WikiPrints – Rendering Enterprise Wiki Content for Printing

Kathrin Berkner*

ABSTRACT

Wikis have become a tool of choice for collaborative, informative communication. In contrast to the immense Wikipedia, that serves as a reference web site and typically covers only one topic per web page, enterprise wikis are often used as project management tools and contain several closely related pages authored by members of one project. In that scenario it is useful to print closely related content for review or teaching purposes. In this paper we propose a novel technique for rendering enterprise wiki content for printing called WikiPrints, that creates a linearized version of wiki content formatted as a mixture between web layout and conventional document layout suitable for printing. Compared to existing print options for wiki content, WikiPrints automatically selects content from different wiki pages given user preferences and usage scenarios. Meta data such as content authors or time of content editing are considered. A preview of the linearized content is shown to the user and an interface for making manual formatting changes provided.

Keywords: Wiki, pagination, print formatting, content linearization, graph model, arborescence

1. INTRODUCTION

Wikis have become a tool of choice for collaborative, informative communication. In contrast to the immense Wikipedia that serves as a reference web site and typically covers only one topic per web page, enterprise wikis are often used as project management tools and contain several closely related pages authored by members of one project. The number of pages and authors are much smaller than in Wikipedia. In such project management scenario it is useful to print closely related content. For example, it may be valuable to review all project materials in their current state, or to export portions of the wiki to a pdf file for usage outside the work group or by auditors.

When today’s wikis allow for printing of their contents, the typical method is to provide a template for a page, e.g. via a cascaded style sheet, and force the user to print each wiki page individually, or to have the user manually select content of interest and then provide formatting for the collection selected. The first solution has the drawback that wiki pages, that link to the page currently being viewed, are not included in the formatted version, and that printing of parts of a page using the print template is not possible. The second solution requires a lot of manual work by the user. In both solutions, a lot of manual work is necessary to include content linked to the main page. Furthermore, meta data such as authors or time of creation are not considered during formatting.

In this paper we propose a novel technique for rendering enterprise wiki content called WikiPrints, that creates a linearized version of wiki content formatted as a mixture between web layout and conventional document layout suitable for printing. The details of the mix are determined by the usage scenario and user preferences. For example, a person who has no interaction with the online wiki may prefer traditional document formatting including a Table of Contents and section numbering. A contributor to the wiki, in contrast, may prefer having online navigation information included in the format. WikiPrint technology performs the following four steps. First, a graph model is built that contains all data needed to support automatic content selection in various usage scenarios. In a second step, content is transformed into a document tree. Finally, linearization of the content is performed via breadth-first walking of the tree nodes.

After the initial WikiPrint version had been developed, user feedback was collected from content contributors and content consumers. Based on that feedback a revised WikiPrint version was developed that included a user interface in a browser to enable the user to preview the formatted version, provide feedback, and make changes before printing.

* berkner@rii.ricoh.com; phone 1 650 496-5714; fax 1 650 854-8740; rii.ricoh.com
The paper describes details on the WikiPrint graph model, specifying vertices, edges, and edge weights in Section 3. Furthermore, criteria for pruning the graph into a tree will be presented and discussed. In Section 4, layout designs for different usage scenarios will be described. Section 5 reports user feedback and demonstrates examples of WikiPrints. The paper concludes in Section 6.

2. WIKI STRUCTURE AND PRINT CHALLENGES

Wikis have emerged as a powerful web 2.0 tool with a dominantly on-line presence. They are considered an editable repository of information, usually predominantly textual, that multiple users share, edit, annotate, and use. It has become clear that in large wikis it becomes very difficult to obtain an overall summary picture of the wiki content without doing some extensive analysis. The technology WikiNavMap tries to support such task by visualizing Tickets, Wiki pages, and Milestones in the Trac environment. A mixture of textual and visual representation of wiki content is also proposed in The Visual Wiki. Those solutions are on-line tools. They do not support printing of wiki material.

In contrast to those online navigation and visualization techniques, little effort has been spent on creating useful printouts from wiki content. Print solutions for material created via certain Web 2.0 tools have been in use for quite some time. The most popular printed Web 2.0 content are blogs. People create blog entries over some period of time and then want to publish them in printed form, e.g. as a book. Many popular blogging sites offer a blog printing service, such as BlogBooker for the blogs WordPress and Blogger. In contrast to blogs, that are typically created and maintained by a single user, may have other reader’s comments collected, and are structured in linear form in the time dimension, wikis have a more complex structure. Many users contribute, introduce cross references by hyperlinks, and edit different content pieces at various points in time.

Current practice today for printing of generic web pages is by use of a style sheet. A single web page is rendered with a style sheet and presented to the user as a print preview. Typical layout changes from the online version are removed navigation menus, print-friendly fonts, etc. Printing of a collection of web page segments is supported by HP’s Smart Web Printing software. With such a print solution only material from a single webpage is printed. Content that is connected to that webpage via hyperlinks is not included in the printout.

The company PediaPress provides a printing solution for wikipedia articles. The user selects the articles manually. The linearization of the content is performed by concatenating the selected pages in the order they have been selected or the order that the user has determined. Chapter groupings can be inserted, and references are collected in an appendix.

In the experiments reported in this paper the Dokuwiki enterprise wiki software was used. For this wiki software the only option for printing content from the wiki was to use the create a printout of a single wiki page via the included stylesheet. Such printing on a per-page basis is not a sufficient solution for some tasks that need to be performed with enterprise wiki content. In our experiment, the wiki was used to document a software transfer project. Managers had to review the project, and employees not working on the project wanted to learn about it by reading some kind of “documentation”. For those tasks that are related to consumption of wiki content and not just to creation of wiki content, the single-wiki-page print solutions were not effective. A paginated version of the entire wiki content was desired. Creating such complete version could only be created with much manual work by collecting pages and formatting them to printed form individually. Figure 1 shows an example page of the project wiki used in the experiments reported in this paper.
Our goal is to derive a printing framework for wikis that is suitable for the printing needs of project- and manual-type wikis. Task-dependent patterns for consumption of the wiki content should be taken into account as well as the multiple-author characteristics and non-linear organization in the time dimension that distinguish wikis from blogs. First a graph model is derived that supports linearization of content via tree extraction. Then different usage scenarios are investigated and task specific layouts for printing described that take the linearized content and render it in a form suitable for a given usage scenario.

3. WIKIPRINT GRAPH MODEL

A wiki can be interpreted as its own small internet universe within the big World Wide Web with pages containing links to other wiki content, but also links to web pages external to the wiki. In our experiments with enterprise wikis we observed that users often organize their content according to traditional logical document rules using sections and subsections with titles as well as paragraphs and lists. That means logical relationships are expressed through certain text formatting conventions, not necessarily through insertion of hyperlinks. In our experience, insertion of hyperlinks between wiki pages by users was not used very often as a method to link content. Figure 2 shows the example of the link structure of an enterprise project wiki visualizing only hyperlinks between wiki page content, but ignoring links included in the wiki’s navigation menu. A hierarchical tree-like structure is apparent. Only very few nodes have more than one incoming link. This phenomenon is very different from encyclopedia-type wikis such as wikipedia, where many more crosslinks are inserted and a tree-like structure is not visible\textsuperscript{7,8}.

Traditional document layout structure, however is also organized in a hierarchical tree-like way with chapters being followed by sections and subsections, etc. Formatting such hierarchically organized information for printing is a well known task in the document layout and pagination world. Our idea is to use the fact that the enterprise project wiki shows a hierarchically organized structure to model the entire wiki content as a tree structure that can then be formatted for printing using conventional layout rules.
To follow such an approach we need to answer two questions: (1) How do we transform the tree-like linkage structure of the wiki into a real tree, and (2): How do we merge web page linkage and traditional document layout structure into one common tree structure?

Our solution to the first question of transforming the wiki graph into a tree is to extract a subgraph that has certain properties related to minimal spanning trees. For a connected, weighted, undirected graph, the minimal spanning tree is a subtree that connects all vertices and has combined weight less than or equal to the weight of every other spanning tree. In the case of the wiki graph we have a directed graph. For a connected, weighted, directed graph, the equivalent to a minimal spanning tree is called an arborescence\(^6\). It is a subgraph that is a tree, connects all the vertices of the original graph together, and has minimum combined weights.

Up till now we have not assigned any weights to the wiki graph. Depending on the usage scenario, different weight distributions are assigned to the graph. Weights distributions may depend, for example, on the temporal attributes of the content (creation and alteration dates), author identities (who created the content), or semantic attributes such as association with a certain keyword or topic. Specific examples of weight distributions will be discussed in Section 4. Given a certain weight distribution, the graph representing the hyper-linkage between wiki content turns into a connected, weighted, directed graph whose arborescence can be extracted with, e.g., the Kleinberg-Tardos algorithm\(^9\). An arborescence extracted from the graph in Figure 2 is shown in Figure 3.

Our solution to the second question regarding the merging between the hyperlink structure and the document layout structure is to interpret the individual wiki pages as chapters, the page content as chapter content and to create a combined document tree that includes the chapters and chapter content. Since we have pruned the wiki graph into a tree already, we do not have to resolve any ambiguity of the content being linked to more than one chapter. The resulting tree represents hyperlinks and document layout structure of all the wiki content. An example of the tree created for the arborescence from Figure 3 is shown in Figure 4. Even though individual numbers are not readable in the graph in Figure 4, the main tree structure is apparent.

---

Figure 2. An example of hyperlink structure extracted from a project wiki. Nodes represent individual wiki pages. Edges between nodes represent links from one wiki page or wiki page content segment to another. Nodes circled by ellipses are nodes that have more than one incoming link.

Figure 3. The arborescence extracted from the wiki linkage graph in Figure 2. Each of the circled nodes has only one incoming link.
4. ADAPTATION TO USAGE SCENARIOS

In this section we analyze a few usage scenarios for consumption of wiki content that we observed in our research lab. A manager may want to get a quick overview of the status of the project, an active contributor to the wiki may want to review the recent changes, and a colleague not working on the project may want to get familiar with the entire project by reading a project documentation. The base content for all those tasks is the same, but the paginated format should look very different in each usage scenario. In the “quick review” scenario only main headings from each point may be important, not every detail written in paragraphs or material attached as pdf files. In the “review recent changes” only newly added or edited content needs to be formatted. And in the “learning the project” scenario the user may want to have the content available in traditional document form with a Table of Content (TOC) and an index.

Given those usage scenarios the weight distributions on the initial wiki graph may be assigned in specific ways. In the “quick review” scenario, links connecting content positioned close to the top of the page may have higher weights than links connecting to content further to the bottom of a page. In the “review recent changes” scenario, links created closer to the review date have higher weights than links created further back in time. And in the “learning” scenario it may be important to weigh a link to a content heavier if it was created earlier in time than a link to the same content created later in time, assuming that the later link was rather a refined than an important initial conceptual link.

According to those weight distributions an arborescence is created from the wiki graph and a complete document tree created. For the example aborescence in Figure 3, the weight distribution was chosen such that links created by one author who also authored the linked content are weighted higher than links connecting content from different authors. Reasoning for this choice of distribution is that one author created a main story that is refined by others and that the links connecting the main story are weighted heavier than the refinement links.

Given the document tree for the wiki content, a linearization for printing is performed by crawling the tree from the root to the leaves following the depth-first strategy. Given that the tree already has a structure similar to a classical document structure with chapters, section, subsections, etc, the depth-first strategy follows conventional layout rules for a typical document. For the “learning the project” usage scenario, a Table of Contents was created and numbering of sections and subsections computed.

5. EXPERIMENTS AND USER FEEDBACK

In a first experiment, the wiki document tree structure and the wiki content were converted into an XML document using a DOM model. Applying style-sheets, three different layouts of the same XML wiki document tree were created. Those layouts differ in the amount of original online wiki navigation information still contained in the linearized form. The content of the created WikiPrints that are discussed in the following, is in all examples extracted from the same project wiki that was used to document a software transfer project. That project did not involve much coding, but rather alignment of people resources, documentation of milestones, etc. In order to keep names of people and specific project details anonymus, we blurred some parts of the documents shown in the figures in this section. That blurring was introduced artificially and is not part of the WikiPrint technology.

In the first layout all online wiki navigation information is removed. Instead, a table of content is included at the beginning of the linearized document and section numbers are introduced. Whenever there was a hyperlink in the
original wiki linking different pieces of content together, but that link was removed during the extraction of the arborescence, the link is referenced by section number. That way the connection between the originally hyperlinked pieces of content is still maintained. Figure 5 shows the table of content and a referenced link.

Figure 5. First layout representing a traditional document style. The top part of the Table of Contents created for the linearized WikiPrint format (left) and a hyperlink replaced by a reference to a section number (right). The highlight in the right image is added only in this paper for better visibility.

In the second layout some online wiki navigation information is included in the linearized document by signaling insertion points for a particular wiki page through inclusion of the original ToC of the wiki page. Figure 5 shows an example of such a wiki-page ToC included in the linearized document. Also images attached to the wiki pages are rendered in the linearized form as shown in Figure 6 as well as thumbnails of the first page of attached pdf files.

Figure 6. Second layout: Original wiki-page tables of contents are included at the top of each section that originates from a wiki page. Images attached to the wiki are inserted in the linearized form (right).
The third layout includes the largest amount of online wiki navigation information. In addition, color codes are used to indicate a certain hierarchy level in the document tree. A table of wiki pages is added at the beginning of the document displaying the associated color codes used in the following. That table reflects the structure contained in the arborescence. Links that are present in the wiki graph, but not in the arborescence are displayed in differentiating formatting style. The ToCs of each wiki page are added to the left of the related wiki content. Figure 7 shows an example of this third layout.

![Table of Wiki Pages](image)

These different layouts were presented to five people who had contributed to the wiki content and feedback was collected. The comments demonstrated that additional control was necessary in order to allow individual preferences for viewing the linearized wiki content. One user liked the positioning of the ToC in the margin of the third layout, another user disagreed with the positioning since the ToC became too narrow and suggested shortening of the titles. Different opinions also existed on using color codes. For one user the color was distracting, for another one it was very helpful.

One feedback, that was uniform across users, was that too many levels in the tree hierarchy are confusing, leading to too deep subsection numbering, as seen in the main ToC in Figure 3. This feedback led to the creation of a revised version that included an upper bound on the depth of the document tree. In order to include the content a hierarchy level beyond that bound we decided to move that content to newly created appendices at the end of the main document. Figure 8 shows a document tree that has been constructed given a tree depth constraint. In contrast to the tree in Figure 4, the last branching on the left side has been moved to a newly created appendix forming a new node to the right side of the graph.

![A revised document tree](image)

Another topic raised in the user interviews addressed the issue of how to seamlessly navigate from the linearized version to the original online version and vice versa. Ideas to overcome this problem included adding barcodes to parts of the content, as in the Videopaper technology. Such solution, however, has not been implemented yet.

The revised version included a small interface in a browser that lets the user see the ToC of the original wiki tree and select to include or not include certain elements of the tree. First the user is presented with an initial version of the
linearized form, then user input is gathered, and a revised linearized version including the user preferences is created and converted to PDF for final printing (see Figure 9).

The proposed WikiPrint solution created for the need of reviewing wiki content on paper seemed to have satisfied the user need in our research lab. The available data set, however, was small, and is by no means representative for a large class of project wikis. Getting data to extend the experiments was very difficult, since project wiki data contain often confidential material and are confidential. Software coding projects documented using the Trac wiki\textsuperscript{11}, are an exception. But for software code it seems to be less likely that users would want to print content. We would have also liked to port our wikiprint solution to other wiki software, such as Mediawiki\textsuperscript{12}. That wiki and the Dokuwiki used in our experiments have still a different enough structure such that some work would have been necessary to adapt the WikiPrint solution to the Mediawiki format. That effort was beyond the scope of our experiments. Despite the small dataset used in our experiments, however, we are convinced that we demonstrated a useful method for creating a content structure via the arborescence and document tree that enables a useful linearization of wiki content for printing.
6. CONCLUSIONS

In this paper we described a solution to the problem of how to linearize content created by multiple users in a wiki used for project documentation to eventually print the content. Since the project wikis in our experiments were observed to have a rather tree-like structure compared to a complicated general graph structure, we chose to extract a specific subtree by adding a task-specific weight distribution to the graph and extracting an arborescence, transform that arborescence into a document tree including section, subsection, table of contents, etc., and perform the linearization by crawling the tree in a depth-first fashion. For different usage scenarios different layouts of the wiki content were created and feedback from the users collected. Based on the feedback the WikiPrint technology was revised by addressing the users comments regarding the depth of the section numbering, color coding and optional manual user refinement.

REFERENCES

11. http://trac.edgewall.org/