

Using Attention and Context Information for Annotations in a Semantic Wiki

Malte Kiesel, Sven Schwarz, Ludger van Elst, and Georg Buscher

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Trippstadter Straße 122, 67663 Kaiserslautern, Germany

{firstname.lastname}@dfki.de

Abstract. For document-centric work, meta-information in form of annotations has proven useful to enhance search and other retrieval tasks. Since creating annotations manually is a lot of work, it is desirable to also tap less obtrusive sources of meta-information such as the user's context (projects the user is working on, currently relevant topics, etc.) and attention information (what text passages did the user read?). The Mymory project uses the semantic wiki *Kaukolu* that allows storing attention and context information in addition to standard semantic wiki metadata. Attention annotations are generated automatically using an eyetracker. All types of annotations get enriched with contextual information gathered by a context elicitation component. In this paper, an overview of the Mymory system is presented.

Shortened version. Please see <http://www.dfki.de/mymory> for the full paper.

1 Annotations in document-centered Work

Generally speaking, an *annotation* “is extra information asserted with a particular point in a document or other piece of information”¹, and as such, it is a widely used element in document-centered work: We put general comments (“This is similar to xy!”) or ratings (“Important!”) of text passages into annotations or annotate text with imperative statements (“Verify!”). Thus, annotating is a means to individualize and personalize documents which, for the rest, might be alike for a group of information consumers. *Formal* annotations as they are commonly applied in the Semantic Web context, in contrast, contain mainly *meta-data*, i. e., they primarily aim at *machine*-understandability of a document's content in order to enable automated information services (see [7, 2] for overviews of the role of annotation in document-centric Knowledge Management and the Semantic Web). For personal knowledge management, in addition to annotations that formalize document semantics, personal annotations along with attention and user context information are needed. Mymory addresses this by supporting

¹ <http://en.wikipedia.org/w/index.php?title=Annotation&oldid=175839314>

additional types of annotations: **Conceptual annotations** are used to classify (Web 2.0 speak: “tag”) document passages. Mymory relies on a Personal Information Model (Ontology) (PIMO), i.e., the user(s)’s conceptualization of his/their (knowledge work) world. Instead of tagging passages with simple text labels, PIMO concepts are used to annotate and classify the passages. **Attention annotations** are used to store how much attention the user invested for which parts of the document. Mymory uses several user observation techniques to gather attentional evidences. For example, an eye tracking device is used to recognize which passages the user really *reads*, which ones he has *skimmed over*, and which ones the user did not seem to have viewed at all. **Highlightings and comments** are comparatively simple annotations. However, *context-sensitive* management (i.e., context-sensitive storage and retrieval) of such annotations conveys a context-sensitive view of the document as the user (himself) had in the past when he created these annotations. It allows a quick “flashback” and reminds him of his past understanding and usage of that document. **Contextualized annotations** are annotations containing meta-information about the user’s context at the time of the creation of the annotation. This allows context-sensitive views of a document and is a way to enable scalable, massive usage of annotations in a multi-user/multi-context scenario as is typical in shared document repositories.

2 Overview over Mymory Components

The **PIMO: The Personal Information Model (Ontology)** [3] provides a vocabulary for describing information elements on an individual desktop, thereby comprehensively reflecting a user’s personal view on his information landscape. The Mymory PIMO offers rather general concepts of knowledge work (Person, Organization, Location, Document, etc.) and allows for extensions of these upper models with more specific group or personal concepts (e. g., concrete project types or organizational structures). It provides the basic vocabulary for annotating concrete information elements and for describing a user’s context.

Automatic User Context Capturing — Mymory aims at transforming the knowledge worker’s workplace into a *context-sensitive* document-centric work environment. Automatic user observation generates a continuous stream of contextual evidences which are then fed into a context elicitation framework². That way, the user’s context is captured without disturbing the user. A detailed description of the user observation and context elicitation framework is beyond the scope of this paper. See [4–6] for an overview of modeling, using, and accessing user context for knowledge management scenarios. The **User Observation Hub**³

² Technically, this is achieved by observing the usage of a number of applications such as the web browser and editor using plug-ins, keeping track of events within these applications, and matching these events and the content displayed in the applications against a number of rules that are partially autogenerated from the user’s PIMO.

³ <http://usercontext.opendfki.de/wiki/UserObservationHub>

is an open-source (Java) project responsible for the gathering and distribution/forwarding of user observation data.

Our scenario assumes that the knowledge worker's world is conceptualized and modeled using the PIMO. Mymory's context elicitation automatically gathers evidences to estimate for every PIMO concept a degree of attention/relevance with respect to the user's current context. Thus, the context as elicited and processed by the system consists of a map associating such an attention level with every PIMO concept. As the user works and his behavior continuously adapts during his current task, the attention levels of the concepts adapt accordingly.

The Mymory Workbench AKA Kaukolu Wiki — The wiki component of Mymory is implemented by Kaukolu, a semantic wiki research prototype. Kaukolu is an extension of JSPWiki⁴. Its new frontend features are implemented using Dojo⁵, a JavaScript framework that allows highly interactive user interfaces.

Typically, semantic wikis associate wiki pages with semantic resources, and allow links between pages to get typed, possibly according to some ontology known to the wiki. However, while this approach is elegant in terms of simplicity and ease of use, there are several drawbacks:

- The rigid mapping between wiki pages and semantic resources imposes severe limits on the possible use cases. Mapping complex ontologies to the wiki or creating proper instances for these ontologies in a wiki is as difficult as capturing the knowledge present in a large wiki page (which would correspond to a resource with hundreds of properties).
- Handling of existing documents, be it existing wiki pages or other documents, is difficult. Metadata has to be added into the page text, changing the actual document. For texts such as law documents or finalized versions of documents this might not be desirable.
- Handling of further information concerning annotations such as provenance or context information is difficult. Personal annotations are not supported.

For Mymory another way of creating annotations has been implemented. Annotations can get created for any text part⁶—in this regard, annotations in Kaukolu are similar to annotations or *notes* created in a standard word processing application. These are displayed in connection with the text they are associated with but do not show up as text characters or markup in neither editing nor viewing mode unless requested by the user.

An example can be seen in figure 1 that depicts a software license text and its annotations. This text contains lots of separate mentions of certain facts—after all, a license is a collection of legal statements. Expressing semantics of this type of document is impossible using the page-resource mapping technique. Only

⁴ <http://www.jspwiki.org/>

⁵ <http://dojotoolkit.org/>

⁶ Technically, these annotations are implemented by creating an *AnnotationAnchor* for each RDF annotation, associating the RDF resource of the annotation with a part of the wiki markup by means of storing character offsets of the annotated text. Offsets are updated on markup edits by a modified text diff algorithm.

license1	General Public License 2.0	
<i>label "GPL 2.0"</i>	Preamble	preamble1
<i>has_preamble</i>	...	
preamble1		
<i>terms_cond</i>	Terms and Conditions	
definition1,	0. ...The "Program", below, refers to any...	definition1
permission1	1. You may copy and distribute...	permission1
activity1		<i>activity</i> activity1
requirement1	provided that you...keep intact all notices...	<i>requirement</i> requirement1

Fig. 1. Annotating a software license in Kaukolu.

annotating a license with a few rather generic information fragments would be possible, de facto using an ontology with one or few classes and lots of properties, and creating one large instance.

Using the text annotation approach, fine-grained annotation is possible. Both text decomposition and assigning complex fact representations to individual text fragments can be done. Since a tight connection between annotations (or more precisely, the information contained therein) and the text exists, text passages concerning or expressing certain facts can be retrieved quite easily.

3 Features of the Mymory Workbench

Creating Annotations in Kaukolu: In Mymory, annotations are created in two different ways: Users may manually create annotations, and an eyetracker component can automatically mark read or skimmed passages, keeping track of the user context in the course⁷. From user perspective, creating manual annotations in Kaukolu is pretty straightforward. Once a text part to be annotated is selected, right-clicking opens an annotation window where possible annotation types are displayed. These types and corresponding dialogs are fetched from ontologies loaded in Kaukolu's RDF repository or, if configured accordingly, also from external sources using a custom implementation.

Using Annotations in Search: All data and metadata found in the system can be queried in Kaukolu's advanced search feature. Advanced search is implemented as a faceted search paradigm: The user can select one or more restrictions on text and annotation characteristics graphically ("Search for paragraphs I read yesterday in context of project Firestart"). Search always returns wiki text paragraphs as results; searching for standalone RDF resources or authors directly is not supported. This was done to keep the system simple and to keep some resemblance with a normal wiki search in which users expect text passages to be returned.

⁷ Information on the algorithms used for reading detection using the eyetracker can be found in [1]. Note that the eyetracker used is built-in in the display and does not require wearing glasses or other equipment.

Creating Documents based on Search Results: Text passages found in search can be used to create new documents. The idea here is that this way it is possible to “remix” texts to form documents that fit to new requirements. Passages are copied, not referenced—however, provenance information indicating the source of copied paragraphs is kept as a new annotation.

Using Annotations for Personalized Views: Kaukolu also supports filtering a page’s display based on annotations during normal page view. While many possible implementations of this can be imagined, we currently implemented two major ways of filtering. One can filter wiki pages by attention information—in practice, this means that passages bearing a *read* or *skimmed* annotation are displayed as usual while all other passages are grayed out. This helps finding (formerly) relevant passages in large documents. Filtering by context information is implemented differently: As context information is meta-information about annotations, we filter the annotations to display in the wiki page. For example, one can choose to display only annotations created in the context of a certain project or topic.

Acknowledgement

The Mymory project is funded by the Bundesministerium für Bildung und Forschung under grant 01 IW F01 and by the IST Programme of the European Union under grant FP6-027705.

References

1. Georg Buscher. Attention-based information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (doctoral consortium)*, 2007.
2. Siegfried Handschuh. *Creating Ontology-based Metadata by Annotation for the Semantic Web*. PhD thesis.
3. Leopold Sauermaun, Andreas Dengel, Ludger van Elst, Andreas Lauer, Heiko Maus, and Sven Schwarz. Personalization in the epos project. In M. Bouzid and N. Henze, editors, *Proceedings of the International Workshop on Semantic Web Personalization, Budva, Montenegro, June 12, 2006*, pages 42–52, 2006.
4. Sven Schwarz. A context model for personal knowledge management applications. In Th. Roth-Berghofer, St. Schulz, and D. B. Leake, editors, *Modeling and Retrieval of Context, Second International Workshop, MRC 2005, Edinburgh, UK*, volume 3946 of *Lecture Notes in Computer Science*, pages 18–33. Springer, 2006.
5. Sven Schwarz and Thomas Roth-Berghofer. Towards goal elicitation by user observation. In A. Hotho and G. Stumme, editors, *Proceedings of the LLWA 2003*, pages 224–228, Karlsruhe, oct 2003. AIFB Karlsruhe, GI.
6. Roza Shkundina and Sven Schwarz. A similarity measure for task contexts. In *Proceedings of the Workshop Similarities - Processes - Workflows in conjunction with the 6th International Conference on Case-Based Reasoning*, 2005.
7. Victoria S. Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4(1):14–28, 2006.