

**Technische Universität München
Fakultät für Informatik**

Bachelor's Thesis in Information Systems

**Design and Implementation of an
Integrated User Interface for a Web
Collaboration Platform**

**Konzeption und Realisierung einer
integrierten Benutzeroberfläche für eine
webbasierte Plattform zur Teamarbeit**

Author: Jakob Stoeck
Supervisor: Prof. Dr. Florian Matthes
Advisor: Dr. Thomas Büchner
Submission Date: 18 August 2008

I assure the single handed composition of this bachelor's thesis only supported by declared sources.

Contents

0.1	Abstract	5
1	Actions	7
1.1	Actions	8
2	Datasheets	11
2.1	Error Pages	12
2.2	Generic Data	16
2.3	Profile	19
2.4	Statistics	23
3	Feedback	27
3.1	Messages	28
3.2	What Can I Do	32
3.3	Where Am I	34
3.4	Who Am I	36
4	Forms	37
4.1	Edit Forms	38
4.2	Login Form	43
4.3	Password Update	45
4.4	Search Forms	47
4.5	Add Media	50
5	Listings	53
5.1	Alphabetical Listings	54
5.2	Recent Activities List	56
5.3	Flat Listings	58
5.4	Grouped Lists	61
5.5	Hierarchical Order Listings	63
6	Navigation	65
6.1	Breadcrumb	66
6.2	Buttons	69
6.3	Dashboard	72
6.4	Navigation Through Different Document Types	74
6.5	Links	76
6.6	Menu	77

6.7	Search	79
6.8	Title	81
6.9	URL	83
6.10	Image Language	87
6.11	Layout	90
6.12	Typography	97
7	Terminology	101
7.1	Dates	102
7.2	Documents	105
A	Used Design Libraries	109
	Bibliography	111

0.1 Abstract

Team collaboration software helps project teams synchronize themselves to the newest information and to create "communities of interest". [Fi01]

Working together in a team is enhanced by providing wikis, blogs and file sharing.

This thesis inquires, creates and refines best practices for an integrated interface design of a web application integrating wikis, blogs and corporate file sharing functionality for smaller teams.

Problem

Current web collaboration software is often built around one document type instead of providing a truly integrated solution. Although the enterprise wiki software "Confluence" has multiple document types, it is clearly visible that wiki pages are the basis of them. Blogging with Confluence is not as convenient as it is with the WordPress. The latter however, even with powerful extensions, is a blog post administration system at its core. Those systems cannot provide wikis, blogs and file sharing in an integrated way and therefore are missing functionality or combination possibilities in at least one of those three functionalities.

Existing fully integrated solutions like the IBM Lotus family are too comprehensive for smaller teams and have a steep learning curve.

A system which combines all those functionalities but conserves the ease of use of non-integrated solutions is needed.

Approach

As mentioned above, non-integrated systems are easy to use which makes them popular. Writing a blog post in WordPress is very intuitive. Sharing a file with the backup and synchronizing service "Dropbox" is very straight-forward. Why are those single systems so easy to handle? And how can this be combined in an integrated web application? In this thesis I look at three of the top single systems in terms of popularity and usability: WordPress for blogs, Confluence for wikis and Dropbox for corporate file sharing. I look at the problems which they had to solve in their domain and how they did it.

Realization

The importance of user interface design is apparent today but there exists no acknowledged way to test and develop it. Tests vary from time-to-task, ergonomic considerations, number and weight of arising errors and observing ISO guideline 9214 [SB06] to user satisfaction, abiding principles of design [Ni94] and extensive usability tests [RH94]. Some say it is best to build a system iteratively with consultation of the users [Kr00] others advice you to design first to the best of ones knowledge and re-adjust the interface only when you observe errors whilst operation of the system. [LM90]

Following the consequence that "there is a direct relationship between the quality of design and the willingness of the designer to take on mindful explorations of what lies beneath a beautiful interface" [La03], I developed a map of interface parts which cover the most important aspects of a web application for collaboration according to above sources.

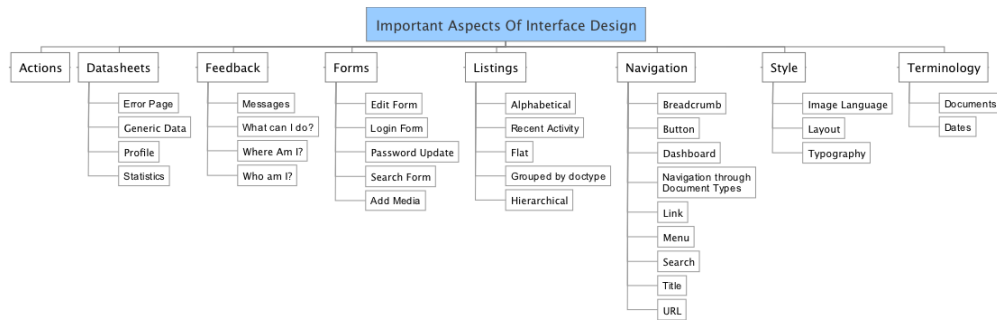


Figure 1: Thesis Sitemap

Those are explored on the following pages. Each chapter consists of an exploration of the three mentioned systems, showing up the advantages and disadvantages of their particular way of handling the current interface design category. Each chapter includes an "Integration" subchapter which shows up best practices in this category derived from this exploration and from other mentioned sources.

Implementation

The thesis can be used as a best practices guide by looking only at the integration sub-chapters. This guide is used to investigate the integrated collaboration platform "Tricia" and to put them into practice mainly with JavaScript, HTML and CSS programming.

Chapter 1

Actions

1.1 Actions

How are actions displayed?

WordPress

With the exception of "Add", actions are always displayed as buttons. There are three kinds of actions:

1. associated to a form field, e.g. an input field or a drop down menu: The button is on the right side of the field (e.g. "Search")
2. associated to selected documents from a list: The button is on the left top of the list, right above the checkboxes which select the documents (e.g. "Delete")
3. associated to a form: the button is on the right side of the form or below of it on the left (e.g. "Save")

Specialties:

- "Add": It is either a link right beside the heading of the document type or part of the main menu. One exception is the dashboard, where also the "Add" functionality is a button.
- "Edit": This functionality has no special button but is invoked automatically when clicking on the title of an editable document type.
- "Search": A search in WordPress is better called a filter as it searches the current shown list for the search terms. It is located next to the current headline.

Confluence

Actions have multiple reflections. They are links, drop down menus or buttons. Their colour, position and their size varies.

- "Save" and "Cancel": Always present in bigger forms. The actions are reflected by two buttons at the top and bottom right of the form. Actions associated with small forms are represented as grey links besides the form head.
- "Edit", "Add" and "Tools": There is a ubiquitous menu on the top right of the window. It consists of those three main menu items. The last two are actually drop down menus. The add-functionality can add "Pages" and "News". "Edit" and "Tools" are only available when a wiki page is openend.
- "Search": The search is located at the top right corner throughout the application. By default, it searches through all document types except mail.

Integration

How To Show Actions

There are four GUI elements which can be used for actions:

- Buttons

- Drop Down Lists
- Toolbars
- Tabs

Do not use other elements, e.g. links. "To most people underlined links mean 'more information,' not 'do something'." [FS04]

Where To Put Actions

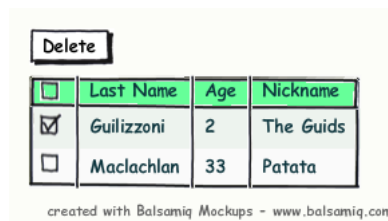


Figure 1.1: List with checkbox column and associated button

Actions affecting multiple rows (e.g. "Delete", "Mark As Read") should be at the top left of the list, above the checkbox column, if available. [FS04]

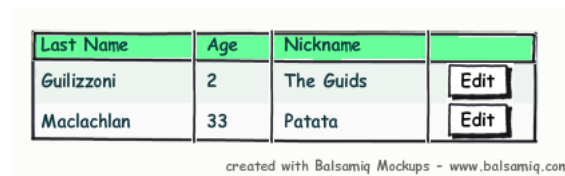


Figure 1.2: List with actions

Actions which affect only one row (e.g. "Edit") should be located in this row. Most of the times in the rightmost column. [We08]

Actions which affect the whole page (e.g. "Save") should always be on the top and the bottom of the page. Because of desktop applications many people are accustomed to use buttons at the bottom of the form, even when there are the same actions at the top where you don't have to scroll. [FS04]

Ubiquitous Add Menu

The ubiquitous Add-menu has a big advantage since you can create a new document from anywhere inside the application. It should stay next to the headline.

Search

It has become a convention to put the search functionality at the top right corner of the screen. It doesn't need a search button anymore. An input field labeled "Search" is enough for most users.

Chapter 2

Datasheets

2.1 Error Pages

WordPress

Not Found

The requested URL /wordpress/wp-admin/kj was not found on this server.

Apache/2.0.59 (Unix) PHP/5.2.5 DAV/2 Server at localhost Port 80

Figure 2.1: Standard apache error page

WordPress uses the standard apache error page.

Good

- Loading time is very fast.
- The page is recognized immediately.
- The entered URL is repeated which can help in finding the cause of the error.

Bad

- There is no solution offered in getting on the track again.
- The user feels left alone.
- Does not fit into the rest of the application design.

Confluence

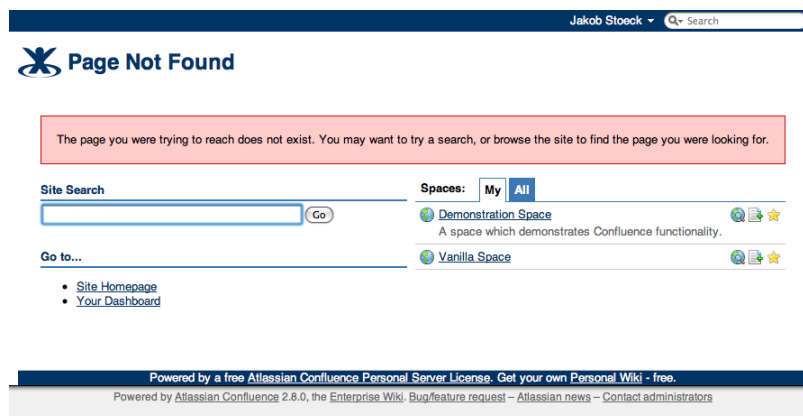


Figure 2.2: Confluence error message

Confluence has two different error pages. The first one is displayed when the URL is misspelled in terms of wrong space short name, wrong page name and so on.

You get

- a page heading "Confluence (Page Not Found)",
- an error message in red,

- an overview over all spaces, similar to the overview on the dashboard,
- a site search box and
- links to the homepage and dashboard (those are actually the same by default but you can customize that)

There is no breadcrumb.

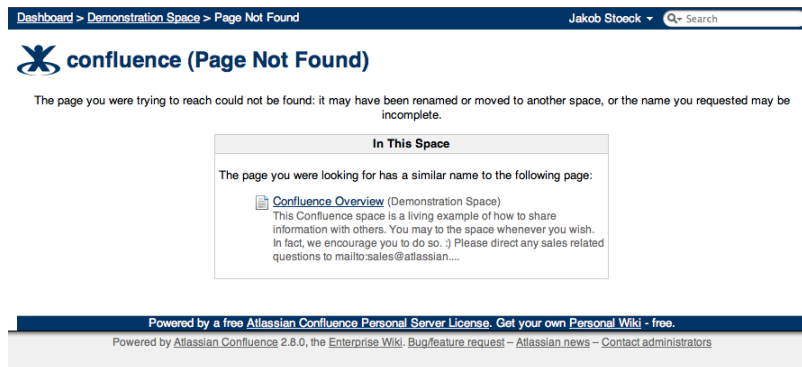


Figure 2.3: Confluence warning message

The Second error page is displayed under the condition that everything of your URL is correct until the last part which resembles the page's title. Additionally, the misspelled title has to be alike enough to an existent page.

You get

- as page heading "Page Not Found",
- an error message,
- an "In This Space" box

This box contains the description "The page you were looking for has a similar name to the following page". Beneath it is the title and an excerpt of the page which are linked to the actual page.

There is also a breadcrumb.

Good

- Offers a search box and important links of the application on the search page.
- Offers a link and an excerpt to a similar page.
- Search box is focused on page load.

Bad

- The error pages differ a lot in terms of their heading, content elements and layout elements. If you do not enter a space short name there it is even worse. But this seems like a bug so it is not further discussed.
- Recognition of similar page names is very bad. It is worse than the built-in search engine for the same terms.
- The additional search box on the error page confuses. Does it search differently than the normal search box at the top?
- Error messages are formatted differently and do not follow a colour code.

Suggestions

- Have only one "Page Not Found" page which can handle both situations described above.
- Only have one search box on the page. If there are two, show the user why, e.g. because there are already search results displayed.

Integration

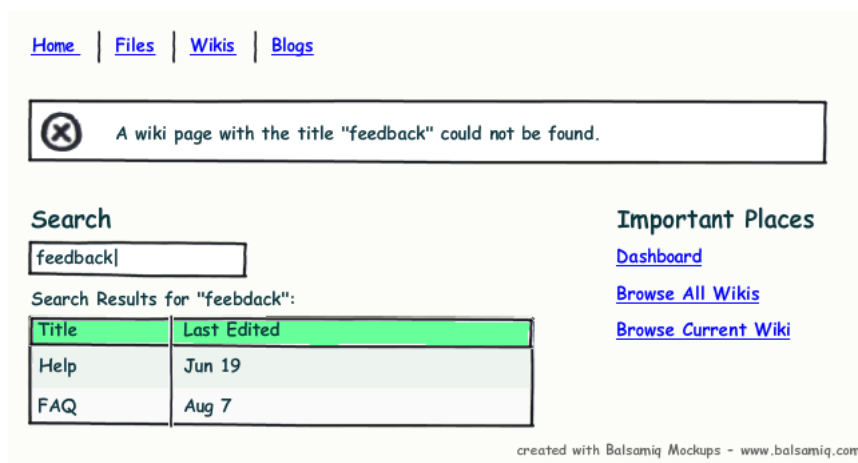


Figure 2.4: An error page with search, important places and the normal web application's navigation bar

Show An Error Message

This means an accordingly formatted one. If the error occurred is because of a potential misspelling, repeat the entered URL to make finding the misspelling easier. See →messages.

Search In Advance

You can use your application search engine to search for the entered terms and display the results on the error page. Also display a message if there are no results for those search terms. You likely take down one step from the user.

Display the search box above the results. It should have the search terms already entered in and be focused.

Send A 404 Not Found Header Reply

If you use a custom error page you have to make sure you send the according HTTP header information. This supports other applications like batch download applications and search engines.

Show Important Places

Show a list of important pages and sections of the application. Especially:

- Homepage / Dashboard
- Browse current wiki/blog
- Browse all wikis/blogs

The User Is Now Open For Suggestions

Most users use a web application in a stubborn way. If they found a way how to achieve their goal they stop searching for another, possibly better solution. But when there is an error, there is a small time frame in which users search for a new guide. [FS04]

You can use this window to advertise functionality or parts of your application which you want to be used. If the user sees an advantage in this functionality to reach his page or if it looks more interesting than the page he originally wanted to go, he is likely to choose your suggested route.

Use Your Layout

Sometimes error pages are ripped out of context of the rest of the application. Do not do that. Use the same layout with the same functionality like other pages offer to make it easy for the user to browse away from this page with the navigational tools he is used to.

2.2 Generic Data

How can a document's meta data be displayed?

WordPress

The screenshot shows the WordPress dashboard with the following meta data sections:

- Right Now:** Write a New Page | Write a New Post. You have 2 posts, 1 page, 2 drafts, contained within 1 category and 0 tags. You are using WordPress Default theme with 0 widgets. Change Theme. This is WordPress version 2.5.1.
- Recent Comments:** See All. "There's the next comment!" From admin on Bachelor's Thesis ready! #
- Incoming Links:** See All | Edit | RSS. This dashboard widget queries Google Blog Search so that when another blog links to your site it will show up here. It has found no incoming links... yet. It's okay — there is no rush.

Figure 2.5: Meta data on WordPress' dashboard'

There is no explicit page for meta data. It is displayed at the top or bottom of the actual page.

The dashboard is an exception. It consists of rectangular modules which display meta information about the whole blog, such as incoming links and statistics. Each consists of a heading, the actual data and actions: See all, Edit, RSS. You can set how many entries and which RSS feeds are displayed. Modules are arranged in a two-column layout.

Confluence

The screenshot shows the info page of a Confluence wiki page with the following sections:

- Title:** Confluence Overview. View Source
- Author:** Anonymous. Nov 15, 2004
- Last Changed by:** Jakob Stoeck. May 25, 2008
- Tiny Link:** (useful for email) http://localhost:8080/x/E4
- Export As:** PDF | Word
- Operations:** Copy
- Hierarchy:** Children (10)
 - Example Index
 - Creating pages and linking
 - Formatting Content
 - News
 - RSS Capabilities
 - Search Confluence content
 - Source code
 - Tasklist
 - Thumbnail gallery
 - Using Spaces
- Labels:** There are no labels assigned to this page.
- Recent Changes:**

Time	Editor	
May 25, 2008 20:55	Jakob Stoeck	view changes
May 09, 2008 14:55	Jakob Stoeck	

 View page history
- Outgoing Links:**
 - External Links (6)
 - confluence.atlassian.com/display/DOC/Troubleshooting
 - confluence.atlassian.com/display/DOC/Features+other+wikis...
 - atlassian.com/about/customers.jsp
 - atlassian.com
 - confluence.atlassian.com/display/DOC/Features+other+wikis...
 - en.wikipedia.org/wiki/Wiki
 - Demonstration Space (10)
 - Search Confluence content
 - Source code
 - Formatting Content
 - Tasklist
 - Using Spaces
 - RSS Capabilities
 - Creating pages and linking
 - Thumbnail gallery
 - Example Index
 - Jira integration

Figure 2.6: Info page of a Confluence wiki page

There are explicit pages for meta data for each wiki page. They contain the following information:

- Main Info
 - Title
 - View Source
 - Author
 - Last Changed By
 - Tiny Link (for email)
 - Export as PDF/Word
 - Copy
- Hierarchy
 - Parent
 - Children
- Labels
- Recent Changes
 - View Changes
 - View Page History
- Outgoing Links
 - External Links
 - Internal Links

Almost every entry is linked, i.e. people to their profile, children and parent pages, version dates to this version of the page etc. While it predominantly shows information it is also a platform to navigate to other places of the application and out of the application.

Good

- Two-column view makes the page very lucid.
- Every information is above the fold. (With a screen resolution of 1024×768 pixels)

Bad

- You cannot edit the page from here. You have to first click "View" then "Edit". See →Layout.
- "Recent Changes" is a table headline and a link at the same time. It's not clear where the link routes to because the "recent changes" are presumably already under this headline. (It routes to the page history.)
- There are no icons in front of a person document type.

Suggestions

- Instead of mixing headlines with links, set a link next to the headline which actually says where the link routes to. In this case "Page History" or "See All". Mixing links and headlines is often a bad idea because generally the content, where the text of the headline presumably routes to, should already be underneath it.
- A link to a profile should always accompany an icon.

Integration

Use Modules

Split the information into modules. Each module has a heading. Two to three modules should not reach more than 800 pixels in height, i.e. they are above the fold for standard resolutions.

Use Two Columns

An informational page should be compact and easily navigable. You can reach this goal by using two columns for a bigger amount of information modules. This way it is more likely that the information will stay above the fold. Additionally the user recognition is improved. It is easier to remember that e.g. user groups are "in the lower right box" than "in the lower third of the page, second or third last" as it would be with one column.

Append Actions To Modules

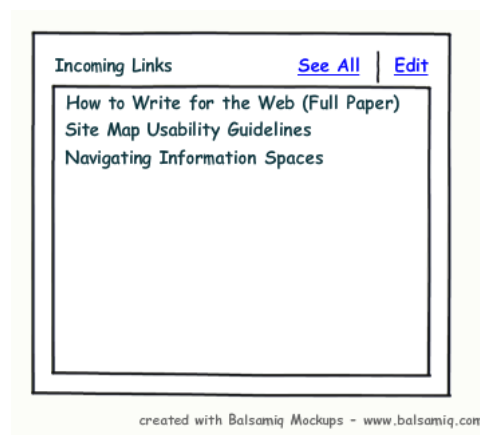


Figure 2.7: Actions associated to a information module

Next to the heading of a module you can append actions which correlate with the information of this module. In a recent changes box those could be "See All" and "RSS" which routes to an automatically updated feed of the latest changes of this document.

Labels Belong To The Left

Like in forms, lists which are longer than three entries, should have their labels aligned to the left of the information to allow faster skimming. [VD07]

2.3 Profile

WordPress

Personal Options

Use the visual editor when writing

Primary Blog:

Name	Contact Info
Username: (no editing) <input type="text" value="admin"/>	E-mail: (required) <input type="text" value="kontakt@stoeck-it.de"/>
First name: <input type="text" value="Jakob"/>	Website: <input type="text" value="http://"/>
Last name: <input type="text" value="Stoeck"/>	AIM: <input type="text"/>
Nickname: <input type="text" value="j"/>	Yahoo IM: <input type="text"/>
Display name publicly as: <input type="text" value="Jakob"/>	Jabber / Google Talk: <input type="text"/>

Figure 2.8: Profile page of WordPress

You can edit the displayed information right on the profile page since everything is displayed with form fields.

Five sub-sections are available:

- Personal Options

You can choose whether you prefer the visual editor over a textarea and choose your main blog.

- Name

Consists of username (not editable), first name, last name and nickname. You can choose which one or which combination of them is displayed under a post.

- Contact Info

Here you can enter your e-mail, website, AIM, Yahoo IM and Jabber/Google Talk info.

- About Yourself

This is a single textarea for a little info about yourself.

- Update Your Password

Bad

- The page doesn't tell how the entered information is used and displayed, e.g. which parts are shown on a publicly available page.
- There is no option to upload an avatar.

Confluence

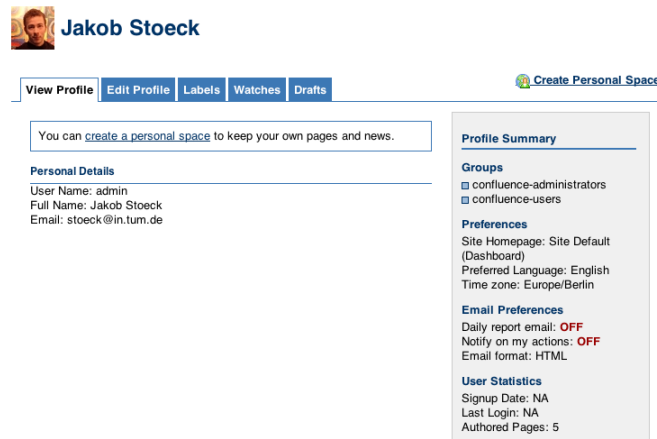


Figure 2.9: Profile page of Confluence

The profile is divided into five sections. You can navigate through them using a tabbed navigation:

- View Profile
- Edit Profile
- Labels
- Watches
- Drafts

The headline represents the particular sub-section you are in and extends the name of the menu item text a little bit. In the "View Profile" section it's the person's full name. In "Edit Profile" > "Profile Picture" it is "Edit My Profile Picture".

View Profile

In the "View Profile" section you have a summary of all profile options on the right hand side.

Edit Profile

The section is subdivided has a submenu to the following pages:

- "Your Details": You can enter your full name and Email-address in two input fields.

- "Profile Picture": You can choose a profile picture from previously uploaded pictures. To upload one, you can use a standard html file input. The picture is automatically resized to 48 x 48 pixels. You can choose from 16 pre-installed generic profile pictures.
- "Password": The current password and the new one (two times) have to be entered in three input fields.
- "General": You can change the following site defaults with drop-down menus:
 - "Site Homepage": The range of options spans the site default, the available spaces, the available personal spaces or the dashboard. There are the space keys in parentheses after the name.
 - "Preferred Language": Can be set to "Site Default", "Deutsch", "English", "français".
 - Select Time: The select menu options are arranged in the popular format, e.g. "GMT +0100 Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna"
- "Email":
 - Subscribe to daily updates [checkbox]: subtitle "The daily updates email contains a summary of all changes made to spaces you can see."
 - Notify me on actions [checkbox]: subtitle "By default, you are not sent an email when you add or change content. Check this option to be notified when this occurs."
 - Email format [select menu]: "HTML" or "Text", "Only the HTML format will contain images and formatted content."
- Labels, Watches and Drafts: The profile page is also the place to see your personal labels and the ones you used lately, pages you watch and pages you have not yet saved.

Good

- In each section, a helping text exemplifies the used terms and what you can do within this section.

Dropbox

You can edit the displayed information right on the profile page since everything is displayed with form fields. There are no profile sub-pages.

The different sections are grouped vertically with left-aligned labels:

- how much disk space is occupied
- which computers are linked to your dropbox
- how many invites are left
- your full name which resembles the display name
- your email-address
- change your password

Account Settings

Quota info | Account type: Beta (2.0GB)

- normal files - 1.1MB
- shared files - 1.1MB
- free space - 2.0GB

Computers | You have 1 Mac linked to Dropbox.
[Manage computers](#)

Invites | You have 9 invites left.

User info

Display Name:

E-mail Address:

Password reset

Old password:

New password:

Confirm password:

[Change settings](#)

Figure 2.10: Profile page of dropbox

Integration

Grouping

A profile page is structured in different sections. The observation of common web applications resulted in the following sections:

- User info
 - Username
 - Real Name
 - E-Mail
- Profile Picture
- Password
- Application-feature-specific (like preferred language etc.)

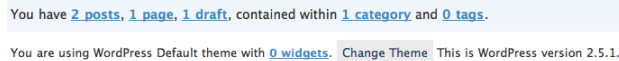
Labels For Input Fields

The de-facto standard for forms is a list of form fields with their labels aligned to the left. Each label should end with a colon and have the appropriate `<label>`-tags. Both help screen readers associate the label to its form field. [FS04]

Many experts suggest table-less forms. In my opinion, table-less styling can be too dogmatic in this situation. If you don't have control over the length of your labels (e.g. because of translations or users are able to rename some), I suggest table-supported forms as long as not all browsers have put the CSS 3 table specification into practice. Tables are good because columns expand automatically if a label changes its length. That is not the case with CSS 2 styled forms. That being said, table-less styling has many advantages not only performance- and style-wise. [Wo08] If you are aware of your label lengths use CSS. If not, it is advisable to use tables at the moment.

2.4 Statistics

Wordpress



You have [2 posts](#), [1 page](#), [1 draft](#), contained within [1 category](#) and [0 tags](#).
 You are using WordPress Default theme with [0 widgets](#). [Change Theme](#) This is WordPress version 2.5.1.

Figure 2.11: Statistics in WordPress are purely textual

The only available statistics are textual. They are located on the dashboard and have this format:

”You have x posts, y pages, contained within z categories and a tags.
 You are using *Theme Name* theme with b widgets.”

Confluence

There are two graphical statistics. One is about the amount of viewing pages and news, the other one about the amount of edited pages, news and comments.

- In the viewing statistics, pages and news views are shown in one area chart.
- In the editing statistics the three different document types are visualized with a line chart with three lines.

Below the graphical representations you have three top ten lists of ”Most popular content (Views)”, ”Most active content (Edits)” and ”Most active contributors (Edits)”. Behind every entry the actual amount is displayed in brackets.

Integration

Show activity by blog (or wiki)

A global activity view which goes beyond what is displayed on the dashboard holds few important information for the common user. Those are:

- Most popular blogs/wikis
- Most active blogs/wikis

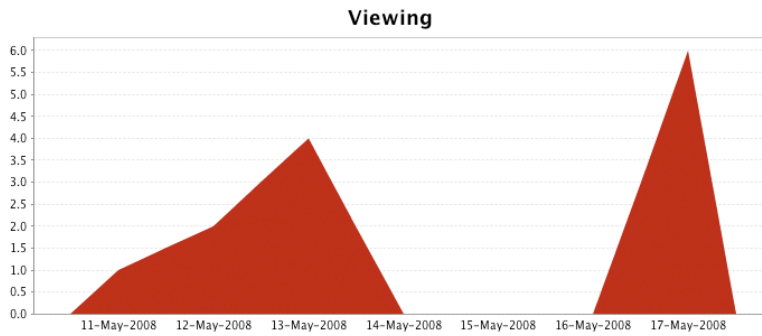
Because those are very short lists, they can be displayed also on the dashboard. More important is the activity of a certain blog or wiki. It should answer the following questions:

- Is this alive? Are there recent posts in it?
- Who are the authors?
- Do others access the blog?
- Which content is popular?

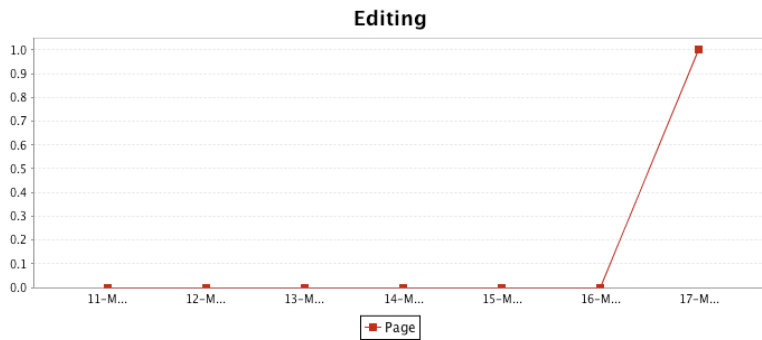
Those questions can be answered with the following:

Activity for week starting 11 May 2008

Period: [month](#) | [week](#) | [day](#) Week: [<< Previous](#) | [Now](#) | [Next >>](#)
 See also: [Global Confluence activity](#)



This graph shows how many times pages and news posts have been viewed over the current time period.



This graph shows how many pages, news and comments have been created or updated over the current time period.

Most popular content (Views)	Most active content (Edits)	Most active contributors (Edits)
1. News (7)	1. .bookmarks (1)	1. Jakob Stoeck (1)
2. Index (6)		

Figure 2.12: Recent activity graph in Confluence

- A graph with the axes page views over time
- A graph with the axes updates and inserts over time
- A top list of the most viewed content
- A top list of the most active content
- A top list of the most active editors

The common period for this information is a week. With less used systems it should be a month.

Make Graphs Customizable

If the graph information is of a higher priority in your application, give users the power to aggregate data at their will.

A user who knows how to work with graphs can aggregate much useful information with a customizable time on the x-axis and aggregating artifacts on the y-axis. [FS04]

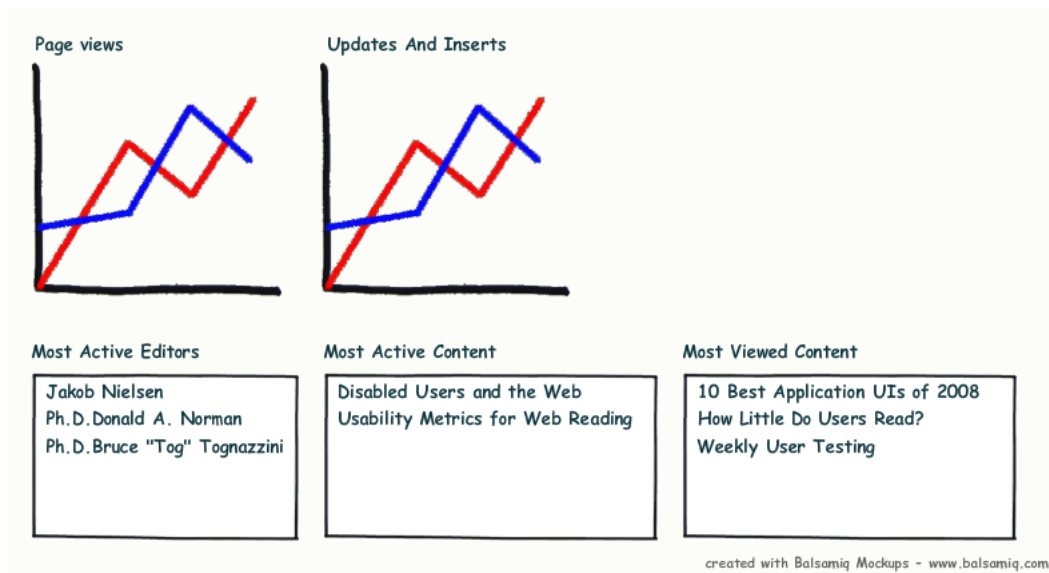


Figure 2.13: Activity statistics for a wiki

Chapter 3

Feedback

3.1 Messages

WordPress



Figure 3.1: WordPress flash

Right beneath the main menu opens up a message which fades once on page load from light yellow to yellow. It is primarily used to contain confirmations after saving a configuration or a document. In the theme chooser it is used for a goal-oriented help (See [LM90] for the five basic types of help) with a link to navigate to the right place.

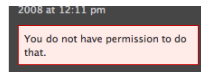


Figure 3.2: WordPress error message

There is also an error message which shows up very rarely but the WordPress team did a good job to show it very rarely. One occasion is when a post is left open in editing mode and its auto-saving cannot work anymore because you were logged out in the meantime. Another one can be seen if you enter the wrong credentials in the login form.

Good

- Because of the transition the flash addresses the motion sensors of the eyes and gets immediate attention. Since it just flashes once per page load, after the user has seen it, he can concentrate on the rest of the page.

Bad

- There is no way to hide it.
- Does not use the convention colour system.
- The design of the message says nothing about its severity.

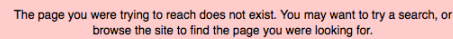
Suggestions

- If the Flash only contains a confirmation, make it vanish after several seconds.
- Use the according colours.

Confluence

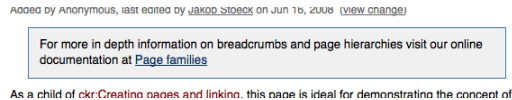
There are three types of messages:

The error message is displayed when something is not found, e.g. a pag. It is also displayed inside a wiki page if an included RSS feed cannot be found.



The page you were trying to reach does not exist. You may want to try a search, or browse the site to find the page you were looking for.

Figure 3.3: Confluence error message



Added by Anonymous, last edited by Jakob Stober on Jun 16, 2008 (view change)
 For more in depth information on breadcrumbs and page hierarchies visit our online documentation at [Page families](#)
 As a child of [ckr:Creating pages and linking](#), this page is ideal for demonstrating the concept of

Figure 3.4: Confluence info message



You have not configured a mail server. Mail notifications will not work properly. [Configure Mail Servers](#). ([Do not remind me again](#)).

Figure 3.5: Confluence warning message

The informational message is displayed at the top of the content or at its right side. It is most of the time coloured blue but there are also yellow and grey ones. It contains deeper information about a topic like an explanation, a summary or syntax help.

The warning message is displayed in administrative areas. It shows up when settings can be improved. Since they often are not important to every wiki installation (In a local installation you get the warning that the wiki is set up at "localhost" which could be not available on the network) they sometimes have a "Do not remind me again" link which let the message vanish and prevents it from being displayed again.

There is no feedback in form of a message when an action succeeded, e.g. saving a page or a preference.

Integration

All three systems use rudimentary approaches to a convention for user feedback messages. There is a trend identifiable in using different colours. However, the messages often do not obey the convention. They sometimes look different, have different positions and behave differently from page to page.

A web application has five types of help [LM90]:

1. Goal-oriented: "What kinds of things can I do with this program?"
2. Descriptive: "What is this? What does this do?"
3. Procedural: "How do I do this?"
4. Interpretive: "Why did this happen?"
5. Navigational: "Where am I?"

In a typical web application, feedback messages are the main source for helping the user. Since they provide different ways of helping there should be different kinds of feedback messages. This can be seen in the examined web applications.

If those message kinds are used in the same design throughout the application the user quickly gets comfortable in using its functions. It is important not to overlook an important message in a web application. The color code secures that the user is always informed about the state of the action. This "visibility of the system status" is one general principle for user interface design [Ni94]. When using messages on the frontend they are most of

the time not that important and should be smaller and less obtrusive, e.g. by removing the associated icon.

Error

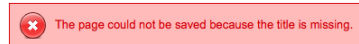


Figure 3.6: Best practice error message

This message is displayed if the last invoked action failed. The colour is red. The icon an X or stop sign.

”Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.” [Ni94]

Examples

- This attribute is already in use please change its name.

Attention

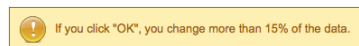


Figure 3.7: Best practice warning message

Coloured yellow, this message makes clear that you should pay attention to it in order for your next action to succeed. Since yellow is also the conventional colour for notes and remarks like post-it styled messages or little help boxes, the user could think those are equally important to the warning message. To avoid this, make the notes easily recognizable by styling them less serious. This is often done by mimicking post-its.

Examples

- Search results: No pages found

Confirmation

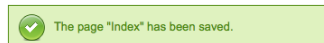


Figure 3.8: Best practice confirmation message

The colour is green. The associated icon is a hook. They optionally can fade out after some time, if you prefer having more free space. Notice that this is the only message type which could fade out.

If it is obvious that the action succeeded you do not need a confirmation. This is the case when you logged in and reside on your dashboard now or you saved a document and are now facing the frontend view of it.

Only display confirmations when it is not easily recognizable that the action succeeded, e.g. if you saved but stayed on the edit page or you logged out and stayed on the frontend which does not change much depending on your login status.

Examples:

- Document saved.

Information

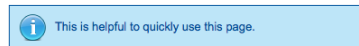


Figure 3.9: Best practice information message

Those messages are just further information for the to invoked action. Its colour is blue and indicates that it is not that important but more than the usual content. An iconographic "i" show its helping function.

Examples

- Search was executed in the following places: [. . .]
- If you do not enter a title, the first sentence of the main text is used.

3.2 What Can I Do

WordPress

Beneath a form field there are usually one or two sentences which explain its behaviour or give a sample input.

Next to the section headline there is often an action to create a new document.

There is an action palette floating on the right side which has always the same layout: Blue header, light blue content, dark grey footer, related section. This palette shows up whenever you edit a document of some kind.

Good

- Explaining text right beneath form fields speeds up its fill in and makes the user more confident as he is assured which part of the site he is editing.

Confluence

Almost always the Confluence toolbar is displayed in the upper right part of the screen. It accumulates most of the actions that can be done on a page.

There are action links next to headlines which correlate with this section.

Actions in menus often are accompanied by an icon.

Integration

Use Icons With Text

There are multiple icons which are instantly recognized, e.g. a pencil to edit a document, a plus sign or a document with an asterisk at the top right for new document etc. See →Image Language.

Most of the time however, icons should be accompanied by their description. That is the case if an action is only displayed once on a page. If that is not the case, e.g. in a list, the text can vanish most of the time.

Use The Sides

Actions are almost always displayed on the right or left side of the page or, with headings, at the right respectively immediately beneath them. Use these four conventional positions. The top and bottom of the page are usually only used for navigational or informational purposes.

Different Levels Of Exposure

”Let the user see clearly what functions are available.” [To92] Actions which are in the toolbar are completely exposed. Actions in menus have to be easily accesible. The only exception are advanced controls which should show up only demand.

Use Descriptions

Use the title attribute to further describe the action if it is not totally clear by its action text what the action does. But do not use it for obvious actions. This creates confusion with the user who already made a decision but has to consider it once more when the title appears.

Do not repeat something already written on the page.

If the space suffices it is better to write a short description directly next to the action than putting it into the title tag.

Use Conventional Action Names

Save, Edit, Delete, View, New, Site Map, Search, Browse and so on, are conventional names which have determined actions. Do not use your own words for those determined actions like "Find", "Look Up", "Erase", etc. The user would have to think if those actions do the same like he is used to. He then is not feeling confident in using those actions in the application. See →Terminology.

Keywords First

Buttons should have their keywords first. Instead of

- Insert Image
- Edit Link
- Add footnote

use

- Insert
 - Image
 - Link
 - Footnote.

Although not completely correct as you do not "insert" a link if it's already there, user tests have shown that this solution is more efficient as the user can spot the action faster. [To03]

3.3 Where Am I

WordPress

Which blog you are editing is displayed very prominently as the first headline.

Which section of the application you are using is shown by highlighting the according menu items.

What the actual page is about, like "Manage Posts" or "Current Theme" is in its title which is the second headline of the page.

Which application you are using is displayed at the bottom left of the screen with text and an additional icon.

Confluence

The information which section of the application you are using and which space you are editing are combined in the breadcrumb at the top left.

The space is also indicated by an icon to the left of the headline. Which is however the Confluence logo as long as you do not change it in the space.

The headline itself shows the title of the page. This can be the title of a wiki page but also the title of an administrative page. It is also shown at the end of the breadcrumb.

Bad

- Headlines are meaningless. What do you expect from a page with the title "Home"? In Confluence's case this is the profile page. What do you think is behind the headline "Advanced"?
- The breadcrumb has design flaws. If you move from your profile to the advanced settings page of a space you get this: "Dashboard > People > Jakob Stoeck > Browse Space > Space Details". Here "Jakob Stoeck" is used as the private space name but is located under "People". Furthermore "Space Details" is not the headline but the active section in the submenu. That is confusing to the user.

Suggestions

- In the headline tell what the page is about. If you want to go with "Home" for your profile say "Home Of *Username*" instead. "Advanced" should be "Advanced Space Settings"

Integration

If You Have A Breadcrumb, Do It Right

In a breadcrumb there should be fix locations for document types. You should always be able to tell where you are in your application and what you can do there by looking at the breadcrumb. See →Breadcrumb.

Highlight Active Menu Items

An active tab should fuse with the active content. See →Main Menu.

With multiple menus use similar highlighting methods to show that those are highlighted menu items and not active actions.

Use The Left, Top And Bottom

Information about the location of the user, i.e. which application, which section and which page, are always located on the left side of the screen. Most of the time they are located at the top and at the bottom but this differs if you have a vertical submenu which sometimes spans the whole content section. See →Layout.

Every Page Has A Headline

Do not use pages without headlines. That is also true for popup windows and modal dialogs, although with small windows the window title can be used as headline.

Tabs and breadcrumbs are no replacement for a headline. They fulfill other purposes. The headline says what you actually can do on the page. A tab says where you are. The headline therefore is most of the time different to the tab text but similar to the window title text.

3.4 Who Am I

WordPress

The mentioning of who you are in this application is very compact in WordPress. At the top right there is the text "Howdy, *username*". Which is linked to the profile.

The username is not changeable and says by default nothing about your status in the application. There is the possibility to add a nickname which can be displayed instead of the username.

The rights you can have are given with a role. There are five of them:

- Subscriber
- Administrator
- Editor
- Author
- Contributor

The only place to look up your role is the users table. It shows which role and therefore which rights you and the other registered users have.

Confluence

Your full name is displayed at the top right of the screen.

Dropbox

It is only apparent in your profile. There is no information in the layout with which role or username you are logged in.

Integration

Name At The Top Right

Always show who the logged in user is. Display his name as it is displayed to others in the application, i.e. full name or username depending on your application. This is especially important if a user can be logged in with multiple usernames or roles. The top right corner is an established convention for displaying this information. To make clear that this is your account, many applications greet the user with, e.g. "Hello *username*".

Chapter 4

Forms

4.1 Edit Forms

WordPress

Posts and Pages have very similar edit modes. Though the options are different: Pages are missing tags and categories, but it does have additional page parent, page template and page order options.

The edit mode consists of four main areas:

- Content with WYSIWYG editor
- Options beneath the content section
- An action palette with the most important actions at the right side
- A related section with links to lists of related document types beneath the action pallet

In the content section you can put the title in an HTML input field. The content can be edited in a WYSIWYG editor which can be switched to HTML mode. Tags are written comma-separated in another HTML input field. When adding them, they show up in an inline list with prefixed buttons that delete them again.

By selecting one of the "Add media" buttons, a lightbox window appears which opens the media upload, image gallery and media library in tabs. See →Upload Forms.

The options are located in multiple small compartments which first have to be expanded by clicking their header. They each accommodate very few controls. Often just an HTML input field and a small description.

Throughout the Write View simple comma-separated text in HTML input fields is used to allocate multiple values. It has a very simple structure.

The main actions are:

- View this post
- Save
- Delete post

When saving you can set its status to

- Published
- Pending Review
- Unpublished

and independent of the above settings

- Private

You can set a publish date.

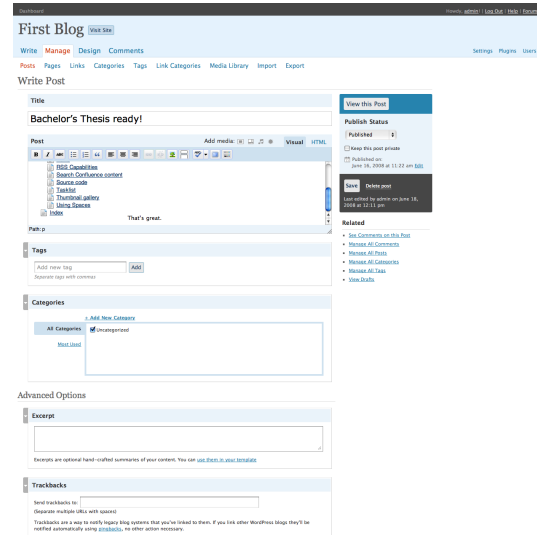


Figure 4.1: WordPress edit form

Good

- very clean interface allows concentration on the text
- all possible options on a single page
- WYSIWYG editor lets you switch to a manual HTML mode
- The edit mode defaults to a write mode, i.e. creating a new post/page, by selecting "Write" accessible from the first level top menu. This makes writing as easy as one click.
- The title text field is focused on page load which allows instant typing without clicking or tabbing to the field.

Bad

- Grouping of options is too finely grained. Often enough the minimized option header takes nearly as much place as the content of the option group.
- You have to remind of the tags and categories you want to denote or go to another web page and have a look. Additionally the risk of mistyping remains.
- There is no save button at the bottom of the page. You have to scroll up again after you have done advanced settings.

Suggestions

- A grouping which is so fine-grained does not make sense. Because of the bold headings it does not even save space. Think about grouping relevant options and position them according to usage.
- For attaching multiple values, instead of the current error-prone solution, use the already very popular Auto Complete design pattern as it is used in many email clients.
- Always put a save button on the top and bottom of a form.

Confluence

The Confluence edit view is of a rich functionality. It combines

- an input field for the title of the page,
- a save and cancel button,
- text editing in wiki mode with an AJAX preview,
- a help section on the left which explains the wiki syntax,
- a place to leave comments of the current change,
- an option to mark this change as minor, therefore not notificating others of the change

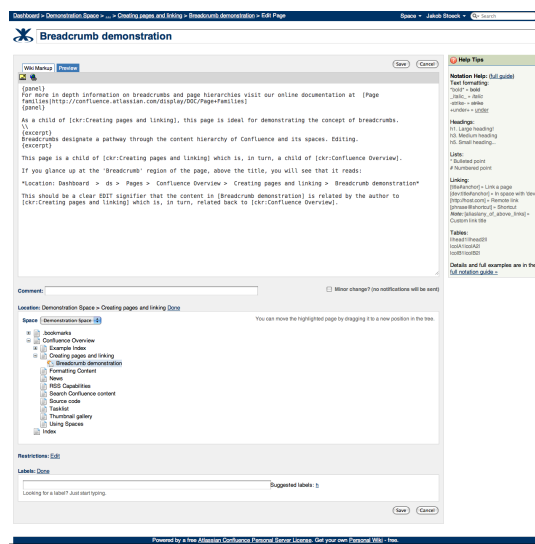


Figure 4.2: Confluence edit form

As an alternative way for wiki syntax there is also a rich text editor. You can choose which one to prefer prospectively with one click.

The following options are collapsed by default:

- a control to place the current wiki page in the space hierarchy
- a control to set viewing and editing restrictions
- an auto complete labels field

The collapsed option controls have the same boundary conditions. They can be expanded by clicking the adjacent "edit". It then changes its text to "done" which collapses the control again. The done has to be clicked in order to confirm changes made in the control. If there are made changes, the done will flash once while collapsing the control.

Good

- All controls are sufficiently labeled and easy to use
- the wiki notation help does not stop at trivial formatting options. It is well-structured.
- switching between editing and previewing is AJAX-based, therefore instant. It allows a fluent workflow.
- auto-completion within the labels field allows faster choosing and prevents misspelling
- combining collapse/expand controls with a confirmation save space and clutter

Bad

- In the text editor there are only buttons for image and link insert. Well-known actions like selecting a text and clicking e.g. a bold button are not possible in wiki mode.

Integration

The Sections

The main sections are the following in both edit modes:

- Editing section with
 - a header input field,
 - a WYSIWYG Textarea,
 - a button to switch to code mode (Wiki syntax or HTML) and
 - a button to switch to a preview of the page.

Adding Media should be easily doable in this section.

- An option section with the following options:
 - Tags

Figure 4.3: Minimal edit form

- Allow Comments
- Set viewing and editing privileges
- An action palette with the following actions:
 - Save button
 - Set publish status (Drop down menu)
 - Show last edited time.
 - Show autosaved time.
- An related section:
 - Comments/Discussions on this post/page

Differences between page and post edit

Wiki pages and blog posts are different in the following matters.

Blog Posts

- Posts have publishing dates and normally do not change after that.
- Posts should support trackbacks and pingbacks.
- Posts are associated to a single user.
- Comments can be made.

Wiki Pages

- Wiki pages change over time.
- A wiki page is often a conglomeration of the work of multiple users.
- The progression of the change should be viewable.
- There is a wiki syntax.
- Often, there is no comment system but a discussion system on a different page.

This leads to the insight that the edit modes have to be customised to fit those differences. There are only changes in the options section.

Code Mode

For experienced users it is often more precise and faster to write parts of the page directly in code mode. Additionally, there are complexity levels a WYSIWYG editor often cannot handle.

In Wiki pages this is wiki code, in blog posts this normally is HTML code. The best solution would be to provide both modes for both document types.

Autosave

Do an autosave in the smallest interval which can be done efficiently and indicate that with the exact date and time above the fold near the top save button or the textarea.

Examples:

- Draft autosaved on 17.06.2008 11:47
- Draft Saved at 11:47 am

Also autosave on page closing. This could be a browser crash. Next time the page opens ask which version to use if they differ.

Focus The Title Input Field

The cursor should be at the rightmost position. Focussing on page load allows faster editing and a better user experience because it eliminates one step.

4.2 Login Form

WordPress



Figure 4.4: WordPress login form

Under the big WordPress logo are located:

- Two input fields labeled "Username" and "Password",
- a "Remember me" checkbox,
- a "Login" button on the bottom right and
- a link asking "Lost your password?".

The ubiquitous dark grey bar on the top is already introduced and features a link back to the frontend of the blog.

Confluence



Figure 4.5: Confluence login form

The base layout of Confluence is already introduced. In the dark blue bar you have the two links "Log In" and "Sign Up".

There is a small explanation what this form is for and beneath:

- Two input fields "Username" and "Password",

- a "Remember my login on this computer" checkbox,
- a "Log In" button,
- a "Signup" link and
- a "Forgot password?" link.

Integration

Introduce Layout

The login screen should use already the layout of the application. Header and footer with their links should display.

Fields

It should contain at least the following fields with their according labels:

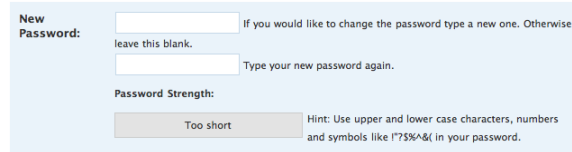
- Logo of the application
- Username
- Password
- Remember me checkbox
- Forgot password link
- Sign up link
- Log in button

Route To The Dashboard Or An Action

On a submit, the log in form should by default route to the application's dashboard. If an action was invoked that requires log in, the action should execute automatically after successful log in and redirect to the according page.

4.3 Password Update

WordPress



New Password: If you would like to change the password type a new one. Otherwise leave this blank.

Type your new password again.

Password Strength: Too short Hint: Use upper and lower case characters, numbers and symbols like !?%*^&() in your password.

Figure 4.6: WordPress password update

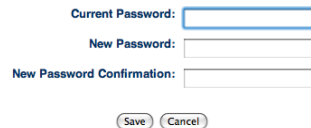
In the user profile you have a section called "New Password" with the following fields:

- New Password
- field to type password again
- Password strength indication while typing (values: Too short, Bad, Good, Strong)

To confirm you "Update Profile".

Confluence

Enter your current password, and the password you want to become your new password in the form below, and click Save.



Current Password:

New Password:

New Password Confirmation:

Figure 4.7: Confluence password update

In the user profile, section password, you have three input form fields:

- Current Password
- New Password
- New Password Confirmation

You can "Save" or "Cancel".

Dropbox

The section is called "Reset Password" and like Confluence consists of three parts:

- Old Password
- New Password
- Confirm Password



The image shows a password reset form. On the left, the text "Password reset" is displayed in a light blue font, with a vertical blue line extending downwards from it. To the right of this line, there are three input fields stacked vertically. Each field is preceded by a label: "Old password:", "New password:", and "Confirm password:". The input fields are empty rectangular boxes with thin borders.

Figure 4.8: Dropbox password update

Integration

Ask for the current password

Only letting the password change when the user knows the old password prevents abuse to some degree.

Of course if you do that, you have to provide an alternative way to change the password if the user has no chance on getting to the old one. This could be e.g. a description on how to change the password directly in a configuration file, assuming only the user has access to this part of the program.

Indicate Strength But Don't Prevent Easy Passwords

A password strength indicator is helpful for inexperienced users.

But do not interfere when they choose easy guessable passwords. Especially in team work environments also a hard password can be socially hackable. Letting them choose easy-to-guess passwords at least prevents the infamous post-it under the keyboard.

Fields

The form should consist of a small description and the following fields:

- Current Password
- New Password
- New Password Confirmation
- Password strength indicator

4.4 Search Forms

WordPress

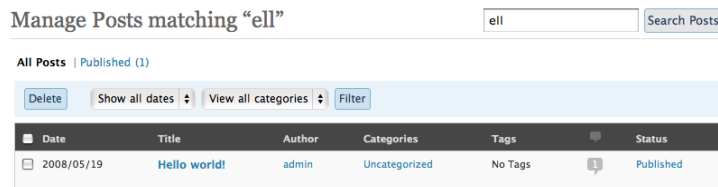


Figure 4.9: WordPress search box

There is no search form which searches through all document types but a simple search box on top of each list of a document type. If the search is executed it acts as a full text filter for the list and displays only the matched entries but still in the exact same layout.

Good

- Easy to search for a particular document type.
- The familiar way of displaying the document type is adopted for the search results.
- Since the result page uses the exact same layout as the normal listing does, the familiar actions like delete, filter etc. are reachable.

Bad

- No way to further specify the filtering than just a full text search.
- No displaying of multiple document types, which would match with a full text search, possible.

Confluence

The search form is only reachable by entering a search term in the ubiquitous search form field. Once invoked, it offers a sidebar which narrows down the search results by a space, a modified date and/or document type and offers the option to group them according to their document type or their associated space. Each search result features a snippet of its inheriting text with the search term highlighted, a link to it, a link to the space and its last modified date. On top of the search results is a box which informs where the search searched, which options were chosen and which search terms were used.

Good

- When you arrive at the search form the first search is already invoked which means you possibly do not need to use the options.
- All options are grouped in one sidebar.
- All document types have their according icon.
- Narrowing the search by date is possibly by easy comprehensible text links.

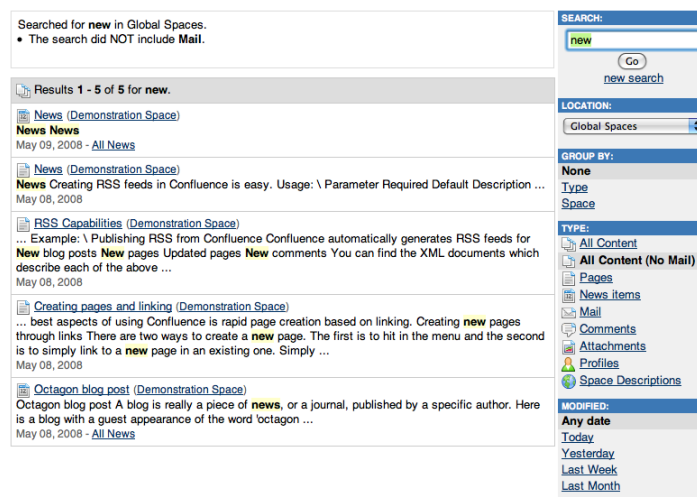


Figure 4.10: Confluence search box

Bad

- The outlook of the box which informs about the executed search options does not look like an information box.
- Line breaks, the usage of lists and usage of bold text in this box is used in a confusing manner.
- Date does not expand in detail view on hover like in the restments of the application.

Suggestions

- Use the layout convention for info boxes: blue color and an icon with an exclamation mark.
- Further formalize the display of the chosen options in this options so it can be grasped faster.

Integration

Site-wide Search With No Options At The Beginning

The default setting for a search functionality in a web application has to be application-wide, i.e. over all document types. It should be located on every page on the top right corner. showed that it is best to have a simple search box for the first search and to give advanced search functions only after the first search on a particular page.

Flat Search Results

The search results should be displayed in a sortable flat list ordered by relevance. Because in such a list, hierarchy is not displayable, the results should have an adjacent breadcrumb visualizing their position inside the application.

The list should have at least the following headers:

- Name

- Last Modified Date
- Document Type

Excerpts of search results should not have their own column as you don't need to sort them. Additionally they widen the table which makes it unclear. Print those under the title of the page like common web search engines.

Grouping And Cropping

Narrow down the search results by adding links which crop by at least by document type, specific blogs or wikis and modified date. Since most of the time users do not know the exact time, a document was written, provide relative modified dates. Popular are: "today", "last week" and "this year".

4.5 Add Media

WordPress

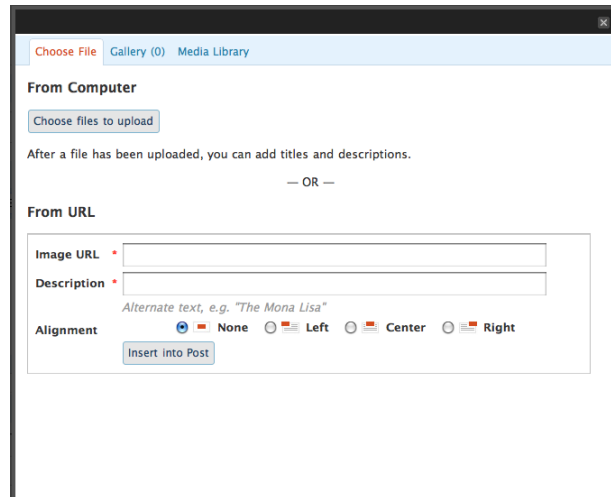


Figure 4.11: WordPress add media

WordPress differentiates between adding "Images", "Video", "Audio" or "Media". There are only small differences between them, e.g. images can have an alignment, while videos do not. If you add one of those types, a lightbox modal window appears which gives you the following options:

- upload from your computer via an HTML file control
- upload from an external URL
- select the gallery (which consists all uploaded media files of this post)
- select one out of the media library

If you have chosen the wrong media type, the modal window will change accordingly after the upload.

Dependent on the media type you can insert a subset of those attributes: title, caption, description, URL, alignment, size

Good

- usage of AJAX makes adding a three-click-process without changing the page.

Confluence

Confluence makes a difference between inserting media and attaching media. You can only insert images into a page. On invoking this action a pop-up window opens and let you insert the following options:

- upload a file by with a HTML file control
- choose one out of the attachments of the page



Figure 4.12: Confluence add media

- filename
- thumbnail (yes/no)
- alignment (left, centre, right)

While you can only insert images, you can attach all kinds of media. The control is also a HTML file control, which can be replicated with a click on a button to upload more than one file at once.

Integration

Although the termini vary, the applications have much in common.

Differentiate Media Type On Insert

Different buttons for different media types.

Stay In The Main Window

Do not change the content of the main window. Use lightbox modal windows or at least pop-up windows.

Offer Options

Those are:

- Title
- Alignment (left, center, right)
- Size (Best: thumbnail, medium and fullsize. At least: thumbnail and fullsize)

Offer Two Media Pools

Attachments: All media associated with the current document should be available in an own overview. The WordPress term "Gallery" is not obvious enough. Confluence's "Attachments" is known by most users e.g. via e-mail software.

Media Library: Apart from that a pool of application-wide uploaded media should be offered. The term "Library" is well-known through UNIX and various Apple Software like iTunes.

Chapter 5

Listings

5.1 Alphabetical Listings

Confluence

















All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z		
Title	Last Updated By	Updated
 .bookmarks	Jakob Stoeck	May 17, 2008
 Breadcumb demonstration	Anonymous	May 08, 2008
 Confluence Overview	Jakob Stoeck	May 25, 2008
 Creating pages and linking	Anonymous	May 08, 2008
 Email archiving	Anonymous	May 08, 2008
 Example Index	Anonymous	May 08, 2008
 Formatting Content	Anonymous	May 08, 2008
 Index	Anonymous	May 08, 2008
 Jira integration	Anonymous	May 08, 2008
 News	Anonymous	May 08, 2008
 RSS Capabilities	Anonymous	May 08, 2008
 Search Confluence content	Anonymous	May 08, 2008
 Source code	Anonymous	May 08, 2008
 Tasklist	Anonymous	May 08, 2008
 Thumbnail gallery	Anonymous	May 08, 2008
 Using Spaces	Anonymous	May 08, 2008

Figure 5.1: Confluence alphabetical listing

The alphabetical ordered listing is only available for pages. By default it shows all pages but you can isolate the display by clicking one of the letters of the above horizontal bar. In this bar the leftmost button shows again all pages in alphabetical order. The columns are: Title, Last Updated By and the time of the last update. Clicking on the title or the name leads you to the page or respectively to the profile of this user.

Good

- Clear and simple constructed.
- Pages have their corresponding icon.
- Uses the table design which is used throughout the page.
- Dates are human-readable and very easy to compare due to their lack of granularity, e.g. May 17, 2008
- If there are no pages with a selected letter, a message is displayed which says so instead of just displaying nothing.

Bad

- Users do not have their corresponding icon. This violates the common behaviour of having each document type always displayed with its icon.
- The message displayed when there are no pages should be in a message format. → see messages
- There is no visual clue of which letter is selected.

Suggestions

- Highlight the selected letter in the horizontal bar.

- Append an icon to each document type like in the rest of the page.
- Use the message convention to display them.
- The date could be formatted in an even more human readable like relative dates.
See →dates

Integration

Do You Need Them?

Often alphabetical listings are not very practical. If you want to search for a title like in a book index you are much faster by using the search function. They do make sense in environments where no other attribute of a document is so important that it changes document order. That is the key with non-hierarchical wiki pages. Just because one wiki page is newer or has more content does not mean it should be displayed at another position in a generic list.

There are use cases a alphabetical separated list makes sense but many of them can be also realised with alphabetically sorted flat lists.

Display All Content By Default

By default the list should display all content. It should have all the letters of the alphabet as buttons aligned horizontally on the top which filter the list items by starting letter.

5.2 Recent Activities List

Confluence

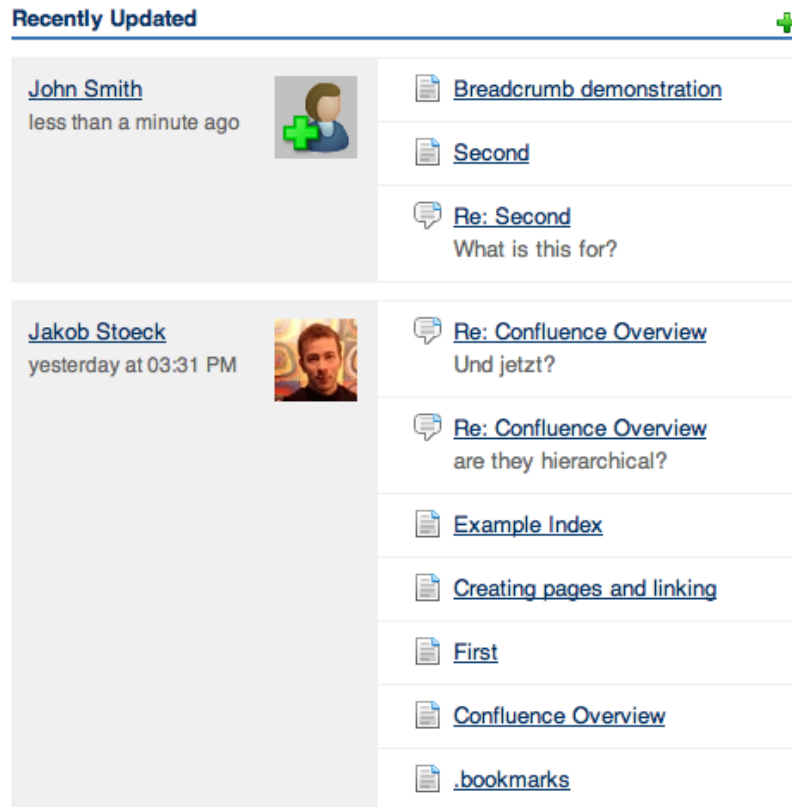


Figure 5.2: Confluence recent activities list

The list of recent changes is sorted descending by date and then grouped by its editor. Every document type is associated to its icon.

Clicking on the plus sign on the top right corner extends the list by the next 10 edited documents. This is repeatable at will.

This view is used in the dashboard and in the pages list.

Bad

- The date of the last change is displayed by user, not by document.

Suggestions

- Display edit date besides the documents entries.

WordPress

It is used only on the dashboard for recent comments. The first 105 characters of the first comment are displayed. Beneath it its author, post and a direct link.

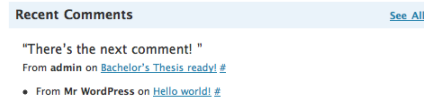


Figure 5.3: WordPress recent activities list

The next comments are displayed in a simple html list without the content. On the top right is a link to all comments ordered by date.

Integration

In a web collaboration application the view of Confluence is very helpful to quickly get an overview of the last changes and by whom they were made.

This is right for the wiki and file system part, but the blog part is usually just from one author. It does not make much sense then.

Differences Between Wiki, File System And Blog List

In the wiki and file sharing part of the system, a user-based grouping provides a clear overview of recent changes in a team collaboration environment. The developers of Confluence have made a good job. Though it cannot be copied 1:1 when incorporating file system changes, too. More than any other document type, file system changes emerge in big numbers. If a user adds 200 files, the first 200 lines of the list would contain those files, which is presumably not what another user wants to know, when he looks at recent activities. The solution would be to truncate the list per user to a fix amount. E.g. if a user makes more than 25 changes, only the last 20 are displayed with a link beneath ("show all 216 changes...") which shows all changes. This way, other users see a bigger change, but can get a quick overview nevertheless.

This type of view should only be used on the dashboard or other recent activity lists. Since a blog of a team collaboration application is usually practiced by one author, a user-based grouping does not make sense.

Every Document Type Has An Icon

Every document type should use its application-wide icon.

Always Show A Date

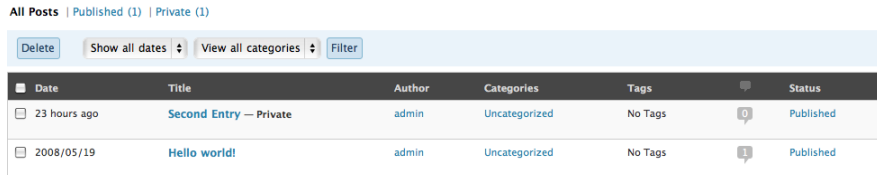
In a list, always associate a document with its last modified date.

Let The List Extend

Provide a way at the top right of the list that either routes to a list with more entries or that extends the list in place by e.g. 10 more entries.

5.3 Flat Listings

WordPress



The screenshot shows a WordPress interface with a flat listing of posts. At the top, there are navigation links: "All Posts", "Published (1)", and "Private (1)". Below these are buttons for "Delete", "Show all dates", "View all categories", and "Filter". The main content is a table with the following data:

Date	Title	Author	Categories	Tags	Status
<input type="checkbox"/> 23 hours ago	Second Entry -- Private	admin	Uncategorized	No Tags	Published
<input type="checkbox"/> 2008/05/19	Hello world!	admin	Uncategorized	No Tags	Published

Figure 5.4: WordPress flat listing

The "Manage" section uses a flat list for all of WordPress's document types:

- Posts
- Pages
- Links
- Categories
- Tags
- Link Categories
- Media Library

The list consists of

- a filter search box with document-dependent options on the top,
- a table where each row represents an document

At the left border are checkboxes to choose documents. The chosen ones can be deleted by the button above the checkboxes. The top checkbox checks all documents.

Good

- It abstracts from the actual document, so managing another document feels the same.
- It is very easy to learn.
- Managing many documents is fast and easy.
- It can be displayed instantly with many documents
- Table heading names are uniform across different document types (e.g. the "author" is the creator of a document independent wether it is a post or page)
- Dates are easy-readable ("[Updated] 23 hours ago")

Bad

- It does not cope with hierarchy very well. Although it displays it with indentation of the rows, you cannot change it or filter by it.
- No possibility to add a new document to the list
- Filtering happens after form submit only.
- Since every listing of a document type looks nearly the same you have to make sure, you are in the right section.

Suggestions

- Users are accustomed to instant filtering. At least focus the input field again after a submit.
- Allow a relocation of rows if there is a hierarchy.
- Make visual clues for different document types like site-wide used icons.
- Display those parts of the documents which are common, like titles, dates, editor etc. in the same way. Start designing from a general view and only change those parts which simplify navigation for this special document type.

Confluence

It has a delete icon on the right of every row. Above the tabel is a "delete all" button. The list is sortable.

Integration

Batch Support

In the leftmost column are rows of checkboxes. The header row checkbox selects and deselects all other rows.

Action At The Top And Bottom

The button or drop down menu of the actions should be placed at the top and the bottom of the list. It is important that it is placed directly above respectively below the checkboxes, not just near to it. This helps the user associate the checkbox with the action button.

Sortable

Since those are lists with tabular data, it is a good idea to make them sortable. The header text should indicate this with underlined text. Clicking again reverses the order. A triangle with the tip at the top (ascending) or at the bottom (descending) indicates the order.

Zebra Tables Are No Cure

YEAR	NUMBER, IN THOUSANDS			RATE PER 1,000		
	BIRTHS	DEATHS	MARRIAGES	BIRTHS	DEATHS	MARRIAGES
1960	3,632	1,452	1,667	24.1	9.6	11.1
1965	4,097	1,529	1,531	25.0	9.3	9.3
1967	4,300	1,633	1,518	25.3	9.6	8.9
1960	4,258	1,712	1,523	23.7	9.5	8.5
1965	3,760	1,828	1,800	19.4	9.4	9.3
1970	3,731	1,921	2,159	18.4	9.5	10.6
1971	3,556	1,928	2,190	17.2	9.3	10.6
1972	3,258	1,964	2,282	15.6	9.4	10.9
1973	3,137	1,973	2,284	14.8	9.3	10.8
1974	3,160	1,984	2,230	14.8	9.1	10.5
1960	3,632	1,452	1,667	24.1	9.6	11.1
1965	4,097	1,529	1,531	25.0	9.3	9.3
1967	4,300	1,633	1,518	25.3	9.6	8.9
1960	4,258	1,712	1,523	23.7	9.5	8.5
1965	3,760	1,828	1,800	19.4	9.4	9.3
1970	3,731	1,921	2,159	18.4	9.5	10.6
1971	3,556	1,928	2,190	17.2	9.3	10.6
1972	3,258	1,964	2,282	15.6	9.4	10.9
1973	3,137	1,973	2,284	14.8	9.3	10.8
1974	3,160	1,984	2,230	14.8	9.1	10.5

Figure 5.5: Better alternative to zebra tables

When a table alternates its row color between two colors with each row, it is called zebra table. It is used in many applications and is supposed to make finding information easier. Although it does not help much in finding the right value with longer rows. You have to really follow the stripe to get to the right place on the other side of the list. It is as slow and error-prone process as without stripes [En07].

But users like striped lists.

A better working alternative is to always group the next three rows instead of a single row and separate this group with a horizontal line. For aesthetic purposes you can additionally stripe every second group to get a similar zebra table effect. This way, stripes are further apart which makes it easier to overview long rows in complex lists.

5.4 Grouped Lists

In a grouped list, there is at most one sub list for each document type.

Confluence

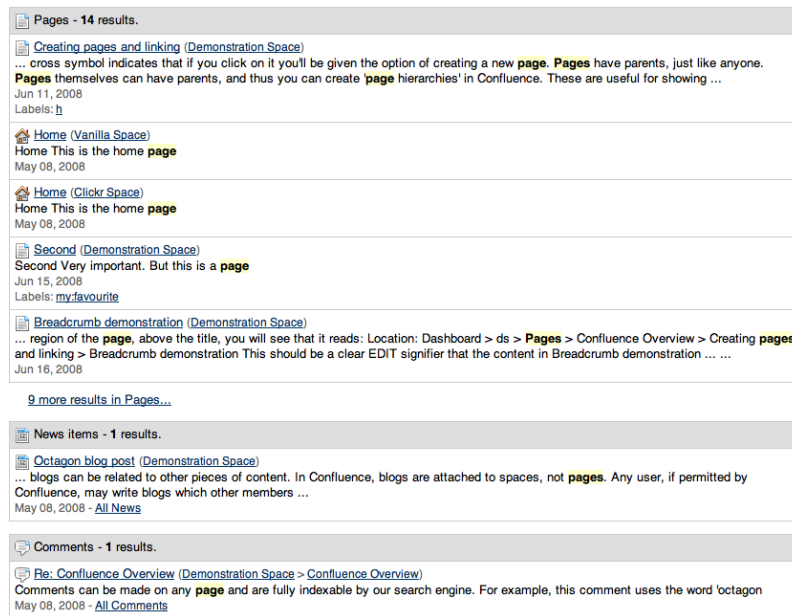


Figure 5.6: Confluence grouped list

The only grouped list exists on the search result page. It is by default not grouped but can be by document type or space.

When there are more results for one document type, the first five are displayed and beneath the list a similar link to this: "x more results in Document Type A. . .".

Good

- Sub lists are truncated with a "more results" link.
- The amount search results for each sub list are shown.

Bad

- A sub list gets always truncated to five results, even if the overall results are not many.
- Browser back button does not work.
- Clicking a "more results" link routes to a page where only those document types are displayed. You have to orientate once more.
- It is hard to spot the desired sub list. The heading fuses with the rest of the table.

Suggestions

- Truncating the results list to five results per document type should only happen when it is useful, i.e. the overall amount of search results is more than 20 items. It does not make sense, truncating a sub list to five when the total amount of search results is just eight.
- The browser back button is very important. Ensure it works on all important pages. If it does not work, provide a clear way to get one page back.
- Do not route to another page if someone clicks the "more results" link. Instead load them in place.
- Like labels in forms it is easier to browse through such a list if all the headings are left to their actual sub list.

Integration

With Apple's Spotlight search feature and the Microsoft Windows Vista search feature, grouped lists moved in the popular operating systems world. They are useful when searching something and you do not want to specify the document type before.

But their usage is narrow. They are good for results at first sight. However, longer lists are better searchable with sortable flat lists.

You should pay attention to the following topics:

Truncate, But Only If It Is Useful

Set a maximum length of items for the whole list, e.g. for five document types set it to 15. Amount of searchable document types multiplied by three is a good thumb rule. Always make sure you use this maximum length. If 6 pages, 5 blog posts and 4 files have been found you do not need to truncate at all. It is the same with e.g. 7 pages and 8 blog posts. Set the maximum length of a sub list to $\text{ceiling}(\text{maximum total length} / \text{total amount of found document types})$. This of course raises the total length by one in extreme cases but you assure not wasting free slots in your list.

Labels Belong To The Left

In a grouped list, you should not use the sub lists like tables with headings but more like groups in a form. To allow faster browsing, the heading belongs to the left.

More Results Appear In Place

If a user clicks the "more results" link the results should just appear where he clicked the link. Do not navigate to another page. If you have to reload the page, make sure to jump to the place the user clicked by adding an HTML anchor to the url.

5.5 Hierarchical Order Listings

Dropbox

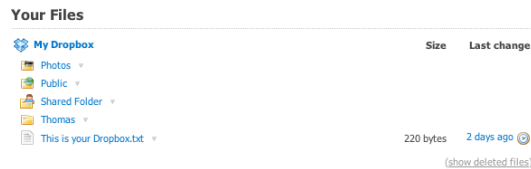


Figure 5.7: Dropbox hierarchical listing

Due to its nature updating and deleting files in Dropbox is outsourced to the file system. Managing comes down to viewing, uploading and restoring. The main place where this takes place is the browse files category. It's different designed then the usual file trees as it only shows

- the root,
- the current subtree root
- and its children.

Clicking on a folder therefore shows the folder and its first level contents. Two additional columns show the size and the date of the last change. The date is expressed in days since the last change. If you click on the date, it presents a listing of the previous document versions. Clicking on text files shows them in the browser. Images are shown in a lightbox. Next to each folder or document lies a gray arrow which accommodates the following options:

- Upload here
- Download as zip
- Restore files
- View as gallery (only for folders containing images)
- Share
- Share details (only for shared documents/folders)

Bad

- Last change is written in days. Although this gives a quick overview of the last changes, a time would be appreciated with more recent changes.
- Because only the current subtree is displayed, navigation is harder than it should be.

Confluence

The only documents which are hierarchically ordered are pages. Those are displayed throughout the application in the same fashion: A document tree. There are no folders, pages can contain additional pages. You can rearrange the document order by dragging the page to its new position. With small plus signs you can expand the document content.

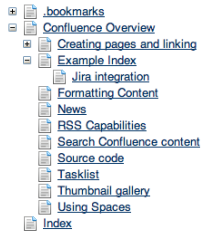


Figure 5.8: Confluence hierarchical listing

Integration

Few Actions, Few Document Types

A hierarchical list is different from the rest of the lists. It is not designed for information about particular documents but to let the user navigate in it. There are many examples like the Windows Explorer or Confluence's hierarchical view that in hierarchical lists you append nearly no information to the documents.

It should contain

- Document type icon
- Name of the document
- Plus/minus sign depending if there are sub items and if it is open or closed.

It should support

- Relocate
- Delete

Make Relocation Easy

You can relocate a document on its present hierarchic level or put it several levels up or down. Those two actions should be displayed differently. For the first one, there is often usage of a line which would represent the new location when you drag over. For the second one, set the background of the new parent item to a fitting colour.

If You Need More Information, Display Only One Level

If you have multiple document types, the rare information about them in a hierarchical list sometimes does not suffice to navigate efficiently between them. In this case use a semi-hierarchical list:

The parent item is displayed with a link back to its level, the content is displayed as a flat list. Documents with more documents as content should display so by adding a plus sign to them.

Chapter 6

Navigation

6.1 Breadcrumb

Confluence

Dashboard > Demonstration Space > Confluence Overview

Figure 6.1: Confluence breadcrumb

In contrast to other popular Wiki software like Mediawiki, Confluence's pages are organized hierarchically. The breadcrumb is used to visualize this structure, e.g. main and sub pages, but also to show the action, which is invoked on the current page, e.g. Dashboard > Space Title > Page Title > Edit Page.

As soon as there are at least five items in the breadcrumb all but the first two and the last two get substituted by a "...". Clicking on the dots expands the breadcrumb to its original size.

(Example: Dashboard > Demonstration Space > ... > Example Index > Jira integration)

Good

- Breadcrumb is always visible.
- Position is on top, which is preferred by many experts.
- Greater than signs are used, which has proven to be the best way to show users a "inherits the following" association.
- Every content except the current page is clickable.
- Current page is not clickable.

Bad

- It is mixing navigation, like page titles with actions, like edit, e.g. when you navigate to your profile, you get this breadcrumb: Dashboard > People > Jakob Stoeck > View Profile, where everything except the last part is a link. If you click the name, you will not navigate further but to the exact same page.
- Some links are not useful as they load the exact same page where you are.
- Sometimes the last part of the breadcrumb is used as the title of the page. There is no other heading.

Suggestions

- A breadcrumb should be used to visualize where you are and how you get there, not what you are doing at the moment. This mixture in Confluence confuses the user and hinders the user to get a mind model of the website. Do not put actions into breadcrumbs.
- Do not use a breadcrumb to replace the heading. A breadcrumb should only be a further addition to a web application.

Dropbox

There are only two levels throughout the site.



Figure 6.2: Dropbox breadcrumb

Good

- Current page is not clickable
- Usage of »-character. It's a convention to use this or the "greater than"-sign >.

Bad

- Is sometimes used as an exchange of a headline. This is bad practice.

Suggestions

- Just use for what it's for: Indicate where you are. Titles are for headlines.

Integration



Figure 6.3: A best practice breadcrumb

A breadcrumb is important to show a hierarchy of the page, so the user can build his mental model upon it. It should have the following characteristics:

Above The Fold

Experienced users often use the breadcrumb more often as the main navigation so it should be positioned very easily reachable.

The Separator

The usage of the character > or » are a convention that should not be violated. They show a hierarchy and are easily understandable. The choosing is more aesthetically than semantic. Although the first one implies a "greater than" i.e. a hierarchy where the other ones semantic is more "leads to" which can be used when the displayed breadcrumb does not display a fixed hierarchy, e.g. a trace path.

The Rightmost Item

The page title should be always the last item. Most of the time it is congruent to the heading on the webpage. It should be no link and emphasized. The convention is bold text.

The Leftmost Item

In most web applications this would be the dashboard as it is the entry point.

Always At The Same Place

A breadcrumb is only usable if it is always there. If there is a situation where you cannot use it, chances are high that you should not use a breadcrumb in the whole application.

6.2 Buttons

WordPress



Figure 6.4: WordPress button



Figure 6.5: WordPress button in hover state

When this text was written four months ago, WordPress was at Version 2.0 which had lots of different kinds of buttons. With version 2.5 they have unified their buttons to one type. Because it is technically not really a button, but a link which looks like a button, there is more styling possible.

- Many have accesskeys
- Hover states
- Overall outlook

They now follow the design rule of links: The text is in blue color which becomes orange on hover. The text colour of buttons which delete something is red on hover. Buttons are used when it:

- initiates an action connected to a form (edit forms, filter, search)
- deletes documents
- switches out of the backend (view page, visit link)

There are no buttons to cancel a current action.

Good

- Their content always says what the button is doing if there could be a misunderstanding. Instead of a generic "Search" you have "Search Pages".

Bad

- Because they have no active state, you do not see whether you clicked the right button, as long as you have not released the mouse button.
- "Add"-functionality is no button. That is incoherent behaviour.

Suggestions

- Make buttons have an active ("clicked") state.

Confluence

There are very few buttons in Confluence. The only occasion where you see some is when there is something to save, like a page or a setting. Those are labeled "Save", "Post" or "Add" and have "Cancel" besides them. They always appear combined and in that order. They have no further styling.

Bad

- Button text is too generic. Especially when there are multiple instances of it on the page.

Suggestions

- Try to communicate what the button does, e.g. "Save page", "Cancel changes", etc.

Dropbox

Like Confluence, Dropbox uses buttons to confirm a form.

Good

- Default button is highlighted.
- Buttons have meaningful text.

Integration

A button should be used instead of a link if one or more of the following cases are true:

- a form is submitted
- a document add or delete action
- by clicking it, you leave the backend.
- it is the main action of the page

Meaningful Text

Buttons should have meaningful text. Generally considered bad are e.g. "OK", "Cancel" etc. Use instead "Save", "Back" etc. If there is not absolutely clear what document type will be saved, add a suffix to the button text. E.g. "Save Post", "Back To Post List".

No Cancel

There are some problems with cancel buttons.

- The user has to guess what action invokes "Cancel": Is there a possibility to save it as a draft, like there would be when he closes the window?
- How much of the current action is canceled?
- Where will he be sent afterwards?

Cancel buttons are like reset buttons in forms generally a bad idea. Most of the times, their invoked action can be substituted by hitting the back button in the browser. In those cases where you need one, address the aforementioned issues.

Button = Link + Javascript

The three used techniques for a button are

1. HTML input buttons
2. Links styled as buttons with JavaScript
3. Flash buttons

The first one is the easiest to handle by the administrator and most recognizable by the user.

The second one is a trend in the web nowadays. They are more stylable, are treated like normal text in the browser (find, copy etc. works) and are browser independent displayed equally. Since the action is invoked by JavaScript you get over the restriction of one submit button per HTML form.

The third one is often used if the invoked action takes a longer time. E.g. in an upload form, this button is aware of the status of the upload and can display accordingly.

Most web applications do not work accordingly without JavaScript. If you are in that environment where JavaScript is surely activated, the second solution has most advantages. Otherwise solution one should be chosen. Solution three should only be used selectively and with a fallback solution of type one.

6.3 Dashboard

Today, it is a de-facto standard to use a dashboard as starting point of a web application. They usually show recent events grouped by document type.

Wordpress

Used as the backend starting page, this page allows a quick overview of the blog. It consists of 7 content sections

- Right now: with New Page and New Post action and statistics about the blog
- Recent Comments
- Incoming Links from other blogs
- Most Popular Plugins
- WordPress Development Blog: Latest Headlines
- Other WordPress News

Whereas the last two sections can be replaced by any other RSS feed.

Good

- Many different tasks can be started with one click where otherwise navigation through menus would be necessary

Bad

- Not configurable enough. The most used tasks and their preferred layout are not the same for every user.

Suggestions

- Provide modular additions to the dashboard which can be included voluntarily.

Confluence

Confluence has five sections:

- Explanation what you can do with that page and their terms.
- Tabbed navigation for the spaces

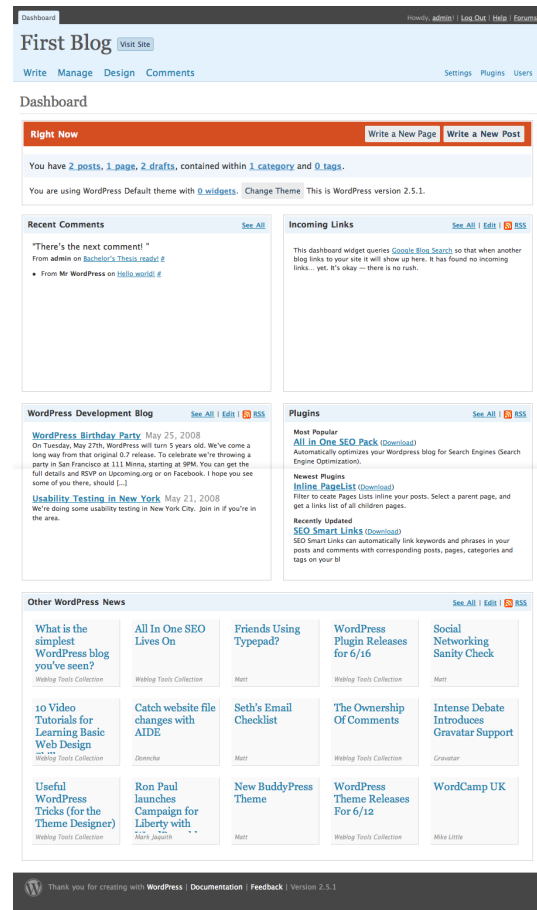


Figure 6.6: WordPress dashboard

The screenshot shows the Confluence dashboard interface. At the top, there is a navigation bar with the user name 'Jakob Stoock' and a search bar. The main content area is divided into several sections:

- Welcome to Confluence:** A introductory message stating that Confluence is an enterprise wiki designed for sharing information.
- Where do I start?:** A section explaining that content is organized into spaces and providing a list of available spaces: 'Clickr Space', 'Demonstration Space', and 'Vanilla Space'. It also includes links for 'Create a space', 'Feed Builder', and 'People Directory'.
- Recently Updated:** A list of recently updated pages, including 'Breadcrumb demonstration' by Jakob Stoock, 'Second' by John Smith, and several 'Re: Confluence Overview' entries.
- Favourite Pages:** A list of pages recently added to the user's favourites, including 'Example Index', 'Confluence Overview', and 'Second'.

At the bottom of the page, there is a footer with the text: 'Powered by a free Atlassian Confluence Personal Server License. Get your own Personal Wiki - free. Powered by Atlassian Confluence 2.8.0, the Enterprise Wiki. Bug/feature request - Atlassian news - Contact administrators'.

Figure 6.7: Confluence dashboard

- Quick links to create a space, build a feed or browse groups
- Recently Updated
- Recently Added To Favourite List

Integration

Sections

A dashboard should cover the following aspects:

- Recent Activity
- Browse Blogs/Wikis
- Quick Links for the main actions

6.4 Navigation Through Different Document Types

Confluence

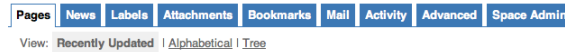


Figure 6.8: Confluence tabbed navigation through document types

The document types (pages, news, attachments, mails, etc.) are grouped by their types and accessible from a tabbed navigation. Almost each document type has a special way of being displayed. Often, you can change the kind of grouping in a document type section. You will find links which are entitled by the prospect beneath the tabbed navigation of the document types.

Good

- All document types are accessible under one navigation.

Bad

- The document types have very different display modes. It is awkward to have to spot the way they organized this particular document type and its associate actions each time.
- Since the menu is not part of the main menu you cannot switch document types when you are in the a document view but only on this navigation page.

Suggestions

- Find a generic way to display most information and places to display possible actions. Except for wiki pages, all other document types should be displayed in a table. There have to be conventions on how to display tabular data, and their invocable actions.

WordPress



Figure 6.9: WordPress tab-like navigation through document types

The document types are grouped under the tab "Manage". Every document type has its own tab-like section.

It is part of the main menu.

Integration

Customize By Document Type

All examined applications split the management of the different document types into their own sections. Each section is customized to support the management of the particular document type.

Part Of The Main Navigation

The links to the sections are collected in one navigation bar which is part of the main menu (e.g. WordPress) or has its own page (e.g. Confluence)

Tabs

Navigation through document types is tab-like: The main navigation stays at its place, the content area changes.

6.5 Links

Confluence

A general link inside a document is blue and underlined. There is no differentiation in link, visited, hover or active state.

Links which indicate an action like "Edit Labels", "Edit" or "Add". Those are grey.

When a link's font size is bigger than common text like linked headlines or the standard actions, there is no underlining.

WordPress

Links are light blue. On hover and active they become orange. Visited links are not differentiated in the backend.

Links who stand for themselves, like actions or content of a table cell, have no underlining. There are exceptions: Links who are part of a text are displayed with underlining. This is also true for links in headlines or for links who are sentences, like "Manage all your posts".

Dropbox

Links are light blue. They become underlined on hover.

Links which represent an action are grey and underlined.

The menu items look the same as links but do not get underlined on hover. They get a lighter blue background.

Integration

Links = blue and speaking

A text link has to be blue. Furthermore they have to speak for themselves (i.e. not "Click here" etc.)

Links In Text = blue + underlined + visited status

Links as part of a text are underlined, are the same font size and weight of the text and should indicate a visited status.

Action Links = no visited status, differentiate them

Links which represent actions should disregard the visited status. To differentiate them from the above, they can have no underlining or another color or both. Grey text color is often used to indicate actions.

If on page load a link is active, i.e. its action has been invoked, it should have no underlining.

6.6 Menu

Confluence



Figure 6.10: Confluence main menu

The main navigation is a horizontal menu on the top of the page consisting of five sections:

- Breadcrumb which resembles document hierarchy
- Drop down menus to navigate to the edit views of documents and to personal preferences
- a simple search box
- edit, add and tools
- heading with added and last edited information

There is a relatively large amount of links hidden in drop down menus. It occupies ca. 90px vertical space.

Good

- Experienced Users: 30 actions can be invoked on a ca. 200px by 40px space without sacrificing overview for new users.
- Clickable regions are big enough
- Drop down menus open on hover
- Advanced Users: Access keys are used. Ctrl-q highlights search box, ctrl-e moves to edit the page
- Access key functionality is signaled at least once by underlining the "E" in "Edit".
- Search box without options in the menu, so you do not have to choose where you search.

Bad

- Other useful access keys are not shown although they exist.

Wordpress

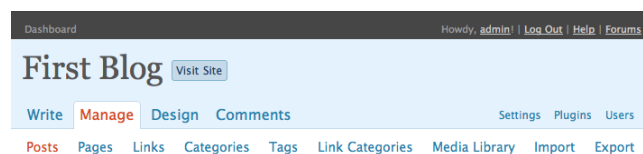


Figure 6.11: WordPress main menu

The main navigation is a horizontal menu on the top of the page consisting of four sections:

- the top part with links to the
 - Dashboard,
 - User Profile,
 - Log Out,
 - Help and
 - Forum
- the name of the blog and a button to visit its frontend
- a tab navigation with an one-level horizontal submenu
- a preferences section with the following entries:
 - Settings
 - Plugins
 - Users

It occupies ca. 160px vertical space.

Integration

Horizontal

The main navigation should be horizontal. This makes it possible to have section-specific vertical menus. Since vertical space is valuable it should be as slim as possible. With a resolution of 1024 x 768, a 120px tall navigation takes already more than one fifth of the usable vertical space.

Indicate Active Section

Use breadcrumbs or, if applicable, highlighting of the main menu entries to indicate the active section you are working in. If you choose the second option, you have to use a sub menu beneath the main menu to get an indication of the right granularity.

6.7 Search

This chapter covers search functionality which is offered on a regular page for quick navigation purposes. It does not cover the results page and advanced search forms. Please see [Forms > Search Form](#) for those.

Wordpress



Figure 6.12: WordPress search box

Search functionality is offered on all subcategories of "Manage" and "Comments". It searches only for the document type in which subcategory you are navigating. The button adjacent to the search form expresses this by its value "Search *Plural of document type*". It consists of a input field, which is empty by default and a button to the right.

It is positioned on the vertical level of the subcategory heading to resemble the affiliation to it.

Dimensions (without button): 170 x 17 pixels with additional 3 pixel padding on each side.

Confluence



Figure 6.13: Confluence search box

The search functionality is part of the header. Its position in the upper right is consistent and it is visible on all pages. It searches through the global spaces for the search terms.

It is an html search field (proprietary addition by apple), which degrades to an input field in other browsers besides those which use webkit (Safari being the most popular). This changes the behaviour to apples search box: It has rounded corners, a magnifier icon which offers a search history on click and a "x" appears when you type text, to reset the search.

Its default value is "Search" which disappears when you focus it and reappears if you move the focus somewhere else with an empty search field.

Its text color is a light grey which stays this way when you enter search terms.

You can focus it by typing the access key "q" on an arbitrary page.

Dimension: 160 x 50 pixels with additional 1 px padding on top and bottom.

Good

- Usage of proprietary html search field EXPLAIN
- Search functionality can be invoked from every page.
- Access key support.

Bad

- Light grey is difficult to read.
- Access key is not indicated.
- Access key "q" is not the convention for searching. It should be "f".

Suggestion

- Change the text color to black when someone is typing.
- Indicate the access key e.g. with a title tag on the search field.

Dropbox

No search functionality.

Bad

- Although the OS's file system search excels a web search fairly easy, a web search function could search for special data, which is not available in the file system, e.g. text in revisions.

Integration**Top Right, No Buttons**

The best practice today for applications and web applications is to position the search field at the top right corner. \citeauig The input field should say "Search" in grey letters which disappear on focus of the field.

Access key: F

Since the field is one the most-often used functionalities it should have an access key. The default key combination for "Search" is "ctrl-f" on the PC, "command-f" on the Mac. Therefore I suggest to use the access key "alt-f" on the PC, "ctrl-option-f" on the Mac. Confluence uses "alt-q" (PC), respectively "ctrl-option-q" (Mac) apparently to avoid confusion with the very similar key combinations above. I think the user who uses access keys is experienced enough to use "f" with different command keys for both, the browser internal search function and the web application search function.

6.8 Title

WordPress

A frontend title looks like this

Name Of The Blog » Blog Archive » *Title Of The Post* – Wordpress

In the backend it looks like this

Name Of The Blog > *Name Of Current Action* – WordPress

Bad

- Frontend: In a bookmark list the first two parts will most likely hide the title of the post. But the post title is likely the most important part of the bookmark.
- Backend: The formulation of the current action differs to it on the page.

Confluence

Page heading - *Space name* - Confluence

Good

- Page name is first element. That conducts to good bookmark titles.

Bad

- You cannot distinguish between a bookmark to the site or to the edit view of this site by just looking at the title.

Suggestions

- Include the current action into the title. (e.g.: "Edit: *Page heading* - [...]")

Dropbox

Dropbox - *Activated main menu item* - Secure backup, sync and sharing made easy.

The placeholder being one of the following: Home, Files, Photos, Share, Upload, Inbox, Account

Although the name of the application, Dropbox, is mentioned as first Element, this has no Disadvantage, because there is never a title so long, the page name would get truncated.

Good

- Most unimportant information (the claim) is on the right side. No important information will get truncated this way.

Bad

- Title does not contain sub-folders in file browsing mode. (This is due the URL not changing because of AJAX calls. It is not possible to bookmark a certain location.)

Integration

The title of a page is important because it appears in bookmarks and in the browse history. Depending on the line-up of the different parts of it you can control their importance.

More Content-Centric

Page heading - Location - Name of the System

Confluence uses this scheme.

This way, the user sees the name of the document first. This is most likely the most important information if he had set a bookmark to it. Separators are hyphens.

It makes searching in long lists like the history easier, because the eyes can retain at the left side of the list while scrolling down.

More Advertising

This solution is presented because it is largely used throughout the web.

By ordering their title WordPress has decided that the name of the blog is more important than its content. Because of the nature of a blog the name of it, like the name of a newspaper, is arguably more important than a certain story. If there is much interesting information on a website, that the user ends up with a bookmark list like this (the titles of the important blog being the first, third and fifth):

- Important Blog > Why a...
- An even more Import...
- Important Blog > The o...
- Design - why it matters
- Important Blog > Becau...

Even if it is not truncated, the eye must go away from the left side to read the title which slows down the reading flow.

Which One To Choose?

The second type is typically used on websites because a user normally goes into a website, clicks some sites and goes out of it again. It is easier for him to spot the website again in the history list if the name is repeated in front of the title.

The first option is preferred if a user clicks through many pages of a website. This is typical for web applications.

The separator should be a greater then sign > or the, to some more appealing, »-sign.

6.9 URL

A URL can be used to spot the current location and to shortcut to known places. Furthermore it's important in SEO. It should reflect the structure of the web site to help the user build it's mental model of it. It should be short and unambiguous enough so it can be handed on over other media than the internet like telephone.

WordPress

Frontend URL

WordPress gives the user the choice between four different URL schemes:

- Default: <http://example.org/wordpress/?p=123>
- Day and name: <http://example.org/wordpress/2008/06/30/sample-post/>
- Month and name: <http://example.org/wordpress/2008/06/sample-post/>
- Numeric: <http://example.org/wordpress/archives/123>

The choice of Default is more a technological one: It does not need Apache's `mod_rewrite`. There is also an option to create your own scheme.

Good

- All: Human-readable
- Day and name, Month and name: It lets the user build a mental model of the system. If he wants to see other posts from this day, he can just remove the title-of-the-post. By deleting any further elements he can get to the archives of the month or of the year. Of course this also works the other way round. To get to messages of a special day you can just type in the date in this format.
- Numeric, Default: Easy to hand on e.g. per telephone

Bad

- Day and Name: Fairly deep hierarchy. Search engine rating drops with additional slashes. [Fi08]

Backend URL

- <http://example.org/wordpress/wp-admin/edit-pages.php>
- <http://example.org/wordpress/wp-admin/page.php?action=edit\&post=2>
- <http://example.org/wordpress/wp-admin/themes.php>

The URLs in the backend are plain flat, i.e. every document is a direct child of `wp-admin`.

Bad

- Usage of GET-variables without `mod_rewrite` [Wo08]
- Suffix `.php` is unnecessary [Wo08]

Suggestions

- Get rid of the suffixes.
- Use `mod_rewrite`, e.g. `page/edit/2`

Confluence

A typical URL looks like this:

- wiki pages: `http://example.org/display/spaceshortname/Page+title`
- for news: `http://example.org/display/spaceshortname/2008/05/09/News`
- when editing a page: `http://example.org/pages/editpage.action?pageId=393221`

Good

- URL has few slashes, this helps SEO.
- URL is human readable. The content is guessable by looking at it.
- Instead of using spaces of the title in the URL (which would be translated to "%20") it uses another symbol.
- Removing the news title or the day of the URL shows the news archive of the day/the month.

Bad

- Many times the suffix `.action` is added. This has no advantage for the user, nor is it important for a web server.
- URLs often do not change although its content changes.
- Although using plus signs for spaces is better than using encoded spaces, it is still hard to read.
- You cannot display a years' archive by removing day and month of the URL.

Suggestions

- Remove suffixes.
- Show actions invoked on documents in the URL. Confluence does a good job with `/display/Page+title`, why not use `/edit/Page+title` respectively?
- Use underscores or hyphens to encode spaces in URLs.
- Make date URLs fully navigable, i.e. removing parts of the date must show the archive of the entered date.

Integration

Frontend URL

The important points for a frontend URL are the following:

Human Readability By just looking at the URL it should be guessable where it leads. This is especially useful when the user hovers over a link which description is not helpful like "click here".

Mental Model A URL should always inherit the underlying model of the website. That way he can get a first and quick impression how the website is structured and can optimize foray through it. A blog is strictly chronologically ordered. This has to be shown in the URL. A wiki has usually a very flat hierarchy which should be represented that way.

Shortcuts "Provide both concrete and abstract ways of getting a task done." [To03] Experienced users often use the URL to shortcut quickly to another place when they understood the inherited model. At a blog they can e.g. just change the date. Or they could change the language. This is especially useful when the wished functionality does not exist on the current page and website navigation has to be executed before having the ability to do so. Navigation through the website is this way just based on the website semantics and therefore completely without using the mouse or understanding the graphical concept of the page. There should also be "guessing shortcuts". E.g. if a user types only the first word of a title, it should automatically direct to a convenient page like a search page with the input as search terms or the most logical place for this input. [Ni94]

Don't Change URLs

It is best practice not to use suffixes for web pages. It does not add an advantage for the user and unnecessarily lengthens the URL. It even would change the URL in case the underlying technology changes but the content stays the same. Do not use suffixes for web pages. Instead instruct the web server to open pages with the right mime type. [Wo08]

If a title of a document changes which is part of the URL, update the URL but permanently redirect the old URL to the new one e.g. with a .htaccess-file or redirect information in the content of the old URL.

Special Cases

Dates

SEO-optimized: Dates are displayed this way: <http://example.org/year-month-day>

More user friendly (favoured): Dates in URLs should be displayed this way: [year/month/day](http://example.org/year/month/day). Although many slashes ranks search results less, the user can select the portions of the date more easily. Additionally, the hierarchy is more visible.

Files

More user-friendly: <http://example.org/actual/path/to/the/file.jpg>

More secure: <http://example.org/files/fileid>

In case folder hierarchies should not be displayed in the browser history and server log.

Several Languages

SEO-optimized and easier to comprehend for a local audience:

- <http://example.org/de/seitenname> (localized title of the page name)
- <http://example.org/en/page-title>

Better for experienced users (favoured):

- <http://example.org/de/page-title>
- <http://example.org/en/page-title>

This makes changing languages and staying on the corresponding page very easy. In a team collaboration environment this also makes exchanging links easier. It does, however, drop the search engine rating, because the search term is no longer component of the URL in other languages than english.

Example

- Blog page: <http://example.org/blog/johns-blog/2008/06/17/the-importance-of-user-guidelines>
- Wiki page: <http://example.org/wiki/johns-wiki/the-importance-of-user-guidelines>
- File Sharing: <http://example.org/file/actual/path/to/the/file.jpg>==== Style =====

6.10 Image Language

WordPress

Few images are used. They split up into:

Branding

There are two logo variants. The W in a circle and the longer one with the WordPress writing. The long one is used at the log in screen. The short one is displayed in the footer at the bottom left of every page.

WYSIWYG Editor



Figure 6.14: WordPress add media icons



Figure 6.15: WordPress calendar icon

Besides the typical WYSIWYG editor pictograms for e.g. bold text or align right, there are four unique pictograms for inserting media: image, audio, video, media. Another one is the calendar symbol.

Preview

The theme editor and the media library use thumbnail previews to make the selection easier.

Comment



Figure 6.16: WordPress comment icon

The amount of comments to a certain post in list view is displayed inside a bubble symbol.



Figure 6.17: WordPress avatar

Comments on the frontend are accompanied by an avatar.

Confluence

There is a rich image language for document types and actions.

Document Types

The icon for the document type is set left to the title of the document. Together they are a unit and appear always like that.

There are different images for all document types:

- Spaces
- Pages
- News
- Labels
- Comments
- Attachments
 - Images
 - PDF Files
 - Excel Worksheets
- Bookmarks
- Mail
- Users

Actions

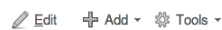


Figure 6.18: Confluence toolbar

Menu Items

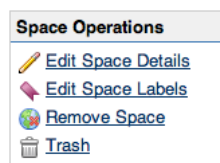


Figure 6.19: Icons in an confluence action menu

Administrative menus always have icons to the left of their menu item text.

Space Logo

Each space has a unique logo which is displayed left to the headline. By default that is the Confluence logo.

In a list all spaces have the same globe icon seen under section "Document Types".

Dropbox

Branding

The Dropbox logo with writing is displayed at the top left of every page.



Figure 6.20: Dropbox logo

Messages

Informative messages have a small !-icon to their left.

Document Types

Each one has its own icon. Invoked actions like creating or editing are displayed change the document type icon in the bottom right corner to reflect the action.

Help

For further information a ?-sign is displayed on the right of the concerning paragraph.

Relative Dates

They have a small clock symbol on the right

Actions

A group of context actions is hidden under a small triangle. Each action has a small pictogram to the left.

6.11 Layout

Each Application has a frame in which all actions and contents take place. Out of which problems were they constructed? How can they be integrated?

WordPress



Figure 6.21: WordPress layout

Its layout is very simple and clean. It consists of five sections:

- Dark header with links to the dashboard, profile, log out and help
- Main Menu
- Heading
- Content
- Dark footer with WordPress-specific links and a version info.

Although it is a stretched layout, i.e. it always occupied 100% of width and height of the browser window, the content section expands to at most 995 pixels width and stops after that.

The website is comfortably usable to a resolution down to 800 pixels width. It is even well usable with limited comfortability down to a browser window width of just 550 pixels.

From the top, the first line of content is at 225 pixels.

There is one image used at the bottom left. It is the WordPress Logo. The application is completely usable with disabled images with nearly no impact on its comfort.

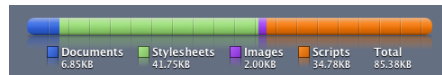


Figure 6.22: The layout weighs around 90 kb, consisting of ca. 12 files. A typical page only adds 10 kb of HTML which is non-cacheable.

Good

- Usability with small window sizes makes window-splitting scenarios accomplishable.

Bad

- The main menu wastes too much horizontal space with the blog title.

Suggestions

- Make the blog title much smaller or use the space around it more efficiently. I would add it to the dark header next to dashboard.

Confluence



Figure 6.23: Confluence layout

This layout consists of four sections:

- Dark header with a breadcrumb, browse space functionality, administration, profile, log out and search functionality
- Heading with space logo and toolbar
- Content
- Dark footer with Atlassian-specific links and version information.

The width of all sections expands to 100% of the browser window width. The height does not stretch. It varies on content length.

There is a lot of functionality just in the header, heading and footer. On average you can invoke 40 actions or links from here. This is partially achieved by using drop down menus.

What differences this layout from the rest of the examined ones is that it is not as static as the rest of them. The heading changes a lot. In edit mode you can e.g. edit the text of it. The toolbar changes a lot depending on which site you browse. Sometimes it is not even there.

The edit mode can be used down to 600 pixels window width. In viewing mode there is no real limit. Even at 350 pixels width you can read the text just fine thanks to low margins to the left and right.

The first line of content is usually 120 pixels from the top.



Figure 6.24: Confluence's breadcrumbs break

With smaller window width, longer breadcrumbs get line breaks.

Used images are:

- at the top left the space logo.
- images for the toolbar
- a shadow in the footer

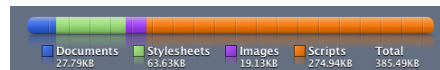


Figure 6.25: The layout weighs around 350 kb splitting into ca. 40 files. A typical wiki page adds 50-100 kb of non-cacheable content.

Good

- Because the breadcrumb is an important navigation tool in Confluence it makes sense breaking it down in multiple lines when horizontal space gets rare with the trade-off of less vertical space.

Bad

- Because the right header side does not get additional line breaks, the breadcrumb is unnecessarily soon full of line breaks or even unusable.
- Usage of the heading for both wiki page headlines and administration headlines is confusing. In edit mode there is no real headline at all because it is inside an input form field.

Suggestions

- On rare horizontal space, additionally insert a line break before the header search form field.
- In edit mode, add a headline like "Write Page" and put the edit input form field beneath it to circumvent the confusion between application headlines and wiki page headlines.

Dropbox

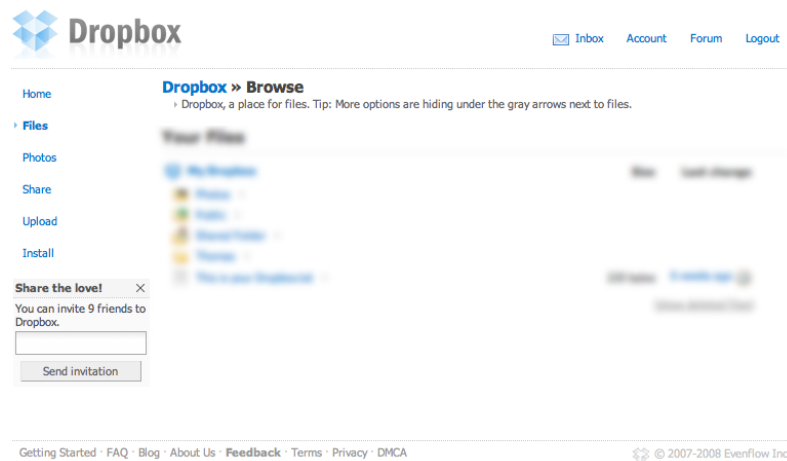


Figure 6.26: Dropbox layout

Dropbox uses a fixed 770 pixels wide layout. It is horizontally centered. Because the main menu is on the left, the first line of content can be found at approximately 160 pixels from the top although there is a generous usage of spacing. It consists of four section:

- Header with logo, links to Inbox, Account, Forum and Logout
- Main menu (vertical) with space to add special announcements
- Headline which mixes with a breadcrumb
- Content
- Footer with Dropbox-specific links, copyright and company information

Bad

- Menu item allocation is unclear. Why is "FAQ" at the top right and "Forum" in the lower left. Why is a link to an install page part of the main menu?

Integration

Footer Is About The Application

The footer should always tell which application the user is using, a link to it and the currently used version.

It should provide a way to get into contact and if applicable submit bugs or feature requests.

Always Have A Headline

It is very important to always have a headline which stays at the same place.

A headline should tell the user which section of the application he is using, *not* which document he is viewing, e.g. "Write Post" or the name of a wiki instead of the actual title of the post. The title belongs to the content and is no part of the headline. With very context-centric pages like wiki pages, the title of the document and the headline become one string. On other pages however they are strictly divided. This is as a consequence of Krug's "trunk test", [Kr00] where you should be able to quickly tell what a page is about without seeing how one navigated there.

There Are Two (Three) Menus

The main menu consists of:

- Header with Dashboard administrative and helping links
- Headline

If applicable a ubiquitous toolbar. It should be usually fused with the main menu.

The footer menu with

- application-specific links,
- version,
- company and
- copyright information

Make Layout Stretchable But Not Too Much

At least the content section should stop stretching further at around 990 pixels width. This makes the pages look similar at different resolutions from 1024 x 768 pixels on which is important in a team collaboration situation where one member wants to show something "in the middle of the page". Additionally the lines do not get too long to read them comfortably.

Make Layout Shrinkable

Also support shrinking the window to at least 600 pixels. This way copying between windows is easy and reading long texts is more comfortable with shorter lines. Additionally users who are working with a resolution of 800 x 600 pixels are supported.

Shrinking width can be done four ways:

1. Using more horizontal space
2. Cutting elements
3. Making them only accessible through a horizontal scrollbar
4. Reduce (letter) spacing

The first type is usually used for the content.

The second is for elements of lower importance. Often the left and right sides of an element are of more importance than the middle, e.g. if it is a file with a suffix or the title, first name and surname of a person. In this case you have to cut the middle.

The third is used for elements who can be useful but are used rarely like a syntax help besides a textarea.

The fourth is used rarely in a browser because it is not as easy to achieve as the other types, but you can reduce the spacing of an element not only to its surrounding but also inside the element. In case of text this is letter spacing. This technique is usually used before cutting an element. It should not be used combined with other techniques at the same time.

Do Not Change Positions Of Elements

All elements of a layout should be designed to be ubiquitous and usable nearly everywhere inside the application. Do not change position, outlook and behaviour of parts of the layout.

Do Not Remove Elements

If an element is not usable on one page think of disabling it instead of removing. If you have to disable it too often, you probably have a concept flaw in the layout.

There are buttons which are replaced depending on the state of the application, e.g. edit/view in confluence, plus/minus in Windows Explorer or reload/stop in Safari.

While this can be helpful in the content section like in Windows Explorer, but you should not use that in the layout. A layout should be a frame for the rest of the application. It should not change.

In the case of Safari this has been discussed and accounted with its introduction. There is almost always a trade-off between the efficient use of space and showing both buttons. This is not always foreseeable: Safari was designed before the AJAX boom. Now imagine an AJAX page which often loads new content. The overloaded button can now be implemented two ways:

1. It reacts on new AJAX calls and toggles to a stop button. To reload this page you have to actually push the button twice. Once for stop loading and once for reloading. And be quick so the next call is not toggling the button again.
2. New AJAX calls do not toggle the reload button. You can reload with one click but you cannot stop AJAX calls. (current implementation)

This is why Firefox still has two buttons for it.

Provide A Printer-Friendly Version

Most of the time you do not have to write a new printer-friendly output page. CSS is nowadays supported well from most browsers to use it for printing. Just use another CSS file for `media="print"` than you do for `media="screen"`. This might not work in some situations where the user can dynamically change reports of the system. [FS04]. Remove

banners and unnecessary navigation. If it makes sense, use CSS to expand links in the way that their URLs are displayed behind the Text. (Example: "Go to the Index (/app/index)") CSS files without a media declaration are interpreted for all output types.

6.12 Typography

How do fonts look like? Which sizes should they have? Why are different fonts used.

WordPress

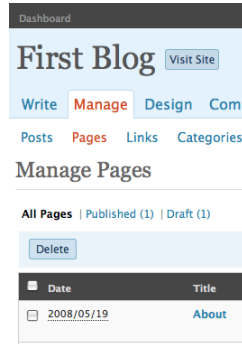


Figure 6.27: Font style of WordPress

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0

Figure 6.28: Georgia is used for headlines

The serif font "Georgia" is used for the blog title and the section headings. It's alternatives are Times New Roman and Times. The blog title is displayed in 36 pixels size, the section heading in 24 pixels size.

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m
n o p q r s t u v w x y z
1 2 3 4 5 6 7 8 9 0

Figure 6.29: Lucida is used for body text

The rest of the page is in a nice readable sans-serif font called "Lucida Grande" (alternatives in this order: "Lucida Sans Unicode", Tahoma, Verdana). The size of normal body text is 11 pixels. The main column of a list and labels of a form are 13 pixels. The main menu is 14 pixels.

Text with title tags have a bottom border.

Everything is in normal font weight except the header row, the main column of lists and form labels are typeset in bold font face. There is no other usage of bold text.

The usage of big sized fonts and the mix of serif headlines with sans-serif body text looks very friendly. Through the small usage of bold font the application looks elegant.

Confluence

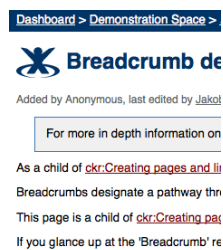


Figure 6.30: Font style of Confluence

ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 abcdefghijklm
 nopqrstuvwxyz
 1234567890

Figure 6.31: The makers of Confluence typeset the whole application in Helvetica (alternative: Arial).

The page title is bold and 18 points in size.

Body text is 13 points in size. If it is meta information on the same page like last edited information it is 0.9 em in size.

The toolbar text has 1.1 em in size.

Bold text is used for all headlines, table headers, activated sections of a menu and form labels. Additionally some actions are typeset bold.

Dropbox

Dropbox uses a sans-serif font.

Bold text is only used for headlines.

Integration

The principle of aesthetics is one of the 13 principles of user interfaces according to Tognazzi, so it should be treated well. He quotes writer William Roster to "[n]ever do anything that looks to someone else like a mistake." [To03] Know your users and their taste. There are no real restrictions to typography except providing accessibility.

Chapter 7

Terminology

7.1 Dates

WordPress

In a list there are relative dates to a certain point in the past with its units being:

- *i* mins ago (diff < 1 hour)
- *i* hours ago (diff ≤ 24 hours)
- Date in format "yyyy/mm/dd" (diff > 24 hours)

With *i* being an integer.

The same format is used for dates in the future (like publishing dates). Just by substituting "ago" with "from now".

If you hover over a relative or otherwise shortened date, a tooltip appears with the date in this format:

2008/06/15, 11:22:23 am

In an edit form the last change is displayed in this format:

Published on: June 15, 2008 at 11:22 am

Confluence

In an alphabetical list the date is only shown in this format:

- Jun 15, 2008

In other lists the date has these units:

- just a moment ago
- less than a minute ago
- a minute ago
- *i* minutes ago
- about an hour ago
- *i* hours ago
- yesterday at 03:05 PM
- Jun 15

With *i* being an integer.

If you hover over a relative or otherwise shortened date, a tooltip appears with the date in this format:

Jun 15, 2008 11:22

In an edit form the last change is displayed in this format:

last edited by Jakob Stoeck on Jun 15, 2008

Dropbox

Dates are displayed only in file lists as last change. They are displayed in these units:

- *i* secs ago
- *i* mins ago
- *i* hrs ago
- *i* days ago
- *i* weeks ago

With *i* being an integer. If *i*=1 the unit is displayed in singular.

From this list always the biggest unit is chosen for which the date can be represented as an integer. It always rounds off. (e.g. 17 secs, 5 mins, 1 hour etc.)

Because Dropbox is not Open Source I cannot say for sure that there are no bigger units. After 5 weeks of testing the date is still counted in weeks, therefore i think it is the biggest unit.

Integration

Dates should vary its format depending on their surrounding.

One Date On A Page

The examination of the applications has shown that when there is only one date displayed for the current element its format is much longer. Usually it is accompanied by an explaining sentence:

Last edited by Jakob Stoeck on Jun 15, 2008 at 11:22 am

There is always the time connected with it, even if this does not show up in the rest of the application.

This is no real best practice, just an often used convention. It can make sense to just use the short date version from the next subchapter throughout the application.

Lists with dates

If there are more dates on one page like in a list, dates should be formatted human readable. The applications agreed to using only integers and (with few exceptions) the biggest unit available. Do not say "70 minutes ago" but "1 hour ago". The integers are calculated by rounding off. If the integer gets 0 after this calculation and there is no smaller unit left it is defined as 1. E.g. 119 minutes are 1 hour.

The units are:

- *i* minutes ago
- *i* hours ago
- 15 May (add ", 2008" only if the list is not ordered by date)

When lists are chronologically sorted, do not display a year.

Always Add A Title Attribute

Always use the title attribute of a relative or otherwise shortened date to display the whole date in such a format:

15 May, 2008 11:23

7.2 Documents

WordPress

- Dashboard - entry point
 - Posts - The main content of the blog
 - Pages - Pages are like posts except they live outside of the normal blog chronology and can be hierarchical.
 - Comments - Can be made to pages and posts. They are associated to a user or a name and a date.
 - Categories - A page or post can have a category. The default for a new document is "uncategorized". You can add any category. They can be nested into each other. They can be selectively converted to Tags.
 - Tags - known usage as keywords
 - Comments - Can be attached to a post or page also by not logged-in users.
 - Uploads/Attachments
 - Blogroll - When you have a list of links in your sidebar to other blogs, it's called a "blogroll".
 - Themes - only for the frontend
 - Latest Activity - comments, posts, pages and storage space
-
- Published - publicly accessible
 - Unpublished - can only be seen in the backend
 - Pending Review
 - Private

Bad

- Categories do not have an advantage over tags. They can be disregarded.

Confluence

- Dashboard - Entry point after you signed in. Root of all spaces.
- Spaces - A wiki with an own theme, users and pages
- News - Chronologically listed content
- Pages - Hierarchically ordered wiki pages, the main document type of a confluence installation
- Comments
 1. You can comment on pages and news. They can be replies to comments done before. They are associated to a user and a date.
 2. You can comment on a change you made in a document.
- Labels - Usage like tags
 - Personal labels
 - Related labels - labels can be in a relation to another one
- Attachments - Can be made to news and pages

- Bookmarks - Can be made of every URL, also external ones. A bookmarklet is provided.
 - Mail - is dragged from real mail accounts
 - Orphaned Pages - which are not linked to by other pages
 - Undefined Pages - has been linked to from within Confluence, but that does not yet exist
 - Favourite Pages - Pages can be made favourite
 - Templates
 - Themes - Overall layout into the given ones
 - Trash - Space-dependent place where removed documents go
 - Watches - If you watch a document you get an e-mail notification on a change. You can also watch whole spaces.
 - Drafts - A draft is a unsaved document. If something e.g. the browser crashed while writing a new page, the page will be still in drafts.
 - Recently Updated - New and updated documents
-
- Remove - Put a document in the space trash
 - Move - Changing the position of a document in its space or move it to other spaces

Bad

- The term "comment" is used for two different things.

Suggestion

- When you change a wiki page you can append one comment why the change was made. That is confusing because also user comments can be done. Call the comment of change something like "Reason of change".

Dropbox

- Home
- Recent Events
- Files
- Photos
- Shared Folders
- Inbox - stores invites to shared folders
- Accounts
- Computers - connected to dropbox
- Invites
- Deleted Files

Integration

A uniform terminology is very important in an integrated web application. My proposition based on most familiarity to the user:

- Dashboard - the entrance page with actions how to go on
- Pages - a document type where hierarchy is more important than other attributes like chronology
- Wiki Pages - Pages with versions and wiki syntax
- Posts - a blog entry
- Comment - a text associated to a post, page or wiki page
- Drafts - an auto-saved document, not saved explicitly by the user
- Tags - the well-known keywords
- Trash - Deleted files go here
- Recent Activity
- Delete, Move, Add Files
- Files
- Folders

Appendix A

Used Design Libraries

While writing this thesis I discovered some very interesting web sites about user interfaces. Since many of them cannot be found in the bibliography, I decided to put them here. They look at interfaces from different viewing angles and are always part of my first search for new interface design possibilities.

Web Patterns: A UC Berkeley Resource for Building User Interfaces

http://groups.ischool.berkeley.edu/ui_designpatterns/

UI Patterns: User Interface Design Pattern Library

<http://ui-patterns.com/>

Wikipatterns

<http://www.wikipatterns.com/>

Not directly about interfaces but short textual paradigms which help improving a wiki. Many tips and anti-tips where even experienced wiki developers can learn something.

Yahoo! Design Pattern Library

<http://developer.yahoo.com/ypatterns/>

Well-structured library of controls like auto-completion or lists with short videos. Often accompanied by a YUI demo.

Guidebook: GUI Gallery

<http://www.guidebookgallery.org/>

The development of GUI design with many screenshots. Although this thesis is about web application you can quite compare them with older GUI applications. It is interesting to see how those interfaces evolved during the last ten years to look out for possible evolutions in the web application domain.

Interaction Design Pattern Library

<http://www.welie.com/patterns/>

With code examples and sometimes a helpful comment section.

Apple Human Interface Guidelines

Being a desktop software-only company in the past, Apple has advanced more and more into the internet. It sits in various web syndicates of one which prepared HTML 5. An integral part of the OS is written in JavaScript and HTML. Although not inventing them they made many graphical expressions popular like reflections below text or images, accordion navigation and very clear interfaces. Apple has several sources for interface design. Some of them can be found in the bibliography.

Bibliography

- [En07] Enders, J.: Zebra striping: does it really help? Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments, pages 319–322, 2007.
- [Fi01] Fischer, G.: Communities of Interest: Learning through the Interaction of Multiple Knowledge Systems. 24th Annual Information Systems Research Seminar In Scandinavia (IRIS'24), Ulvik, Norway, pages 1–14, 2001.
- [Fi08] Fishkin, R. e. a.: Search Ranking Factors, 2008. 12 Aug 2008 <http://www.seomoz.org/article/search-ranking-factors>.
- [FS04] Fowler, S.; Stanwick, V.: Web Application Design Handbook: Best Practices for Web-Based Software. Morgan Kaufmann, 2004.
- [Kr00] Krug, S.: Don't make me think!: a common sense approach to Web usability. New Riders Publishing, 2000.
- [La03] Laurel, B.: Design research: methods and perspectives. MIT Press, 2003.
- [LM90] Laurel, B.; Mountford, S.: The Art of Human-Computer Interface Design. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990.
- [Ni94] Nielsen, J.: Heuristic evaluation. John Wiley & Sons, Inc. New York, NY, USA, 1994.
- [RH94] Rubin, J.; Hudson, T.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons, Inc. New York, NY, USA, 1994.
- [SB06] Sarodnick, F.; Brau, H.: Methoden der Usability Evaluation. Verlag Hans Huber, Bern Schweiz, 2006.
- [To92] Tognazzini, B.: TOG on Interface. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1992.
- [To03] Tognazzini, B.: First Principles of Interaction Design, 2003. 12 Aug 2008 <http://www.asktog.com/basics/firstPrinciples.html>.
- [VD07] Van Duyne, D. e. a.: The Design of Sites: Patterns for Creating Winning Websites. Prentice Hall Ptr, 2007.

[We08] van Welie, M.: Patterns in Interaction Design, 2008. 12 Aug 2008 <http://www.welie.com/>.

[Wo08] World Wide Web Consortium:, 2008. 12 Aug 2008 <http://www.w3.org/>.

List of Figures

1	Thesis Sitemap	6
1.1	List with checkbox column and associated button	9
1.2	List with actions	9
2.1	Standard apache error page	12
2.2	Confluence error message	12
2.3	Confluence warning message	13
2.4	An error page with search, important places and the normal web application's navigation bar	14
2.5	Meta data on WordPress' dashboard'	16
2.6	Info page of a Confluence wiki page	16
2.7	Actions associated to a information module	18
2.8	Profile page of WordPress	19
2.9	Profile page of Confluence	20
2.10	Profile page of dropbox	22
2.11	Statistics in WordPress are purely textual	23
2.12	Recent activity graph in Confluence	24
2.13	Activity statistics for a wiki	25
3.1	WordPress flash	28
3.2	WordPress error message	28
3.3	Confluence error message	29
3.4	Confluence info message	29
3.5	Confluence warning message	29
3.6	Best practice error message	30
3.7	Best practice warning message	30
3.8	Best practice confirmation message	30
3.9	Best practice information message	31
4.1	WordPress edit form	38
4.2	Confluence edit form	39
4.3	Minimal edit form	41
4.4	WordPress login form	43
4.5	Confluence login form	43
4.6	WordPress password update	45
4.7	Confluence password update	45
4.8	Dropbox password update	46

4.9	WordPress search box	47
4.10	Confluence search box	48
4.11	WordPress add media	50
4.12	Confluence add media	51
5.1	Confluence alphabetical listing	54
5.2	Confluence recent activities list	56
5.3	WordPress recent activities list	57
5.4	WordPress flat listing	58
5.5	Better alternative to zebra tables	59
5.6	Confluence grouped list	61
5.7	Dropbox hierarchical listing	63
5.8	Confluence hierarchical listing	64
6.1	Confluence breadcrumb	66
6.2	Dropbox breadcrumb	67
6.3	A best practice breadcrumb	67
6.4	WordPress button	69
6.5	WordPress button in hover state	69
6.6	WordPress dashboard	72
6.7	Confluence dashboard	73
6.8	Confluence tabbed navigation through document types	74
6.9	WordPress tab-like navigation through document types	74
6.10	Confluence main menu	77
6.11	WordPress main menu	77
6.12	WordPress search box	79
6.13	Confluence search box	79
6.14	WordPress add media icons	87
6.15	WordPress calendar icon	87
6.16	WordPress comment icon	87
6.17	WordPress avatar	87
6.18	Confluence toolbar	88
6.19	Icons in an confluence action menu	88
6.20	Dropbox logo	89
6.21	WordPress layout	90
6.22	The layout weighs around 90 kb, consisting of ca. 12 files. A typical page only adds 10 kb of HTML which is non-cacheable.	91
6.23	Confluence layout	91
6.24	Confluence's breadcrumbs break	92
6.25	The layout weighs around 350 kb splitting into ca. 40 files. A typical wiki page adds 50-100 kb of non-cacheable content.	92
6.26	Dropbox layout	93
6.27	Font style of WordPress	97
6.28	Georgia is used for headlines	97
6.29	Lucida is used for body text	97
6.30	Font style of Confluence	98

6.31 The makers of Confluence typeset the whole application in Helvetica (al-
ternative: Arial). 98