2008

Semantic Wikis: Conclusions from Real-World Projects With Ylvi

N Popitsch
University of Vienna, niko.popitsch@univie.ac.at

B Schandl
University of Vienna, bernhard.schandl@univie.ac.at

r Ross
University of Vienna, ross.king@univie.ac.at

SEMANTIC WIKIS: CONCLUSIONS FROM REAL-WORLD PROJECTS WITH YLVI

Popitsch, Niko, University of Vienna, Department of Distributed and Multimedia Systems, Liebiggasse 4/3-4, A-1010 Vienna, Austria, niko.popitsch@univie.ac.at

Schandl, Bernhard, University of Vienna, Department of Distributed and Multimedia Systems, Liebiggasse 4/3-4, A-1010 Vienna, Austria, bernhard.schandl@univie.ac.at

King, Ross, University of Vienna, Department of Distributed and Multimedia Systems, Liebiggasse 4/3-4, A-1010 Vienna, Austria, ross.king@univie.ac.at

Abstract

Wikis have proven to be a useful tool for knowledge creation, sharing, and archival in a number of application fields. Their unstructured, flat nature makes them an easy-to-use tool for many purposes, while simultaneously imposing a significant problem: especially when wikis grow in size, content becomes hard to find and navigate. Semantic wikis enrich traditional wikis with semantic annotations like categories, attributes, and typed relationships. Although a number of promising research prototypes have been developed, little has been reported on how semantically enabled wikis are used in production environments. In this paper we describe relevant aspects of our semantic wiki implementation Ylvi and present a first analysis of scenarios in which a semantic wiki has been successfully used in the real world. We describe three different applications in the areas of education, project management and content management and identify experiences, lessons learned, and future directions for research on semantic wikis.

Keywords: Application Scenarios, Case Studies of Real-World IS Deployments, Collaboration, Emerging Technologies, Knowledge Sharing, Ontologies, Tools, Usage.

1   INTRODUCTION

Wikis today are gaining increasing attention as tools for collaborative knowledge capture, knowledge management, and learning. Among various other applications fields, they have already been employed as lightweight content management tools for a number of information- and knowledge-centric scenarios including project management, e-learning, encyclopaedias, dynamic websites, and public and internal archives. It seems clear that wiki-based systems can constitute considerable alternatives to conventional (commercial) computer systems in these and many other areas as reported e.g. in (Majchrzak et al., 2006). The shortcomings of traditional wikis, like the very popular MediaWiki, in capturing semantic information about and semantic relationships between resources have motivated the development of semantically enabled wikis, or semantic extensions to existing systems (such as Semantic Wikipedia, see Völkel et al., 2006).

In this paper we report on the experiences we have gathered with our own semantic wiki implementation *Ylvi*[1] with which we have implemented several projects within the last 18 months. We have selected three projects (described in Section 3), two of which have been in use for more than one year, that largely differ in their thematic orientation and their target user groups. We believe that empirical feedback and case studies from real-world scenarios can help the research community a great deal to shape its research and development strategies for semantic wikis.

In Section 4 we report some of the conclusions we have derived from these projects that we believe would also apply to other semantic wikis in real-world projects with similar requirements. In Section 5 we outline future directions for our own research and implementation.

Related Work. In this section we consider related work that reports on wiki-based projects. Other related work is cited throughout the paper where appropriate.

Although a number of promising research prototypes have been developed[2], semantically enabled wikis are still rarely used in production environments, and little is reported about how users utilize and benefit from the semantic features they provide. We suspect one reason for this may be that these features are new to a majority of users, and their benefits may not be immediately apparent. For this reason, it is incumbent upon the semantic wiki research community to clearly indicate use cases, scenarios, and projects that show the increased benefit of semantic wikis.

This is, to a lesser extent, also the case for traditional wikis. Although many companies and scientific institutions now include wikis in their intranet landscape, surprisingly few scientific papers have been published in this area of research. A reasonable discussion about the particular requirements of intranet wikis can be found in (Stenmark, 2005). The conclusions described there can be applied to semantic wikis as well, particularly when they are used in an intranet-like manner, as in our described project management scenario (cf. Section 3.3). In such a case semantic wikis can help to make the co-worker's knowledge more explicit and, even more importantly, machine-processible.

In (Buffa & Gandon, 2006) the authors present their semantic wiki implementation together with a number of relevant references to (rather large-scale) real-world wiki projects. They also describe how semantic technologies can address a significant number of the problems and shortcomings of traditional wikis; still, there is no indication whether the implementations have really delivered what they promise. (Bruns & Humphreys, 2005) report on the process of building up a public, special-interest encyclopaedia with a wiki. Lund & Smørdal (2006) and Wang et al. (2005) highlight certain aspects of the role of wikis in collaborative learning based on two projects they have established: one in an upper secondary school in Norway, the other at a public college in Taipei. These projects can be

---

[1] http://www.cs.univie.ac.at/ylvi
[2] Refer to http://ontoworld.org/wiki/Semantic_Wiki_State_Of_The_Art for an overview.

considered similar to our PoWiki project (cf. Section 3.1) where we attempt to analyze the usability of a semantic wiki in an educational setting.

2   AN INTRODUCTION TO YLVI

We have deployed a number of real-world projects based on our semantic wiki implementation *Ylvi*. Ylvi is a flexible semantic wiki toolkit with a special focus on media management and provides three major semantic features: *resource typing* (where resources are both wiki articles and media objects), *link typing* and *assigning attributes*. Further information about Ylvi can be found in (Popitsch et al., 2006); some of its relevant characteristics are described in the following.

Plug-In Framework. The flexibility of Ylvi originates to a great deal from the plug-in framework of the underlying multi-media middleware METIS (King et al., 2007). METIS constitutes the technological foundation of Ylvi and provides convenient methods for extending the semantic features as well as the functionality of the system at runtime.

Semantic artefacts, like article types or link types, may be added to or removed from Ylvi at runtime via a convenient web-interface. Such artefacts may also be grouped into so-called *semantic packs* (or *sempacks*), atomic units that group semantically related artefacts in an XML namespace-like fashion. Sempacks are used to establish the basic semantics of the application: a basic administrative ontology is contained in an Ylvi sempack that is uploaded during installation. This ontology contains fundamental concepts of our semantic wiki, like *Article*, *Media*, or basic link types. Semantic packs may be hierarchically organized and make use of other sempacks in order to re-use already existing models. Semantic artefacts can be used to annotate Ylvi articles by adding special markup elements directly to the article source code; for instance, `<<concept>>` for assigning ontology concepts to articles, or `~attribute=value~` for assigning attribute values.

While sempacks provide customizations of the underlying data model, *kernel plug-ins* provide functional extensions for the system. Kernel plug-ins are basically Java packages that extend the functionality of the system and communicate with the METIS kernel via an event mechanism.

Flexible User Interface. Ylvi makes direct use of the METIS plug-in framework and extends it by introducing two specializations of kernel plug-ins: *render plug-ins* and *toolbox plug-ins*. Render plug-ins provide a method for mapping from Ylvi's wiki markup language to various output formats (e.g., HTML or LaTeX, but binary output formats are also possible) through a modularized pipeline mechanism. Note that this mechanism makes Ylvi also very extensible in terms of additional markup elements that are favoured in a given application scenario.

Toolbox plug-ins are visual kernel plug-ins that extend the default Ylvi user interface. "Visual" means that unlike render plug-ins they include their own user interface. When such a plug-in is deployed on an Ylvi instance, a link is automatically added at certain predefined locations on the default Ylvi user interface.

Collaborative Ontology Building. One of the recent extensions we have developed for Ylvi is *OntoYlvi*, a plug-in that provides collaborative ontology building features. The OntoYlvi extension consists of a semantic pack and a kernel plug-in. The semantic pack extends Ylvi by a meta-model that is used to model new ontologies. This meta-model contains for example article types like *Concept* or link types like *Sub Concept Of* for the modelling of concept hierarchies. The kernel plug-in converts articles and links typed with such artefacts to a corresponding model instantiation (types, attributes, and relations). When this plug-in is activated it incrementally updates any prior OntoYlvi ontology, allowing for gradual, collaborative ontology building. Ontological concepts can be associated with categories, which are in turn reflected in the automatically generated search interface, as described in the faceted search section below. A comprehensive discussion of OntoYlvi is beyond the scope of this paper, refer to (Hepp et al., 2006) for a comparable approach.

Role Based Access Control. Ylvi benefits from another feature of the METIS multimedia framework, the role base access control (RBAC) mechanism. In METIS, access to all data objects can be restricted based on user roles. Policies can be uploaded to the system in the XACML (Moses, 2005) format. Ylvi uses these features to restrict access rights (read, write, delete) for articles and media objects. Privileged users can set access rights for these objects via the Ylvi web interface.

Faceted Search and Navigation. Ylvi exploits its semantic data model mainly to provide improved search and navigational functionality. Ylvi makes direct use of the METIS query processor in order to build up its faceted search interface. Users may search for content via multiple dimensions, including types, attribute values, relations, and full text. Searching in Ylvi is carried out by adding and removing search filters to/from a filter container that represents the current search process. Whenever a filter is added or removed, an Ylvi search is triggered and the results are listed.
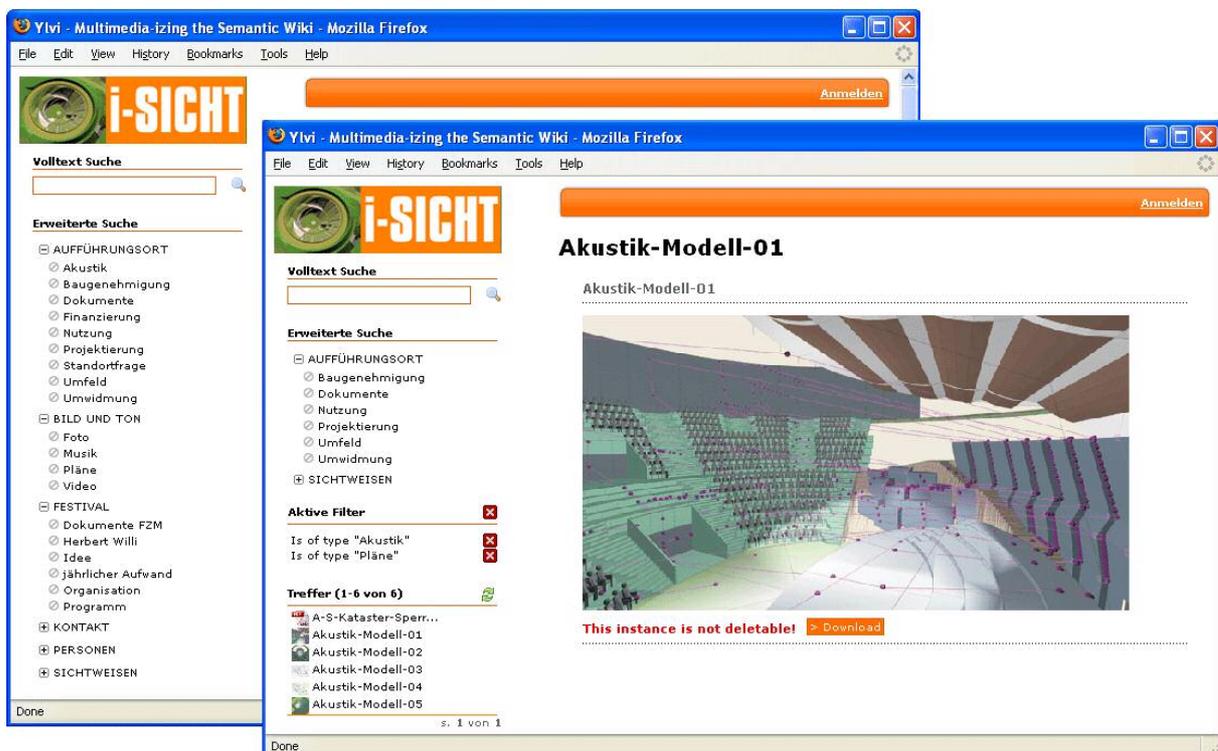
Figure 1: Screenshots from the I-Sicht scenario demonstrating one of the core concepts of Ylvi, faceted search. The window in the back shows all available concepts the user may search for. The window in front shows the search interface after two type-filters were added by clicking on the respective concepts. Note that now only those concepts that would lead to a further refinement of the result set remain available in the search interface.

The core concept behind Ylvi's search interface is to guide the user in the search process until she arrives at a manageable list of search results that can be individually checked for relevance. This query refinement technique is supported in the following manner: The Ylvi search interface is constantly adapted to the current result set: only those facets (article types, link types, etc.) that are capable of further refining the current result set are made available in the search interface. At the start of the search process, the universal set of all Ylvi objects is considered, thus all facets that are referenced by any Ylvi object are displayed. Once a user has defined a search filter, only the calculated result set is considered by the Ylvi search interface; thus some search facets will disappear because they are not

capable of further refining the result set (because none or all contained objects reference this facet). Consider Figure 1 for an illustration of this mechanism.

A detailed discussion of the Ylvi search interface and algorithm is beyond the scope of this paper. Based on our initial feedback, this approach for the category search appears to be quite intuitive for the user, despite its apparent complexity.

Inline queries. It is possible to embed queries directly in article source code. Whenever such an article is rendered, the embedded query is executed and the links to the corresponding result set objects are listed. Embedded queries are defined using the common query language (CQL)[3], a simple standardized query language, and allow the definition of dynamic link lists for directory articles, index articles, or similar navigational aids.

Other features. There are two other useful Ylvi features that are not common in wiki implementations: First of all it is possible to rename articles and media[4]. Furthermore, names in Ylvi are not unique, so it is possible to have multiple articles or media items with the same name. In this case, accessing a named link (that is, a link that references an Ylvi object by name) results in a disambiguation context menu which provides further information to the user about the possible link targets. Both of these features were identified as beneficial by Lund & Smørdal (2006, p. 43).

3 APPLICATION SCENARIOS

During the past 18 months we have deployed a considerable number of projects that were technologically based on Ylvi. In the following we report on three of these projects that differ substantially in both their target audience and their general character. In all three presented projects, we acted only as technology providers. We set up the system and demonstrated how to use Ylvi, but we did not interfere with how the authors created their respective content and used Ylvi's semantic features. It was our goal to discover how non-expert users would make use of a semantic wiki in a real world environment.

3.1 Scenario 1: PoWiki

The PoWiki project (http://powiki.univie.ac.at) was started in August 2006 with the goal of building an encyclopaedia for political science in Austria (see Figure 2). From the beginning, PoWiki was designed to act as both a publicly accessible encyclopaedia and as a platform for the e-learning support of university courses in the field of political science.

Using the extension mechanisms of Ylvi, we set up an environment that enables students to register with PoWiki for a particular course, as there were six courses running in parallel. Students were assigned their own home article, which they worked on throughout the course. This article was readable but not writable by their colleagues, who could however add comments. On specific dates during the semester the course instructors locked the articles for reviewing, after which no further changes by the students were possible. After this evaluation phase the articles were made anonymous and publicly accessible. From then on the students were able to contribute to the articles of their colleagues, e.g., to improve linking. Although this final collaborative phase was on a voluntary basis, a

---

[3] http://www.loc.gov/standards/sru/specs/cql.html
[4] Whereas in MediaWiki, "renaming" an article results in copying the content to a new article and then redirecting from the old to the new article. Although this feature is also present in Ylvi, renaming articles proved its usefulness in early project stages when many articles do not have their final names yet.

significant number of students contributed and this greatly improved the content quality and usability of the system.

Ontology. During the first semester of PoWiki, a small and simple domain ontology was developed in a collaborative effort among the course instructors. This ontology consisted of ten resource types and three link types. Furthermore, a concept was created for every course in order to allow the tagging of an article with its corresponding course. As concepts in OntoYlvi are also represented as articles, these articles were also used to describe the course itself. In addition, a small ontology was deployed that enabled Ylvi to store the mentioned user comments.

Users and User Groups. The PoWiki project consists of four distinct user groups: *administrators* (a mentoring tutor and us), *editors* (so far ten different course instructors), *students* (about 300 registered users so far), and *anonymous users* (visitors from the web). All user groups have different access rights for various PoWiki resources, and some users belong to multiple user groups.
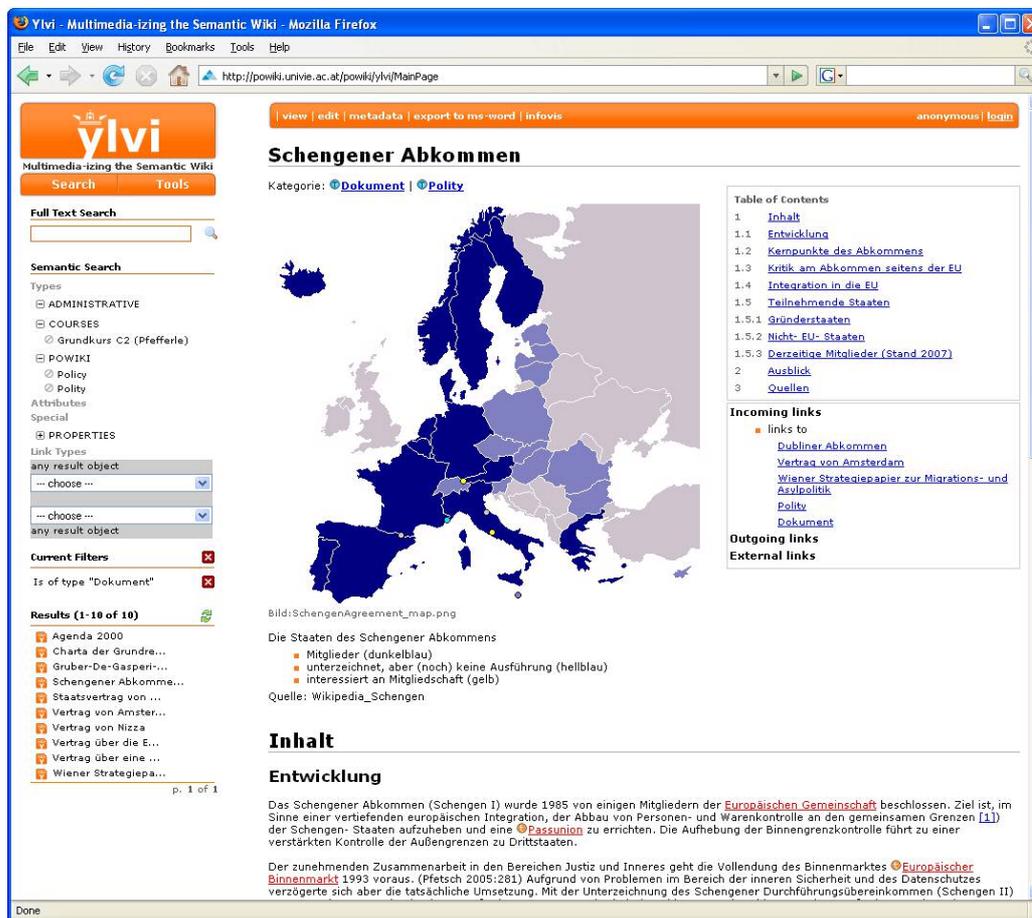

Figure 2: PoWiki screenshot

Administrators and editors were added to the system manually, whereas students registered themselves via a registration system that was developed in the form of an Ylvi plug-in. This system allowed student self-registration and required them to enter the name of their home article, i.e., the encyclopaedic entry they would work on. Based on this data the registration plug-in automatically created an article stub that was already tagged with the concept of the corresponding course. This immediately allowed users to find all articles associated with a particular course and was also used for an automatically created list of these entries in the course's article itself. The latter was realized using

an inline query in the article corresponding to the course concept, which displayed a list of all entries corresponding to the course.

Training. After the project was set up, an initial group of course instructors and the mentoring tutor were trained for approximately two hours on how to use the system. From then on they trained their colleagues themselves. Students were introduced to the tool during the first lecture of a given course. A set of help articles was created in order to support the authors and the students. Informal polls indicated that very few students have previously authored a wiki article.

Preliminary Results. So far around 300 students from nine different basic and advanced courses from the field of political science have contributed approximately 450 entries to PoWiki. No hierarchical relationships between the ontology elements were defined, as this was not requested by the course instructors. Thus the ontology basically consists of a controlled vocabulary that was used by the students for article and link typing. At the time of this writing, 236 of 443 entries (53%) were tagged by at least one (non-administrative) concept.

Link typing was by far less often applied. The vast majority of links were untyped (1599 of 1647, or 97.09%). From the PoWiki domain ontology only the link type *see also* was used for 29 links, or 1.76% of all links, while the other two types, *is part of* and *consists of*, were hardly used. The few remaining typed links were taken from the administrative ontology.

3.2   Scenario 2: I-Sicht Wiki

The I-Sicht project (http://www.i-sicht.eu) was started in late September 2007. The goal of the I-Sicht consortium was to establish a new music festival in Vorarlberg/Austria—detailed information can be found on the website (in German). The consortium had been active for over six years before they approached us with the request to help them present all the documents and media files they have collected so far in a single, information-centric website. Thus the goal of the project was to present the current status as well as the project history to the media, politicians, and citizens. Furthermore the platform should from now on be used as the central project management tool by the I-Sicht team. Thus I-Sicht is an example for the application of a semantic wiki as a web-content management system. A crucial factor for this project was the time pressure it was under—in order to meet a specific deadline, the application had to be online about two weeks after we have met the project team for the first time.

Users and User Groups. The project required only two user groups: *administrators* or content providers (the I-Sicht project team), and *anonymous users*, that is, external visitors of the web site. The project team used the RBAC features of Ylvi to protect internal and draft documents from unauthorized access.

Training. Due to the mentioned time pressure, we introduced the team to the features of our semantic wiki implementation and trained them during a two day workshop. At this time, the core of the domain ontology used by I-Sicht was developed in a collaborative manner and deployed using OntoYlvi. Also, a significant fraction of the available content was imported. After the workshop, we provided only technical support, as the project team was fully capable of authoring articles, importing the remaining media, extending the ontology, and adjusting the access rights.

It should be noted that the I-Sicht administrators were not expert computer users although all had experience with computers from their professions (e.g., one member was an architect). None had previously authored a wiki article.

Ontology. Because the project team wanted to present the content from various points of view (a "cultural view", an "architectural view", and so on) and due to the diversity of the content in form and content, a fairly extensive ontology (compared to the number of content items) was developed. So far the applied domain ontology consists of 30 concepts from six different categories for about 370 content items. Apart from this, only the Ylvi administrative ontology was imported to the application.

The character of the ontology is widely flat, but some hierarchical relations were also defined: for example, the concept *Politician* was defined as a sub-concept of the concept *Person*.

Preliminary Results. At this time the I-Sicht instance consists of about 120 articles and 250 media items. The media items include PDF documents, images, videos, and audio files. The authors extensively used the possibility of tagging articles and media objects with concepts from the domain ontology—nearly 88% of the content was classified with a domain concept, in most cases with more than one concept.

In contrast, link typing and attribute values were not used at all. The content authors stated that manually entering semantic relationships between objects would require considerable effort, and they doubted that the typed-link search interface would be used by many users as it seemed not sufficiently self-explanatory to them. Therefore they doubted the usefulness of explicit semantic relationships in their content and fully relied on the content tagging features for grouping semantically related objects. Defining metadata attribute values, in turn, seemed not very useful in this scenario: no obvious attributes were identified by the project team, aside from the automatically defined creation and modification dates.

In this scenario, use of the semantic features was in principle comparable to MediaWiki's category system[5] but with the additional possibilities for faceted navigation and semantic search that are currently missing in the MediaWiki software. See (Völkel et al., 2006) for a sound discussion about the shortcomings of MediaWiki's categories and a current approach to overcome them. We also consider the fact that it was possible to realize this project in such a short time as evidence for the applicability of semantically enabled wikis in similar scenarios.

3.3 Scenario 3: SemDAV Ylvi

Wikis are more and more often being used as collaborative platforms for the management of distributed projects; for example, wikis are emerging as the technology of choice for managing EU-funded projects, in which at least three countries must be involved. We are presently using Ylvi as a communication platform for the SemDAV research project (http://www.semdav.org). This project involves three Austrian partners with separate locations and IT infrastructures; in this context, Ylvi offers an ideal common information exchange platform, available at any location with an internet connection. Meeting minutes, progress reports, design documents, and related reference materials can all be shared and annotated through Ylvi. The semantic search features in combination with the full-text search supports rapid access to important project-related information.

In order to use Ylvi as project-internal wiki, no customization was necessary—Ylvi was deployed "as is" from the release distribution with standard visualization plug-ins. Because there were a known and limited number of users, roles were managed through the underlying METIS administration interface, although in principle the registration plug-in developed for the PoWiki scenario could have been used for this purpose as well.

The project-internal Ylvi instance has been used during all phases of the project. During the proposal phase, it was used to collect project ideas and drafts, to store information about the project partners, to organize references to related work, and for the actual authoring of the proposal. For project management activities each project team member created a personal page, and meeting minutes were entered on the fly. Publications were partly edited directly within the wiki; in cases where this was not possible the wiki was used to gather ideas and text blocks for the final documents.

Users and User Groups. The SemDAV project itself deals with the use of ontologies and extended expressive semantic concepts in the application domain of the Semantic Desktop. Hence the majority

---

[5] http://www.mediawiki.org/wiki/Categories

of the ten project team members were familiar with ontologies, classes, attributes, and so on. Two members of the SemDAV project team were also involved in the design and development of Ylvi, thus the features and usage of Ylvi were well-known to them. These two experts were assigned the Administrator role. In this Ylvi instance, administrators were the only user group that were allowed to modify the ontology and to manage users.

The remainder of the core project team was assigned a registered user role. Registered users were allowed to create and modify articles, add media items, and use semantic features like classification and assigning attribute values.

A third user group were associated students. Students contributed to the project during their assigned courses. Only restricted access was granted to students: they were allowed to monitor the project progress and read technical documents and meeting minutes, but were not allowed to modify any of the data. Anonymous access was disabled in this Ylvi instance.

Training. The level of training was different for the various user groups. For the administrators, no training was required since they were already familiar with the functionality and the user interface of Ylvi. For the regular project team, a four-hour training session was held. In this session the administrators demonstrated the use of the main functionality of wikis, as well as specific functionality of semantic wikis: creation and modification of articles, semantic annotation, and search.

During the training session, not only the usage of Ylvi but also the underlying semantic concepts were discussed. In the course of this discussion we noticed that the vocabulary used to describe the Ylvi concepts (e.g., articles, article types, links and typed links) are, in principle, understood by all users, but often their semantics was interpreted differently. These misunderstandings led to follow-up discussions about naming in semantic systems, but were clarified later when the project team members actively used the different features. This experience indicated that knowledge about the value and potential of semantic technologies can probably best be derived from active work, not necessarily from classroom teaching.

Ontology. No predefined ontology was used for the project; instead, project team members created ontological concepts on demand. If a user wanted to introduce a new concept (for example *Technical Report*), she simply used it. Because it did not yet exist in the ontology, it would appear in red as an unknown class in the article text. To resolve this, she notified an administrator, who then elevated the concept to the formal ontology and, if applicable, integrated it into the class hierarchy.

During the initial project phase, the ontology grew to about 20 concepts, 5 attribute types, and 7 link types. The ontology consists of classes typical for projects like, for example, *Person*, *Meeting*, *Related Work*, and *Use Case*. Where applicable, ontology concepts have been integrated in a small class hierarchy (e.g., the class *Meeting* has the subclasses *Technical Discussion*, *Jour Fixe*, and *Steering Committee Meeting*). This phase lasted about three months from the project start. After that time, no major changes or additions to the ontology have been made; thus we consider the ontology as stable for the purposes of a small research project. Since then it has been used for efficient search and structured archival of project-related information.

Preliminary Results. So far the SemDAV project wiki contains 140 articles; however as the project has reached its midpoint at the time of writing we expect this number to grow significantly. 64 articles (46%) have not been typed with any concept from the domain ontology. Additionally, around 50 media objects have been uploaded; these include technical diagrams and sketches, as well as project-related text documents, but also relevant publications from external sources.

It is important to notice that in this instance the *embedded queries* feature is heavily used. Embedded queries change the way a user looks for information since it gives an overview of articles that meet certain fixed criteria. We also believe that these queries provide a measure of self-organization that adds necessary structure to otherwise unstructured wiki content.

4 CONCLUSIONS

Semantic Features. First, we note that the semantic features of Ylvi were used in every project we have deployed so far although this was not required of the content authors. Apparently these features were considered advantageous in some way. In particular, the ability to type articles was extensively used in all projects—even though many authors had not used semantic tagging technologies before.

Content tagging seems to be a quite intuitive concept, usable for content structuring and improved searching. Accordingly, the developed ontologies tended to be rather broad but quite shallow, similar to tag lists. We ascribe this also to the lack of experience of our test users with the process of ontology building, which is already known to be quite a complex one. The usefulness of more advanced ontologies for improved content classification and searching was not doubted by any content author we have worked with so far, but even the definition of shallow, hierarchical taxonomies required a controlled authoring process lead by expert users, something that might not be feasible in every project scenario.

Typed linking was successfully applied only in the SemDAV wiki scenario, which included among its authors expert users who were familiar with formal ontologies. In general, feedback regarding search for typed links has been mixed, and the interface for this semantic facet requires further investigation and improvement. It seems to be quite suitable for advanced editors and delivered considerable benefits for searching and browsing the content as well as for the creation of auto-generated lists. Such lists (embedded queries) were successfully adopted in our application scenarios and, according to informal polls, greatly help users to get an overview over available content.

Being aware of the ongoing discussions about folksonomies vs. ontologies (cf. Gruber (2005), Echarte et al. (2007), Buffa & Gandon (2006)), we believe that our scenarios confirm that there exists no simple answer to this question. Generally speaking, it appears to be satisfactory to start with very simple tag lists, but later (or in more knowledge-oriented scenarios) it might be advantageous to incrementally strengthen the expressed semantics of the system, e.g., by interrelating the tags. We consider our OntoYlvi extension an appropriate tool for this, as it allows improvement of the semantic model of an instance at any time without the fear to losing semantic information.

Attribute value assignments were used in different ways in the projects we have implemented with Ylvi so far. In most scenarios we observed that users did not enter metadata values directly, although this was the case when expert users were involved, as in the SemDAV scenario. We ascribe this not only to deficiencies in the user interface, but also believe that the benefits that authors obtain from entering such values – like improved search – are less obvious during authoring, compared to content tagging. However, even when attribute values are not manually entered via the default Wiki user interface, we believe the capability of the Ylvi data model to store metadata attributes in a structured manner is beneficial: it enables Ylvi to store information derived automatically or entered via alternative user interfaces in a structured way and allows users to search for articles based on attribute values. For example, in PoWiki users enter comments via a special comment plug-in, and the comments are stored as values of a string attribute assigned to the respective content article.

In summary, we conclude that wiki implementations that should be adapted to diverse real-world projects should not only support intuitive and simple content tagging features but also the possibility for advanced users and automated system components to express semantics more precisely. Naturally, this requirement has an impact on the user interface of such wiki implementations. It is our observation that the discussed semantic features – when used appropriately – are capable of greatly enhancing the navigational capabilities of a wiki (e.g., by providing multi-faceted search interfaces or automatically generated link lists).

Other Features. Although it somehow contradicts the original idea of wikis, all projects we have implemented so far required some sort of access control to protect certain content, the ontology, or some user interface functionality. This requirement was also identified by others (Wang & Turner,

2004), (Elrufaie & Turner, 2005). We have had very positive experiences with our role based access control model as it greatly reduces administrative effort in scenarios with large numbers of users, as was the case in the PoWiki scenario in particular.

Semantic wikis are a novel technology and (not surprisingly) users do require some training by those responsible for the configuration of a semantic wiki instance, like the users who contribute to the domain ontology. On the other hand, even non-technical users can make use of semantic features like content tagging with minimal training. In summary, it seems quite simple to learn how to use a semantic wiki, perhaps due to the collaborative nature of this technology. Whether the authors employ an available semantic feature seems to be a question of whether they are aware of the benefits of applying such features.

The possibility to rename content items (cf. Section 2) was especially helpful in situations in which the content has evolved over time, as it was the case in the SemDAV and in the PoWiki scenario. A resulting requirement that has been noted several times is that links to content items should automatically be updated when content items are renamed, a feature we have not yet implemented.

5   FUTURE WORK

The Ylvi semantic wiki is under continuous development. At the moment, we are pursuing numerous lines of research and development that are inspired by our experiences with our productive semantic wiki instances. First, we are investigating means by which semantic tagging and linking can be made simpler for the user, as a present drawback of the system is that it is not immediately clear to an author which terms are available within the ontology. One approach here is to provide suggestions or auto-completion during user input; another is to include some form of graphical ontology browser in the interface. We also hope to improve the search interface for typed links – the present interface is confusing to many users and may be one reason why links are tagged far more seldom than articles.

Second, we plan to extend the dynamic ontology building tool (OntoYlvi) with features like versioning and locking mechanisms. At present, individual concepts are automatically versioned (because they are also articles) and can also be locked through the access control mechanism, but this is not sufficient (see Fischer et al., 2006). We are also working on user interface enhancements and plan to develop import and export functionality from and to standardized ontology formats like OWL[6].

Finally, we wish to extend the whole semantic annotation approach to include segmented multimedia. At this time, it is possible to annotate images and image areas in Ylvi; we intend to extend this feature to include documents (Word, PDF), audio, and video.

In addition to these Ylvi developments, we are already planning and implementing new scenarios in the application areas of archaeology and medical history that will make use of these new features, in particular media annotations.

Acknowledgments

References

Bruns, A. and Humphreys, S. (2005). Wikis in Teaching and Assessment: the M/Cyclopedia Project. In WikiSym '05: Proceedings of the 2005 International Symposium on Wikis, San Diego, California pp. 25-32, ACM New York, NY, USA. DOI: 10.1145/1104973.1104976

---

[6] http://www.w3.org/2004/OWL

Buffa M. and Gandon F. (2006). SweetWiki: Semantic Web Enabled Technologies in Wiki. In WikiSym '06: Proceedings of the 2006 International Symposium on Wikis, Odense, Denmark, pp. 69-78, ACM New York, NY, USA. DOI: 10.1145/1149453.1149469

Chau, T. and Maurer, F. (2005). A Case Study of Wiki-based Experience Repository at a Medium-sized Software Company. In K-CAP '05: Proceedings of the 3rd International Conference on Knowledge Capture, Banff, Alberta, Canada, pp. 185-186, ACM New York, NY, USA. DOI: 10.1145/1088622.1088660

Echarte, F. and Astrain, J. J. and Córdoba, A. and Villadangos, J. (2007). Ontology of Folksonomy: A New Modeling Method. In Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop (SAAKM2007) located at the 4th International Conference on Knowledge Capture (KCap 2007), Whistler, British Columbia, Canada

Elrufaie, E. and Turner, D. A. (2005). A Wiki Paradigm for Use in IT Courses, ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, Las Vegas, pp. 770-771, IEEE Computer Society, Washington, DC, USA. DOI: 10.1109/ITCC.2005.54

Fischer, J. and Gantner, Z. and Rendle, S. and Stritt, M. and Schmidt-Thieme, L. (2006). Ideas and Improvements for Semantic Wikis. In The Semantic Web: Research and Applications, Vol. 4011/2006 of Lecture Notes in Computer Science, pp. 650-663, Springer Berlin/Heidelberg, DOI: 10.1007/11762256

Gruber, T. (2005). Folksonomy of Ontology: A Mash-up of Apples and Oranges. First On-Line Conference on Metadata and Semantics Research (MTSR'05). http://mtsr.sigsemis.org/ (2005).

Hepp, M. and Bachlechner, D. and Siorpaes K. (2006). OntoWiki: Community-driven Ontology Engineering and Ontology Usage Based on Wikis. In WikiSym '06: Proceedings of the 2006 International Symposium on Wikis, Odense, Denmark, pp. 143-144, ACM New York, NY, USA. DOI: 10.1145/1149453.1149487

King, R. and Popitsch, N. and Westermann, U. (2007). METIS: A Flexible Foundation for the Unified Management of Multimedia Assets. In Multimedia Tools and Applications, pp. 325-349, Springer Netherlands. DOI: 10.1007/11042-007-0101-3

Lund, A. and Smørdal, O. (2006). Is There a Space for the Teacher in a WIKI? In WikiSym '06: Proceedings of the 2006 International Symposium on Wikis, Odense, Denmark, pp. 37-46, ACM New York, NY, USA. DOI: 10.1145/1149453.1149466

Majchrzak, A. and Wagner, C. and Yates, D. (2006). Corporate Wiki Users: Results of a Survey. In WikiSym '06: Proceedings of the 2006 International Symposium on Wikis, Odense, Denmark, pp. 99-104, ACM New York, NY, USA. DOI: 10.1145/1149453.1149472

Moses, T., ed. (2005). eXtensible Access Control Markup Language (XACML), Version 2.0, OASIS Standard, Feb 2005; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

Popitsch, N. and Schandl, B. and Amiri, A. and Leitich, S. and Jochum, W. (2006). Ylvi — Multimedia-izing the Semantic Wiki. In SemWiki '06: Proceedings of the 1st International Workshop on Semantic Wikis, Budva, Montenegro, ESWC2006

Stenmark, D. (2005). Knowledge Sharing on a Corporate Intranet: Effects of Re-Instating Web Authoring Capability. Proceedings of ECIS 2005, Regensburg, Germany, May 2005

Völkel, M. and Krötzsch, M. and Vrandecic, D. and Haller, H. and Studer, R. (2006). Semantic Wikipedia. In WWW '06: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, pp. 585-594, ACM New York, NY, USA. DOI: 10.1145/1135777.1135863

Wang, H.-C., Lu, C.-H., Yang, J.-Y., Hu, H.-W., Chiou, G.-F., Chiang, Y.-T., et al. (2005). An Empirical Exploration of Using Wiki in an English as a Second Language Course. Paper presented at the Fifth IEEE International Conference on Advanced Learning Technologies, 2005 (ICALT 2005).

Wang, C.-M., and Turner, D. (2004). Extending the Wiki Paradigm for Use in the Classroom. Paper presented at the International Conference on Information Technology: Coding and Computing (ITCC'04), pp. 255ff., IEEE Computer Society, Washington, DC, USA