

# Contextualized Knowledge Acquisition in a Personal Semantic Wiki

Ludger van Elst, Malte Kiesel, Sven Schwarz,  
Georg Buscher, Andreas Lauer, Andreas Dengel

Knowledge Management Department  
German Research Center for Artificial Intelligence (DFKI) GmbH,  
Kaiserslautern, Germany  
`{firstname.surname}@dfki.de`

**Abstract.** The use of semantic representations in document-oriented environments – as formal annotations or embedded instances of a formal knowledge base – is seen as an enabling technology for intelligent services which may help knowledge workers in tasks like finding, structuring, or assessing information. Also, a high level of formalization has potential to directly support problem solving, e.g., by the application of inferencing services. A coupling of textual and formal representations in document-centered knowledge work raises, amongst others, two questions: i) How can the *acquisition of formal knowledge in such an environment* be facilitated? ii) How can the potential *complexity of formal annotations* during a document’s life and use cycle be adequately handled?

We present the Mymory workbench as an approach to investigate and tackle these challenges. Mymory is based on a semantic wiki system and supports manual as well as automated annotations of wiki documents. These annotations can be framed by automatically obtained models of the user’s work context, establishing situation-oriented structuring of annotations which can be exploited in semantic search and adapted document presentation.

## 1 Motivation: Document-centered Knowledge Work

Knowledge work is often to a large extent document-centered work: Knowledge workers receive and process documents which contain information that is related to their tasks, experiences, attitudes, or expectations. They assess the relevance of entire documents or, more often, parts of them to extract information, incorporate it into their personal body of knowledge, or solve problems at hand. These documents may be related to persons, things, events, topics, *etc.* in the realm of the knowledge worker. Such concepts are part of the knowledge worker’s mental representation of her work domain and often already occur in her digital environment, *e. g.*, their documents, as names of file or email folders, *etc.*

However, the exploitation of relations between documents in personal information webs is still quite restricted as they are i) rarely explicated during document generation, ii) hardly ever captured during document consumption, iii) to a large extent context-dependent.

The use of semantic representations in document-oriented environments – as formal annotations or embedded instances of a formal knowledge base – is a technical means to make the relations between texts and a knowledge worker’s conceptual world more explicit. Thereby, they can be seen as an enabling technology for intelligent services which may help the knowledge worker in tasks like finding, structuring, or assessing information. A very high level of formalization may even more directly support problem solving, e.g., by the application of inferencing services.

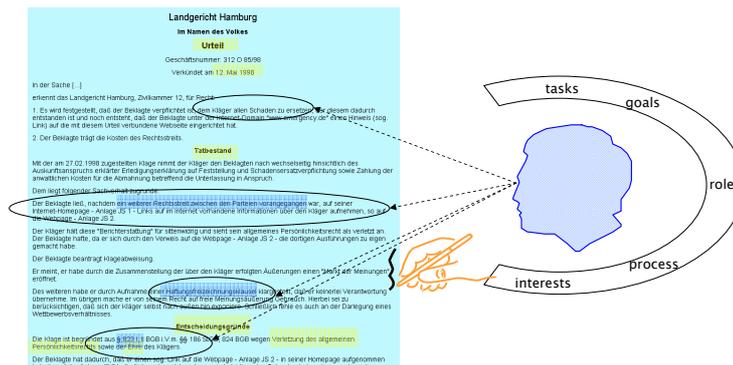


Fig. 1. Ontological and Subjective Annotations in a Verdict Document.

Figure 1 shows an exemplary document together with potential annotations and other meta-data. Such meta-data might refer to clearly ontological concepts (e.g., a statement that the document indeed is a verdict document and that the verdict was pronounced in May 1998) or about its structure (e.g., that paragraph 2 describes the case facts while paragraph 3 describes the decisive factors). However, in processing the document, a lawyer might construct much more relations, for example between this old verdict and her current case. Though she will often not make these relations explicit in a formal sense, her interaction with the document – e.g., manual notes in the margin or marks with a highlighter – might be a good indicator which of the many possible relations are actually relevant to her in her current work context.

An extensive coupling of textual and formal representations in document-centered knowledge work raises, amongst others, two questions:

1. How can the *acquisition of formal knowledge* in such an environment be facilitated?
2. How can the potential *complexity of formal annotations* during a document’s life and use cycle be adequately handled?

In the Mymory project<sup>1</sup>, our central assumption for tackling these questions is that *information about a user’s attention and interaction with documents* may

<sup>1</sup> <http://www.dfki.de/mymory>

be a good guideline to find out which relations are worth being established and how they can be utilized for improved information handling. Therefore, the proposed Mymory architecture for document-centered knowledge work employs technologies for *unobtrusive user attention estimation* during document production and consumption in order to automate the process of creating meaningful, context-oriented relations between information items as far as possible. Attention and context information can also be used to generate situated views in order to structure the information space during retrieval or information browsing. In the remainder of the paper, we present the basic principles of this approach and show how it was integrated into the *Kaukolu* semantic wiki component<sup>2</sup>. The semantic wiki provided us with the technological basis for semantically enriched text documents while integrating document consumption and document generation as well as collaboration facilities. We think that the resulting workbench is a powerful tool for continuous acquisition of formal, contextualized knowledge in document-centered knowledge work. A system walkthrough indicates some of the utilization potential of the acquired knowledge.

## 2 Contextual Annotations as Knowledge Acquisition

The motivation for classical knowledge-based systems is often that valuable knowledge is in the head of relatively few and potentially expensive experts; in a knowledge acquisition phase this knowledge is being formalized for automated use, typically by people who are not the experts from whom the knowledge was acquired. In *personal* knowledge management scenarios, the motivation is a bit different from that: First and foremost, the “expert” and the knowledge user are the same person, performing her regular job. This means that acquiring and generating knowledge might belong to the knowledge worker’s daily business, but processes for formalizing this knowledge are additional, second order activities, with no special budget and time dedicated to them. Therefore, the main motivation to perform such formalization activities will be a better subjective contentment with handling the personal knowledge space and an increased individual productivity. Only then, organizational goals like documentation, knowledge reuse by colleagues, *etc.* become relevant. This has a direct effect on technology that requires acquisition of formal knowledge in such settings as well as for the systems that try to support these acquisition steps: Only formalizations which are obtained at no additional cost or directly pay-off for the knowledge worker will actually be generated. The approach presented in this paper accommodates for these factors with the following design principles:

1. *Tight coupling between business work and knowledge acquisition:* As there is no extra time for a special knowledge acquisition phase available, the document-centered work itself is the trigger for formalization processes which are intertwined with text consumption or production. An additional advantage is that in this situation the respective part of knowledge is already highly activated in the knowledge worker’s mind.

---

<sup>2</sup> <http://kaukoluwiki.opendfki.de/>

2. *As many automation as possible, as unobtrusive as possible*: Obviously, the tight coupling of operative and acquisition processes has an immanent risk of disrupting the primary workflow. Also, the knowledge worker might be reluctant to spend additional formalization effort. Therefore, we aim at exploiting work which is done by the user anyway and complement it with (semi-)automated formalization techniques.
3. *Multiple levels of formalization, the user drives transitions between levels*: The level of formalization normally constrains the potential services that can exploit the knowledge; annotations with text tags only allows for simple retrieval algorithms, annotations with formal concepts may facilitate semantic search, and even higher levels of formalization make it possible to employ automated problem-solving based on reasoning services (see [9] for an application of problem-solving methods in a wiki). In addition to automatically and casually obtained knowledge we don't compel the user to reach a fixed level of formalization, but offer a wide spectrum, from plain text to formal instances.

In Mymory, coupling between business work and knowledge acquisition is achieved by using a personal wiki as the main workplace for generating, consuming, and manipulating documents. We thereby acknowledge that in many areas the means of producing documents has dramatically changed in the last few years, and we expect this trend to continue. While huge amounts of texts are still produced as stand-alone documents with standard word processing tools, more and more text is directly produced as hypertext with explicit links to other information elements, be it on the personal desktop or somewhere on the World Wide Web. By employing a *semantic wiki*, we allow for multiple levels of formality in the system, from plain text and tag-like annotations up to instances of an ontology.

Generally speaking, an *annotation* “is extra information asserted with a particular point in a document or other piece of information”<sup>3</sup>, and as such, it is a widely used element in document-centered work (see [7] for a broader discussion and study): 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. This individualization is typically quite situation-specific. Therefore, it is often hard to retrospectively analyze the meaning of such annotations, especially if they are non-verbal (*e. g.*, underlining or highlighting) and it is even more difficult to utilize these annotations for value-adding information services (see [14]). *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 [16, 4] for overviews of the role of annotation in document-centric Knowledge Management and the Semantic Web). For two

---

<sup>3</sup> <http://en.wikipedia.org/w/index.php?title=Annotation&oldid=175839314>

reasons, these meta-data predominantly represent knowledge which holds relatively context-independently: *i*) The technique is mainly applied in the open web scenario, *i. e.*, the things started formally should be true for all users in all situations. *ii*) The acquisition of this formal knowledge is quite expensive, so the effort only pays off if there is a high chance of re-usability. Consequently, this approach is most successful in domains like bibliographic or encyclopedic information which both aim at capturing rather stable knowledge (*e. g.*, the Semantic MediaWiki [6]).

In enterprise knowledge management, annotations have different characteristics than in those open scenarios: *a*) Annotations are often created by people *not* being the author of the document that gets annotated. *b*) Annotations often have a very subjective character—it is perfectly possible that an annotation created by person A is very helpful for person A, but does not make any sense to person B. *c*) Often, the nature of annotations being separated from the actual document is not only due to technical limitations (cf. marker pen and a paper document) but a necessity even if the document exists in editable (electronical) form. Consider the original document being a finalized document with the one reading it only wanting to scribble some personal notes on it.

In Mymory, we allow for several types of annotations: *Highlightings and comments* can be seen as perspective (*i. e.*, subjective and situated) interpretations of the text; *attentional annotations* also represent a user’s perspective on the text by storing how much attention the user invested for which parts of the document; *conceptual annotations* mainly aim at better retrieval. In contrast to traditional annotation techniques, Mymory uses *context-sensitive* annotations. This means all annotations contain meta-information about the user’s current context, that is, the context at the time of the creation of the annotation. This allows a context-sensitive view (markup) of a document and is an important means to enable a scalable, massive usage of annotations. Such contextualized annotations are also the enabling technology to allow context-sensitive understanding and sense-making of a document. Context-sensitive management (*i. e.*, context-sensitive storage and retrieval) of highlighting and commenting annotations convey a context-sensitive view of the document; it allows a quick “flashback” and reminds the user of his past understanding and usage of that document. Attentional annotations remind the user even more of his past understanding and usage of a document.

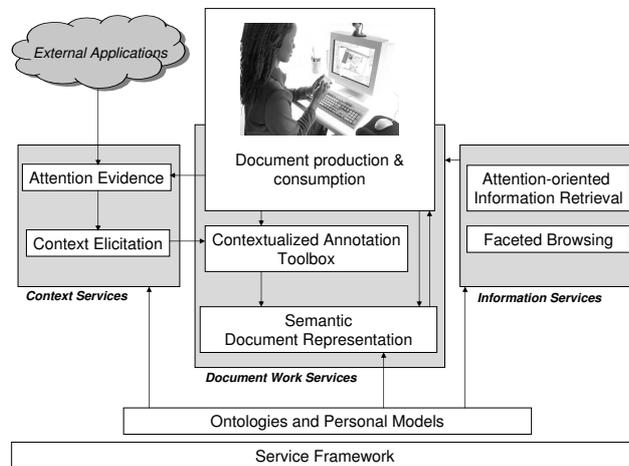
In the following section, we will present the main features of the Mymory workbench, especially with respect to the annotation representation facilities in the wiki system.

### 3 The Mymory Document Workbench

Figure 2 sketches the overall architecture of the Mymory workbench as a platform for document-oriented knowledge work. The workbench consists of three main elements which are realized on top of a light-weight service framework<sup>4</sup>: i)

<sup>4</sup> <http://servicia.opendfki.de>

*Document Work Services* support core activities like document production (writing, annotating, *etc.*) and document consumption (reading). ii) *Context Services* realize the acquisition of user context and attention information. iii) *Information Services* comprise tools which aim at supporting a user's current task with relevant information (*e. g.*, search). The vocabulary which is used by these services is provided by ontologies (modeling context and annotation types) and a Personal Information Model (PIMO) which represents the mental concepts used by the knowledge worker to get his work done. On a sound formal basis, a PIMO provides a vocabulary for describing information elements on an individual desktop (for example persons, projects, and locations), thereby comprehensively reflecting a user's personal view on his information landscape. It comprises relatively informal-tag or topic map-like-elements as well as more formal aspects with the expressivity of RDF/S. For a more detailed description of the PIMO, see [10].



**Fig. 2.** Building Blocks of the Mymory Workbench.

### 3.1 Document Work Services: Kaukolu Wiki

The core of the Mymory workbench is implemented by the semantic wiki system Kaukolu, a prototypical extension of JSPWiki<sup>5</sup>. While conceptually we use the wiki only as a representative for many typical document work environments, this is not just a (technical) simplification. We rather think that wikis already naturally integrate three main aspects of our concept, namely text consumption, text production, and open, flexible annotations. Building a *semantic* wiki [11] provided the necessary extensions for having informal and formal knowledge and annotations within one environment.

<sup>5</sup> <http://www.jspwiki.org/>

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

**Fig. 3.** Annotating a software license in Kaukolu.

In order to cope with the special requirements of Mymory—especially massive annotation and contextualization of annotations—the state-of-the-art approach for handling semantic annotations in semantic wikis had to be extended: 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 is not desirable.
- Handling of further information concerning annotations such as provenance or context information is difficult. Personal annotations are not supported.
- Context-sensitive annotations are not possible due to scalability issues: A materialization of a multitude of annotations in the wikitext will render the actual text unreadable.

While Kaukolu is able to embed semantic statements in its wiki markup [5], for Mymory an additional way of creating annotations has been implemented. Annotations can get created for any text part—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. 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.

An example can be seen in figure 3 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 standard page-resource mapping technique. Only 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.

### 3.2 Generation of Annotations

The Mymory system supports multiple annotation types which can represent knowledge at different levels of formalization, interpretable by humans or the system. These annotations are created in two different ways—users may *manually* create annotations or annotations are *automatically* being added to a wiki document. Such automated annotations contain either *context information* or *attention information*. The latter mark read or skimmed passages, keeping track of the user context in the course. The annotation types are represented in a simple, extensible annotation ontology. Currently, the wiki supports the following types of manual annotations:

- *Highlightings and comments* are typical annotation types known from the paper world. Comments can be free text or graphical annotations.
- *Ratings* based on a pre-defined rating ontology are an example for typed comments.
- *Provenance annotations* help in maintaining the history of origins of a document, thereby connecting a new text with its “parent documents”.
- *Conceptual annotations* are used to classify document passages. Mymory can build on a Personal Information Model (PIMO), *i. e.*, the user’s conceptualization of his (knowledge work) world [10], or on other imported RDF/S ontologies. Thereby, it is for example possible to express that some text on a wiki page is the name of a particular person stored in the PIMO, or that a text is of a certain document type (*e. g.*, a memo or a license document).

From a user’s 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.

Since typically the system knows many possible annotation classes, there are some shortcuts for potential interesting types, exploiting inherent annotations characteristics: i) If the text that is to be annotated lies within a text that has already been annotated, RDFS ranges of properties of the surrounding annotation instance are shown. This facilitates text decomposition: For example, if a

text segment is annotated as being a “collection of requirements” (with this class naturally having a *hasRequirement* property with *Requirement* as range), then this is taken as a hint to display *Requirements* (instances) more prominently. ii) Another way of limiting the class tree shown is looking at the history of recently used classes and showing the class hierarchy in their vicinity only.

**Attention Annotations** The idea of attention annotations is to store how much attention the user invested for which parts of the document. Mymory uses several user observation techniques to gather attentional evidences. In addition to simple techniques like the analysis of scrolling behaviour, an eye tracking device<sup>6</sup> 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.

Roughly spoken, eye movements are composed of fixations and saccades. During a fixation the eyes are steadily gazing at one point. A saccade is a quick movement from one fixation to the next. Since the sequence of fixations and saccades is very characteristic during reading behavior, it is possible to detect whether a person is currently reading or skimming a text. See [3] for details on the algorithm. Hence, when a person reads a document on the wiki, the read and skimmed text parts automatically get annotated appropriately.

### 3.3 Context Services for Embracing Annotations

The increasing generation of automated and manual annotations leads to the question of how this massive amount and complexity of annotations can be adequately handled during a document’s life and use cycle, not only from a technical viewpoint but also from a user’s perspective. Mymory’s answer to this question is to group these annotations by meaningful contexts. Therefore, Mymory aims at raising the knowledge worker’s workplace towards a *context-sensitive* (document-centric) work environment. Automatic user observation is applied to enable a continuous stream of contextual evidences, *i. e.*, mainly system interactions which might give a hint which context a user is currently dwelling on. These contextual evidence are then fed into a context elicitation framework in order to build and maintain a model of the user’s current work context. The **User Observation Hub**<sup>7</sup> is an open-source (Java) project responsible for the gathering of user observation data. Using this technology, the user’s context is captured without disturbing the user. A detailed description of the user observation and context elicitation framework are beyond the scope of this paper. See [12] for an overview of modeling, using, and accessing user context for knowledge management scenarios. Further information on eliciting context from user observations can be found in [13, 15].

As already stated, the knowledge worker’s world is conceptualized and modeled using a Personal Information Model (PIMO). During his work, the user

<sup>6</sup> We apply a Tobii 1750 desk-mounted eye tracker which has a data generation frequency of 50 Hz and an accuracy of around 40 pixel at a resolution of 1280x1024.

<sup>7</sup> <http://usercontext.opendfki.de/wiki/UserObservationHub>

puts more or less attention on the entities of his conceptualized world view. Mymory's user context elicitation automatically gathers evidences to estimate the degree of attention posed on every PIMO concept during the user's work. As the user's behavior continuously works and adapts towards the user's current task, the attention levels of the concepts adapt accordingly. Followingly, Mymory's user context model keeps track of this dynamic attention distribution. As the knowledge worker is typically working for multiple, different tasks in parallel, Mymory takes into account that the user is performing or contributing to *multiple* contexts, called *context threads*. However, concerning computer-free tasks as well as tasks done with the computer, the user can not handle these contexts simultaneously. We assume that human beings can only focus on but one single knowledge work task at once. Working on multiple tasks is conceptualized as if he continuously switches back and forth between them. The Mymory system switches between context threads accordingly.

For each context thread, Mymory keeps track of such an *attention distribution* of PIMO concepts. That means, that for each context, a different (dynamic) distribution is kept. Hence, attention for one context is not "polluted" by attentional behavior done in other contexts. Only one of these context threads can be active at a time. When the user switches to another context thread the attention distribution switches to the new context thread's distribution accordingly.

This model of the user's work context is used as contextual meta-information for annotations. Technically, this meta-information consists of two parts: i) the context thread that was active when the user created the annotation and ii) the PIMO concepts that were relevant when the created the annotation. The latter are taken directly from the context thread's attention distribution of PIMO concepts.

## 4 Utilizing Annotations in Mymory

An obvious way to use document annotations is for *re-contextualization purposes*. Often, one re-opens a document after some time in order to find exactly the same information as during the first time. Especially when the document has dozens or hundreds of pages, the time needed to find the specific piece of information can be annoyingly long. Since one often knows from the first time viewing the document that the information has to be in there somewhere, it would be helpful to see which document parts have been viewed before. So, attention annotations can be used as a document-internal filter. Such a filter would be especially useful for people like lawyers who often have many ongoing cases in parallel. Such cases often include a number of long documents where only a few paragraphs really matter to the case (context). When a lawyer switches from one case to a previous one, it might be very helpful to get a quick overview of the text parts that mattered before, i.e., that had been viewed in the past.

Another possibility is to distinguish different search modes: Does the user want to re-find information or to find new information? In the former case, the search strategy should ignore all not viewed text parts of the documents in the

search process. In the latter case, it should only consider not viewed or roughly skimmed text parts. Such a filter function would also be useful in a lawyer scenario: Some lawyers have very large collections of law comments on their computers (*i. e.*, comments on how the different laws should be interpreted). Such collections are normally used as reference books. Therefore, a search that can explicitly distinguish between searches for sighted and unsighted content could be very helpful. The former case could be useful when the lawyer wants to make sure a remembered fact. The latter case might be applied when she wants to obtain information (*e. g.*, a new argument) that she is not aware of.

The implementation of these features in the Mymory system can be found in Kaukolu's semantic search feature. 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.

In Mymory, three filters searching paragraphs that match selected criteria have been implemented so far:

- i) A *page filter* supporting standard wiki search (full text page content search, search by author, search by modification date).
- ii) An *annotation filter* searching for paragraphs with a matching Kaukolu annotation. Annotations can be filtered using facets derived directly from their RDF representation which is useful when handling annotations based on arbitrary ontologies. Filtering by the annotation's autor and creation date is possible, too.
- iii) A *context filter* implementing filtering by an annotation's context, either searching for a specific context, or an activated PIMO concept within the context.

Filters can be combined using AND and OR operators. Shortcuts that utilize substring matches when entering context concepts and annotation class names exist. E.g., for the page filter, the user can choose between page content (text), page author, and modification date. Then, a restriction value for this facet can be chosen. If the range of the facet is discrete, only values that exist in the wiki are displayed. For the annotation filter, facets and restriction values are derived from actual RDF annotations in the wiki.

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. Here, also provenance information is attached to the newly generated document.

In addition to the filter-oriented semantic search briefly presented above, we also implemented and evaluated the utilization of attention annotations for standard information retrieval technology. [1] shows a study on the use of eye tracking data for implicit relevance feedback. In [2], we present an evaluative study on query expansion using gaze-based feedback on the subdocument level. Details of these approaches are beyond the scope of this paper.

## 5 System Walkthrough

In the following, we will demonstrate the Mymory system using several examples. There are two big groups of annotations, respectively, their use cases:

*Context-Free Annotations:* The first use case uses annotations to classify text passages with a *domain-dependent annotation schema*. An example for this is the semantically enriched software license documents including license-specific annotations of text passages that allow fine-grained retrieval based on the semantics of texts. These annotations are created manually by an expert. As a consequence, such annotations are “correct”, precise, complex, not high in number, and typically context-free.

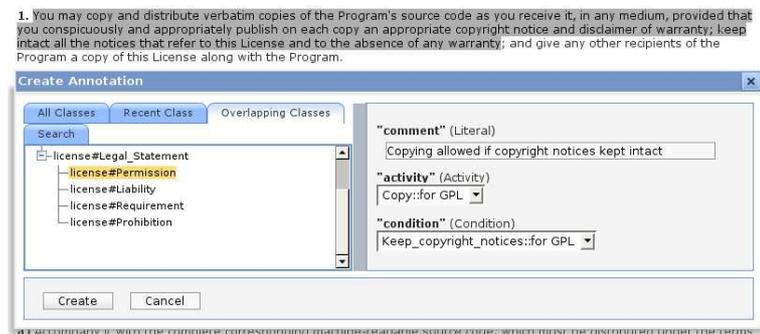
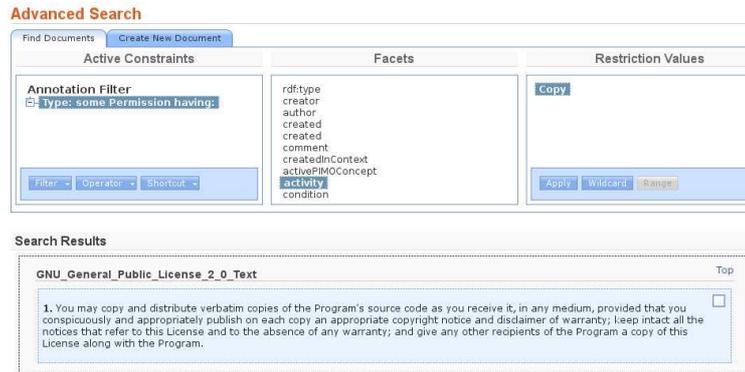


Fig. 4. Creating a complex annotation for a software license document.

As an example of Kaukolu’s annotation functionality, in figure 4 you see the software license text already known from figure ???. Most of the text is already annotated; the user is currently about to create an annotation describing the license term that permits the licensee to copy and distribute the software given that copyright notices are kept intact. When Kaukolu opens the annotation type chooser, it detects that an annotation surrounding the currently selected text—`license1`, instance of class `License`—has a `has_legal_statement` property whose range it uses to narrow down the set of annotation classes to display. `Permission` is a subclass of the `Legal_Statement` class and, thus, gets displayed.

Once texts have been annotated, detailed querying gets possible. Let us say the user wants to get an overview of all license text sections dealing with `Permissions` that concern copyright notices. In figure 5 the corresponding query is shown as created in Kaukolu’s advanced search feature. The user chose the annotation filter since paragraphs with a special semantic annotation are of interest. Any annotation of type `Permission` that points to a `KeepCopyrightNotes` instance is considered. Matching paragraphs are shown at the bottom of the search page at any time.



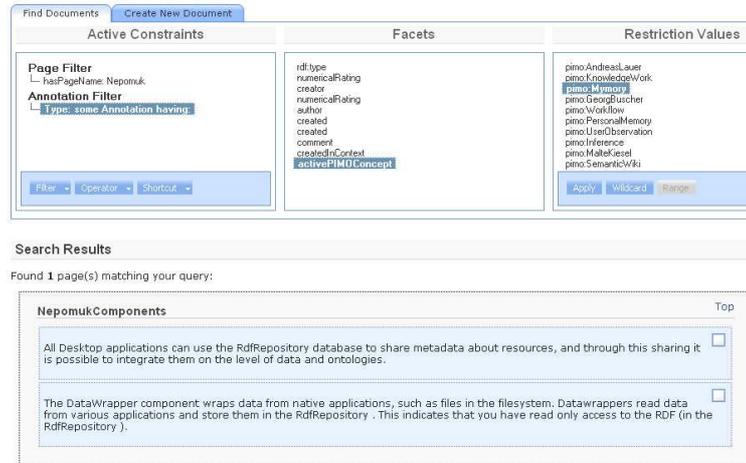
**Fig. 5.** Searching for annotated paragraphs.

If the user wants to, a digest of the paragraphs found can be created as a new page using the “Create new document” tab. In the new document, the link to paragraphs the document has been created from is kept using *Provenance* annotations for each paragraph.

*Personal Annotations:* The second class of annotation usages is to use *standard annotation types* like highlighting, rating, comments, margin bars, but also attentional annotations (user has really read a passage). These annotations do not provide very precise meta-information to be used directly for information retrieval. Besides, they typically occur in very high number and are highly context-sensitive, that is, they have been created in different contexts and their “meaning” only holds in their creation contexts. Even though these imprecise annotations cannot be used to answer *precise questions*, they can help to find information when the user remembers more the context of an information (text) than its concrete classification. For example, you can use these annotations to find passages that have been *important* (due to a rating annotation) for a certain software component (due to the contextual meta-data of the rating annotation).

Combinations of both annotation types (context-free + personal) enables sophisticated retrieval mechanisms. The example above can be extended towards the retrieval of all “licence documents” (license ontology and annotations) marked as “important” and relevant for “one specific software component” (a specific `pimo:Thing` in the context of the ranking annotation).

*Example 1:* Imagine Arthur, a member of the “Nepomuk” project, has been given the task to identify (happened) collaborations with the “Mymory” project. As both projects use the Kaukolu wiki, he decides to “ask” the wiki for text passages about Nepomuk with Mymory context. In other words, he searches the wiki using two filters: i) passage is part of a page with a name containing `Nepomuk`; ii) passage has an annotation with `activePimoConcept Mymory` in its context. Figure 6 shows two found text passages in a Nepomuk page with Mymory context.



**Fig. 6.** Searching for text passages in Nepomuk pages with Mymory context.

*Example 2:* The other day, Arthur searches for colleagues that he can discuss logical programming issues with. He searches the wiki for annotations of passages about logical programming. In other words, he searches the wiki using the following filter: passage classified as **about logical programming** or context of annotation contains **logical programming**. However, the result is used differently to example 1: He uses the annotations to get information about which persons have created them.

## 6 Summary and Outlook

In contrast to traditional knowledge-based systems where some expert's knowledge is formalized in order to support other (less trained) people's work, today's knowledge workers are often expert, knowledge engineer, and system user in one person. Technology for personal knowledge management (PKM) should support this type of work. With the Mymory workbench, we presented an exemplary PKM system for document-centered knowledge work. Figure 2 gives a comprehensive overview of our approach: A wiki system serves as a workbench for document generation and consumption. A semantic extension of this wiki allows for enriching the wiki text with formal annotations based on a personal information model and dynamically loaded ontologies. Annotations are created manually as well as automatically. Two types of automated annotations are possible: Context annotations, based on user observation providing a stream of contextual evidences, describe the knowledge workers situation with the vocabulary of his personal information model; attention annotations (automatically gathered from an eye tracker) differentiate between read, skimmed, and unread parts of a text.

The annotations can enhance the understanding and management of documents (currently restricted to wiki pages), especially when the documents are used by more than one person and for more than one context. Furthermore, annotated documents allow retrieval in use cases where a keyword-based search is not enough to find a document viewed a long time ago. Moreover, annotations allow additional filters to be applied to search, which can enhance retrieval precision—this holds particularly for contextual annotations, respectively contextual filtering. We have demonstrated how storing the user’s context as part of the annotations enables contextual search or filtering in retrieval scenarios.

The system is fully implemented and integrated into a physical demo workplace. However, there are still some technical drawbacks to be handled in the future: *Annotations* can be created within the wiki only (currently). While there exists an HTML importer and a scrapbook feature that allows quick import of other web content, it is certainly desirable to be able to annotate external websites and other document types such as PDFs without having to import them. Since this is a major technical effort that also raises many non-trivial issues (what if external annotated content changes?), we focused on on-system annotations for now. *Usability* needs to be improved. Still, far too many technical artifacts end up in the user interface. *Contextualization* will be extended. Currently, annotations get contextualized when being created. Many of the user’s interactions should leave context annotations automatically in the wiki though (clicking on an inter-page link *etc.*). Customizing rendering of documents according to the current context might also be worth looking into. From a conceptual point of view, also research concerning the (re-)use of annotations between contexts, *i. e.*, between users or for one user but at a different time or for a different task, has to be accomplished. [8] shows some work in this direction.

Currently, the system is being applied to two real-world use cases: Firstly, it replaces a standard wiki which has been in use for personal knowledge management in research for a couple of years now. In addition, the context-oriented document representations and presentations are applied in a service center scenario where situated views on documents have to be generated for different user groups. Here, we also gather experience with the acquisition of formal knowledge from and about the documents with offline use of a pen-based interface for manual annotation in the wiki.

**Acknowledgements** This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW F01, Project Mymory: Situated Documents in Personal Information Spaces) and by the European Union IST fund (Grant FP6-027705, Project NEPOMUK).

## References

1. G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2991–2996, New York, NY, USA, 2008. ACM.

2. G. Buscher, A. Dengel, and L. van Elst. Query expansion using gaze-based feedback on the subdocument level. In *Proceedings of the The 31st Annual International ACM SIGIR Conference 20-24 July 2008*, Singapore, to appear.
3. G. Buscher, A. Dengel, L. van Elst, and F. Mittag. Generating and using gaze-based document annotations. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3045–3050, New York, NY, USA, 2008. ACM.
4. S. Handschuh. *Creating Ontology-based Metadata by Annotation for the Semantic Web*. PhD thesis.
5. M. Kiesel. Kaukolu: Hub of the semantic corporate intranet. In *Proceedings of the 1st Workshop on Semantic Wikis in conjunction with the 3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 2006.
6. M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 935–942. Springer, 2006.
7. C. C. Marshall. Annotation: From paper books to the digital library. In *DL'97: Proceedings of the 2nd ACM International Conference on Digital Libraries*, Digital Scholarship, pages 131–140, 1997.
8. C. C. Marshall and A. J. B. Brush. Exploring the relationship between personal and public annotations. In H. Chen, H. D. Wactlar, C. chih Chen, E.-P. Lim, and M. G. Christel, editors, *JCDL*, pages 349–357. ACM, 2004.
9. J. Reutelshöfer, J. Baumeister, and F. Puppe. Ad-hoc knowledge engineering with semantic knowledge wikis. In C. Lange, S. Schaffert, H. Skaf-Molli, and M. Völkel, editors, *Proceedings of the 3rd Semantic Wiki Workshop (SemWiki 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, volume 360 of *CEUR-WS*, 2008.
10. L. Sauer mann, L. Elst, and A. Dengel. Pimo - a framework for representing personal information models. In T. Pellegrini and S. Schaffert, editors, *Proceedings of I-MEDIA '07 and I-SEMANTICS '07 International Conferences on New Media Technology and Semantic Systems as part of TRIPLE-I 2007*, J.UCS, pages 270–277. Know-Center, Austria, 2007.
11. S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel. Semantic wikis. *IEEE Software*, 25(4):8–11, Jul/Aug 2008.
12. S. Schwarz. A context model for personal knowledge management applications. In T. Roth-Berghofer, S. 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.
13. S. Schwarz and T. 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.
14. F. Shipman III, M. Price, C. C. Marshall, and G. Golovchinsky. Identifying useful passages in documents based on annotation patterns. In T. Koch and I. Sølvberg, editors, *ECDL*, volume 2769 of *Lecture Notes in Computer Science*, pages 101–112. Springer, 2003.
15. R. Shkundina and S. 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.
16. V. S. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. 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.