Ylvi - Multimedia-izing the Semantic Wiki

Niko Popitsch[1], Bernhard Schandl[2], Arash Amiri[1],
Stefan Leitich[2], and Wolfgang Jochum[2]

[1] Research Studio Digital Memory Engineering, Vienna, Austria**
{niko.popitsch,arash.amiri}@researchstudio.at
[2] University of Vienna, Department of Distributed and Multimedia Systems
{bernhard.schandl,stefan.leitich,wolfgang.jochum}@univie.ac.at

Abstract. Semantic and semi-structured wiki implementations, which extend traditional, purely string-based wikis by adding machine-processable metadata, suffer from a lack of support for media management. Currently, it is difficult to maintain semantically rich metadata for both wiki pages and associated media assets; media management functionalities are cumbersome or missing. With Ylvi, a semantic wiki based on the METIS multimedia framework, we combine the advantages of structured, type-/attribute-based media management and the open, relatively unstructured wiki approach. By representing wiki pages as METIS objects, we can apply sophisticated media management features to the wiki domain and provide an extensible, multimedia-enabled semantic wiki.

1 Introduction

Wikis facilitate simple, efficient, collaborative document creation and evolution. Based on our experience, we believe that wikis are a promising approach and will spread out to new application fields, such as corporate intranets, collaborative knowledge management systems, and e-learning scenarios. However, while traditional wikis (for example MediaWiki[3] or MoinMoinWiki[4]) are unable to semantically structure information, current developments in the field of semantically-enabled wikis [5, 1, 3] suffer from unsatisfactory support for the management of multimedia data, like videos, audio, or complex media presentations.

In this paper, we present Ylvi, a wiki implementation based on the METIS media management framework[2]. With Ylvi, we combine the collaborative properties of traditional wiki systems with strong semantic features that characterize a structured media management framework, and extend the wiki with typed articles and media instances, typed article links, and advanced query processing.

2 Semantic Features in Ylvi

Ylvi makes extensive use of the METIS framework as its underlying media object management layer. METIS is a middleware component for the rapid development

** This work was supported by the Austrian Federal Ministry of Economics and Labour.

[3] MediaWiki: http://www.mediawiki.org
[4] MoinMoinWiki: http://moinmoin.wikiwikiweb.de/

of multimedia applications with a focus on metadata processing. Its data model is comparable to RDF/RDF Schema and can be extended by complex data types (dynamically loaded Java classes) that are essential for the development of advanced multimedia applications. Media objects can be typed, thereby inheriting strongly typed attributes that can be used to describe the media. Semantic models (or *ontologies*) can be defined using an available Protégé[4] interface. METIS provides a plug-in framework for extending its core functionalities and semantics, including plug-in media-locators, data types, functions and predicates as well as sub-data models (so-called *semantic packs*, e.g. for meta data standards like Dublin Core, Exif[5], ...). A query engine can be used to search along multiple dimensions (metadata values, semantic types, media features), and the Apache Lucene[6] full-text engine was incorporated for indexing text-based content. METIS implements its own multi-channel publishing strategy—basically a complex pipeline of XSLT transformations—for media aggregation.

Ylvi treats both wiki articles and multimedia objects in a uniform way: Both are modeled as METIS media objects that can be typed and attributed and may participate in directed, typed links. An overview of the semantic features provided by Ylvi is depicted in Figure 1 and described in more detail below.

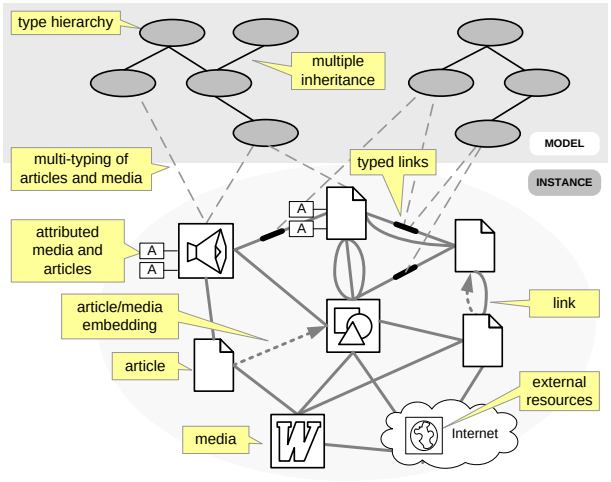

Fig. 1. Features of the Ylvi Semantic Wiki

Type Hierarchy – Ylvi is able to use ontologies (e.g. formulated in OWL) for the typing of articles and links. These can be imported by using the abovementioned Protégé interface.

[5] Exif: Exchangeable image file format for digital cameras
[6] Apache Lucene: `http://lucene.apache.org`

Multi-Typing – In Ylvi, each article/media object can be an instance of an arbitrary number of types that are defined in the ontology.

Attributed Media and Articles – Types in Ylvi are associated with a set of strongly typed attributes that can be defined using an expressive model (e.g. cardinality restrictions, default values, derived attributes). Each article/media object that is an instance of a type may define values for the respective types' attributes (e.g. picture dimensions, e-mail address, document author).

Typed Links – Traditional wikis support simple, purely navigational, unidirectional linking of wiki pages, and minimal inclusion of media objects (mostly images) into wiki pages. Ylvi allows the definition of multi-typed links between articles and media objects; consequently, Ylvi also supports the embedding of media objects or other articles in Ylvi articles.

In contrast to other semantic wikis, Ylvi does not only relate articles as a whole, but also retains the exact position of a link within the source code of an article. Multiple links between the same article pair are not collapsed, but are kept as multiple navigational and logical connections. This allows for example the enrichment of query results with excerpts from the articles in the result set, or ordering of result sets based on the links' textual context.

Typed Links to External Resources – As Ylvi does not distinguish between internal and external links, external resources (e.g. other wikis or web resources) can be integrated by using the same syntax, and these links can be typed and queried as well.

Sophisticated Synonym Handling – Traditional wikis use the page name as a unique identifier within the scope of one wiki instance. In this case, the wiki must rely on the manual definition of *disambiguation pages*[7], which are not machine-processable. Many semantic wikis, like IkeWiki[5], approach this problem by using an URI as an article identifier. However, this imposes two drawbacks: (*1*) in most cases, URIs are not intended for human consumption and are hard to read and to remember, and (*2*) the wiki has no mechanism to automatically detect ambiguous pages. In Ylvi, we use both a non-unique page name and an internal, auto-generated, unchanging page number as identifier. Users may link to a page using the page name (then, Ylvi automatically creates a disambiguation page), or may eliminate the ambiguity and link to a page using its internal page number.

Search – Ylvi implements a hybrid semantic search, enabling queries for articles and media objects along multiple semantic dimensions (full-text, types, attributes, links).

3 Enriching a Wiki User Interface with Semantic Features

Markup – Wiki content is usually expressed in a simple markup language that is easily adopted by non-technical users. Unfortunately, so far no standardized wiki

[7] see e.g. `http://en.wikipedia.org/wiki/Wikipedia:Disambiguation`

markup language exists[8]. As it was our intention to develop an open and flexible system, Ylvi does not implement a particular markup language but rather provides the possibility to configure rendering pipelines consisting of plug-in components that convert markup elements into rendering directives for the chosen output channel (e.g. HTML or LaTeX). The single exception to this is the markup for link definition and semantic annotations. For the definition of (typed) links, Ylvi uses a MediaWiki-like syntax; for typing and attribute intstantiation, two new syntax elements are introduced (see Fig. 2).

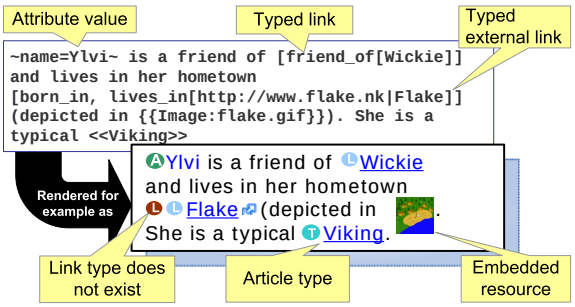

Fig. 2. Ylvi syntax and rendering example

Semantic annotations of Ylvi articles are expressed by using the shown markup elements (and some variations) – no special user interface is required for this. This has several advantages:

- A single, coherent input paradigm is used for content and annotations.
- Semantic annotations are part of an article's content and therefore benefit from all functionality applicable to text-based content in a wiki (versioning, diff, merging, quick copy/paste, ...).
- Semantic annotations remain in the article's source code even if the corresponding model elements (e.g. types, attributes) are removed. This makes sense as these annotations may still represent useful metadata of the article and may be automatically reused if the corresponding elements are added to the system again.

Rendering – To display an Ylvi article, its source code is passed through the rendering pipeline. The plug-ins interpret the markup (including the semantic annotations) of the article, transform it to a suitable output format (e.g. HTML) and enrich it with additional information that is relevant to the user.

[8] although there are already ongoing initiatives, see
`http://tikiwiki.org/tiki-index.php?page=RFCWiki` and
`http://www.usemod.com/cgi-bin/mb.pl?WikiMarkupStandard`

For instance, one may introduce a plugin that converts a designated markup of GPS coordinates into a list of articles annotated with nearby coordinates.

Ontology Manipulation – As described above, we define wiki markup only for semantic annotation, not for the definition and manipulation of semantic concepts (i.e. *ontology editing*). We do this because ontology development is a non-trivial problem and requires sophisticated user interfaces and tool support. Although Ylvi provides simple online ontology editing features (e.g. adding a new type to the ontology), we heavily rely on the available METIS extension for Protégé, which allows the user to transfer Protégé models into METIS, and makes the structures defined therein available for Ylvi.

4 Conclusion

In this paper, we have presented Ylvi, a semantic wiki that has extended functionality for media management. We described how we have merged the traditional wiki approach of collaborative content creation, the extended functionalities of semantic wikis (typed links, typed articles, attributes), and the power of a sophisticated media management framework. We demonstrated how we allow the user to integrate semantic markup directly into a wiki page and how we can use these context-conserving annotations to improve search results. With Ylvi, we have realized a wiki implementation that combines elaborate semantic features with an open, extensible architecture.

In the future, we intend to extend Ylvi's functionality by introducing user management and more accurate markup-based annotation and querying features, and we will extend Ylvi's semantic features by allowing links to be attributed, which supports the expression of more meaningful relations between articles. We consider Ylvi as a suitable framework for the creation of semantic, media-centric (intranet) applications and will continue to develop it into a solid knowledge management and exchange platform.

References

1. D. Aumueller. Towards a semantic wiki experience - desktop integration and interactivity in WikSAR. In *Proceedings of the ISWC 2005 Workshop on The Semantic Desktop*. Galway, Ireland, November 6 2005.
2. R. King, N. Popitsch, and G.-U. Westermann. METIS - A Flexible Database Foundation for the Unified Management of Multimedia Content. In *Proceedings of the 10th International Workshop on Multimedia Information Systems (MIS 2004)*, 2004.
3. M. Krötzsch, D. Vrandecic, and M. Völkel. Wikipedia and the semantic web - the missing links. In *Proceedings of Wikimania 2005*. Wikimedia Foundation, July 2005.
4. N.F. Noy, M. Sintek, S. Decker, et al. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems*, 16(2), 2001.
5. S. Schaffert, A. Gruber, and R. Westenthaler. A Semantic Wiki for Collaborative Knowledge Formation. *Semantics 2005, Vienna, Austria*, 2005.