Reasoning in Semantic Wikis

Markus Krötzsch[1], Sebastian Schaffert[2], and Denny Vrandečić[1]

[1] AIFB, Universität Karlsruhe (TH), Karlsruhe, Germany
{kroetzsch, vrandecic}@aifb.uni-karlsruhe.de
[2] Salzburg Research Forschungsgesellschaft, Salzburg, Austria
sebastian.schaffert@salzburgresearch.at

Abstract. Semantic wikis combine the collaborative environment of a classical wiki with features of semantic technologies. Semantic data is used to structure information in the wiki, to improve information access by intelligent search and navigation, and to enable knowledge exchange across applications. Though semantic wikis hardly support complex semantic knowledge and inferencing, we argue that this is not due to a lack of practical use cases. We discuss various tasks for which advanced reasoning is desirable, and identify open challenges for the development of inferencing tools and formalisms. Our goal is to outline concrete options for overcoming current problems, since we believe that many problems in semantic wikis are prototypical for other Semantic Web applications as well. Throughout the paper, we refer to our semantic wiki implementations *IkeWiki* and *Semantic MediaWiki* for practical illustration.

1 Introduction

Semantic wikis enrich wiki systems for collaborative content management with semantic technologies. Annotations added to wiki pages are stored in a knowledge base, and possibly connected with background ontologies. Based on these annotations, semantic wikis provide enhanced navigation, search, and retrieval, and often also contextual adaptation of the presentation of the content. Collaborative authored semantic content is exported to the Semantic Web in standard formats, such as RDF [1] and OWL [2].

Semantic wikis have been successfully applied in real-world scenarios, and various implementations exist – see [3] for an overview of current research. Prominent systems like Semantic MediaWiki [4,5], IkeWiki [6,7], and SemperWiki [8] managed to disseminate semantic technologies among a broad audience, and many of the emerging semantic wikis resemble "small Semantic Webs." Indeed, wikis are characterised by their dynamic, open nature with many different contributors who independently create different "pieces of knowledge" that need to be integrated.

In spite of their success, semantic wikis often use very simple semantic structures and hardly employ complex inferencing procedures. We argue that this is not at all due to a lack of practical applications of advanced reasoning: though a little semantics might go a long way, we believe that many worthwhile goals won't be reached without additional expressivity. There are, of course, complications when moving from shallow semantic data to more complex ontologies, and we will discuss requirements, existing challenges, and open problems. Motivated by concrete use cases for inferencing within semantic wikis, we outline viable ways for improving reasoning support in wikis. Based on our experience with developing IkeWiki and Semantic MediaWiki, we provide many examples for simple inferencing within these systems, and give an outlook on upcoming enhancements related to reasoning.

The goal of this article is on the one hand to introduce into semantic wikis as an interesting testbed for evaluating Semantic Web technologies, including reasoning, and on the other hand to illustrate in practical application scenarios – based on semantic wikis – where reasoning can lead to interesting applications and where reasoning still faces challenges that need to be addressed.

After giving a short outline of semantic wikis in Section 2, we briefly describe our two semantic wiki systems, Semantic MediaWiki and IkeWiki, with a particular focus on the way knowledge is represented. We then show general uses of reasoning for enhancing *browsing*, *querying*, *editing*, and *validating* in Section 4. In Section 5, we outline three different concrete application scenarios for reasoning in semantic wikis and from that derive selected challenges that we consider as relevant for future reasoning systems. In Section 7, we conclude by sketching next steps towards improving reasoning support in our wiki systems.

2 Semantic Wikis

The term "semantic wiki" encompasses a broad range of applications that are using machine-readable data with a well-defined semantics to augment the functionality of a wiki-based content management system. To clarify this rather unsharp definition, we now present some ways of classifying semantic wikis and present typical implementations and usage scenarios. Possibly the most important characteristic of each semantic wiki is its approach towards collecting semantic data. Classical wikis are primarily designed for collecting textual information, and the collaborative creation of this content is supported in many ways, e.g. through simple user interfaces, extensive versioning support, or discussion pages. Semantic wikis need to find ways of also obtaining machine-readable data, without sacrificing the core strengths of classical wikis. Three main sources for semantic data are usually considered:

Manually provided content. Virtually all current semantic wikis enable users to directly enter and modify semantic data via the wiki's editing interface or an extension thereof. Implementations range from providing text-forms for RDF data, over the use of a simplified wiki-markup within existing texts, to the use of additional interactive user interfaces that support annotation. Basically all classical wikis provide simple annotation support in the form of tagging or categorisation. Semantic wikis extend these capabilities.

Automatically collected metadata. Many wikis already collect large amounts of metadata that is required for normal operation. This includes details about authors, version histories, licensing, and hyperlinks. This data can be converted into machine-readable formats that enable interchange and reuse, as was done, e.g., with Wikipedia's metadata in the *Wikipedia*3 project[3]. Another source of metadata is information associated with multimedia files, used e.g. in IkeWiki to gather information on uploaded images.

External background knowledge. Semantic wikis are not always the only source for semantic data, and can be used in conjunction with other semantic data sets. In this case, the wiki should be able to import and reuse ontologies or RDF datasets, in order to take this background-knowledge into account for editing and presentation. A prototypical implementation of such functionality was described in [9].

Semantic wikis often use more than one of those sources, but some way of manually providing semantic content appears to be a core feature of all current implementations. Independent of its source, the use of the available semantic data can be very different. Though the boundaries are not strict, the following two approaches are prominent:

"Wikis for semantic data." Wikis that sufficiently support the editing of ontological information can be used as collaborative ontology editors. The system in this case can help domain experts and ontologists to cooperate in one system, while the text-content of each wiki page is used to create a human-readable specification in parallel to the formal ontology. Even if the wiki does not provide means of editing complex schema information, it can still be used to develop and document ontological vocabularies in a similar way. The use of wikis for such tasks facilitates the dynamic, evolutionary development of ontologies, and supports the gradual lifting of informal textual descriptions to formal conceptualisations. In the project *Dynamont*[4], IkeWiki is used for this purpose.

"Semantic data for wikis." Wikis are already used successfully in many applications where the primary task is to collect textual content. In those cases, semantic data is used to support the current usage of the wiki, e.g. by simplifying the retrieval of information through semantic search functions. Semantic data in this case either is used to make some of the wiki's contents machine-processable, or to simplify wiki maintenance by exploiting metadata. A major goal is to retain the known working principles of the wiki, and thus to not introduce too many new interfaces or functions. Examples of this use are given by many sites running Semantic MediaWiki, such as ontoworld.org [4]. Another typical scenario for wiki usage is internal knowledge management in working groups.

3 Knowledge Representation in Wikis

In this section, we briefly introduce two popular semantic wiki systems, *IkeWiki* and *Semantic MediaWiki*, which will also be our main objects of examples in later sections. Here, we are mainly interrested in the forms of semantic knowledge that each of those applications supports, and on the approaches for representing this data within the wiki. Advanced uses of this data will be discussed in later sections.

[3] http://labs.systemone.at/wikipedia3
[4] http://dynamont.factlink.net

3.1 Semantic MediaWiki

Semantic MediaWiki (SMW) [4] is an extension of the popular wiki software *MediaWiki*[5] that is used by Wikipedia and many other sites. SMW is typically used as a modular enhancement to existing MediaWiki installations, and thus aims at a seamless integration of semantic features into the existing user interfaces. The internal knowledge model of SMW is closely related to OWL DL, although just a small fragment of the expressive means of this language is actually available within the wiki.

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

Most of the annotations that occur in SMW correspond to simple *ABox statements* in OWL DL, i.e. they describe certain individuals by asserting relations between them, annotating them with data values, or classifying them. The schematic information (*TBox*) representable in SMW is intentionally shallow. The wiki is not intended as a general purpose ontology editor, since distributed ontology engineering and large-scale reasoning still faces various open challenges, some of which we will consider below.

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

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

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

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

It is obvious that the processing of Attributes requires some further information about the *Type* of the annotations. Integer numbers, strings, and dates all require different handling, and one needs to state that an attribute has a certain type. As explained

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

above, every ontological element is represented as an article, and the same is true for categories, relations, and attributes. This also has the advantage that a *user documentation* can be written for each element of the vocabulary, which is crucial to enable consistent use of annotations.

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

When using SMW as a wiki, users rarely are confronted with complete URIs. Instead, SMW generates unique URIs from the titles of the respective articles whenever this is necessary, especially for exporting semantic data as OWL/RDF. URIs of entities within the wiki thus are typically "local" to the given wiki, and are generated dynamically as the wiki is extended. It is also possible for users to assign given existing URIs to concepts that occur in the wiki, e.g. to associate a wiki's category `Category:Person` with the URI of the concept `foaf:Person` provided by the FOAF vocabulary. Such reuse of existing URIs is constraint to vocabularies that have been explicitly *imported* by the wiki's administrators, such that no abuse of existing URIs is possible (e.g. `foaf:Person` should never be used as the URI of a binary property).

Besides such explicit annotations, SMW does not consider any other sources of semantic data. Prototypes for using expressive ontologies within SMW have been studied [9], but the usual approach is to export semantic content from the wiki, and to further process it within some other semantic system that can also incorporate additional ontologies.

3.2 IkeWiki

IkeWiki is a complete reimplementation of a semantic wiki as a Java web application. The name *IkeWiki* is derived from the Hawaiian words *ike* – meaning "knowledge" – and *wiki* – meaning "quick". Although now also considered in different settings, IkeWiki has originally been developed as a prototype tool to support knowledge workers in collaboratively formalising knowledge. IkeWiki's design principles are an easy to use, interactive interface, compatibility with Semantic Web standards, immediate exploitation of semantic annotations, support for different levels of formal expressiveness (from RDF to OWL), and reasoning support.

IkeWiki uses the Java-based Semantic Web framework *Jena* as a backend and therefore offers support not only for plain RDF but also for the representation of different kinds of ontologies (RDFS and OWL). Different reasoning mechanisms can be employed; the default installation uses OWL-RDFS reasoning (essentially subclass and type inference), but we have also successfully integrated the Pellet reasoner with full OWL-DL support. We are also experimenting with a rule-based reasoning engine that would allow users to add user-defined rules to the knowledge base. One of the goals of

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

IkeWiki is to support the formalisation of knowledge all the way from informal texts to formal ontologies.

In contrast to Semantic MediaWiki, IkeWiki relies on background ontologies that are pre-loaded in the knowledge base. As a consequence, it is not possible to annotate a link with a predicate that is not defined as an OWL ObjectProperty in the knowledge base. Likewise, IkeWiki only offers AnnotationProperties and DatatypeProperties as textual metadata fields. This can be seen as a "restrictive" approach as compared to Semantic MediaWikis "open" approach.

Every page in IkeWiki corresponds to a resource in the knowledge base and is associated with a type (concept). A new page in general has the page `rdfs:Resource`, which can be further refined by the user. IkeWiki's reasoning component automatically infers all types of a page that follow by subclass relationships or relations to other pages. As this can be somewhat confusing to users, IkeWiki by default only offers link annotations that are compatible with the currently associated page types.

Unlike Semantic MediaWiki, IkeWiki stores all semantic metadata separately from the page content. When rendering a page, a so called *rendering pipeline* assembles content from the article database and metadata from the knowledge base to create an "enriched article" that contains the relevant semantic annotations. This approach has the advantage to make the maintenance of the knowledge base easier, but on the other hand does not allow versioning of the metadata and the locality of metadata is abandoned.

The IkeWiki interface makes heavy use of AJAX[7] technologies to provide an interactive interface. Page content is entered using a WYSIWYG editor. Semantic annotations are created in a separate "annotations editor"; the rationale behind this decision is that IkeWiki was designed as a collaborative tool that allows users with different levels of experience to work on the data in a defined workflow, and that editing content and metadata are separate processes in such settings.

The annotations editor allows to add types to a page and the links it contains via a guided, dialogue-based interface (see Figure 2 on page 11). When a user clicks on the +-symbol behind a link or behind the page types, he is presented with a list of all applicable properties or classes from which he can choose what he thinks appropriate. Reasoning is used to restrict this list to only the relevant ontology concepts. IkeWikis interface therefore prohibits to add link or page types that are not applicable, but supports the user by showing those types that *are* applicable. This conflicts to some extent with the "wiki philosophy" but results in a more homogeneous knowledge base with no inconsistencies with respect to the ontologies.

In addition to normal page annotations, IkeWiki's annotation editor also supports to modify or extend the loaded ontologies themselves. Specific support is given for adding sub-/superclasses, range/domain of properties, inverse relations, etc. There is currently no support for advanced OWL concepts like cardinality constraints and other restrictions.

---

[7] AJAX = *Asynchronous JavaScript and XML*, a technology that allows Web-applications to look and feel like desktop applications

3.3 Maintaining Ontologies in Wikis

Since advanced reasoning features are typically based on information that is more complex than simple RDF data, semantic wikis need to provide means for editing ontological axioms or logical rules as well. This is not a trivial task, since wikis and ontological knowledge bases are structured in different ways. In this section, we discuss how ontological information can still be maintained in wikis, and which kinds of such information are most suggestive for seamless integration in current usage.

Semantic assertions in wikis typically refer to the contents of the wiki, which usually are represented by wiki pages or parts thereof. Simple assertions can thus be made at the place in the wiki to which they refer: assertions truly are annotations to the text of the wiki. If more complex axioms are taken into account, it becomes apparent that the structural models of ontologies and wikis do not match so nicely any more:

- Wikis are based on the notion of *articles* which might be further separated into sections.
- Ontologies are based on the notion of *axioms* or *rules*.

So how should ontological axioms be distributed among articles? Wiki articles often play the role of the underlying *vocabulary* used in ontological axioms, but each axiom might refer to many different articles. Different approaches towards solving this problem exist:

- **Decoupling pages and axioms.** Instead of treating axioms as annotations of wiki pages, it is also possible to consider them as independent content in their own right. Some semantic wikis therefore provide two input forms for each page: one for page text and one for axioms. While this allows users to freely organise axioms within the wiki, it has the disadvantage that there is no direct way of finding the page within which a particular axiom was given.
- **Strict coupling of pages and axioms.** If the possible kinds of ontological axioms are known, and if their structure is sufficiently simple, then it makes sense to couple them with one of the affected pages. For example, MediaWiki allows users to state subsumption relationships between categories.[8] Although two categories are involved in each subsumption, the axiom must be stated on the page of the subsumed category. Similar solutions can be found for many typical types of axioms. For complex axioms that could be expressed in syntactically different ways, however, the assignment of pages could still be ambiguous.
- **Loose coupling of pages and axioms.** Another solution is to couple a particular axiom to all pages that it affects. Thus, one axiom might be displayed on many pages, while being separated from the wiki text, and each page displays all axioms that are directly relevant for it. This solution is attractive for keeping an overview of all axioms, but it significantly affects the architecture of the wiki. Axioms do no longer belong to articles, and editing, versioning, and discussion of particular axioms needs additional mechanisms. For simplicity, axioms should be editable by editing the article in which they appear – but then concurrently editing two different articles might yield editing conflicts.

[8] Which, in some of their uses, can be viewed as ontological classes.

Each of the above solutions has its advantages and disadvantages, and a major future design task for semantic wikis will be to develop combined solutions that further push the expressivity of semantic wikis without sacrificing simplicity and usability.

4 Using Reasoning in Wikis

Reasoning is the process of deriving conclusions from formal symbolic knowledge, as constituted by the semantic content of a semantic wiki. In the case of a semantic wiki, the premises are the underlying knowledge base together with the actual page content. Reasoning can be applied in semantic wikis for various different effects and benefits. The "conclusions" of reasoning can be of many different kinds, e.g. deriving of additional relations, but also the adaptation of the presentation based on information in the knowledge base. In the following, we describe four areas where reasoning could provide actual benefit to wiki users. Note that all four areas have been investigated separately in other contexts; however, semantic wikis provide a useful testbed to show the possible benefits of all areas in a single application.

4.1 Browsing and Displaying the Wiki Contents

Knowledge that is made explicit within the wiki enables the system to provide additional ways to browse, explore, and display the wiki's content. This is relevant not only as a support for finding information within the wiki, but also as a basic way for inspecting its semantic contents. In contrast to advanced search functions that are discussed in the following section, browsing generally relies on information that is *local* to the data that is currently displayed.

Displaying semantic data. A typical way of browsing data is to simply display, for every page in the wiki, the pages and concepts that it directly relates to. Often, this is done in textual form to inform users about the semantic contents of a page without requiring them to view the page source. For instance, Semantic MediaWiki inserts a *Factbox* below every article, showing related articles and attribute values, and displaying links to those elements, as well as to specific search functions and the article's RDF export. It also provides a browsing feature that displays incoming and outgoing relationships as well as filtering by relations and attributes.

Domain specific visualisations. Besides generic browsing functions, there are cases in which semantic information can be used to generate customised displays for specific types of data. For example, IkeWiki displays symbols for indicating licensing information, and provides special display for EXIF metadata that is extracted from image files. Semantic MediaWiki supports conversions between units of measurement which in turn are specified by semantic annotations within the wiki. In each of those examples, it is necessary that the wiki is aware of certain domain specific annotations that are treated in a special way. Existing vocabularies are suggestive as a basis for such special features, and it might be necessary to incorporate further ontological background knowledge to properly support those knowledge models within the wiki.

Graphical visualisation. Besides textual displays of a resource's semantic context, graphical visualisations can be helpful for exploring a wiki's content. IkeWiki, for example, can render a graph that shows all resources within a certain relational distance to the current one. The main advantage of graphical displays is their ability to convey more information directly without lengthy textual descriptions. Colours, shapes, and spacial grouping can carry relevant information. The main problem, on the other hand, is that automatically generated layouts of graphs are easily confusing if many resources are involved: visualisations often do not scale to larger amounts of data.

Filtering and selection. The above ways of displaying data are challenged when the amount of semantic content is large. For example, the article "Germany" in English Wikipedia has more than 30,000 incoming links, many of which could represent certain semantic relations. Filtering, grouping, and ordering of results is needed to usefully display such information. While ordering often requires extralogical information, filtering and grouping correspond to semantic retrieval and classification. Criteria for those operations might be provided by the given usage context, user preferences (i.e. semantic statements about the user), or by explicit queries (e.g. as in faceted browsers).

Some of the above visualisation methods could be realised with little or no reasoning effort. But as soon as semantic wikis incorporate semantic knowledge that is not restricted to the most simple facts, even the immediate relationships of some resource need to be computed first. The reason is that browsing should usually visualise the semantic model of the wiki, not just statements given explicitly by editors.

4.2 Querying the Knowledge

Besides the individual display of wiki pages, semantic wikis provide mechanisms to directly query their knowledge base. The query language used in IkeWiki is SPARQL. Semantic MediaWiki provides its own simple query language whose syntax is based on MediaWiki's mark-up language for editing. Semantically, this built-in language is easily seen to correspond to concepts of the description logic $\mathcal{EL}^{++}$ [10], with addition of (Horn) disjunction that does not add to the complexity [11].[9] Therefore, IkeWiki's native query language is more powerful but also computationally more complex than that of Semantic MediaWiki. Both wikis export their data in standard formats, so that existing tools can be used to query the data in further languages.

Besides the technical aspect of the employed query language, the use of queries in wikis strongly depends on the provided user interfaces. Both Semantic MediaWiki and IkeWiki enable editors to embed query results into wiki pages. The obvious advantage is that users that are not familiar with the query language can still view results of predefined queries. In Semantic MediaWiki, the query language is extended to allow advanced formatting of embedded results, e.g. for displaying times and dates within dynamic Timeline views (cf. Figure 1). In addition to embedded queries, it is also possible to directly evaluate queries and to browse the results.

[9] The query language extends $\mathcal{EL}^{++}$ by allowing concrete domains that are *non-convex*, but this does not increase expressivity in our case due to the absence of existential assertions on concrete roles.

Below is the complete ISWC workshop programme for 6th of November 2006. Please refer to the events' webpages for latest information on timing and registration.

	Title	deadline	start	end
SAAW2006	Semantic Authoring and Annotation Workshop	2006-08-10	2006-11-06 08:00:00	2006-11-06 12:00:00
Terra Cognita 2006	Terra Cognita 2006 - Directions to the Geospatial Semantic Web	2006-07-21	2006-11-06 08:00:00	2006-11-06 18:00:00
SWUI2006	The 3rd International Semantic Web User Interaction Workshop	2006-08-11	2006-11-06 08:30:00	2006-11-06 17:45:00
SEBIZ2006	First international workshop on Applications and Business Aspects of the Semantic Web	2006-08-05	2006-11-06 09:00:00	2006-11-06 12:30:00
SemanticDesktopWS2006	Semantic Desktop and Social Semantic Collaboration Workshop	2006-08-10	2006-11-06 09:00:00	2006-11-06 17:30:00
SSN2006	Semantic Sensor Networks Workshop	2006-07-10	2006-11-06 09:00:00	2006-11-06 12:30:00
SWESE2006	2nd International Workshop on Semantic Web Enabled Software Engineering	2006-08-10	2006-11-06 09:00:00	2006-11-06 18:00:00
WCMHLT2006	Web Content Mining with Human Language Technologies workshop 2006	2006-08-01	2006-11-06 13:00:00	2006-11-06 17:00:00

The same information in a timeline:

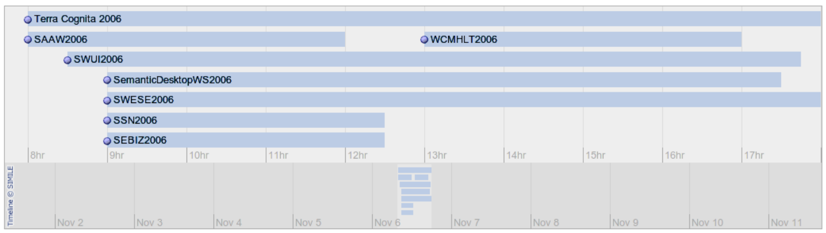


Fig. 1. Semantic MediaWiki can display the same query in different formats, such as the tabular view and timeline view shown above.

Strong reasoning allows the query engine to answer more complex queries. But when designing the wiki query interface it must not be forgotten that we are still working on a wiki: queries should be easy to write, save, and share (like inline queries in *Semantic MediaWiki*), and answers should be traceable to the content of the wiki, which might require advanced explanation features to assist users in finding the source of a particular result.

4.3 Editing Support

Reasoning can not only be used to enhance the content presentation, but also to support editing the (textual and semantic) content. Typically, semantic annotation can be supported by exploiting ontological information: if a resource was classified to belong to a certain class, properties whose domain is a subclass of this class might be suggested for annotation. This is currently done in IkeWiki to simplify annotation by providing a list of suitable properties, see Figure 2. An AJAX-based interface allows to dynamically load and specialise such suggestions. Suggesting adequate annotations based on statistical and linguistic analysis will be subject to future research. Upcoming versions of Semantic MediaWiki will feature special annotations that can be used for marking deprecated properties. Explicatory messages will be displayed on any page on which deprecated properties are used, thus supporting convergence towards a unified knowledge model.

Reasoning can also be used to *avoid* inconsistent content in advance, by suggesting appropriate values for statements that are constrained by the existing knowledge base. In IkeWiki, templates that contain SPARQL queries can be used to dynamically compute suggested values. For example, we successfully used this technique to create a conference wiki based on IkeWiki for the “Social Skills durch Social Software” conference in

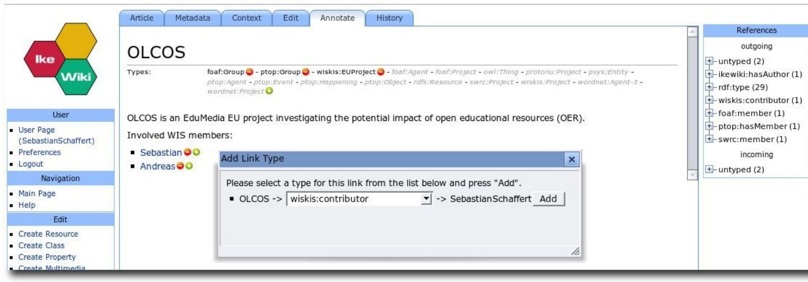


Fig. 2. Annotations in IkeWiki are suggested based on the types associated with the two pages and the domain/range definitions of relations (in this case `wiskis:contributor`).

Salzburg, 2006. The conference wiki contained pages for every talk with information about the speaker, the abstract of the talk, etc. (see Figure 3). Most of this information was filled from the knowledge base. Furthermore, the talk pages where associated with "sessions" having start and end time, and the sessions with "tracks" having a room. The wiki used this information to automatically fill in time, session, and track information in the talk pages. More complex suggestion mechanisms based on more expressive knowledge bases can easily be envisaged, as can reasoning-based methods of conflict resolution in case of contradictory knowledge.

4.4 Validating Formalised Knowledge

Background ontologies can constrain the statements within the wiki, and a reasoner can check if the knowledge does indeed adhere to these constraints. For example, it is possible to add axioms that state that each person may only have one father and one mother. A reasoner can then check if the knowledge is valid with regards to the background ontology.

Note that in order to do these tests, the system must enable average users to intuitively state the constraints they want to say. To this end, it is also useful if the chosen ontology language provides features such as (optional) constraint semantics [12], local closed world assumption [13,14], and default assumptions on uniqueness of names. Further possibilities to create unit tests, and the subtle semantic difficulties that can arise, are described in [15]. Most of the approaches can be applied in the setting of a semantic wiki.

Besides these logical constraints, the semantic wiki can use further resources from its ecosystem for more sophisticated checks. Often, it is possible to verify the content against the information in the knowledge base at the time it is entered. For instance, in case of the conference wiki described in the previous section, the system could automatically verify whether the participants associated with an event are available during this slot. Similar consistency verification can also be used for verifying the structure of

Fig. 3. Conference wiki of the "Social Skills durch Social Software" conference. Reasoning is used to automatically fill templates for talk, session, and track pages.

the human readable content. For example, meeting minutes of a project meeting might be required to always conform to a certain schema.

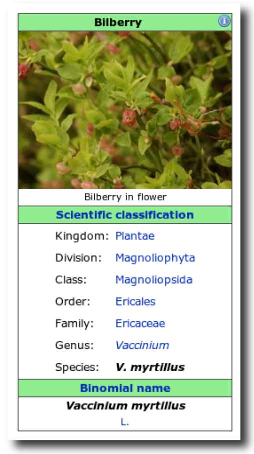
5 Scenarios for Reasoning in Semantic Wikis

We illustrate our use cases with a number of concrete scenarios where reasoning in semantic wikis is necessary and beneficial. Because semantic wiki engines currently do not implement all of the conceived reasoning functions, some of the tasks in these scenarios are currently not supported. To highlight these problems, we describe one implemented scenario, one possible but not implemented scenario, and one currently impossible scenario for reasoning in semantic wikis. One of the goals of this article is to help the readers realise what advancements are needed in reasoning in order to be able to address these issues.

5.1 Implemented: the Biology Taxonomy Scenario

In the Wikipedia, there is a huge collection of biological articles, ranging from *kingdoms* over *families*, *genres*, to *species*. In the current, non-semantic Wikipedia, this collection and the relationships (e.g. between a species and its genre) are maintained manually, which leads to redundancies and inconsistencies. In a (currently hypothetic) Semantic Wikipedia, reasoning can be used to address these issues.

Automatic inference of implicit taxonomy relations. Biology articles in Wikipedia display a "taxonomy box" in the top right corner of the page, listing the relationships between a species, its genre, its family, etc. An example of such a taxonomy box is given on the right.



In today's Wikipedia, all this information is entered manually, for each biology article. Semantic wikis can improve the situation in various ways:

- using a biology taxonomy as background ontology, a single relation from an article (e.g. a species) to the article describing its super-concept (e.g. a genre) would suffice to automatically infer all other relations (e.g. from the species to the family, order, class, kingdom),
- using appropriate inline queries, most of the taxonomy box can be computed, even without a sufficiently axiomatised background ontology,
- from the knowledge that the article describes a biological concept, semantic wikis can automatically infer that it a taxonomy box with relations to super-concepts should be displayed.

These applications exploit the methods described in the use cases *browsing and displaying*, and *querying* in Section 4.

Annotation support: offering only *relevant* relations as link types. In larger semantic wikis, there are often many different relations in the background model, most of which are irrelevant for a article. The wiki can use available information about a relations range and domain to suggest suitable annotations. In the biology taxonomy, for example, there might be a relation `has Genre` with domain `Species` and range `Genre`, and a relation `has Family` with domain `Species` ∪ `Genre` and range `Family`. For annotating a link from the page of the species *Bilberry* to the page of the genre *Vaccinium*, the IkeWiki's AJAX-based annotation editor (cf. Figure 2) can thus suggest the relation `has Species`. This is an example of the *editing support* use case of reasoning.

Query support: structured queries over the knowledge base. Reasoning based on the the biology taxonomy also allows to ask more complicated queries. For example, a user might ask for all species in the family *Ericaceae*, skipping the various genuses like *Vaccinium*. Such a query is not possible in a non-semantic wiki, and requires more extensive maintenance in a semantic wiki without reasoning support.

Challenges and Issues. The "biology taxonomy" scenario has been implemented successfully as a demonstrator in IkeWiki and will be shown in the tutorial.[10] Wikipedia's content, including all biological data, has also been imported into an installation of Semantic MediaWiki, augmented with automatically generated annotations. Nonetheless, there are a number of problems that we encountered:

- *Rule-based reasoning.* As IkeWiki and SMW rely on RDFS (or OWL DL) reasoning, there is no simple solution for axiomatising relations to biological super concepts, and IkeWiki currently implements the reasoning task in Java. Rule-based

[10] Online demo available at http://ikewiki.wastl.net:8080/

Fig. 4. A simple example query on a semantically enhanced copy of Wikipedia, showing football players with their association.

reasoning would be one solution for this problem. Axioms in many ontology languages can conveniently be represented as rules, but support for genuine rule languages is more problematic. If only restricted forms of rules are allowed, a suitable presentation to the user must be found to assist editing.

– *Representation of layout information.* There is currently no obvious way to infer a visual representation (such as "if the article is a biological concept, display a taxonomy box"). In IkeWiki, this functionality has been implemented in XSLT, but a proper integration into the knowledge base would be preferable. An existing approach for specifying layout in RDF is Fresnel[11], but from our experience expressing layout information in this format is rather hard and limited to RDF data (whereas the displaying of the taxonomy box affects the rendering of the page content).

5.2 Possible: Semantic Wikipedia

Augmenting a large-scale wiki such as English Wikipedia with semantic technologies would already be possible, but some challenges need to be addressed to make this extension successful. In contrast to the previous scenario, Wikipedia has a very broad scope, and no existing vocabulary or ontology is available to cover the whole domain. Moreover, Wikipedia is particularly large, both in terms of users and in terms of content. The possible uses of semantics in this scenario are also intractable, and we only give a few typical examples below.

Query answering. Being a general purpose encyclopaedia, Wikipedia gives rise to a number of interesting queries. Currently, those can only be answered by reading articles and manually building pages with the results of this research. This happens in practice, and has lead to the generation of many "list pages" such as the list of the largest cities in Spain. Of course, such list must be manually updated whenever some information

[11] http://www.w3.org/2005/04/fresnel-info/

changes, and inconsistencies are very likely. With semantic wikis, querying could be done dynamically by users through the wiki's query interfaces, or query results could be embedded into articles via *inline queries* to replace the current manually maintained lists (as shown in Fig. 4).

Quality control. Semantic data also provides a high-level view on article contents, enabling the automatic search for possibly wrong or incomplete information in large amounts of articles. This can be done by using domain-specific checks that are executed on parts of the semantic data. Such quality control also must involve the semantic annotations themselves: are relations used consistently? Which annotations are used too infrequently to be useful? Are there cases where multiple relations are used to model the same situation? Such tasks can be performed directly by the wiki, such that all editors can easily help to correct possible problems.

Analysing differences between wikis in different languages. Wikipedia is available in numerous languages, but each language differs in size and content. Comparing them is hardly possible, and usually only done by linking similar articles in different languages. Using semantic wikis, the content of different languages becomes machine-accessible and thereby comparable. Using known mappings between articles in different languages, the whole vocabulary of two wikis can be mapped. For instance, one could compare the population number of Paris as given in the French and Arab Wikipedia, without speaking either language. This can help to spot errors, as well as to understand cultural differences between the different language communities.

Challenges and Issues. The "Semantic Wikipedia" is not a reality yet, but semantically enhanced mirrors of Wikipedia are currently used for testing purposes. We think that the following challenges must be further addressed to enable this project:

– *Performance.* Computation times and memory usage are important for any large-scale web site, but these issues are especially problematic when offering reasoning-based services, such as semantic query answering, to a large audience. Semantic MediaWiki has always been considering performance as a critical success factor, but further improvements are needed to scale up to one of the of largest web sites on earth.
– *Usability.* It is desirable that semantic features are available to all users, even if they are not willing to learn the wiki's query language. This is partly realised by inline queries, but improved visual interfaces for constructing queries are an important next step. Similarly, annotation must be as easy as possible, and further simplification mechanisms might be needed. IkeWiki's current annotation support is a first step into this direction.
– *Interchange of data.* In order to compare data from different wikis, or to reuse such data within external applications, wikis need to export their semantic content in standardised formats. This is currently done for the simple data semantic wikis collect, but can become problematic when extending the ontological expressivity of

each wiki. Indeed, especially rule languages currently hardly support information exchange via the Web (e.g. since they do not support URIs or standard datatypes).
- *Closed and open world.* The wiki itself can be perceived as a closed world for many applications of reasoning. Users are often interested in finding out which information was *not asserted* in the wiki, instead of what information was explicitly *asserted to be false*. For example, one might look for instances of (some subcategory of) the category *Marine mammal* that do not belong to the order *Carnivora*, to find vegetarian sea mammals. In Wikipedia, no species is explicitly classified as *not* belonging to a given order, but still most users would expect a closed-world behaviour in this case.
  On the other hand, any wiki is necessarily open and dynamic, and many kinds of conclusions should not be drawn from the absence of certain statements. For instance, from the fact that some person is not classified as female, one should certainly not conclude that this person is a man. In general, the data within the wiki should always be considered incomplete. A possible solution for dealing with this situation is to explicitly assume (certain) queries to have a closed-world semantics, even if this is not assumed for the knowledge base in general. Local closed world approaches provide a foundation for this [13,14].

5.3 Vision: Project Management Scenario

The "semantic wiki for project management" scenario is a vision of how semantic wikis with reasoning support might support complex social and collaborative tasks like project management. This vision requires significant development work and in some cases involves also reasoning problems that are yet to be solved.

In recent years, wikis have increasingly be used as tools for project management. Major reasons for this are that wikis are easy to set up and use, allow everyone to edit, strongly connect knowledge via hyperlinks, and – most importantly – do not impose a rigid workflow by means of technical restrictions (instead, workflows are usually followed by social convention). Wikis thus provide the flexibility that is needed particularly in collaborative software development, where people and organisations with different cultures and backgrounds work together.

Unfortunately, existing wiki software is not particularly well suited for such tasks. On the one hand, current wiki systems are not really able to work with non-textual content such as spreadsheets, diagrams, or video material. On the other hand, they do not provide support for the semantically rich content that participates in processes like project management and software engineering.

A sufficiently developed semantic wiki system could support project management in a number of ways. Some of them are exemplarily given below.

Visualisation and editing of project data. Project management involves a lot of data about resources, dates, progress, meeting minutes, persons, etc. A lot of different visualisations have been developed to support these tasks, e.g. Gantt and Pert diagrams (see Figure 5), project progress reports, etc. Most of these visualisations are views on the same data. In a semantic wiki, such data could be represented formally in the knowledge base. The semantic wiki could then offer the user different visualisations of the

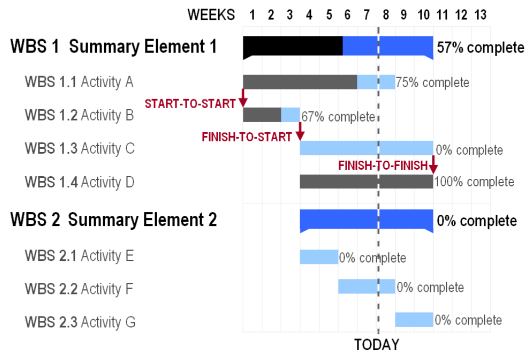


Fig. 5. Example of a Gantt diagram (source: Wikipedia); a semantic wiki for project management could provide such a diagram as a view on the data in the knowledge base for visualisation and interactive editing of project workplans.

data based on reasoning about the context. For instance, a page describing the workplan or project progress could automatically include a Gantt diagram, or a page describing a meeting could automatically show the project calendar and a list of persons that should attend the meeting. Both tasks involve querying and reasoning for retrieving the relevant knowledge from the knowledge base.

Likewise, interactive visualisations could allow to modify the project resources. A Gantt diagram about the project workplan could offer ways to e.g. add resources to a task, move task start and end via drag and drop, etc. Besides user interface issues, this kind of editing also involves reasoning. In this case, reasoning is in a sense “two way” (similar to view updates in relational databases), because on the one hand reasoning serves to transform the data in a suitable visualisation and on the other hand it transforms the visualisation back into the data.

Planning support. Project management also involves a lot of planning. A common planning task in project management is appointment scheduling. For example, a project manager might want to find a possible meeting date for all persons involved in a certain component of a software project when creating the description of the meeting in the project wiki. Obviously, this task requires different kinds of rather complex reasoning, like temporal reasoning for finding out appropriate dates and rule-based reasoning for determining whom to invite to the meeting. It would also be desirable for the system to fail gracefully in case a meeting date where all involved persons are free cannot be determined: in such cases the system should suggest a “good enough” solution.

Similarly, planning support in a semantic wiki can be useful in other project management areas, like resource planning (which requires constraint checking) or issue tracking.

Workflow support. A third area where reasoning in a semantic wiki for project management is important is workflow support. In a semantic wiki, project workflow models could be represented by some kind of ontology (not necessarily OWL DL, specific process modelling languages might be more appropriate). Reasoning would be used to guide the user along this workflow. For example, when taking meeting minutes, the system could automatically offer to enter action items into an action item list and provide a suitable editor for this task. For an action item, the semantic wiki could then offer to move along a defined workflow, e.g. from the state "open" to the state "closed". In the case of bug tracking, specific support for such workflows is already given by the system trac.[12] However, this support is specific to the case of bug tracking and does not follow an explicitly represented model.

Challenges and Issues. In order to realise a semantic wiki for project management, there are many challenges that need to be addressed in the area of reasoning. In the following, we mention a few of the most important ones (the challenges mentioned in the other scenarios are mostly relevant here as well):

- *Temporal reasoning.* Many tasks in project management require to reason over temporal data (e.g. appointment scheduling or task editing). However, current knowledge representation formats and reasoning languages do not offer much support in this area.
- *Constraint verification.* To ensure consistency of the data, a reasoning system needs to support the verification of constraints. Constraint verification, although not "generative", can also be considered a reasoning task, and has been studied much in the context of relational databases.
- *Active and reactive rules.* Many functionalities cannot be sufficiently covered by deductive systems. A system that closely interacts with users needs to support active or reactive behaviour. Active rules (or "Event Condition Action" rules) are a way to address this issue. In the context of the (Semantic) Web, such rules have e.g. been studied in the language *XChange* [16].

6 Related Work

Besides IkeWiki and SMW, a number of other semantic wiki implementations have been created, most of which remain research prototypes [17,18,19,20]. To the best of our knowledge, the most notable (and stable) system currently is *MaknaWiki* [21], which is similar to SMW and IkeWiki with respect to the supported kinds of easy-to-use inline wiki annotations, and various search and export functions. In contrast to SMW, IkeWiki and MaknaWiki introduce the concept of *ontologies* and (to some extent) *URIs* into the wiki, which emphasises use-cases of collaborative ontology editing that are not the main focus of SMW. MaknaWiki is tailored for closed-domain settings where a suitable schema for annotation can be anticipated. It supports the usage of expressive ontologies and integrates according inferencing features, but allows only administrators to change the ontological schema.

[12] http://trac.edgewall.org

Besides text-centred semantic wikis, a number of collaborative database systems have appeared recently. Examples of such systems include *OntoWiki* [22], *OpenRecord*[13], *Metaweb*[14], and *OmegaWiki*[15], most of which are still preliminary or only very recent. Such systems typically use form-based editing, and are used to maintain data records instead of texts. OntoWiki draws from concepts of semantic technologies and provides a built-in faceted (RDF) browser. The other systems have their background in relational databases.

7 Perspectives and Conclusion

In this article, we discussed how reasoning can improve the use of semantic wikis in the areas of *browsing*, *querying*, *editing*, and *validation*, and which requirements and future challenges arise from these use-cases. We also highlighted first simple applications of reasoning within the semantic wiki systems *IkeWiki* and *Semantic MediaWiki*, which we currently develop, and described scenarios for further, more complex applications. Based on the above analysis, we plan to further improve reasoning support in a number of ways.

IkeWiki will be extended by a rule-based reasoning engine that allows to define more complex "views" on the RDF data in the knowledge base. Current experiments with connecting to SWI Prolog have been promising, but this system is not yet stable enough for inclusion in the main distribution. Second, we plan to extend IkeWiki by support for project management tasks, such as appointment scheduling and document management. To this end, we are currently investigating customised editing based on knowledge about the content of a wiki page as described in the third scenario.

Semantic MediaWiki aims at providing a platform that can scale to the size of Wikipedia, and thus powerful reasoning is strongly constrained by performance requirements. The current stand-alone storage model based on MySQL will soon be augmented with optional bindings for (presumably faster) RDF stores. Bindings for reasoners as in [9] are also considered, but are hindered by the restricted query languages of many reasoners.[16] The main goal for extending expressivity are role hierarchies and role composition, as well as efficient DL constraints in the style of [12]. We do not intend to significantly increase the expressivity of the query language, since while sub-polynomial in complexity, queries are already very challenging in real-world scenarios.

We believe that experiences gathered in semantic wikis are highly relevant to many other applications of expressive semantics, since they provide typical situations for a number of problems that still need to be overcome by the Semantic Web community. As semantic wikis have many characteristics of "small semantic webs", many of these issues are equally relevant for the Semantic Web as a whole.

[13] http://www.openrecord.org

[14] http://www.metaweb.com/

[15] The former *WikiData*, http://www.omegawiki.org

[16] One problem is that the current SPARQL specification does not directly support OWL.

References

1. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation (10 February 2004) Available at http://www.w3.org/TR/rdf-schema/.
2. McGuinness, D., v. Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation (10 February 2004) Available at http://www.w3.org/TR/owl-features/.
3. Völkel, M., Schaffert, S., eds.: 1st Workshop "From Wiki to Semantics" (SemWiki'06), Budva, Montenegro (2006)
4. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic MediaWiki. In: Proc. of the 5th Int. Semantic Web Conf. (ISWC'06). (2006) 935–942
5. Krötzsch, M., Vrandečić, D., Völkel, M.: Wikipedia and the Semantic Web – the missing links. In: Proc. of WikiMania2005. (2005)
6. Schaffert, S.: IkeWiki: A semantic wiki for collaborative knowledge management. In: 1st Int. Workshop on Semantic Technologies in Collaborative Applications (STICA'06), Manchester, UK (2006)
7. Schaffert, S., Westenthaler, R., Gruber, A.: IkeWiki: A user-friendly semantic wiki. Demonstration at the 3rd Europ. Semantic Web Conf. (ESWC'06), Budva, Montenegro (2006)
8. Oren, E.: SemperWiki: a semantic personal wiki. In: 1st Workshop on The Semantic Desktop, Galway, Ireland (2005)
9. Vrandečić, D., Krötzsch, M.: Reusing ontological background knowledge in semantic wikis. In: Proc. of 1st Workshop "From Wiki to Semantics" (SemWiki'06). (2006)
10. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05), Edinburgh, UK, Morgan-Kaufmann Publishers (2005)
11. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexity boundaries for Horn description logics. In: Proc. 22nd AAAI Conference on Artficial Intelligence, Vancouver, British Columbia, Canada, AAAI Press (2007)
12. Motik, B., Horrocks, I., Sattler, U.: Integrating description logics and relational databases (Dec 6, 2006) Technical Report, University of Manchester, UK.
13. Rosati, R.: Autoepistemic description logics. AI Communications **11** (1998) 219–221
14. Bonatti, P., Lutz, C., Wolter, F.: Expressive non-monotonic description logics based on circumscription. In Doherty, P., Mylopoulos, J., Welty, C., eds.: Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). (2006) 400–410
15. Vrandečić, D., Gangemi, A.: Unit tests for ontologies. In Jarrar, M., Ostyn, C., Ceusters, W., Persidis, A., eds.: Proceedings of the 1st International Workshop on Ontology content and evaluation in Enterprise. LNCS, Montpellier, France, Springer (2006)
16. Bry, F., Eckert, M., Pătrânjan, P.L.: Reactivity on the web: Paradigms and applications of the language xchange. Journal of Web Engineering **5** (2006) 3–24
17. Tazzoli, R., Castagna, P., Emilio, S.: Towards a semantic wiki wiki web. In: Demo Session at ISWC'04. (2004)
18. Souzis, A.: Rhizome position paper. In: Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web. (2004)
19. Aumüller, D.: Semantic authoring and retrieval in a wiki (WikSAR). In: Demo Session at the ESWC'05. (2005)
20. Kiesel, M.: Kaukolu – Hub of the semantic corporate intranet. In: Proc. of 1st Workshop "From Wiki to Semantics" (SemWiki'06). (2006)
21. Nixon, L.J.B., Simperl, E.P.B.: Makna and MultiMakna: towards semantic and multimedia capability in wikis for the emerging web. In: Proc. Semantics 2006. (2006)
22. Auer, S., Dietzold, S., Riechert, T.: OntoWiki – A tool for social, semantic collaboration. In: Proc. 5th Int. Semantic Web Conference (ISWC'05). Number 4273 in LNCS, Springer (2006) 736–749