also could be queried in order to populate ontologies quickly (one could, for example, pull all soccer players for their soccer ontology).

Another significant advantage is the internationality of such a semantic tagging approach, based on the fact that Wikipedia relates concepts across languages. For example, a user in Bahrain could ask for pictures tagged with an Arabic word, and would also receive pictures originally tagged with equivalent Chinese words. To some extent, semantically interlinking articles across different languages would even allow to retrieve translations of single words—especially when including Wiktionary[17], a MediaWiki-based dictionary that already exists for various languages.

The wealth of data provided by Semantic Wikipedia can also be used in research and development. A resource of test data for semantically enabled applications immediately comes to mind, but social research and related areas can also profit. For example, comparing the semantic data, one can highlight differing conceptualisations in communities of different languages. Also ontology engineering methodologies begin to consider consensus finding processes as a crucial part of ontology creation [23]. Researchers could observe how knowledge bases are collaboratively built by the Wikipedia community and how such communities reach consensus. Wikipedia records all changes as well as discussions associated to articles. Combined with the evolving semantic annotation, this might allow insights on discussion and collaboration processes as well.

7. RELATED APPROACHES

The general concept of using a wiki for collaboratively editing semantic knowledge bases is appealing, and it should not come as a surprise that many approaches towards this goal exist. Most of them are subject to ongoing development.

Platypus Wiki was introduced in 2004 as one of the first semantic wikis [18, 8]. It allows users to add semantic information to a dedicated input field, which is separated from the field for editing standard wiki text. Users provide semantic data by writing RDF statements in N3 notation [5]. This approach influenced later systems, such as the Rhizome wiki [21, 22] that provided very similar functionality. Both approaches are very RDF-centric and probably hard to understand for ordinary users, as they require a good understanding of RDF and a some skill in writing N3.

One year later, the prototype WikSAR [3, 2] was presented, winning the *Best Demo Award* at the European Semantic Web Conference. The system uses a different, more tightly integrated syntax. Users can now simply make semantic statements within the text by writing lines of the form `PredicateLabel:ObjectLabel`. As in our work, statements in this notation are interpreted as RDF-triples with the page title being the subject. A similar approach was taken by [16]. Both approaches improve usability by having the semantic content together with the wiki text. These approaches still lacked the tight integration of explicit machine-readable data and human-readable editable text that we are introducing.

There were approaches that put a stronger emphasis on formal structure, and hence are more geared towards editing ontologies instead of text. For instance, [1] resembles more an ontology editor than a wiki. Although it could be used as a wiki, it lacks an easy way to create links, which is a central issue of common wikis. The unpublished *KendraBase* [12] is similar in scope but provides a more wiki-like interface.

Recently, some personal semantic wikis were proposed [4, 17]. They are designed to be used as desktop applications, and thus do not emphasise the community process that is typical for classical wikis. However, with respect to the way in which they integrate human- and machine-readable content, these systems are comparable to [2] and [16].

With regards to attributes, WikSAR [2] introduced this idea. However, we are not aware of any approach that supports data types and unit conversion to the extent of our present work. This feature allows the for a world-wide integration, spanning different unit systems.

Finally, some other systems offer domain-free machine-readable knowledge bases. OpenCyc[18] collected more than 300,000 assertions (comparable to the typed links in our setting). In spite of its name, OpenCyc is not fully open: it is free for usage, but it does not allow free contributions. A former online project called MindPixel[19] allowed users to provide statements and evaluate them as either true or false. Currently, the project is down for maintenance, but it collected already 1.4 million statements as of January 2004. A similar system is described in [19].

8. CONCLUSIONS AND OUTLOOK

We have shown how Wikipedia can be modified to make part of its knowledge machine-processable using semantic technologies. On the user side, our primary change is the introduction of *typed links* and *attributes*, by means of a slight syntactic extension of the wiki source. By also incorporating Wikipedia's existing category system, an impressive amount of semantic data can be gathered. For further processing, this knowledge is conveniently represented in RDF-based format.

We presented the system architecture underlying our actual implementation of these ideas, and discussed how it is able to meet the high requirements on usability and scalability we are faced with. The outcome of these considerations is a working implementation which hides complicated technical details behind a mostly self-explaining user interface.

Our work is based on a thorough review of related approaches, and we are continuously discussing our proposals with users and editors of Wikipedia. Recent presentations of parts of this work have been received very positively by the community [13], and lead to further constructive exchange of ideas. Considering how quickly earlier extensions, such as the category system, have been introduced into Wikipedia, we thus have strong reasons to believe that our implementation will be used in English Wikipedia by the end of 2006.

Various issues remain topics for future research. For ex-

[17] `http://wiktionary.org`

[18] `http://www.opencyc.org`

[19] `http://mindpixel.org`

ample, the addition of more expressive schema information (inverse and symmetric relations, meta-modelling, consistency checks, etc.) as supported by OWL or RDFS (but not always by both) requires additional discussions. Moreover, in order to get knowledge bases small enough to fit existing tools, the RDF-graph might need to be pruned and relevant subgraphs have to be identified. For specific uses it might be required to map Semantic Wikipedia to existing knowledge bases. In addition to general ontology alignment issues, there will be an extra challenge, because Wikipedia is neither complete nor consistent nor particularly homogeneous.

We have demonstrated that the system provides many immediate benefits to Wikipedia's users, such that an extensive knowledge base might be built up very quickly. The emerging pool of machine accessible data bears great opportunities for developers of semantic technologies who seek to evaluate and employ their tools in a practical setting. In this way, Semantic Wikipedia can become a platform for technology transfer that is beneficial both to researchers and a large number of users worldwide, and that really makes semantic technologies part of the daily usage of the World Wide Web.

Acknowledgements. We wish to thank the community of Wikipedians for their comments. This research was partially supported by the European Commission under contract IST-2003-506826 "SEKT," and FP6-507482 "Knowledge Web Network of Excellence," and by the German BMBF project "SmartWeb." The expressed content is the view of the authors but not necessarily the view of any of the mentioned projects as a whole.

9. REFERENCES

[1] S. Auer. Powl – a web based platform for collaborative semantic web development. In *Proceedings of 1st Workshop Scripting for the Semantic Web (SFSW'05), Hersonissos, Greece, May 30, 2005*, May 2005.

[2] D. Aumüller. Semantic authoring and retrieval in a wiki (WikSAR). In *Demo Session at the ESWC 2005*, May 2005.

[3] D. Aumüller. SHAWN: Structure helps a wiki navigate. In W. Mueller and R. Schenkel, editors, *Proceedings of the BTW-Workshop WebDB Meets IR*, March 2005.

[4] D. Aumüller and S. Auer. Towards a semantic wiki experience – desktop integration and interactivity in WikSAR. In *Proceedings of 1st Workshop on The Semantic Desktop, Galway, Ireland*, 2005.

[5] T. Berners-Lee. Primer: Getting into RDF & Semantic Web using N3, 2005. Available at `http://www.w3.org/2000/10/swap/Primer.html`.

[6] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001. Available at `http://www.sciam.com/2001/0501issue/0501berners-lee.html`.

[7] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/rdf-schema/`.

[8] S. E. Campanini, P. Castagna, and R. Tazzoli. Platypus wiki: a semantic wiki wiki web. In *Semantic Web Applications and Perspectives, Proceedings of 1st Italian Semantic Web Workshop*, Dec 2004.

[9] W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, 2001.

[10] D. C. Fallside and P. Walmsley. Xml schema part 0: Primer second edition, Oct 2004.

[11] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 2006. to appear.

[12] D. Harris and N. Harris. Kendrabase. Available at `http://kendra.org.uk/wiki/wiki.pl?KendraBase`.

[13] M. Krötzsch, D. Vrandečić, and M. Völkel. Wikipedia and the Semantic Web – the missing links. In *Proceedings of the 1st International Wikimedia Conference, Wikimania 2005*, Aug 2005.

[14] F. Manola and E. Miller. Resource Description Framework (RDF) primer. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/rdf-primer/`.

[15] A. Mathes. Folksonomies – cooperative classification and communication through shared metadata. *Computer Mediated Communication*, Dec 2004.

[16] H. Muljadi and T. Hideaki. Semantic wiki as an integrated content and metadata management system. In *Poster Session at the ISWC 2005*, Nov 2005.

[17] E. Oren. Semperwiki: a semantic personal wiki. In *Proceedings of 1st Workshop on The Semantic Desktop, Galway, Ireland*, Nov 2005.

[18] S. E. Roberto Tazzoli, Paolo Castagna. Towards a semantic wiki wiki web. In *Demo Session at ISWC2004*.

[19] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 – Confederated International Conferences DOA, CoopIS, and ODBASE 2002*, pages 1223–1237, London, UK, 2002. Springer-Verlag.

[20] M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Recommendation 10 February 2004, available at `http://www.w3.org/TR/owl-guide/`.

[21] A. Souzis. Rhizome position paper. In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, Sept 2004.

[22] A. Souzis. Building a semantic wiki. *IEEE Intelligent Systems*, 20:87–91, 2005.

[23] D. Vrandečić, H. S. Pinto, Y. Sure, and C. Tempich. The diligent knowledge processes. *Journal of Knowledge Management*, 9(5):85–96, Oct 2005.

[24] J. Wales. Wikipedia and the free culture revolution. OOPSLA/WikiSym Invited Talk, Oct 2005.