

Supporting wiki designers with a Confluence Design Pattern Library

Dan Wendling, wendingd@mail.nlm.nih.gov
National Library of Medicine, NIH/HHS

Accessible version of a PowerPoint presentation for the Wiki Pecha Kucha, 5/14/2009,
<http://wiki.sla.org/display/CEWIKI/2009+05+14+~+Sharing+our+Stories+about+Wikis+the+Pecha+Kucha+Way>.

Introduction

How can we build the core competencies of wiki designers to create effective wikis? One way is through a collection of wiki design tips that we call a Confluence Design Pattern Library (DPL). If you would like to give the wiki users at your institution a quick start on a small number of high-leverage capabilities, consider adding a design pattern library.

Table of contents

Introduction.....	1
Core competencies for “The Designer”	2
Design Pattern Library home page.....	2
DPL home page-list by task.....	3
Content chunking; embedding videos	4
RSS for bringing in content from outside; Tag clouds	5
Creating a slide show/tutorial; footnotes.....	6
Components the design pattern template.....	7
Example: The PageTree design pattern.....	8
Illustration for PageTree	9
Sample code for PageTree.....	10
When PageTree might be useful	11
Accessibility notes for PageTree	13
Consequences, Additional resources	13
Structural scale, Example pages, Related patterns.....	14
9 reasons to implement a design pattern library.....	15
What the designer should focus on (e.g., Davenport’s Hierarchy of Information Engagement)	16
Conclusion.....	17
Key influences	17

Core competencies for “The Designer”

We recognize six wiki roles:

1. The System Administrator
2. The Space Administrator
- 3. The Designer**
4. The Writer
5. The Reader-Commenter
6. The Reader

“The Designer” engages in the art and science of web design to help people accomplish work. Designers architect and re-architect spaces, show users what they need to see, and work within accessibility compliance guidelines. More specifically, the core competencies for designers can be described as:

- Architecting new spaces
- Re-architecting existing spaces
- Highlighting emergent content
- Using information architecture to improve the user experience
- Showing others how to communicate more effectively
- Conforming to Section 508 (accessibility) rules

Design Pattern Library home page

As shown in **Figure 1**, the home page of the DPL in our instance of Confluence has around 25 entries representing what we believe are “core competency” design tasks.

A "design pattern library"

About this area...
It is **not necessary** to spend time in this area before starting to work on wiki pages; however, this page introduces a small number of **wiki concepts and functionalities** that can get you going in a hurry and improve your team's productivity. You might find it convenient to open this page in a second browser window or tab. Another resource is the [Confluence User Guide](#).

<p>Wiki starter kits Get an overview of available functionalities and a "quick start" for a project. Page templates.</p>	<p>Wiki architecture - contents - usage Use "Browse Space" to understand a wiki's structure; find pages via metadata; study usage.</p>	<p>Content chunking Decide whether to aggregate information in a small number of pages or spread it among several or many pages. Known as "Paging vs. Scrolling."</p>
<p>Edit page, add page, upload file Post new content to an existing page, a new page, and/or make a PDF, Excel, PowerPoint, or other file available to others.</p>	<p>Change the location of a page Move a page from one place in the hierarchy to another place.</p>	<p>Page quadrants Increase the density of information on pages by using columns.</p>

Figure 1. Home page of NLM's design pattern library.

Each pattern has a thumbnail image, a link to the pattern page, and a short blurb. Sample entries include:

- **Wiki starter kits:** Get an overview of available functionalities and a "quick start" for a project. Page templates.
- **Wiki architecture - contents – usage:** Use "Browse Space" to understand a wiki's structure; find pages via metadata; study usage.
- **Content chunking:** Decide whether to aggregate information in a small number of pages or spread it among several or many pages. Known as "Paging vs. Scrolling."
- **Edit page, add page, upload file:** Post new content to an existing page, a new page, and/or make a PDF, Excel, PowerPoint, or other file available to others.
- **Change the location of a page:** Move a page from one place in the hierarchy to another place.
- **Page quadrants:** Increase the density of information on pages by using columns.

DPL home page-list by task

Outside the home page's grid design is another column; This presents design patterns by task; "If you want to do x, use y." If you want help starting a new wiki space, go to the starter kits. If you want to update people to keep them informed about what's happening, use a blog; if you want to gather opinions, use a survey. Here is the task listing, without the hyperlinks:

If you want to...

Start a new site/area

- Wiki starter kits
- Edit page, add page, upload file

Understand Confluence structure

- Wiki architecture - contents - usage
- Page quadrants

Organize data

- Content chunking
- Use a page tree for main navigation
- To-do list
- Display old, outdated pages
- Browse by popularity
- Browse by topic: Tag cloud
- Add content without focusing on site structure: Tag cloud

Keep participants informed, confer

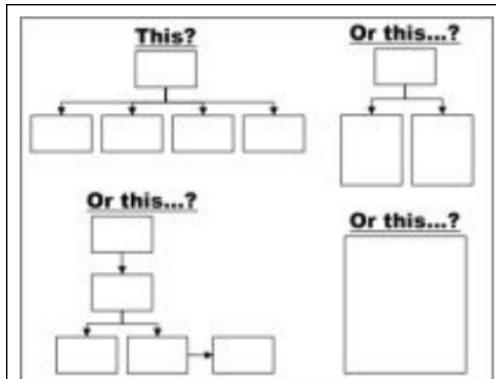
- Blogging

Gather opinions

- Gather data with a survey
- Comment on page content

Content chunking; embedding videos

The **Content chunking** pattern asks, "What is the best structure for a wiki space or area?" It depends on what you're doing. Emergent content will probably be small pieces loosely joined, but FAQs are often on one long page. We tell people what the decision hangs on and we quote Jakob Nielsen and others.




Content chunking

Decide whether to aggregate information in a small number of pages or spread it among several or many pages. Known as "Paging vs. Scrolling."

Figure 2. The "Content chunking" entry on the DPL home page.

Future development

The following YouTube video was referenced in the meeting. It has interesting information about the way wikis work. [Contact Dan for accessibility help](#)



What staff find most about this perspective is that it's not overly technological and the content is clear enough to get you

Embed YouTube, Flickr content

Embed media from YouTube, Flickr, SlideShow, Google Gadgets, and other services.

Figure 3. The “Embed YouTube, Flickr content” entry on the DPL home page.

RSS for bringing in content from outside; Tag clouds


We summarized the Atlassian resources for our enterprise, describing key capabilities by the design problems they solve, and we have *localized* the information. Because authentication issues prevent us from using outbound RSS, our **RSS feeds** pattern focuses on inbound RSS.



RSS feeds

Bring an outside feed into a Confluence page, for a database search, blog, news feed, etc.

Figure 4. The “RSS feeds entry on the DLP home page.



Tag Cloud

2008 is CSS XHTML JavaScript government keynote
 language mobile rss new packaging shdesign survey tool
 track-collaboration **track-content-management**
 track-cyberlaw track-discovery track track-identity track-lead-gen-usage
 track-open-source track-planning-managing track-search-engines
 track-webdesign translation web20 wikipedia


Tag cloud

Add a tag cloud to display labels by popularity; allows users to move through the content without following a specific navigation structure.

Figure 5. The “Tag cloud” entry on the DPL home page.

Creating a slide show/tutorial; footnotes

The core of each pattern is the code to copy and paste, and around the code we wrap implementation advice. Here are the summaries for **Slide show** and **Footnotes**. Other patterns focus on navigating among and within pages, setting access permissions, blogging, to-do lists, charts and graphs, etc.



What is CSS

- Cascading Style Sheets, a W3 standard
- Defines how browsers display areas HTML.
- Specifies look and feel from HTML.

Web Training - CSS Basics

Slide show

Create a slide show that is similar to PowerPoint but is based in the wiki.

Figure 6. The “Slide show” entry on the DPL home page.

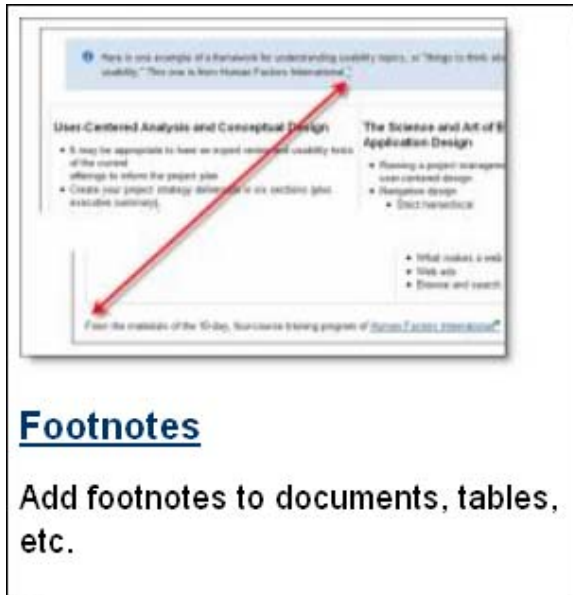


Figure 7. The "Footnotes" entry on the DPL home page.

Components the design pattern template

To create new patterns we use a template that contains the following 12 elements:

Title

(Oriented toward the designer who has a problem, and this is the solution.)

Problem summary

(Brief description that includes context and motivation)

Solution

(Text description of the pattern, coordinated with Illustration/example gif)

Illustration

Example gif:

("Generic" example--thumbnail of the widget, or a wireframe diagram)(Click to expand)

("Long description")

If you wanted to... (Show a use case):

(Gif of use case, thumbnail)(Click to expand)

("Long description")

Sample code

```
{code}  
// Some comments here  
public String getFoo()  
{  
    return foo;  
}  
{code}
```

(Required and optional arguments, tags, etc.)

When this pattern might be useful

(Workflow principles and guidelines that this pattern might support.)

Accessibility

(How to make this pattern accessible.)

Consequences

(Implementation issues, known problems, constraints that users may experience as a result of using this pattern.)

Additional resources

- * Confluence user guide
- * (URLs from Confluence site, NLM staff contact for customized functionality)

Structural scale

(Level of granularity: Page series, Page, Subpage, Widget)

Examples of use

(link)

(link)

Related patterns

(link)

(link)

(end of quoted text)

Example: The PageTree design pattern

For example, the PageTree pattern, shown in **Figure 8**, is about a page and a half printed out. The **title** says you can use it for main navigation. The **problem summary** has a brief description that includes context and motivation. The **solution** is a text description of the problem or problems you can solve.



Figure 8. Title, Problem summary, and Solution for the PageTree pattern

Text version of this image:

Title: Use a page tree for main navigation

Problem summary

1. You want to populate a wiki area quickly by adding new pages
2. You will be building a new set of documents but you don't know how to arrange them.

Solution

Use the PageTree plugin, which puts file listings into a tree form similar to what Windows Explorer does in column 1. When the page is loaded, PageTree will dynamically generate a list of all the pages in the hierarchy that you tell it to look for (usually you will want to look "down" from the current page).

A search blank can be added that is limited to the same hierarchy as the tree.
(end of quoted text)

Illustration for PageTree

The Illustration section, shown in **Figure 9**, has an **example image** of the PageTree in action.

Illustration



The above illustration of PageTree shows content moved from Outlook into a wiki space. Content was copied from Outlook onto the clipboard, then in the wiki Add > Page was selected; the content was pasted in; a page title was added; and the page was saved.

Figure 9. The hierarchy of pages and a use case.

This pattern also shows a **use case** below the image saying how a person can rapidly populate a wiki space from Outlook emails: “The above illustration of PageTree shows content moved from Outlook into a wiki space. Content was copied from Outlook onto the clipboard, then in the wiki Add > Page was selected; the content was pasted in; a page title was added; and the page was saved.”

Sample code for PageTree

Sample code is THE essential component for most of our patterns. There may be multiple samples that describe required and optional arguments, tags, etc. Here we have one suggestion and narrative about what it does as shown in **Figure 10**.

Sample code

The following code will appear like the display above:

```
{ pagetree:root=@self|startDepth=2|searchBox=true}
```

Here's what this code does:

- The tree will be constructed from this page down (@self), i.e., it looks down the hierarchy rather than viewing the entire hierarchy of the wiki space
- It will have the first level expanded (startDepth=2, for two levels showing by default)
- A search blank will search the same content as "root"; in this case it will search down the hierarchy starting from the current page.

Figure 10. Sample code section of PageTree.

Text version of this image:

Sample code

The following code will appear like the display above:

```
{pagetree:root=@self|startDepth=2|searchBox=true}
```

Here's what this code does:

- The tree will be constructed from this page down (@self), i.e., it looks down the hierarchy rather than viewing the entire hierarchy of the wiki space
- It will have the first level expanded (startDepth=2, for two levels showing by default)
- A search blank will search the same content as "root"; in this case it will search down the hierarchy starting from the current page.

(end of quoted text)

When PageTree might be useful

“When this pattern might be useful,” shown as **Figure 11**, includes workflow principles and guidelines that this pattern might support; how this functionality supports knowledge workers. An influence here was Tom Davenport’s book called *Thinking for a Living*, that says to provide a template for action. We also wanted to convey the utility of a course of action.

When this pattern might be useful

1. Populate a wiki area quickly

Because PageTree automatically generates the list of pages, you do not need to link to the new pages. PageTree will keep track.

2. Add many new documents to the wiki without worrying about how to arrange them.

The version of the page tree that administrators see at Browse > Pages is similar in appearance to the PageTree plugin, however in Browse > Pages administrators are able to drag and drop pages to change the hierarchy easily and quickly. So, if you need to bring a lot of information in from email or other sources, you could dump everything in with Add > Page, and after your information is in the wiki you can use Browse > Pages to drag and drop the content into emergent categories.

If you're using this pattern for its capabilities with emergent structure you may wish to combine its use with the tag cloud or page popularity plugins. You may also wish to use this dynamic-structure view in one column of your page and then other types of links, such as links to external pages, in a second column. These uses would be represented better in page quadrants.

The **excerpt** flag described on the [plugin home page](#)² will assist if you are using PageTree to organize a large amount of new data.

Figure 11. When this pattern might be useful.

Text version of this image:

When this pattern might be useful

1. Populate a wiki area quickly

Because PageTree automatically generates the list of pages, you do not need to link to the new pages. PageTree will keep track.

2. Add many new documents to the wiki without worrying about how to arrange them.

The version of the page tree that administrators see at Browse > Pages is similar in appearance to the PageTree plugin, however in Browse > Pages administrators are able to drag and drop pages to change the hierarchy easily and quickly. So, if you need to bring a lot of information in from email or other sources, you could dump everything in with Add > Page, and after your information is in the wiki you can use Browse > Pages to drag and drop the content into emergent categories.

If you're using this pattern for its capabilities with emergent structure you may wish to combine its use with the tag cloud or page popularity plugins. You may also wish to use this dynamic-structure view in one column of your page and then other types of links, such as links to external pages, in a second column. These uses would be represented better in page quadrants.

The excerpt flag described on the plugin home page will assist if you are using PageTree to organize a large amount of new data.

(end of quoted text)

Accessibility notes for PageTree

Designers need more accessibility data about macros, plugins, and functionalities than is currently available, so one of the key reasons we started the design pattern library was to create a place to highlight accessibility information, as shown in **Figure 12**.

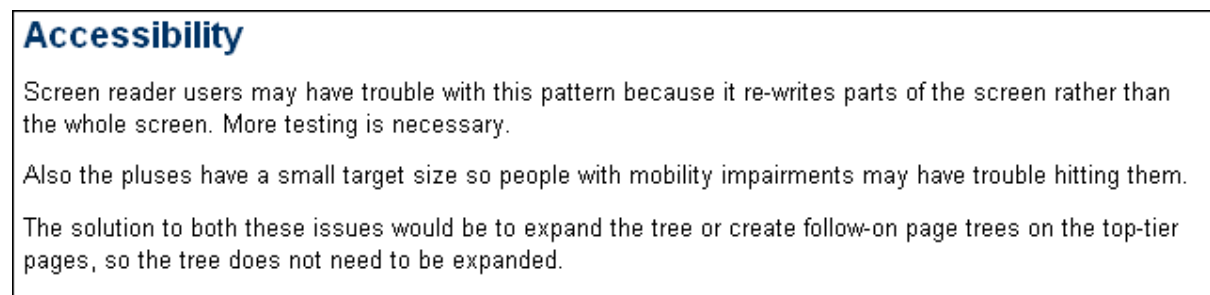


Figure 12. Accessibility information for people using PageTree.

Text version of this image:

Accessibility

Screen reader users may have trouble with this pattern because it re-writes parts of the screen rather than the whole screen. More testing is necessary.

Also the pluses have a small target size so people with mobility impairments may have trouble hitting them.

The solution to both these issues would be to expand the tree or create follow-on page trees on the top-tier pages, so the tree does not need to be expanded.

(end of quoted text)

Consequences, Additional resources

As shown in **Figure 13**, **Consequences** include implementation issues, known problems, and constraints that users may experience when using this pattern. **Additional resources** includes the URL for the macro or plugin page on the vendor site. Often we have a link to the Confluence user guide. We're NOT replacing other documentation; that is not the goal.

Consequences

You won't be able to mix links to external pages into this list.

Additional resources

- <http://confluence.atlassian.com/display/CONFEXT/Confluence+PageTree+Plugin>
- <http://confluence.atlassian.com/display/DOC/Moving+Pages+within+a+Space>
- <http://confluence.atlassian.com/display/DOC/Working+with+Page+Families>
- [Confluence user guide](#)

Figure 13. The Consequences and Additional resources sections of the pattern.

Text version of this image:

Consequences

You won't be able to mix links to external pages into this list.

Additional resources

- <http://confluence.atlassian.com/display/CONFEXT/Confluence+PageTree+Plugin>
- <http://confluence.atlassian.com/display/DOC/Moving+Pages+within+a+Space>
- <http://confluence.atlassian.com/display/DOC/Working+with+Page+Families>
- [Confluence user guide](#)

(end of quoted text)

Structural scale, Example pages, Related patterns

As shown in **Figure 14**, the **Structural scale** means does this apply to a Page series, Page, Subpage, or Widget; **Examples of use** provides links within this wiki where people can see this pattern in use. **Related patterns** are similar patterns; if you like this one, check out these others. We've seen one complete pattern.

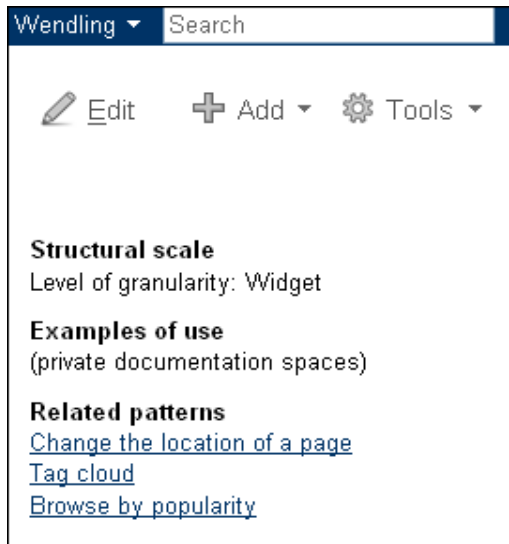


Figure 14. Structural scale, Examples of Use, and Related patterns for PageTree.

Text version of this image:

Structural scale

Level of granularity: Widget

Examples of use

(private documentation spaces)

Related patterns

[Change the location of a page](#) (link)

[Tag cloud](#) (link)

[Browse by popularity](#) (link)

(end of quoted text)

9 reasons to implement a design pattern library

(Derived from Lamantia, 2006, 2007)

1. Be user-centric; address user experience design challenges; create an effective user experience
2. Provide a simple design approach with building blocks
3. Provide a structure for growth and evolution
4. Increase the value of each individual tool by showing context and the differences between functionalities
5. Demonstrate new ways to accomplish work, pre-vetted, to increase and improve content assets
6. Rapidly create larger units of content from smaller chunks of information
7. Ease concerns about problems with enterprise information architecture
8. Provide collaboration functionalities that scale up and help you respond to change

9. Provide sharing and reuse of familiar, defined blocks of functionality among projects of all sizes and complexity.

What the designer should focus on (e.g., Davenport's Hierarchy of Information Engagement)

The reason we went to this trouble is to free up designers to focus on the issues in the model by Davenport and Pruzak shown in **Figure 15**. We want designers and users out of the systems realm and in the knowledge work realm. "Take this code, here's what it's good for, avoid these problems, and go help people collaborate and get their work done." Make their work the focus.

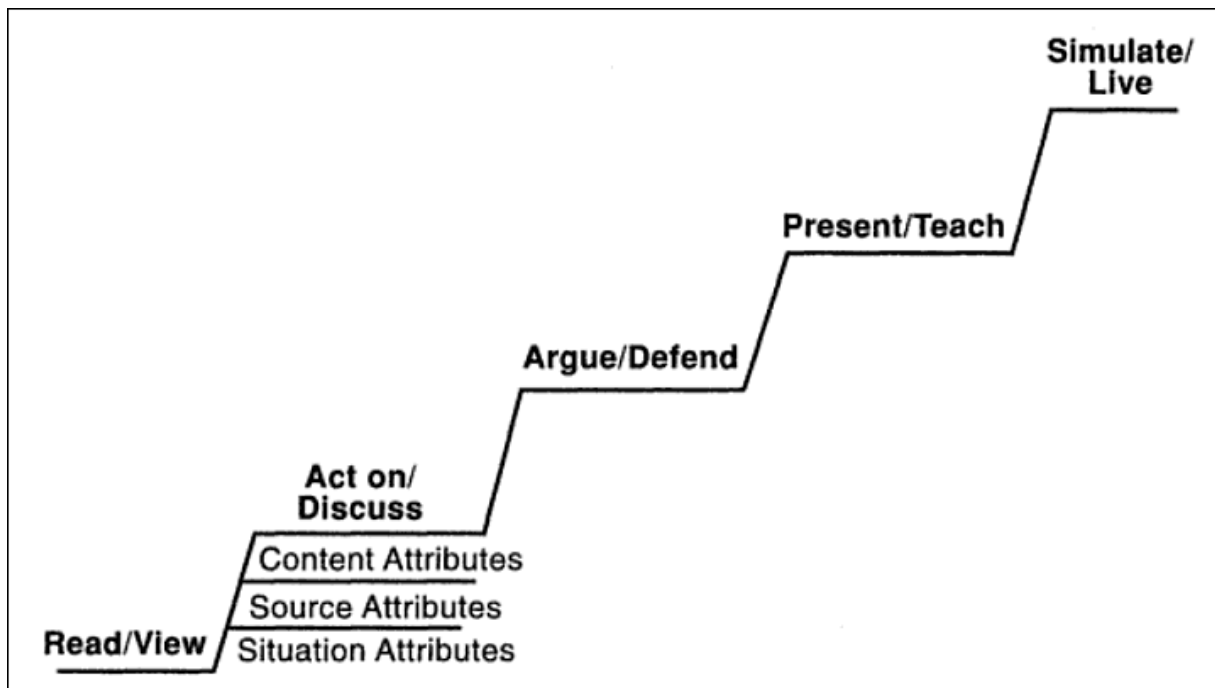


Figure 15. Hierarchy of Information Engagement.

Text description of this image:

More information is communicated in read/view mode than in any other. However, higher levels of emotional engagement and commitment come from the levels shown, in order: when people act on/discuss the information; when people argue/defend the information; when people present/teach the information, and the highest level of engagement, when people simulate/live the information.

(end of description of Figure 15)

Conclusion

Instead of having dozens of options that can be explored on the Atlassian web site we focus on **capability maturity** for around two dozen design topics. We make the tasks of our designers as simple as possible, and if we choose correctly this will make the tasks of all participants as simple and as effective as possible, so they can accomplish their work.

Key influences

- Barker, Dean T. (2007, July 12). User interface patterns: Reusable solutions for design. White paper. Human Factors International, Inc. Online at: <http://humanfactors.com/downloads/whitepapers.asp>
- Lamantia, Joe. (2006, 2007). The challenge of dashboards and portals. Part 1 of 6. Boxes and Arrows, August 2007. Online at <http://www.boxesandarrows.com/view/the-challenge-of> (see all parts)
- Yahoo! Developer Network, Design Pattern Library: <http://developer.yahoo.com/ypatterns/index.php>
- Tidwell, Jennifer. Designing Interfaces: <http://designinginterfaces.com/>
- Davenport, Thomas, and Prusak, Laurence. (1997). *Information Ecology: Why technology is not enough for success in the information age*. New York: Oxford University Press. p. 93, Hierarchy of information engagement based on Linder.
- Davenport, Thomas. (2005). *Thinking for a living: How to get better performance and results from knowledge workers*. Boston: Harvard Business School Press. Chapter 4, Knowledge work processes; chapter 7, Investing in knowledge workers' networks and learning.