Wiki-Templates

Adding Structure Support to Wikis On Demand

Anja Haake, Stephan Lukosch, and Till Schümmer
FernUniversität in Hagen
Computer Science Department
Universitätsstraße 1, 58084 Hagen, Germany
{anja.haake, stephan.lukosch, till.schuemmer}@fernuni-hagen.de

ABSTRACT

This paper introduces the concept of wiki templates that allows end-users to determine the structure and appearance of a wiki page. In particular, this better supports editing of structured wiki pages. Wiki templates may be adapted (defined and redefined) by end-users. They may be applied if found helpful, but need not to be used, thus maintaining the simple wiki editing way. In addition, we introduce a methodology to reuse wiki templates among different wiki instances. We show how wiki templates have been successfully used in real-world applications in our CURE wiki engine.

Keywords

Wiki, template, tailoring, structural editing and viewing

1. INTRODUCTION

Since their introduction by Ward Cunningham in March 1995 [13], wikis achieved sustained success. Wikis are applied in many application domains and many wiki engines have been developed since then [21].

Part of the Wiki success is based on their total freedom, ease of access and lack of structure [17]. Wikis serve as a means for quickly expressing ideas and share information. The easy way to link pages (e.g., by simple framing with characters or by using CamelCase page names) allows the user to create a hypertext *on-the-fly*.

A wiki can therefore be considered as a tool for thought [16] as it helps users to express and share their ideas. The user becomes, as envisioned in legendary MEMEX study on hypertext by Vanavar Bush [3], an active participant in recepting, processing, commenting, and creating information. Giving the power to express their thoughts back to the users is the most significant feature of wikis, as in contrast - for example - to the plain WWW where only the owners of the information can modify it [11].

Humans do structure their thoughts and information, e.g., by relating it to comparable information or dividing the information in classes. These classes are often implicit but users can benefit from means to make them explicit. In educational contexts, the teacher can, e.g., provide students with questions that they should answer in order to understand a specific topic. These questions help to shed light on the subject from different perspectives and break the subject down into manageable chunks of knowledge. As long as the students have not internalized the procedure of understanding the topic, the questions provide valuable support for the student. After having learned the procedure, they will start to make creative use it, following those aspects that are appropriate for the current situation and adapting other parts that do not perfectly fit the current context.

Examples of subjects or practices that can be learned this way are

- literature research, where students have to read and understand a piece of research literature,
- exam preparation, where students learn to state questions and provide and memorize answers to these questions,
- project planning, as it is, e.g., done in the eXtreme Programming planning game [1], or
- pattern writing, where practitioners learn to capture their best practices in a way that makes the problem, the requirements addressed by the problem, and the solution explicit.

Adding additional structure to wikis can help in such cases since it helps the author to state and answer the right questions in order to understand and communicate a specific subject. However, there has to be a good balance between proposed structure and potential freedom in expressing oneself. Two strategies complement one another in this respect: (1) Users should be free to combine less structured parts with structured parts according to their needs using gradually increasing structured documents (that we call structure gradients), and (2) users should be able to tailor the structure to their needs.

Examples where the need for structured representation of content in wikis becomes obvious include (1) more and more complex wiki syntaxes, e.g., as means to express text-boxes as in the WikiPedia [22], (2) provision of means to edit only parts of a wiki-page [4], or (3) the development of special purpose wikis such as the XPSwiki [15], offering built-in

WikiSym '05, October 16-18, San Diego, CA, U.S.A.

page-types with different text input boxes and specialized input fields to support the XP-Planning game. These three examples show the spectrum from free-form wiki pages up to highly structured wiki pages as they can be found in the XPSwiki. Combining both ends of the spectrum as needed is the rationale behind the demand for tailoring and structure gradients.

In this paper, we will compare the application of a traditional wiki with the use of a wiki offering templates as means for end-user defined structured pages on a conceptual level. Note, that our use of the term template is different to other uses that understand a template as a CSS layout mechanism. Instead, we consider a template as a description of a document's structure and its layout.

The wiki we are using in our research is the CURE wiki engine developed at our university to support interaction in collaborative learning. It offers additional support for group formation, group coordination, and group communication [8] to support users in their learning interaction. One of the most important goals of CURE is to enable users to tailor the collaboration environment to their needs while not overstraining them. CURE was intentionally developed to support computer supported cooperative learning (CSCL). However, due to the flexibility and power offered to the users, CURE has been also applied to many computer supported cooperative work (CSCW) tasks in other domains, for example, as a XP support tool during the continued development of CURE or for SIGs and working groups at the FernUniversität. Employing the CURE wiki engine without the support for structured pages for the past two years with more than 1300 users for different use cases, structured editing was frequently requested in various ways. We addressed these needs through so called wiki templates.

Whereas [8] discussed the needs for tailoring in CSCL in general and defines abstract levels of tailoring, this paper introduces wiki templates in a wiki-based collaboration environment. We begin in section 2 with a discussion of general needs for structure in wikis. Section 3 introduces the concept and use of wiki templates, section 4 compares our approach to related work. Finally, section 5 summarized the main contributions and lists issues for future work.

2. REQUIREMENTS

This section analyzes needs for structure support in wikis using real-world examples at the FernUniversität in Hagen: the exploitation of research literature during virtual seminars, and during students' first large research project, their diploma thesis.

We will base our considerations on a wiki with *minimal* functionality. This means that users can create interlinked pages consisting of a page title and a source text. The source text will be rendered using a wiki mark-up language with elements for headings, lists, and text emphasis.

The first approach to literature research was very unstructured. The teachers created a literature page containing a description of the task: students should search literature regarding a specific topic and summarize the found literature. The students were asked to append their results to the literature page using links to new pages if appropriate. The students however were not able to extract relevant information from the found literature. They also were not capable of referencing the literature in a way that contained all information needed to find the literature again.

```
Cite as: __ PUT THE AUTHOR HERE (PUT THE YEAR HERE).__ ~~ PUT THE
TITLE OF THE PAPER HERE.~~ In: PUT THE VOLUME HERE, PUT THE PLACE
HERE, PUT THE PAGES HERE. ~~(Online at: **:e:LINK TO EXTERNAL
URL**)~~

! Summary

!!Research Questions

!!Methods

!!Results

!Comments by other students or teachers
```

Figure 1: Seeding Page for Literature Summary in CURE

2.1 The Need for Structure

The deficits found in the unstructured literature research made the teachers think about providing students with additional guidance.

The educational dimension of understanding research literature has been addressed by others, for example Bergin proposed the Read, Read, Read pattern [2] to support Ph.D. students. It basically suggests that students should answer specific questions when reading a research paper. Thus, the pattern explicitly addresses the first problem observed in our context. Following the pattern's approach, the teachers suggested to their students to use two different kinds of pages: A Literature Summary page describing a paper similar to a note card with an analysis of the paper, and a Literature Pool page for each research project (seminar contribution or diploma thesis) collecting and categorizing all literature for this research project by linking to the respective note cards. Students could follow this guidance, but only few students really structured their work in the proposed way. The problem was that the students were not used to structure their content on their own.

In a next iteration, the teachers defined seeding pages for a Literature Summary (cf. figure 1) and for a Literature Pool. A seeding page intends to support structured editing through suggesting a certain structure of the page content (1) by proposing wiki tags such as headlines and formattings for certain input, (2) by using comments to suggest where to put certain information, or (3) by adding default links [13].

Figure 1 shows the source text of a seeding page for a Literature Summary that has been defined by our teachers. The seeding page uses framings by __, ~~ or ** to define bold text, text in italics, and links to other pages, respectively. Lines starting with exclamation marks define headlines of different levels. The seeding page starts with a proposal to correctly cite the respective paper, followed by sections headings that prompt the students to address aspects like (1) research questions dealt with in the paper, (2) methods applied in the paper to answer these questions, and (3) actual results obtained by the paper. Finally, a section for comments by other students and authors is provided. Students fill this seeding page with actual content (figure 2). After saving the content, the page will be rendered for display (figure 3).

Seeding pages could serve as a solution for the requirement of more structured wiki content. Unfortunately, the seeding page of figure 1 turned out as not being very helpful. Teachers complained that the students still did not stick

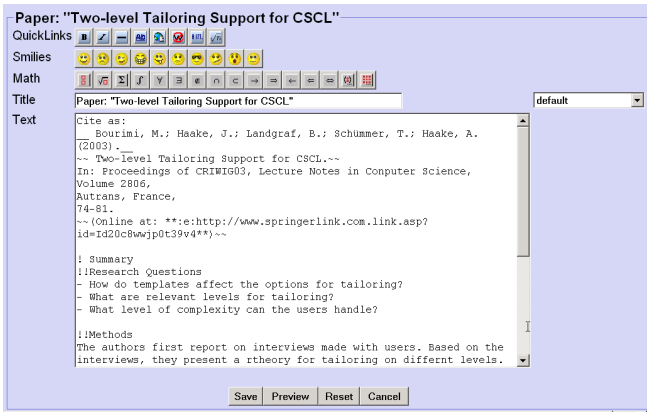

Figure 2: The Filled Literature Summary Seeding Page.

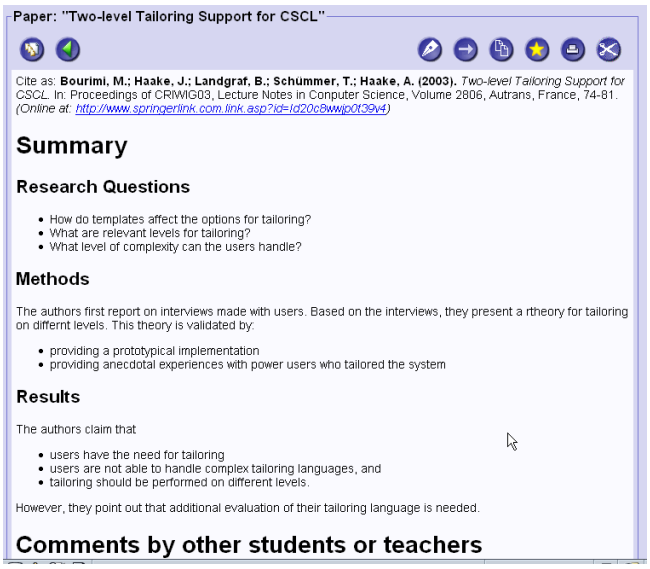

Figure 3: Rendered Version of the Literature Summary Page.

to the required structure of the document: Sticking to the proposed structure would have helped the students to internalize major steps of the process of literature research. By ignoring the structure, students neglected important aspects of literature research.

An analysis revealed four aspects in which users need additional support regarding structured content. They are (1) the readability of the page's source text, (2) the danger of unintentionally changing the structure of the seeding page and thereby loosing the provided help, (3) the need to change the provided structure where the context requires this, and (4) the need to benefit from other users' adaptations. The following sections will discuss each aspect in detail.

2.2 Maintaining the Readability

Most of the students complained that the formatting tags disturbed the readability of the bibliographic data during editing. They admit that the seeding page provides all necessary information and that, in principal, it is possible to use the page thus achieving a nicely rendered page (figure 3). But when editing the bibliographic data, they got confused about the mark-up. Therefore, they tend to ignore the citation rules for scientific papers. The mark-up made it difficult to concentrate on their tasks at hand. The simple formatting tags obviously already violated Leuf and Cunnigham's requirement, that mark-ups should be kept simple to enhance browsing readability without degrading editing readability.

One solution to avoid mark-ups like `~~` for italics or `__` for bold would be to provide extra wiki tags like `<paper-author>` or `<paper-place>`. The mark-up using additional wiki tags would be more meaningful to the author, but still has to stay in the source text. Introducing more and more wiki tags for different purposes would make the wiki-language more and more complex. Especially, learning the new mark-ups is required from all authors and parsing the new mark-ups needs to be implemented in the wiki engine, which makes it impossible for end-users to add new mark-ups.

Another approach would be to provide a special input form to supply the information. Instead of just providing two input fields, one for the title of the page and one for the source text, an input field could be provided for each structural element. In a Swiki, it is possible to associate input forms to pages [17]. Structure is not expressed in terms of mark-up but through graphical and functional design of the input page. However, to define these input forms in a Swiki, programming is required.

One prominent appropriation of a Swiki for a specific application domain is the XPSwiki [15]. It offers built-in page-types to support the XP-Planning Game, a process used during eXtreme Programming. One form, for example, "allows editing of free-format descriptions using text editor panes. Other attributes are edited using standard input fields and combo-boxes." As Pinna et al. state, the structure helps both, users "that are starting to learn the PG, since it obliges them to use the PG" and professionals since it "helps to keep track of project activities" [15].

2.3 Preventing Unintentional Changes of Structures

Next to the problems with respect to the bibliographical data, students partially ignored the structure of the three

questions to answer for each paper. They just left out one or two sections or even deleted sections they could not answer. Reusing such pages as a basis for an analysis of a new paper, they were not even aware of the need to address some (previously omitted) research questions. To support the process of literature research, teachers would have preferred having structural elements that could not be changed or unintentionally disturbed during editing.

Such a requirement contradicts with the general wiki philosophy of freedom. At least, the teachers themselves would require the right to change the structure of a literature summary for their purposes as needed, e.g. while improving their courses. Therefore, it would be desirable to allow changing the set of structural elements of a page at run time. Obviously, there is a tension between predefined, static sets of structural elements for students and flexible, adaptable sets for teachers.

DokuWiki [4] offers section editing that could be helpful for these problems. In a DokuWiki, each headline starts a new section. When a page is editable, a local edit button appears at the end of the section. Using this button loads only this section into the editor. The section editing supports structured editing, but would not prevent the unintentional destruction of page structures since at any time it is possible to edit the whole page source text by pressing the global edit button.

Another approach, at least for the section on "Comments from other students or authors" in our example (cf. figure 2 and 3), would be to allow appending text to the source text instead of editing the whole source text [13]. Appending content avoids the unintentional destruction of content and of structure and imposes a sequential order on the content contributions.

Again, specialized input forms would help, but only, if they are changeable on demand by the users.

2.4 Tailoring

One problem in the literature research example was that the provided seeding page was insufficient regarding the proposed structure. Some students wanted to reference to untraditional media such as video proceedings, television reports, or newspapers. Such media requires different treatment. Additional bibliographic data has to be added that was not anticipated by the teacher who initially defined the seeding page.

Students started to reflect on the task that they wanted to do. They tried to describe the media with the proposed seeding page, but encountered a breakdown in their action (on a theoretic basis, the students lived through a sequence of reflection in action [19]). The easiest way to react to this problem would be to use a blank page instead or simply changing the content of the seeding page (intentionally).

The better reflection and learning, however, would take place if the students would analyze the problem and create a solution for this problem. They could for instance create a new seeding page. This process of creating new structures for capturing knowledge can be considered as tailoring [20]. It complements the tailoring of the collaboration environment that we discussed in [10].

Another reason for tailoring of the Literature Summary structure has its roots in the different scientific backgrounds of the different departments at our university: psychologists had different citation conventions than computer scientists.

Finally, the tailoring of the interaction process in which the Literature Summary was used again required a tailoring of the seeding pages. In one instance, a section for comments from other students was added in order to support a discussion phase in the literature research process. Other teachers wanted their students to consider additional questions when evaluating a paper.

Unfortunately, seeding pages are not sufficient for supporting tailoring in the context of the process. Consider the case where students already created Literature Summaries. Now the teacher decides to enhance the process by adding a discussion field. This will, however, only affect the way how new Literature Summaries are created. Since the seeding page was copied when creating the Literature Summary, it will not be affected by later changes of the seeding page.

2.5 Sharing of Tailoring

Fernandez et al. [5] argued that tailoring of any collaborative environment should be considered as a group activity. Especially, the results of the tailorings should be shared in the group. In the case of changed seeding pages, this could be implemented by providing a wiki area in which these pages are collected. However, since seeding pages are just pages as any other wiki pages, it may be hard to get to know about seeding pages from other users. It is even more difficult to learn about useful seeding pages from other wiki instances [7]. Actually, one motivation to compile a catalog of CoWeb uses [7] was to foster the exchange among different wiki operators.

To group similar wiki pages, the WikiMedia wiki engine [14] introduces name spaces. Name spaces are in particular used to implement a catalog of so-called templates: Templates are used to add recurring messages to pages in a consistent way, to add boilerplate messages, to create navigational boxes, and to provide cross-language portability of texts. All templates in a WikiMedia instance should be put in the template name space. Users do so by preceding the page name with "Template:". The content of a template can be added to a page by typing the `{{templatename}}`. Note that WikiMedia has yet another understanding of the term template that should not be confused with our understanding of wiki templates that will be described in chapter 3.

In principal, name spaces could be used to set up a catalog of seeding pages in a wiki instance. To our knowledge, there is no work considering exchange of seeding pages among different wiki instances, except the catalog of CoWeb uses mentioned above.

2.6 Requirements

From the discussions above, we can conclude the following set of requirements:

R1: Structure. Users want to create well-structured content. However, it is often difficult for untrained users to create good structure without the guiding help of other users.

R2: Readability. Structure and presentation should be separated in order to help the author to concentrate on the content rather than struggling with layout aspects.

R3: Safety regarding unintended changes. The system should make the users aware of changes that they ap-

ply to the structure in order to prevent unintended departure from the proposed way of structuring content.

R4: Tailoring. Users should be free to use different content structures if they think that it is more appropriate. In cases where the users feel the need to create a new structure, they should be able to add this structure as a tailoring to the wiki in order to reuse it later on.

R5: Sharing of Tailorings. User-built structures should be shared with other users in order to help them solving comparable tasks.

3. WIKI TEMPLATES

Our solution to the requirements described in the previous section is the introduction of *wiki templates*. Wiki templates are means for describing both, the structure and appearance of a wiki page. They further determine the interaction elements used to edit and view a page. We have implemented and used wiki templates in our CURE wiki engine. After presenting a case for the application of wiki templates, we will discuss how the requirements were addressed by the different aspects of the templates, both from a technical level and from an end-user's perspective.

3.1 Wiki Templates Applied

A wiki template is a pair of an edit template and an associated display template. When creating a page, the author can select a template from the set of available templates and assign it to the newly created page. This will determine how the page is presented to the user and which fields can be filled by the author in the edit-view.

For supporting the example of the literature summary, the teachers created a LITERATURE SUMMARY template and added it to the set of available templates. From then on, authors could select this template for defining the behavior of their page. The LITERATURE SUMMARY template provided means to structure a page with ten different elements, each containing a source text.

The different fields are presented to the author in two sections (headlines "Bibliographic Data" and "Summary" in figure 4). In each section, the input field for each source text is labeled by a plain text positioned left to the respective input field using a table. The authors now can provide content into the respective input field. Thereby, they may still use the wiki language to control the rendering of the content of the respective source text. In figure 4, e.g., the author uses list elements to structure the content of some source texts (i.e., "Research Questions", "Research Method", and "Results").

The edit template that generated the form of figure 4 is shown in figure 5.

For now, it is not important to focus on the details of the template (this will be subject of the next sections). Two aspects are important from a user's point of view: (1) The user normally only selects which template to use. He will not need to understand the source text of the template if he only uses templates defined by other users. (2) The template definition language is a wiki derivate that is extended with special tags for defining fields. The required learning to move from being able to author a page to being able to define wiki templates is quite low.

Figure 4: Literature Summary supported by the Wiki Template (Edit View).

```
<box>
!!Bibliographic Data
|Author: | <wikiTextInput id="author" />|
|Title:  | <wikiTextInput id="/pageTitle" />|
|In:     | <wikiTextInput id="volume" />|
|Place:  | <wikiTextInput id="place" />|
|Year:   | <wikiTextInput id="year" />|
|Pages:  | <wikiTextInput id="pages" />|
|URL:    | <wikiTextInput id="url" />|

!!Summary
|Research Questions: | <wikiTextInput lines="8"
                        id="research Questions" />|
|Research Methods:   | <wikiTextInput lines="8"
                        id="research Methods" />|
|Results:            | <wikiTextInput lines="8"
                        id="research Results" />|

<submitButton /><cancelButton />
</box>
```

Figure 5: Edit Template for the Literature Summary

```
<pageToolBar />
<box>
<boxTitle>Details for the paper "<pageTitle />"</boxTitle>
<box>
<boxTitle>Cite as:</boxTitle>
<box style="border-width:0px; background-color:#ffffee;">
__<unrenderedText id="author" /> (<unrenderedText id="year" />).__
~~<pageTitle />.~~ In: <unrenderedText id="volume" />.
<unrenderedText id="place" />, <unrenderedText id="pages" />.
~~(Online at: <unrenderedText id="url" />)~~
</box>
</box>
<box>
<boxTitle>Summary</boxTitle>
!!Research Questions
<box style="border-width:0px; background-color:#ffffee;">
<renderedText id="research Questions" />
</box>

!!Methods
<box style="border-width:0px; background-color:#ffffee;">
<renderedText id="research Methods" />
</box>

!!Results
<box style="border-width:0px; background-color:#ffffee;">
<renderedText id="research Results" />
</box>
</box>

!!!Comments by other students or teachers
<box style="border-width:0px; background-color:#ffeeee;">
<renderedText id="comment" />
</box>
<wikiAppendText id="comment" lines="5" cols="90"
                rememberAuthor="yes" />

<div align="right"><editInformation /></div>
</box>
```

Figure 6: Display Template for the Literature Summary

For displaying the page, the display template part of the wiki template is used. It defines, how the fields will be rendered. Figure 6 shows the template code needed in order to define the rendering shown in figure 7.

Compared to the edit template in figure 5, the display template looks more complicated. The reason for this is that the author of the template wanted to use formattings that exceed the capabilities of the simple wiki mark-up syntax.

3.2 Defining Structure

We assume that users create wiki templates in order to represent structural patterns for organizing their content. They identify types (or entities) and model the attributes of this type. This is comparable to identifying classes and attributes in object-oriented analysis. Note that in contrast to typed programming languages, the attributes can store any textual content and are, therefore, untyped.

For each identified type, the users create a wiki template. The attributes are defined automatically by adding input fields to add data to the attribute.

The default type of the CURE wiki (and of all other wikis) is very simple: it provides one field for the page title and one for the source text (cf. figure 2). In general, a wiki template can define any number of attributes.

The introduction of wiki templates enhances the page model and the publishing model of wikis. In the traditional publishing model, an input field for the title and another in-

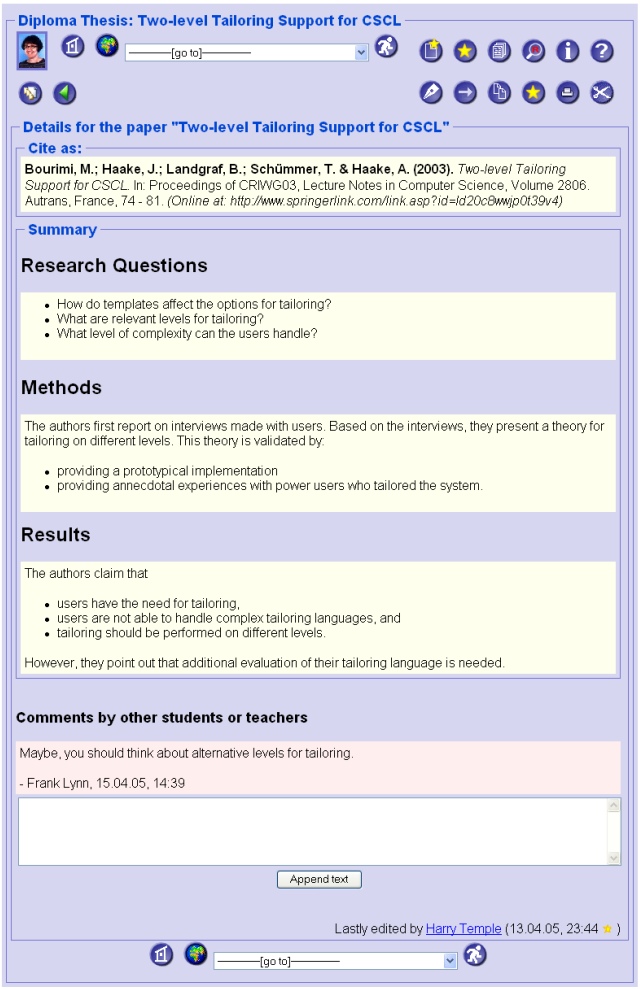

Figure 7: Literature Summary supported by the Wiki Template (Display View).

put field for the source text will be presented to the author.

$$Page := Title \times SourceText$$

The computing of the rendering of a page for reading will be controlled by (1) mark-ups and editing conventions (e.g. CamelCase) according to the wiki language that has been applied to the page content by the author and (2) a display template defining general presentation conventions for wiki pages.

However, such simple pages are not very expressive. They are instances of arbitrary objects that do not support any internal structure.

In the enhanced page model a page consists of a title and a set of identifiable source texts:

$$ExtendedPage := Title \times \{ID \rightarrow SourceText\}$$

The enhanced page model allows the template author to model any type having numerous attributes in the content set. On the implementation level, a page can be represented with a dictionary that maps plain text names to plain texts.

When defining a wiki template, designers have to consider carefully for which structural element/semantic unit of a page an extra attribute should be added to the wiki template. Two situations indicate the need for introducing new attributes: (1) users feel the need that an extra input field should be provided during authoring, and (2) users identify semantic units of a page that should be identifiable for other processing, e.g. for adding formatting style in the display template or for having the source texts addressable during search.

However, not all structural units of the page automatically lead to new attributes. If users, e.g., add headlines to their page content, they should still use the wiki language to mark a line of the content as a heading. The introduction of a new attribute only makes sense if more pages should share the same structure. In this case, the user would move the different sections of the page to attributes of the wiki template.

In the example of the Literature Summary, the template author identified four main sections to structure the summary page. (1) the bibliographic data, (2) research questions, (3) methods, and (4) results. In a first iteration, each of these sections was modeled using a wiki field. In order to allow better formatting and compliance of the citations, the first section was further structured using fields for author, volume, place, year, pages, and url. Note that the field referenced as `/pageTitle` is the standard title of a wiki page.

Fields are created automatically when they are used in a template. The most common way is to define a `<wikiTextInput id="fieldName"/>` tag. This will create an edit-box and store its content in the field specified in the parameter `id`.

Besides the text fields, we added two more types of content on the page model level: (1) Single links to pages or users (shown as selection lists in the edit view and as links to the respective object in the display view) that can be generated with the `<wikiSelectUser />` and `<wikiSelectPage />` tags (not shown in the examples). (2) Selection lists that allow the user to select a string from a predefined set of available options.

While the above means for defining structure controlled the structure of a single page, we also considered the relation between pages as an important issue in defining lager structural units containing more than one page.

Another attribute we implemented in our CURE wiki engine addresses the issue of link support: As mentioned in section 2.1, a Literature Pool should collect all Literature Summaries relevant for a certain topic. A literature pool page should thus have a field `links` that can be used to collect links to new Literature Summary pages. Whenever the user creates a new link in this field (using link framing, e.g., `**My Page**`), the system should automatically create pages related to the Literature Summary template. We support this by allowing an additional optional attribute `template` in the `wikiTextInput` tag, e.g.:

```
<wikiTextInput id="links" template="Literature Summary"/>
```

On creating a link to a new page in the field `links`, CURE will suggest that this new page conforms to the template Literature Summary. A user may deviate from this suggestion, but Literature Summary is supposed to be the default page type in order to support the process of understanding and evaluating research papers.

One could think about introducing further attributes for fields to express field types. For example, currently users may type in any text content into the field "year", not necessarily a correct numerical notation of a year. By allowing the association of regular expressions to fields controlling the text input patterns more type restrictions may be introduced. However, up to now requirements towards this directions have not been uttered by our users. Typing seems to be more important for modelling than for producing quality of content. Whether stronger typing of fields should be introduced or not into wiki templates is subject to future work and user studies.

3.3 Maintaining Readability

Fields of a page can be referenced in two ways: (1) view methods return the field's content either in a formatted or in a plain text representation. (2) manipulation methods return input fields in which the authors can specify content.

From a user's perspective, fields are modified in the edit view of the page defined with field tags discussed in section 3.2. They provide content that is rendered according to the display template. The display template often helps to enhance the readability of data provided by the user. Consider, e.g., the bibliographic data in the Literature Summary page. The user provides the information using different input fields. The content of these fields is formatted according to the first block of the display template in figure 6. A field may be referenced in the display template either as rendered text, i.e. the source text will be interpreted for display with respect to the wiki language (e.g. `<renderedText id="research Questions">`), or as unrendered text, i.e. the source text will be displayed without any further interpretation, (e.g. `<unrenderedText id="url">`).

In addition to these tags that reference fields, we have considered the following fields as useful for structuring the visual representation of a page:

- Standard sets of buttons that are frequently used to manipulate the page (e.g. `<pageToolBar/>` in CURE).
- Special tags that help to access meta information, e.g. `<linksToPage/>` in CURE for back links to this page or

`<editInformation/>` in CURE, displaying the author of the wiki page as well as change date and time for this page.

- Tags for standard design (e.g., `<box>` in figure 6). - At any time, using the wiki language may be useful.

Moreover, when needed raw XHTML may be used to define the presentation, e.g. the definitions in figure 6 use XHTML to define colors. However, as seen in the example templates, the application of the *normal* wiki syntax can be sufficient for most template definition tasks. E.g., the headlines in figure 6 have been formatted using the WIKI syntax (`!!Methods`) instead of a comparable HTML syntax (`<h2>Methods</h2>`).

The fact that most of the formatting code moved from the page source to the page's template improves the readability of the page's source.

One may argue that the introduction of WYSIWYG would even better support readability during editing. This may be an appropriate means to deal with format definitions such as bold etc. However, in general this approach contradicts the WikiWay and eventually breaks down: For example, when expressing links, the structure has to be made explicit and users have to consciously deal with it. WYSIWYG wikis like JotSpot try to unburden their users from these considerations by introducing linking automatism, e.g. when typing a page's title on any other page in a JotSpot wiki "a link will be automatically created - no HTML required" [12]. However, this may lead to unintentional and misleading links, i.e. the creation of unintended structure, and provoke solution such as careful naming of pages and name spaces. In our opinion, eventually authors have to deal with the structure of their content and hiding too much away is not helpful in the end. In addition, covering structure in formats is similar to expressing structure in seedings pages with respect to the aspect that structure is not separated from other forms of input. With such an approach, the danger to unintentionally change the structure grows.

3.4 Safety

Templates help to prevent unintended structural changes of the document. The authors are presented with input fields matching the intended structure. At time of authoring, they are only allowed to change the content. If they want to change the structure, they have to edit the wiki template. A user has to select the appropriate template from a list of available templates and open the template modification page (cf. figure 8).

The template modification page, however, only shows the structure of the page and does not include any content. The user will, thus, become aware of his different role – switching from content author to structure designer.

Finally, CURE is a wiki engine that wraps every wiki instance in a so-called room that defines general properties of the wiki instance such as the set of available templates or access rights for (groups of) users [9]. The rooms are organized with respect to spatial relationships and at any time users may create new wiki instances by adding new rooms to their room environment. Using the access control means of CURE, users may also define who is allowed to change templates for a given wiki instance, i.e. they can define that students or individual users are not allowed to change a certain template whereas teachers are.

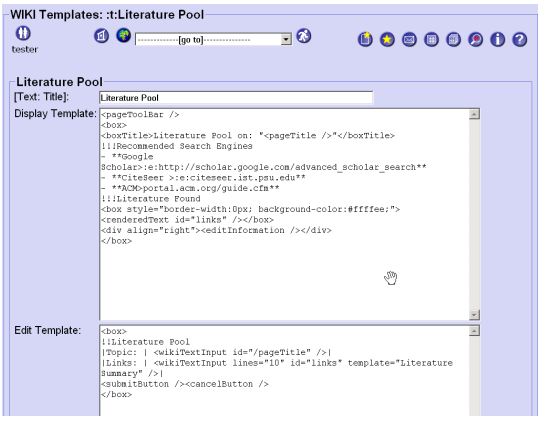

Figure 8: Editing the Wiki Template for the Literature Pool in CURE.

3.5 Tailoring

One problem with structured representation of information is that the structure may be too inflexible, especially in creative contexts where the problem domain is wicked [18] and not well understood.

We therefore argue that the authors should tailor the structure in parallel to the authoring process. Users should be encouraged to reflect on emerging structures (recurring structures in simple wiki pages) and capture these structures in a wiki template.

The pages created manually with the structure will then be used as starting points for defining the respective templates. Users take, e.g., recurring headings and move the text structured by the headings to fields of the new page type.

An important aspect when allowing the authors to change the template at runtime is template evolution. Existing page instances need to change their type at runtime while keeping their information. We assume a two phase transition: In a first step, the tailorer creates an intermediate template that includes all fields from the old and the new page template. Then, the type of the existing pages is changed from the old template to the intermediate template. The edit view will show all fields available in the old and the new template and authors can adapt the content to the new structure. Finally, the page type is changed to the new type with the effect that old fields are no longer shown.

In cases where the page type is only extended (e.g., by adding a new field), the intermediate step is not needed. In this case, old pages associated with the new type just have no content for the added field. If source texts are removed from the template, the content stays in all page instances that were created with the older version of the template. However, the content will no longer be shown if it was also removed in the display template. Again, if the content should be removed, no intermediate step is required.

3.6 Sharing Wiki Templates

As mentioned before, CURE wraps every wiki instance in a so-called room. All templates defined for a wiki instance are listed on the property page of the wiki instance. Sharing of templates takes place in two ways:

1. Templates are *inherited* between rooms. From a desig-

nated root room (the entrance hall of the CURE system), users collect templates from all rooms that they pass when moving to their destination room.

Consider a room structure used in a virtual university. The entrance hall provides university wide templates, e.g., a template for introducing oneself to other students or a template for creating a page to contribute to a car pool (for commuting students). From the entrance hall, students can move to the different faculties. The faculty for computer science may provide templates for diploma theses that include all information needed in this faculty. Finally, there may be rooms for lectures that provide domain specific templates. A lecture on software engineering processes may for instance provide templates for XP planning cards.

A student contributing information in the lecture room has access to all templates, i.e., the university wide templates, the faculty templates, and the domain specific templates of the lecture. The room owner however can restrict the default way of inheriting templates by blocking specific templates. The owner of the software engineering lecture room may, e.g., decide to block the car pool template and thereby not support car pool pages in his room.

2. Templates can be *copied* to other rooms. One room can, thus, serve as a template pool. Users looking for an appropriate structure for their information can browse existing templates in the template room and copy appropriate templates back to their own room. They can then modify the template and tailor it to their needs.

4. COMPARISON TO RELATED WORK

In this section, we will compare the concept of wiki templates to related work along the five requirements identified in section 2.

4.1 Structure

The main goal of wikis as web-based software is to enable all readers of web pages to also edit those pages, as in contrast to the plain WWW. Since editing of raw X(HT)ML is difficult to survey and cumbersome, wikis introduce a wiki language which controls the appearance of a page for reading. Thereby, the styling of the pages in display mode should be a side effect of structural mark-up done during editing [13].

This requirement is consistent with the general SGML [6] respectively XML publishing model. Note, HTML is an application of SGML respectively its successor XML. In this sense, wikis can be compared to other SGML/XML editing tools.

In contrast to SGML/XML, wiki languages do not require the definition of a domain specific DTD. Rather, a simple DTD for a page is defined, i.e. a page consists of a title and a source text. Within the source text, more mark-up may be applied, but this mark-up is not controlled by a user-defined DTD. The rules how to combine mark-ups are at best supported in the wiki engine and applied while computing the display of a page for reading. This gives a lot of flexibility and freedom to the users and is one factor for the success of wikis in general. Depending on the application domain or processes in which the wiki is employed, users are demanding structure as guidance. Wiki templates allow to introduce structure. The edit template of a wiki template can be compared with a DTD and the display template with a XSL.

Compared to SGML/XML, wiki templates allow the definition of more than one source text for a page. These source texts can be considered as simple SGML/XML elements. The ids given to the source text may be compared to element names in SGML/XML. The source texts are ordered sequentially as implied by their order of definition in the edit template. No nesting of elements as in SGML/XML is supported in a wiki by design. We assume that these structures are too complex to use by end-users. Moreover, the source texts defined by wiki tags are not rendered using a syntax editor common in pure SGML/XML applications. Syntax editors have been proven to be too confusing to use for non-computer experts. As for wiki templates, for each element an extra input field is displayed to the user. The display of the edit field may be enhanced using plain text, the wiki language or raw HTML as considered necessary. In this way, a simple structured editor looking like a form to fill out can be offered to the authors.

4.2 Readability

The edit template controlling the editing of the page shown in the figure 5 defines extra input fields for the parts of the bibliographic data. Compared to the use of the seeding page shown in figure 1, editing the bibliographic data is much more readable. Since the edit template defines identifiable source texts, those individual information elements may be presented to a user in a more usable and readable form. Actually, by defining an edit template a simple editor is configured that eases information input. In particular, the information about different structural elements does not disturb the editing of the source text but rather enables to generate a user interface making users aware of the information elements to provide during editing.

Since the structure elements defined in the edit template are named they may be referenced in the corresponding display template. Mark-ups in the source text usually provided to control the display of the page for viewing thus can be moved into the display template, reducing the mark-up in the source text.

Note that the definitions of display templates are usually longer and more extensive than the definitions of the edit template. The visualization of a page for viewing is important for information communication and therefore requires careful and extensive considerations and thus voluminous definitions.

4.3 Safety

The edit template controlling the editing of a literature summary shown in figure 5 defines extra input fields for all aspects of a literature summary. The page structure predefined in the edit template enables the representation of the structure at the user interface for editing. In particular, the pre-defined structure can not be destroyed, as it may happen unintentionally while editing a seeding page (cf. figure 1).

Configuring a simple editor through an edit template is the safest way to protect the structure. Compared to section

editing in a DokuWiki [4], an editing user may destroy the structure unintentionally by editing the whole source text. Moreover, offering an extra input field directly in contrast to offering edit buttons for sections prompts the editing user to think about all information units and not only some.

Looking at the literature research example, our teachers created templates that guide the students through the process of collecting relevant information about a paper. So, in contrast to the use of standard pages that allow free from input, the students now have to fill out specific fields that are relevant for their task.

4.4 Tailoring

Our approach to offer simple tailored editors for information input is consistent with the approach pursued by the XPSwiki [15], which provides specialized input forms to support the XP-Planning game. However, in order to define the input forms of the XPSwiki programming was required. Furthermore, programming will be required to adapt the XPSwiki to changing requirements. Obviously, different applications will require an unforeseeable amount of useful page structures. Therefore, we did not want to define the page types at the coding level. This approach seemed to be too complex for non-computer expert users, of which we have many at the FernUniversität in Hagen.

Wiki templates avoid the need for wiki users to program and allow them to create these page type definitions by themselves. Providing an edit template for editing wiki pages is just a transfer of the concept of display templates for displaying wiki pages that wiki users are already used to. Providing a template editor (cf. figure 8) makes changing templates an integrated, but well defined, task of wiki interaction. Providing the template editor as an extra tool makes defining and adapting structures a conscious process not mixed with content editing, like a single editable source text suggests. Nevertheless, the users may use the wiki syntax to add structure while editing an input field, i.e. they can add structure spontaneously during editing. However, structure can not be destroyed unintentional during content input, but has to be removed intentionally using the template editor.

Moreover, offering users different kinds of input elements, such as direct text input, appending text to a source text or selection lists informs template designers of different styles of information input that can be applied according to the information input needs. The opportunity to tailor the editable pages motivated users to reflect on their activities. Users created new domain specific templates. Users tailored on demand. This is however part of the structure evolution/tailoring idea. The tailoring users should fix a problem of use during use, which is consistent with the wiki philosophy of freedom.

4.5 Sharing of Tailoring

We made wiki templates a property of a wiki instance. In this way, there is a central well-defined place in the system where to look up defined templates. From here, the templates may be used, changed or reused to define more templates.

Name spaces for wikis collecting similar pages are a concept that is close to the idea of collecting all wiki templates as a property of a wiki instance. However, to assign a page in WikiMedia [14], the page title has to be preceded with the respective name of the name space, followed by a colon. This information has to be provided by the user. In contrast, the interaction with wiki templates is linked to the properties of a wiki instance itself. They are by definition associated to this place.

The SWiki wiki engine allows inheritance of wikis. They allow that different wikis share properties provided by the wiki from which the new wiki was created as a successor. However, there is no convenient way to share templates across hierarchies or to customize inherited page styles.

5. CONCLUSIONS AND FUTURE WORK

The questions "When or when not to impose structure?" and "How to define this structure?" have been addressed by Leuf and Cunningham in the first wiki days [13]. In this paper, we discussed the need for structure in wikis and identified five requirements, i.e. structure support, readability, safety, tailoring, and sharing of tailoring. We introduced wiki templates and showed how our wiki engine together with the wiki templates fulfills these requirements.

Wiki templates control the display of a wiki page for a reader and the display of a wiki page for an author. Wiki templates empower the wiki users to define the core structural elements of a wiki page as a set of named source texts. These elements may then be exploited for rendering the wiki page for both, editing and reading. In addition, within each source text a standard wiki language may be used to further control the content display and to add structure while editing. The introduction of wiki templates in principal separates content editing from structure editing making structure editing more conscious. It is central to wiki templates that they may be used by the end-users, but do not have to be used. They, thus, combine freedom and structure for authors. The least common denominator for all wiki pages is the core wiki page model consisting of a title and a single source text. Finally, we have shown how wiki templates may be shared among the users of a wiki instance and among different wiki instances hosted by the same wiki engine.

Future work will focus on additional potential applications of structured wiki pages. Teachers, e.g., complained that they could not benefit from the structure they designed for editing and viewing during search and page content evaluation. For example, when looking for papers on a research method, they wanted to query for pages where the research method appears in the section on "Research Questions". However, if they wanted papers where the method was applied, they wanted to query for pages mentioning the method in the section on "Methods". Using our wiki templates, it will be easy to customize the search function of our wiki engine to allow searching for content in a specific section of a wiki page.

5.1 Acknowledgments

We would like to thank the other members of the CURE development team, Mohamed Bourimi and Britta Landgraf, who contributed to the implementation of the presented approaches. Special thanks are due to Sven Laaks who implemented large parts of the template interpreter. Jörg Haake contributed to the discussion for this paper and challenged us to keep the structure focussed.

6. REFERENCES

[1] K. Beck. *eXtreme Programming Explained*. Addison Wesley, Reading, MA, USA, 1999.

[2] J. Bergin. Patterns for the doctoral student. PRELIMINARY– (Edition 3), July 2002.

[3] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108 (online at: http://www.isg.sfu.ca/ duchier/misc/vbush/vbush–all.shtml), 1945.

[4] Doku wiki. http://c2.com/cgi/wiki?DokuWiki, Mar. 2005.

[5] A. Fernandez, J. M. Haake, and A. Goldberg. Tailoring group work. In J. M. Haake and J. A. Pino, editors, *Proceedings of the CRIWG'2002 – 8th International Workshop on Groupware*, LNCS 2440, pages 232–242, La Serena, Chile, Sept. 2002. Springer-Verlag Berlin Heidelberg.

[6] C. F. Goldfarb. *The SGML Handbook*. Oxford University Press, 1991.

[7] M. Guzdial. A catalog of CoWeb uses. Technical Report GIT-GVU-00-19, Georgia Institute of Technology, Nov. 2000.

[8] J. Haake, T. Schümmer, A. Haake, M. Bourimi, and B. Landgraf. Supporting flexible collaborative distance learning in the cure platform. In *Proceedings of HICSS-37*. IEEE Press, 2004.

[9] J. M. Haake, A. Haake, T. Schümmer, M. Bourimi, and B. Landgraf. End-user controlled group formation and access rights management in a shared workspace system. In *CSCW'04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 554–563, Chicago, Illinois, USA, November 6-10 2004. ACM Press.

[10] J. M. Haake, T. Schümmer, A. Haake, M. Bourimi, and B. Landgraf. Two-level tailoring support for cscl. In J. Favela and D. Decouchant, editors, *Groupware: Design, Implementation, and Use. Proceedings of the 9th International Workshop (CRIWG 2003)*, volume 2806 of *Lecture Notes in Computer Science*, pages 74–82, Heidelberg, September/October 2003. Springer.

[11] A. D. Iorio and F. Vitali. A xanalogical collaborative editing environment. In *Proceedings of the second international workshop on Web document Analysis*, 2003.

[12] JotSpot, Inc. Jotspot intro tour. http://www.jot.com/tours/intro/3.php, 2005.

[13] B. Leuf and W. Cunningham. *The Wiki Way*. Addison Wesley, 2001.

[14] Media wiki. http://c2.com/cgi/wiki?MediaWiki, Apr. 2005.

[15] S. Pinna, S. Mauri, P. Lorrai, M. Marchesi, and N. Serra. XPSwiki: An agile tool supporting the planning game. In M. Marchesi and G. Succi, editors, *Extreme Programming and Agile Processes in Software Engineering, 4th International Conference, XP 2003*, number LNCS 2675, pages 104–113. Springer-Verlag Berlin Heidelberg, 2003.

[16] H. Reingold. *Tools for Thought – The History and Future of Mind-Expanding Technology*. MIT Press, 2000.

[17] J. Rick, M. Guzdial, K. Carroll, L. Holloway-Attaway, and B. Walker. Collaborative learning at low cost: CoWeb use in english composition. In *Proceedings of CSCL 2002*, Boulder, Colorado, USA, 2002.

[18] H. W. J. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4:155–169, 1973.

[19] D. A. Schön. *The Reflective Practictioner: How Professionals Think in Action*. Basic Books, New York, 1983.

[20] G. Teege. Users as composers: Parts and features as a basis for tailorability in cscw systems. *Computer Supported Cooperative Work (CSCW)*, 9:101–122, 2000.

[21] Wiki choicetree. http://c2.com/cgi/wiki?WikiChoicetree, May 2005.

[22] Wikipedia. http://www.wikipedia.org/, May 2005.