Reusing Ontological Background Knowledge in Semantic Wikis

Denny Vrandečić and Markus Krötzsch

{vrandecic,kroetzsch}@aifb.uni-karlsruhe.de
AIFB, Universität Karlsruhe, Germany

Abstract. A number of approaches have been developed for combining wikis with semantic technologies. Many semantic wikis focus on enabling users to specify properties and relationships of individual elements. Complex schema information is typically not edited by the wiki user. Nevertheless, semantic wikis could benefit from taking existing schema information into account, and to allow users to specify additional information based on this schema.
In this paper, we introduce an extension of Semantic MediaWiki that incorporates schema information from existing OWL ontologies. Based on the imported ontology, the system offers automatic classification of articles and aims at supporting the user in editing the wiki knowledge base in a logically consistent manner. We present our prototype implementation which uses the *KAON2* ontology management system to integrate reasoning services into our wiki.

1 Introduction

Wikis allow for the simple and quick editing of web content. They considerably lower the barrier for contributing to a website, and especially allow to fix small glitches and errors quickly. This has lead to the deployment of wikis both for web communities – such as the programming languages pattern group where the idea of wikis originated [10] – and as an extension to existing intranet systems in corporate settings.

As of today, a number of approaches have been developed in order to combine wikis with semantic technologies, and actual implementations of semantic wikis are becoming more and more mature. While the actual goals and methods of existing systems vary greatly, many semantic wikis focus on enabling users to specify properties and relationships of individual elements or articles. Complex schema information, as considered in expressive semantic web languages such as OWL [23], is typically not considered for being edited in the wiki.

Instead of extending semantic wikis into general purpose ontology editors – which is a task where dedicated ontology editors [19, 25, 14] are usually more suitable for – we investigate how semantic wikis can benefit from existing, possibly expressive ontologies. Especially in corporate settings, a fixed schema for metadata is often already in use and part of the internal workflow of the company. Changes to the schema typically are a well managed task. Still, wikis can be a convenient tool for managing or annotating instance data for complex ontologies. Schemas often are quite stable, whereas instance data changes at a much higher pace. In practice, this means that semantic wikis should be able to take existing schema information into account, and allow users to specify additional information based on this schema.

In this paper, we present an extension of Semantic MediaWiki [15, 27] that incorporates schema information from existing OWL ontologies. Based on an imported ontology, the user interface facilitates the reuse of the ontological concepts to categorize articles and the properties to type links. It offers an automatic classification of articles and aims at assisting the user to edit the wiki knowledge base in a logically consistent manner. In our prototype implementation, we employ the KAON2 [12] system to integrate necessary reasoning services into our wiki.

The following section describes different use cases and scenarios, where the presented system would be of advantage. This is followed by the details of our mapping to OWL (Sect. 3) and concrete uses of ontological data (Sect. 4). We then describe how the implementation is working and how our system is used (Sect. 5). Before we conclude our work in the last section, we offer an overview over related work (Sect. 6).

2 Use Cases and Requirements

We start our investigations by discussing typical use cases for the reuse of ontological knowledge in semantic wikis. This motivates our approach and allows us to derive basic requirements for its technical realization.

Since one obvious requirement for any practical system is that it succeeds in performing its computations, we explicitly exclude global scale wikis such as Wikipedia[1] from our target environments. In the context of expressive ontology languages, one is quickly confronted with computational problems that are not tractable, so that semantic technologies typically do not scale indefinitely. However, wikis are also employed very successfully in contexts where the overall amount of data is much smaller.

Since we started to develop Semantic MediaWiki, we have been approached several times by companies that are interested in using semantic wikis in an enterprise setting. Wikis indeed are successfully applied to cooperatively manage knowledge in working groups or project teams. Concurrent access and ease of use are important advantages over other content management systems in this setting. The added value of semantic wikis in this scenario is the ability to leverage the wiki's contents in other enterprise applications.

For a concrete scenario, consider a wiki used for coordinating a particular project team within a company. Using semantic technologies, relevant parts of the wiki data shall automatically be gathered by the company's intranet search engine. In the wiki, project members coordinate their activities, and describe their progress on their deliverables. This data can then be collected from the wiki and reused in other applications, e.g. to create monthly report figures, or even up-to-date status reports that are generated on request. As the semantic wiki reuses the company's metadata schema for documents and respects the associated constraints (e.g. no document must have more than one title and topics must stem from a predefined set of topics), the automatic integration into the corporate information infrastructure works smoothly.

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

Another frequent use case is to use existing ontologies to bootstrap the contents and vocabulary of a semantic wiki. For a concrete example, assume that an international conference wants to use a wiki for gathering information around the event. Participants can use the system to exchange information about accommodation and travel, to coordinate Birds-of-a-feather (BOF) sessions, or to actively provide links to their presentation material. At the same time, the organizers publish official schedules on protected wiki pages. Using a semantic wiki, this data can be queried and extended in complex ways, e.g. to provide a scheduling system that suggests sessions and BOF sessions based on a participants interests. Also, if the conference management system supports some form of RDF export, one can initialize the wiki pages with basic information about accepted papers and participants. The ESWC2006 wiki[2] is based on such a bootstrapped system.

In a third example, we consider the usage of semantic wikis in personal knowledge management [20]. There, the wiki is operated as a desktop application and cooperative editing is not required. Semantic technologies simplify data organization and search, and the machine-processable annotations provide suitable interfaces with other semantic desktop applications. For instance, the wiki can be used to take notes about persons, and one would like to combine this information with address book applications. Using vocabulary from existing ontologies, the wiki becomes compatible with various types of metadata, and thus its information could be used in RDF based desktop tools. Another example would be to import Friend-of-a-friend [8] files directly from the web.

All of the above use cases stem from our practical experience, and we have been asked by companies and organizations to support them. The following requirements are emerging from these use cases:

- referring to existing ontological vocabularies from the wiki,
- incorporating schema information and constraints from external ontologies,
- exporting data from the wiki in a standard ontology language,
- importing data from external ontologies, such that it is represented and edited through the wiki.

Note the difference between fully importing external data into the wiki and merely incorporating external information for editing tasks. The former implies that the imported knowledge is represented in the wiki afterwards and can be edited by users, whereas the latter means that the wiki is aware of additional external data or constraints that must be taken into account, but that cannot be modified within the wiki. This is further described in Sect. 4.1.

We argue that the full import and representation of all kinds of complex schema information into the wiki is not an immediate requirement, and is often not even desirable at all. Firstly, from a user perspective, it is quite complicated to edit such schema information within a wiki, and dedicated graphical user interfaces of full-fledged ontology editors might be much better suited for this task [19, 25, 14]. Secondly, in many of the above use cases the ontological schema should not be changed by users of the wiki at all. In the opposite, ontologies are often considered as vehicles to achieve interoperability with other applications, and this requires that all participants adhere to the original

[2] `http://wiki.eswc2006.org`

schema. The evolution of the schema is usually done by a central board, as described, for example, by the DILIGENT methodology [28]. Thirdly, distributed ontology editing in itself is not a trivial topic. In contrast to the situation in software engineering, it is neither easy nor usual (although desirable) to separate ontologies into independent modules of networked ontologies. Furthermore, small changes in some parts of an ontology can have strong effects on the overall semantics.

3 Semantic MediaWiki in Terms of OWL DL

Semantic MediaWiki is an extension to the MediaWiki system that allows users to add various types of ontological information to the wiki, and which forms the basis of the implementation that we describe in Sect. 5. In this section, we relate the formal information gathered within this wiki system to the Web Ontology Language OWL. In particular, we discuss export and import of OWL DL ontologies.

3.1 Extracting Ontological Knowledge from the Wiki

We first describe the expressive means that are available in the wiki, and specify their semantics in terms of the OWL DL part of the Web Ontology Language. This defines the formal semantics of the annotations that are used in the wiki such that a canonical OWL DL export becomes possible. We remark that the user interface of Semantic MediaWiki does not strictly require the formal interpretation in terms of OWL DL, or the restriction to the expressive means of this language. Since most complex features of OWL are not used in Semantic MediaWiki, one could even argue that the wiki's annotation mechanism might as well be used to author different ontology languages. However, we make use of some characteristic OWL features such as equality reasoning and transitive roles, and we feel that OWL DL's set-based semantics for classes and roles is more intuitive than the semantics of RDFS.

Also recall that OWL DL is conceptually related to description logics. In particular, one can divide ontological elements into *instances* that represent individual elements of the described domain, *classes* that represent sets of individuals, and *roles* which represent binary relations between individuals. The way in which Semantic MediaWiki represents knowledge was partially inspired by OWL DL and one can naturally relate the elements of the wiki, i.e. the individual content pages, to the basic vocabulary of OWL. Technically, the MediaWiki system employs *namespaces* to distinguish several types of content pages, and our semantic interpretation exploits this mechanism of "typing" pages as follows:

OWL individuals are represented by normal article pages. These pages typically constitute the majority of the wiki's contents, mostly contained in the MediaWiki's `Main` namespace. However, there are several additional namespaces, such as `Image` or `User`, which also are interpreted as individuals.

OWL classes in turn have natural counterparts in the wiki in form of MediaWiki *categories*. The category system, which was introduced only in 2004 [29], quickly became the most important feature for classifying articles in Wikipedia. Categories are represented as pages within the `Category` namespace. They can be organized in a hierarchical way, but it is not possible to make a category contain other categories. Thus Wikipedia's category system is more similar to the semantics of classes in OWL DL than to the semantics of classes in RDFS.[3]

OWL properties, i.e. roles in description logic, do not have a counterpart in MediaWiki, and were introduced by the Semantic MediaWiki extension. OWL further distinguishes object-properties (describing relationships between two individuals) from data-properties (associating individuals with values of a given datatype), and a similar distinction is found in Semantic MediaWiki. Object-properties are represented by pages in the namespace `Relation`, whereas data-properties are represented by pages in the namespace `Attribute`.

In addition to the above correspondences, the usage of some namespaces in MediaWiki suggests to ignore them completely for semantic interpretation. Most prominently, this includes all `Talk` pages since they do not represent separate concepts, but are merely used to collect notes and discussions on other articles. Semantic MediaWiki can be configured to ignore annotations given on such pages according to intended usage.

Based on the above mapping to OWL, Semantic MediaWiki allows users to describe various ontological statements within the wiki. An incomplete overview of OWL constructs that can be represented in the wiki is given in Table 1. OWL statements about some OWL individual/class/role are specified in Semantic MediaWiki by providing annotations on the wiki page that corresponds to this element. For example, in order to state that the object-property *is located in* holds between *SemWiki2006* and *Budva*, one writes `[[is located in::Budva]]` within the article about SemWiki2006. Further details on annotation in Semantic MediaWiki and the underlying principles are found in [27].

Semantic MediaWiki includes an export function that generates OWL/RDF documents according to mappings such as the ones in Table 1. The export function also associates URIs with all wiki pages. These URIs bijectively correspond to concrete pages. However, they are not *identical* to the article URLs in order to prevent confusion between ontological concepts (e.g. the city of Budva) and actual HTML documents (e.g. the article about Budva).

3.2 Adding External Ontologies

Table 1 indicates that, besides some rather simple schema information, the wiki is mainly used to provide concrete descriptions of individuals and their relationship. In description logics, this assertional part of a knowledge base is known as the *ABox*, and

[3] Note that the informal semantics of categories as used in Wikipedia varies. E.g. *Montenegro* does not describe the class of all "Montenegros," but the class of all article topics related to Montenegro.

Table 1. Representation of OWL constructs in Semantic MediaWiki.

OWL	**Semantic MediaWiki**
OWL individual	normal article page
owl:Class	article in namespace Category
owl:ObjectProperty	article in namespace Relation
owl:DatatypeProperty	article in namespace Attribute
Statement about element *page*	**Syntax in wiki-source of *page***
object-property	[[property_name::object_article]]
attribute-property	[[property_name:=value_string]]
rdf:type *class_name*	[[Category:*class_name*]] (on article pages)
rdfs:subClassOf *class_name*	[[Category:*class_name*]] (on category pages)

in our case we even restrict to ABox statements without any complex concept terms. Moreover, the annotations in Semantic MediaWiki are restricted in such a way that the exported OWL/RDF cannot be logically inconsistent. In the following, we discuss how the ontologically simple knowledge base of the wiki can be combined with complex schema information from an external ontology.

We wish to combine the OWL representation of the wiki contents with an external OWL ontology. Since merging of OWL DL specifications is trivially achieved by taking the union of their statements, the only problem in doing so is the mapping between elements of the two ontologies. It was already mentioned that URIs for all elements of the wiki are generated automatically. These URIs are very similar to the page URLs of the wiki, and do generally not agree with the URIs used in the external ontology. However, OWL provides us with expressive means to describe that two different URIs represent the same entity (with respect to its extensional interpretation). Depending on the type of entity that one considers, this is achieved with owl:sameAs, owl:equivalentClass, and owl:equivalentProperty. We incorporate this specification into the wiki via a new attribute equivalent URI as shown in Table 2. To give an example, on the page Category:Person one could add the statement [[equivalent URI:=http://xmlns.com/foaf/0.1/Person]], thus stating that every page tagged with this category describes a person in the sense of the FOAF vocabulary.

Table 2. Mapping of concepts to external URIs in Semantic MediaWiki.

OWL	**Semantic MediaWiki**
owl:sameAs *URI*	[[equivalent URI:=*URI*]] (on article pages)
owl:equivalentClass *URI*	[[equivalent URI:=*URI*]] (on category pages)
owl:equivalentProperty *URI*	[[equivalent URI:=*URI*]] (on relation/attribute pages)

While this provides a convenient way of mapping ontologies, this new expressive feature must be handled with care. The reason is that it yields numerous ways of creating undesired OWL/RDF specifications with the wiki. For example, one can create logical inconsistencies by declaring a non-empty wiki-class equivalent to `owl:Nothing`. While this can still be disallowed rather easily, it is also possible to make statements that are not possible in OWL DL but only in the undecidable OWL Full variant of the language. For example, one could assign the same URI to both a class and an individual, or one could even assign the URIs of language constructs such as `rdf:type` to wiki elements. Clearly, this often leads to a meaningless specification.

Both logical inconsistency and the usage of the undecidable language OWL Full prevent effective query answering over the wiki knowledge, and we therefore suggest to use `equivalent URI` only on pages that cannot be edited by arbitrary, possibly anonymous users. Since our use cases evolve around wikis of limited size that are used in a closed community, it is also realistic to assume that problems can be avoided by instructing users accordingly. Another option, discussed below, is to build appropriate checks into the wiki. One could also disallow the use of `equivalent URI` and require an external ontology where a mapping between the URIs of the wiki knowledge base and the target ontology is declared, but this would need to be administrated manually and outside of the wiki.

4 Usage of Ontological Data

In this section, we discuss concrete ways of using the knowledge of an external ontology and the associated challenges.

4.1 Inferencing Tasks

The principal purpose of specifying complex external schema information is to restrict the possible (formal) interpretations of the wiki's contents. Two major practical uses of this information are to impose constraints on the data within the wiki, and to infer additional information about this data.

Constraints are imposed by stating that certain situations are logically inconsistent. For a simple example, one could state that the categories (classes) *Workshop* and *Person* are disjoint. When a user tries to classify an article to belong to both classes, the system detects the inconsistency and can react appropriately. Typically, the reaction should involve a warning, and, if possible, an explanation. In general, inconsistency in OWL DL can arise for very complex reasons, and huge parts of the ontology can be involved in the deduction. The possible solutions to the resulting computational challenges are discussed in the next section. For now, let us note that the high complexity of the required reasoning also means that humans are typically not able to detect all consistencies easily for a complex schema. Since inconsistent ontologies do not represent any knowledge under the OWL semantics, it is clear that consistency checking is required for all applications discussed in Sect. 2.

Another application for ontological inferencing is automatic classification of instances, i.e. the assignment of OWL classes (wiki categories) to instances in the wiki. Automatic classification helps to structure the contents of the wiki, and therefore facilitates the creation and management of the wiki's content. In the case of MediaWiki, categories can be used to browse the wiki pages, and help to organise the wiki in a hierarchical way.

Besides this, classification yields a mechanism for users to understand the consequences of the external schema information. For example, the schema of a company's human resource ontology might define a property *supervised by* to describe a relationship between employees and interns. If a user now erroneously states that a certain project is supervised by somebody, then the automatic classification will classify the project as a *person*. This supports the user in immediately detecting the misconception even before further statements generate an actual inconsistency.

In both cases more expressive ontologies than those that are expressible with the current means of the Semantic MediaWiki are required. Therefor the system needs to refer to an external OWL ontology, that holds these more expressive ontologies. In Sect. 5.3 we describe an example where such an architecture is employed.

4.2 Practical Scalability of Inferencing

Useful as they might be, complex OWL DL inferences also impose huge scalability challenges. Reasoning in OWL DL is NExpTime complete, and thus appears to be unsuitable for a continuously growing knowledge base like a wiki. Yet, there are various possibilities to build practical systems that can still solve non-trivial reasoning tasks. We give a brief overview in the following.

First of all, it must be noted that current OWL DL reasoners are highly optimized systems that can already deal with complex ontologies to a certain complexity and size. So for small wikis, as they arise in many of the use cases from Sect. 2, it is quite feasible to employ standard software systems. Also, many of the system, while (N)ExpTime in the worst-case, have good "pay as you go" characteristics, so that simple ontologies require significantly less resources.

While restricting to simple ontologies can lead to improved computation time in OWL DL reasoners, it will not suffice for larger wikis since it is still not tractable. For such a system, the only feasible solution currently seems to be to restrict the expressive power of the employed ontology language. Various choices exist[4], and OWL fragments such as *Horn-*$\mathcal{SHIQ}$ [13] and *EL*++ [3] are quite expressive while still providing polynomial complexity for important reasoning tasks.

Another approach that is very interesting for reasoning in a wiki environment has been developed for the *KAON2* system [17]. Reasoning there is divided into two separate processing steps: the complex terminological part of an ontology is preprocessed in a first stage, whereas query answering over large ABoxes is performed in a second stage. The preprocessing is still ExpTime complete, but query answering can be performed in NP (wrt. the size of the ABox). If the ontology is restricted to Horn-$\mathcal{SHIQ}$, query answering is even possible in polynomial time. Evaluations show that KAON2 performs very well for ontologies with a fixes schema and large amounts of instance data [18] – which is exactly the scenario we address in this work.

[4] See `http://owl-workshop.man.ac.uk/Tractable.html` for an overview.

4.3 User Interface Enhancements

Finally, some uses of ontological background knowledge are possible without implementing complex reasoning. In particular, one can reuse the predefined vocabulary and known simple schema information to make suggestions during editing. In its simplest form, the wiki could warn users when using annotations that do not belong to the vocabulary that was imported into the wiki. This is not feasible in semantic wikis that are not based on an existing schema, since users there must be able to add new annotations to the system.

An example for more elaborate user interface enhancements are mechanisms for suggesting appropriate annotations to users. This can be realized syntactically by comparing user inputs to existing labels (e.g. in the form of autocompletion). On the other hand, semantic structure such as the specification of property domains can be exploited as well. For example, when editing a page on *SemWiki2006*, the classification of this article as a *workshop* might be evaluated by suggesting the properties *organized by* and *paper deadline* to the user.

5 Implementation

5.1 Ontology import

Our implementation of the ontology import is based on the Semantic Mediawiki extension to the Mediawiki[5] software. For importing the ontology we used the RDF API for PHP package RAP[6]. RAP is a software package for parsing, searching, manipulating, serializing, and serving RDF models.

The ontology import extension loads an ontology using RAP. Each statement in the ontology is checked for two criteria. First, it is checked whether the statement can be represented within the wiki. As we have seen in Sect. 3.1, we primarily regard the assertional information with the ontology, whereas complex class description will not be imported to the wiki. Second, it is checked if the statement already known to the wiki. If so, importing it is obviously unnecessary. The wiki then presents a list of all statements that passed both checks to the user, and allows her to choose which statements to import. For statements with subjects yet unknown to the wiki, the import will create articles, categories, relations, and attributes as appropriate. The text created to capture relations between articles is still very crude and recognizable as being machine created. In the future we imagine sophisticated text generation techniques [6] to augment this functionality. The name of a page is derived from the entity's label, or, if none exists, from the local part of the qualified name [7] of the entity. The comment is used as an initial text for the page, besides the generated text mentioned before.

Although the extension was first designed solely to allow the initial set-up of a wiki with an ontology, it also allows to upload ontologies at a later stage. This would allow to continuously add data from an ontology to a wiki, especially since the wiki is able to detect which parts of the ontology are missing.

[5] http://mediawiki.org
[6] http://www.wiwiss.fu-berlin.de/suhl/bizer/rdfapi/

In order to keep the ontology import simple, we refrained from using inferencing and mapping techniques on the ontology to be imported. We only take into account the basic relations described in table 1, and further only those explicitly stated in the ontology. As the wiki can be linked to any external ontology, inferencing can be part of the later stage as described in the next section and the example below. It is also possible to materialize derived statements within the ontology, or to create materialized mappings, prior to the import, if the wiki should represent this knowledge explicitly.

The implementation of the ontology import is finished and is part of Semantic Mediawiki in the current version.[7] As an ontology import potentially leads to a big number of changes, access to this feature is controlled via Mediawiki's user rights management system.

5.2 KAON2 integration

In order to reason with the ontology, we choose the KAON2 ontology management infrastructure [12], which is written in Java. In order to employ it in the PHP-based MediaWiki, we hooked KAON2 to a simple, Java-based server – Jetty[8] – and defined a simple ad hoc protocol to expose the required features and test the functionality. We will consider DIG [4] as a possible protocol, but DIG does not yet allow for SWRL-based extensions (which are possible with the current architecture), and it is also unclear how well DIG would scale in our example. Semantic MediaWiki uses the *curl* PHP extension in order to communicate via HTTP to the KAON2 server. The results are then integrated into the output of the page.

The separation of the reasoner and the wiki engine on possibly different machines, and the loose coupling via HTTP offer several advantages. First, we do not need to reimplement an OWL reasoner in PHP, which would be a highly non-trivial task, but instead we can rely on a mature inferencing engine that is based on sound and complete algorithms. Second, with a well-defined protocol it would be possible to plug in different reasoners if required. Third, a reasoner may require significant resources (both in terms of processing and memory) which could slow down the wiki considerably. By distributing the tasks to different machines we allow the wiki to remain responsive regardless of the computations required for reasoning. We consider asynchronous calls to a reasoning service as a possibility to combine possibly expensive tasks with a responsive user interface.

Right now, there is a first proof-of-concept implementation of the KAON2 integration, but a more mature prototype is planned for the near future.

5.3 Example

In order to exemplify the workflow in our implementation, we consider a possible semantic wiki about the *SemWiki2006* workshop. Instead of creating a new ontology for the workshop, the wiki administrator decides to reuse the ontology for the semantic web research community, SWRC [26]. The SWRC contains numerous terms that may serve

[7] `http://sf.net/projects/SeMediaWiki`
[8] `http://jetty.mortbay.org`

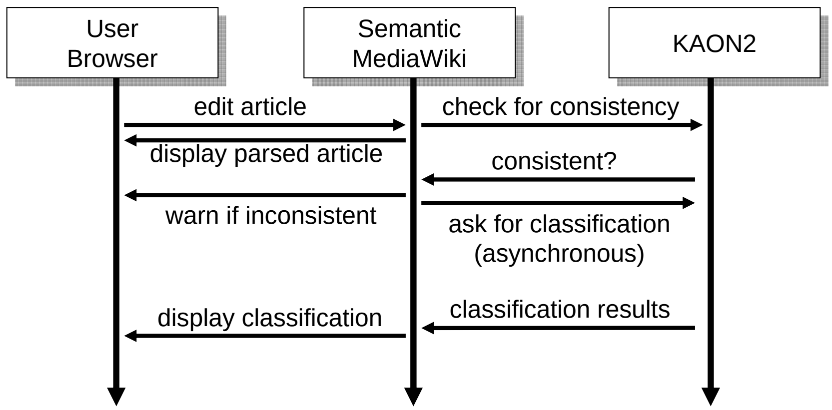

Fig. 1. Sequence diagram of how the SMW extension cooperates with the reasoner

as an initial base for the use case, including concepts such as *Workshop*, *Article*, and *Person*, and properties such as *organiser or chair of*, *member of PC*, and *at event*.[9]

The wiki administrator imports the SWRC ontology to have the initial relations and categories set up appropriately. Now she can start to populate the wiki manually, e.g. to add pages for the members of the programme committee. Assume that the accepted papers are already available in a machine readable format through the conference management system. In this case, it would be possible to export this data to an RDF document that reuses the SWRC ontology. Depending on the available format of the data this might require some simple conversion or mapping. However, if a suitable conversion has been implemented once, it can easily be reused for future events. The RDF version of the accepted papers can now be imported easily into the wiki to kick-start the wiki with existing ontological data. We consider it to be far easier to augment existing pages than to start from an empty wiki.

Now the wiki is set up and the wiki administrator sends its URL to interested parties. Workshop participants are encouraged to enhance the text in the wiki (as the automatically generated one does sound quite awkward as of now), or to add further details and references. One of the presenters may decide to add a link to his supervisor, typing it appropriately with a relation *supervised by* which, for the sake of the example, we assume to be part of SWRC. After this, he creates a new article about his supervisor with a few sentences and a link to the homepage – but without stating that the supervisor is indeed a *Person*.

But when the edit is saved, the automatic classification mechanism automatically adds the supervisor to the category *Person*, since this can be inferred from the ontological definition of the range of the property *supervised by*. After saving, the article on the author's supervisor would indeed state that it is a person, so that other users can find the article when browsing the according category page.

[9] For readability, we omit namespaces and camel case capitalization.

More complicated description could be added easily to the ontology. Imagine the wiki administrator adding the class *Good Paper* as one that is *endorsed by* many, or one that is *cited by* many other papers. The system then could automatically annotate “good papers” based on the semantic description of the article about the respective paper.

Also, inconsistencies in the wiki knowledge base could be detected by the reasoner: if, for example, we know that only a *Professor* may be a supervisor, categorizing a supervising person as a *PhD-Student* can be recognized as being inconsistent with the given knowledge base, and the user will be warned about this. Typically, this supports users in recognizing misconceptions about the intended usage of some relation. In the given example, another relation *tutored by* might be appropriate. Based on a built-in suggestion mechanism, the system can assist the user to find such relations without studying the background ontology.

6 Related work

Since the introduction of *Platypus Wiki* [9] in 2004, a number of semantic wikis have been developed. The focus of early systems such as Platypus or the *Rhizome Wiki* [24] was to provide capabilities for editing RDF content within a wiki environment. Due to the typical splitting of wiki source and RDF, importing data into these ontologies would be possible, whereas advanced features such as consistency checking and classification are mostly out of scope. The main reason is that RDF is treated as a freely edited data format without a semantic relationship to the wiki's content.

More recently, a number of new semantic wikis have been introduced. Since many of these systems are under active development, their exact capabilities and features are still evolving while this paper is written. Some of these systems, such as *IkeWiki* [22], adhere to the strict separation of semantic content and wiki text. In contrast, some wikis integrate semantic annotations into the wiki source code, as it is done in *WikSAR* [1, 2], and *Semantic MediaWiki* [27]. Finally, some wiki systems feature a WYSIWYG interface, that allows users to edit content without editing wiki markup. The only semantic wiki of this type that we are aware of is *SweetWiki*[10] which is still in prototype stage.

Only very few semantic wikis provide any support for inferencing or ontology import. The most advanced system in this respect currently seems to be IkeWiki, which allows users to import data from external ontologies and exploits schema data to provide editing support. The employed ontology language by default is (a subset of) OWL, and the system uses a reasoning engine in the back. To the best of the authors' knowledge, IkeWiki does not employ a complete OWL reasoner, but it provides partial reasoning support to structure wiki content and to browse data.

IkeWiki differs from our system in various ways. First of all, IkeWiki emphasizes the use of external ontologies much more than Semantic MediaWiki. The wiki can be initialized with multiple ontologies, and users choose annotation elements from the according namespaces. In contrast, Semantic MediaWiki uses external ontologies only for RDF export, and users work with internal identifiers. These identifiers might be equal to the abbreviated URIs in an external ontology, but it is also possible to choose

[10] `http://wiki.ontoworld.org/wiki/SweetWiki`

more human-readable names, e.g. on a wiki that is run in German instead of English.[11] On the other hand, IkeWiki provides user-friendly special purpose interfaces for editing annotations, the implementation of which is facilitated by the wiki's separation of RDF and text.

IkeWiki's stronger reference to existing ontologies implies further conceptual differences. For instance, ontological concepts in IkeWiki must be declared or imported before usage. In Semantic MediaWiki, many annotations can be used without prior declaration – necessary URIs are generated from the URL of the wiki system. Declaring references to external ontologies is an added feature that is not enabled by default, since we consider it as problematic in public wikis. In particular, free access to elements from the RDFS and OWL vocabulary enables users to generate OWL Full ontologies and logical inconsistencies, which is an interesting combination since one cannot generally detect inconsistencies in OWL Full automatically. On the other hand, an even tighter integration of *selected* ontologies can be an interesting feature that is planned for Semantic MediaWiki as well.

Other than IkeWiki, we are only aware of *KaukoluWiki*[12] as another wiki system that features ontology import and inferencing. There, the primary ontology language is RDFS. Various related features are planned or currently implemented, but we are not certain about the current status and integration of reasoning support.

Finally, Semantic MediaWiki appears to be the only wiki with extended support for XML Schema (XSD) datatype annotations. The current implementation allows users to provide data values in various syntactic forms, transforms values into XSD conformant representations, and incorporates units of measurement into the RDF export.

7 Conclusions

We have developed and implemented a semantic wiki that meets several requirements:

- it refers to existing ontological vocabularies,
- it incorporates schema information and constraints from external ontologies,
- it exports data in a standard ontology language,
- it imports data from external ontologies, so that it is represented in and editable through the wiki.

In order to meet these requirements, we enabled the wiki to test its knowledge base for inconsistent facts and to classify articles automatically. We also showed how the ontological knowledge could be used for enhancing the user interface. In order to design the system, we have taken results from research in the field of scalable reasoning over web ontologies into account and geared the system towards a fast and interactive user experience.

Various tasks still are left open for future work. Better natural language generation techniques [6] would considerably improve the ontology import function. The inconsistency check right now only tells us that the ontology is inconsistent, but not in what

[11] Semantic MediaWiki supports easy internationalization and is available in various languages.
[12] `http://kaukoluwiki.opendfki.de`

way and how to resolve the inconsistency (or even offer automatic correction capabilities) [11, 21, 16]. It would also be interesting to synchronize a wiki knowledge base with a dynamic ontology outside the wiki, i.e. a continuous import from an ontology. Finally, although we have presented some ideas on how the ontological background knowledge can be reused to enhance the user experience, we could not yet test or properly evaluate how users interact with such a system. Although we are positive that it will be of great help to the user, there are still some pitfalls: frequent inconsistencies or inexplicable automatic classification could lead to frustration about the "stubborn" system.

The paper has also shown that we can use ontologies for both exporting and importing knowledge from and to a wiki. Although the exchange format is far from being as complete or even useful as other wiki exchange syntaxes, it has the advantage of being based solely on the W3C standards RDF for data exchange and OWL for the vocabulary, and thus may not only interact with other wikis, but also with the ever growing set of ontology-based tools. The inconsistency check and automatic classification presented in this paper are a mere example of this.

Whereas previous work on Semantic Mediawiki [27] has presented a way to turn wikis, and especially the Wikipedia, into a major foundation of the Semantic Web [5], in this paper we propose the inverse approach: reusing Semantic Web resources for a wiki. These two technologies together allow to integrate wikis fully into the Semantic Web idea.

Acknowledgments

Research reported in this paper has been partially financed by the European Union in the IST project SEKT (IST-2003-506826), `http://www.sekt-project.com`, and by the German BMBF project "SmartWeb." The views expressed in this paper are those of the authors and not necessarily those of the projects or their institute as a whole. We would like to thank our colleagues for fruitful discussions.

References

1. D. Aumüller. Semantic authoring and retrieval in a wiki (WikSAR). In *Demo Session at the ESWC 2005*, May 2005.
2. D. Aumüller and S. Auer. Towards a semantic wiki experience – desktop integration and interactivity in WikSAR. In *Proc. of 1st WS on Semantic Desktop, Galway, Ireland*, 2005.
3. F. Baader, S. Brandt, and C. Lutz. Pushing the EL envelope. In *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 364–369, 2005.
4. S. Bechhofer, I. Horrocks, P. F. Patel-Schneider, and S. Tessaris. A proposal for a description logic interface. In *Proc. of the 1999 Description Logic Workshop (DL'99)*, pages 33–36, 1999. CEUR Workshop Proceedings `http://ceur-ws.org/Vol-22/`.
5. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001.
6. K. Bontcheva. Generating tailored textual summaries from ontologies. In A. Gómez-Pérez and J. Euzenat, editors, *2nd European Semantic Web Conference, ESWC 2005*, volume 3532 of *LNCS*, pages 241–256, Heraklion, Crete, Greece, May 2005. Springer.

7. T. Bray, D. Hollander, and A. Layman. Namespaces in XML. W3c recommendation, World Wide Web Consortium, January 1999.
8. D. Brickley and L. Miller. FOAF vocabulary specification. Namespace Document 27 July 2005 ('Pages about Things' Edition).
9. 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.
10. W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, 2001.
11. P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proceedings of the Fourth International Semantic Web Conference (ISWC2005)*, volume 3729 of *LNCS*, pages 353–367. Springer, November 2005.
12. U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ description logic to disjunctive datalog programs. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proceedings of the KR2004*, pages 152–162. AAAI Press, 2004.
13. U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. 19th IJCAI*, pages 466–471, 2005.
14. A. Kalyanpur, B. Parsia, and J. Hendler. A tool for working with web ontologies. *International Journal on Semantic Web and Information Systems*, 1(1), 2004.
15. M. Krötzsch, D. Vrandečić, and M. Völkel. Wikipedia and the Semantic Web – the missing links. In *Proc. of the 1st Int. Wikimedia Conf., Wikimania*, Aug 2005.
16. S. J. Lam, D. Sleeman, and W. Vasconcelos. Retax++: a tool for browsing and revising ontologies. In *Proc. of ISWC 2005 Demo Session*, Galway, Ireland, November 2005.
17. B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Universität Karlsruhe (TH), Germany, 2005.
18. B. Motik and U. Sattler. Practical DL reasoning over large ABoxes with KAON2, 2006. available at `http://kaon2.semanticweb.org/`.
19. N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.
20. E. Oren. Semperwiki: a semantic personal wiki. In *Proc. of 1st WS on The Semantic Desktop, Galway, Ireland*, 2005.
21. B. Parsia, E. Sirin, and A. Kalyanpur. Debuging OWL ontologies. In *Proc. of the 14th World Wide Web Conference (WWW2005)*, Chiba, Japan, May 2005.
22. S. Schaffert, A. Gruber, and R. Westenthaler. A semantic wiki for collaborative knowledge formation. In *Semantics 2005*. Trauner Verlag, 2005.
23. M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Recommendation.
24. A. Souzis. Building a semantic wiki. *IEEE Intelligent Systems*, 20:87–91, 2005.
25. Y. Sure, J. Angele, and S. Staab. Ontoedit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics*, 1(1):128–152, November 2003. LNCS 2800.
26. Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The SWRC ontology - Semantic Web for Research Communities. In C. Bento, A. Cardoso, and G. Dias, editors, *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, volume 3803 of *LNCS*, pages 218 – 231, Covilha, Portugal, Dec 2005. Springer.
27. M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. of the 15th int. conf. WWW 2006, Edinburgh, Scotland, May 23-26, 2006*, May 2006.
28. D. Vrandečić, H. S. Pinto, Y. Sure, and C. Tempich. The DILIGENT knowledge processes. *Journal of Knowledge Management*, 9(5):85–96, Oct 2005.
29. Wikipedia. History of Wikipedia – Wikipedia, The Free Encyclopedia, 2006. Online version of 28 March 2006.