Enhancing a Semantic Wiki with Context Annotations

Scenario - A personal workspace for document-centered knowledge work.

Problem - Assisting the user in digesting, navigating, and searching his document repository relies on annotations. Manual annotation of documents is tedious work though. Also, the number of simple annotations (highlightings...) can become very high and render the system unusable.

Idea - Use attention and context information which can be gathered with almost no extra load for the user.

Approach - Extend the annotation features of a Semantic wiki to support attention annotations and contextualizing annotations. Extend browse and search facilities of the wiki to make use of these new annotation types.

Result - A system that helps the knowledge worker while searching and browsing the document repository, introducing only minimal overhead.

Context component
- Plug-ins for several applications gather usage information.
- Observed activities get matched to concepts stored in the user’s Personal Information Model which contains instances of Persons, Projects, Organizations, etc. of importance to the user.
- The activation value of the detected concepts in the current context vector gets adjusted accordingly – context switches get detected.
- Wiki annotations get contextualized by attaching the context vector at creation time.

Attention component
- Infrared LEDs and camera integrated in display.
- Calculates fixations with 50hz frequency.
- A number of processing steps digests this information and outputs reading/skimming events.
- This gets matched to the markup displayed in the browser window.
- Corresponding attention annotations get created in the wiki.

Wiki Annotations
- Each annotation is an instance of the RDFS class corresponding to the respective annotation.
- Each annotation is associated with (part of) the markup of a wiki page on a character basis.
- No change of the wiki markup is necessary for adding annotations.
- Each annotation carries context information (context active during annotation creation).

Advanced Search
- Combines normal wiki full-text search with semantic search (search by annotation).
- Works on a per-paragraph basis and returns a set of paragraphs matching the selected criteria.
- No query syntax – queries are constructed using a customized user interface borrowing techniques known from faceted browsing.
- Shortcuts for often-used searches exist.
- Search results can be used for creating new documents.
- Examples:
  - Find paragraphs I read yesterday while working on project X.
  - Find paragraphs dealing with copying restrictions in software licenses.
  - Find paragraphs I highlighted in pages also mentioning “java”.

Browse
- During normal browsing in the wiki, paragraphs can get hidden or grayed out depending on attention annotations.
- Other annotation types can get filtered by several criteria in order not to clutter the user interface.
- Do not display attention annotations.
- Only display own annotations.
- Only display annotations made in context of a certain PIMO concept.
- UI: Annotation markers can get aggregated in the sidebar of the document, highlighting the annotated text only on mouseover.
- Overlapping annotations are supported.

Examples
- Find paragraphs I read yesterday while working on project X.
- Find paragraphs dealing with copying restrictions in software licenses.
- Find paragraphs I highlighted in pages also mentioning “java”.

Table Preview
<table>
<thead>
<tr>
<th>Annotation</th>
<th>Category</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>java</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Licensing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copyrights</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JavaBeans</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>