

Patterns for Effective Interaction Design

2nd Edition

Designing Interfaces



O'REILLY®

Jenifer Tidwell

Designing Interfaces

Jenifer Tidwell

O'REILLY®

BEIJING • CAMBRIDGE • FARNHAM • KÖLN • PARIS • SEBASTOPOL • TAIPEI • TOKYO

Designing Interfaces
by Jenifer Tidwell

Copyright © 2006 O'Reilly Media, Inc. All rights reserved. Printed in the United States of America.

Published by O'Reilly Media, Inc.,
1005 Gravenstein Highway North,
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editors: Andrew Odewahn and Mary O'Brien
Production Editor: Genevieve d'Entremont
Cover Designer: Mike Kohnke
Interior Designer: NOON

Printing History:
November 2005: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Designing Interfaces* and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

This book uses RepKover, a durable and flexible lay-flat binding.

ISBN: 0-596-00803-1
ISBN13: 978-0-596-00803-1

[C]

[04/07]

ABOUT THE AUTHOR

Jenifer Tidwell is an interaction designer and software developer for The MathWorks, makers of technical computing software. She specializes in the design and construction of data analysis and visualization tools, and has been working on new designs for the data tools in MATLAB, which is used by researchers, students, and engineers worldwide to develop cars, planes, proteins, and theories about the universe. She has been known to design web sites, and was an early enthusiast for rich Internet application (RIA) technology, having helped design and develop Curl in the early 2000s.

Jenifer received her technical education at MIT and her design education at the Massachusetts College of Art, but she's not finished learning yet. She has been researching user interface patterns since 1997. Photography and writing are her creative outlets, and she

spends as much time as she can in the New England outdoors—on a bike, on a boat, on foot, on skis, and on belay.

Jenifer's personal web site can be found at <http://jtidwell.net>.

COLOPHON

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The animal on the cover of this book is a Mandarin duck (*Aix galericulata*), one of the most beautiful of the duck species. Originating in China, these colorful birds can be found in southeast Russia, northern China, Japan, southern England, and Siberia.

The males have diverse and colorful plumage, characterized by an iridescent crown, chestnut-colored cheeks, and a white eye stripe that extends from their red bills to the back of their heads. Females are less flamboyant in appearance and tend to be gray, white, brown, and greenish brown, with a white throat and foreneck.

These birds live in woodland areas near streams and lakes. Being omnivorous, they tend to have a seasonal diet, eating acorns and grains in autumn; insects, land snails, and aquatic plants in spring; and dew worms, grasshoppers, frogs, fish, and mollusks during the summer months.

The mating ritual of Mandarin ducks begins with an elaborate and complex courtship dance that involves shaking movements, mimed drinking gestures, and preening. Males fight each other to win a female, but it is ultimately the female who decides her mate. Mandarin ducklings instinctively follow their notoriously protective mothers, who will

feign injury to distract predators such as otters, raccoon dogs, mink, polecats, eagle owls, and grass snakes.

Mandarin ducks are not an endangered species, but they are considered to be threatened. Loggers continuously encroach upon their habitats, and hunters and poachers prize the males for their plumage. Their meat is considered unpalatable by humans, and they are generally not hunted for food.

Genevieve d'Entremont was the production editor and proofreader for *Designing Interfaces*. Ann Schirmer was the copyeditor. Susan Honeywell was the page compositor. Phil Dangler and Claire Cloutier provided quality control. Kelly Talbot and Johnna VanHoose Dinse wrote the index.

Mike Kohnke designed the cover of this book, based on a series design by Edie Freedman. The cover image is from *Johnson's Natural History*. Karen Montgomery produced the cover layout in Adobe InDesign CS, using Adobe's ITC Garmond font.

NOON (www.designatnoon.com) designed the interior layout. This book was converted by Joe Wizda to Adobe InDesign CS. The text fonts are Gotham Book and Adobe Garmond; the heading fonts are Univers and Gotham Bold. The illustrations that appear in the book were produced by Robert Romano, Jessamyn Read, and Lesley Borash using Macromedia FreeHand MX and Adobe Photoshop CS. This colophon was written by Jansen Fernald.

CONTENTS

PREFACE X

01 WHAT USERS DO 2

A Means to an End 3

The Basics of User Research 5

Users' Motivation to Learn 7

The Patterns 10

1 safe exploration 11

2 instant gratification 11

3 satisficing 11

4 changes in midstream 12

5 deferred choices 13

6 incremental construction 14

7 habituation 14

8 spatial memory 15

9 prospective memory 16

10 streamlined repetition 17

11 keyboard only 17

12 other people's advice 18

02 ORGANIZING THE CONTENT: INFORMATION ARCHITECTURE AND APPLICATION STRUCTURE 20

The Basics of Information Architecture:

Dividing Stuff Up 22

Physical Structure 28

The Patterns 30

13 two-panel selector 31

Put two side-by-side panels on the interface. In the first, show a set of items that the user can select at will; in the other, show the content of the selected item.

14 canvas plus palette 34

Place an iconic palette next to a blank canvas; the user clicks on the palette buttons to create objects on the canvas.

15 one-window drilldown 36

Show each of the application's pages within a single window. As a user drills down through a menu of options, or into an object's details, replace the window contents completely with the new page.

16 alternative views 39

Let the user choose among alternative views that are structurally different, not just cosmetically different, from the default view.

17 wizard 42

Lead the user through the interface step by step, doing tasks in a prescribed order.

18 extras on demand 45

Show the most important content up front, but hide the rest. Let the user reach it via a single, simple gesture.

19 intriguing branches 47

Place links to interesting content in unexpected places, and label them in a way that attracts the curious user.

20 multi-level help 49

Use a mixture of lightweight and heavyweight help techniques to support users with varying needs.

03 GETTING AROUND: NAVIGATION, SIGNPOSTS, AND WAYFINDING 54

Staying Found 55

The Cost of Navigation 56

The Patterns 63

21 clear entry points 64

Present only a few entry points into the interface; make them task-oriented and descriptive.

22 global navigation 66

Using a small section of every page, show a consistent set of links or buttons that take the user to key sections of the site or application.

23	hub and spoke	68	33	center stage	103
	Isolate the sections of the app into mini-applications, each with one way in (from the main page) and one way out (back to the main page).			Put the most important part of the UI into the largest subsection of the page or window; cluster secondary tools and content around it in smaller panels.	
24	pyramid	71	34	titled sections	107
	Link a sequence of pages with Back/Next links. Combine this sequential presentation with a main page that links to and from all pages in the sequence.			Define separate sections of content by giving each one a visually strong title, and then laying them all out on the page together.	
25	modal panel	74	35	card stack	109
	Show only one page, with no other navigation options, until the user solves the immediate problem.			Put sections of content onto separate panels or “cards,” and stack them up so only one is visible at a time; use tabs or other devices to give users access to them.	
26	sequence map	76	36	closable panels	111
	On each page in a sequence, show a map of all of the pages in order, including a “You are here” indicator.			Put sections of content onto separate panels, and let the user open and close each of them separately from the others.	
27	breadcrumbs	78	37	movable panels	114
	On each page in a hierarchy, show a map of all the parent pages, up to the main page.			Put different tools or sections of content onto separate panels, and let the user move them around to form a custom layout.	
28	annotated scrollbar	80	38	right/left alignment	116
	Make the scrollbar serve double-duty as a map of the content, or as a “You are here” indicator.			When designing a two-column form or table, right-align the labels on the left, and left-align the items on the right.	
29	color-coded sections	82	39	diagonal balance	118
	Use color to identify which section of an application or site that a page belongs to.			Arrange page elements in an asymmetric fashion, but balance it by putting visual weight into both the upper-left and lower-right corners.	
30	animated transition	84	40	property sheet	120
	Smooth out a startling or dislocating transition with an animation that makes it feel natural.			Use a two-column or form-style layout to show the user that an object’s properties are edited on this page.	
31	escape hatch	86	41	responsive disclosure	123
	On each page that has limited navigation options, place a button or link that clearly gets the user out of that page and back to a known place.			Starting with a very minimal UI, guide a user through a series of steps by showing more of the UI as he completes each step.	
			42	responsive enabling	125
				Starting with a UI that’s mostly disabled, guide a user through a series of steps by enabling more of the UI as each step is done.	
04	ORGANIZING THE PAGE: LAYOUT OF PAGE ELEMENTS	88			
	The Basics of Page Layout	89			
	The Patterns	99			
32	visual framework	100			
	Design each page to use the same basic layout, colors, and stylistic elements, but give the design enough flexibility to handle varying page content.				

43	liquid layout	128
	As the user resizes the window, resize the page contents along with it so the page is constantly “filled.”	

05 DOING THINGS: ACTIONS AND COMMANDS 130

Pushing the Boundaries 133

The Patterns 136

44	button groups	137
	Present related actions as a small cluster of buttons, aligned either horizontally or vertically. Create several of them if there are more than three or four actions.	
45	action panel	140
	Instead of using menus, present a large group of related actions on a UI panel that’s richly organized and always visible.	
46	prominent “done” button	144
	Place the button that finishes a transaction at the end of the visual flow; make it big and well-labeled.	
47	smart menu items	146
	Change menu labels dynamically to show precisely what they would do when invoked.	
48	preview	147
	Show users a preview or summary of what will happen when they perform an action.	
49	progress indicator	149
	Show the user how much progress was made on a time-consuming operation.	
50	cancelability	151
	Provide a way to instantly cancel a time-consuming operation, with no side effects.	
51	multi-level undo	153
	Provide a way to easily reverse a series of actions performed by the user.	
52	command history	156
	As the user performs actions, keep a visible record of what was done, to what, and when.	
53	macros	158
	Macros are single actions composed of other, smaller actions. Users can create them by putting together sequences of actions.	

06 SHOWING COMPLEX DATA: TREES, TABLES, AND OTHER INFORMATION GRAPHICS 160

The Basics of Information Graphics 161

The Patterns 173

54	overview plus detail	174
	Place an overview of the graphic next to a zoomed “detail view.” As the user drags a viewport around the overview, show that part of the graphic in the detail view.	
55	datatips	176
	As the mouse rolls over a point of interest on the graphic, put the data values for that point into a tooltip or some other floating window.	
56	dynamic queries	178
	Provide ways to filter the data set immediately and interactively. Employ easy-to-use standard controls, such as sliders and checkboxes, to define which parts of the data set get shown. As soon as the user adjusts those controls, the results appear on the data display.	
57	data brushing	181
	Let the user select data items in one view; show the same data selected simultaneously in another view.	
58	local zooming	184
	Show all the data in a single dense page, with small-scale data items. Wherever the mouse goes, distort the page to make those data items large and readable.	
59	row striping	187
	Use two similar shades to alternately color the backgrounds of the table rows.	
60	sortable table	189
	Show the data in a table, and let the user sort the table rows according to the column values.	
61	jump to item	191
	When the user types the name of an item, jump straight to that item and select it.	
62	new-item row	193
	Use the last row in the table to create a new item in place.	
63	cascading lists	195
	Express a hierarchy by showing selectable lists of the items in each hierarchy level. Selection of any item shows that item’s children in the next list.	

64	tree table	197	73	autocompletion	227
	Put hierarchical data in columns, like a table, but use an indented outline structure in the first column to illustrate the tree structure.			As the user types into a text field, anticipate the possible answers and automatically complete the entry when appropriate.	
65	multi-y graph	198	74	dropdown chooser	230
	Stack multiple graph lines, one above the other, in one panel; let them all share the same X axis.			Extend the concept of a menu by using a drop-down or pop-up panel to contain a more complex value-selection UI.	
66	small multiples	200	75	illustrated choices	233
	Create many small pictures of the data using two or three data dimensions. Tile them on the page according to one or two additional data dimensions, either in a single comic-strip sequence, or in a 2D matrix.			Use pictures instead of words (or in addition to them) to show available choices.	
67	treemap	203	76	list builder	235
	Express multidimensional and/or hierarchical data as rectangles of various sizes. You can nest those rectangles to show the hierarchy, and color or label them to show additional variables.			Show both the “source” and the “destination” lists on the same page; let the user move items between them.	
			77	good defaults	237
				Wherever appropriate, prefill form fields with your best guesses at the values the user wants.	
			78	same-page error messages	239
				Place form error messages directly on the page with the form itself; mark the top of the page with an error message, and if possible, put indicators next to the originating controls.	

07	GETTING INPUT FROM USERS: FORMS AND CONTROLS	206
----	---	-----

The Basics of Form Design	207	
Control Choice	209	
The Patterns	218	
68	forgiving format	219
	Permit users to enter text in a variety of formats and syntaxes, and make the application interpret it intelligently.	
69	structured format	220
	Instead of using one text field, use a set of text fields that reflect the structure of the requested data.	
70	fill-in-the-blanks	222
	Arrange one or more fields in the form of a prose sentence or phrase, with the fields as “blanks” to be filled in by the user.	
71	input hints	224
	Beside an empty text field, place a sentence or example that explains what is required.	
72	input prompt	225
	Prefill a text field or dropdown with a prompt that tells the user what to do or type.	

08	BUILDERS AND EDITORS	242
----	----------------------	-----

The Basics of Editor Design	244	
The Patterns	248	
79	edit-in-place	249
	Use a small, dynamic text editor to let the user change text “in place”: position the editor directly over the original text, rather than using a separate panel or dialog box.	
80	smart selection	251
	Make the software smart enough to automatically select a coherent group of items, rather than making the user do it.	
81	composite selection	253
	Use different gestures—or mouse clicks in different screen areas, such as the composite’s edges versus its insides—to determine whether you should select a composite itself or allow its contained objects to be selected.	

82	one-off mode	255
	When a mode is turned on, perform the operation once. Then switch back automatically into the default or previous mode.	
83	spring-loaded mode	257
	Let the user enter a mode by holding down a key or a mouse button. When the user releases it, leave the mode and go back to the previous one.	
84	constrained resize	259
	Supply resize modes with different behavior, such as preserving aspect ratio, for use under special circumstances.	
85	magnetism	261
	Make the objects “magnetic” to the things a user positions them against. When the user drags an object very near one of these things, it should stick.	
86	guides	263
	Offer horizontal and vertical reference lines to help users align objects.	
87	paste variations	266
	Provide specialized paste functionality in addition to the standard paste operation.	

09

MAKING IT LOOK GOOD:

VISUAL STYLE AND AESTHETICS 268

Same Content, Different Styles	270	
The Basics of Visual Design	279	
What This Means for Desktop Applications	287	
The Patterns	290	
88	deep background	291
	Place an image or gradient into the page’s background that visually recedes behind the foreground elements.	
89	few hues, many values	294
	Choose one, two, or at most three major color hues to use in the interface. Create a color palette by selecting assorted values (brightnesses) from within those few hues.	

90	corner treatments	297
	Instead of using ordinary right angles, use diagonals, curves, or cutouts for some of the interface’s box corners. Make these corner treatments consistent across the interface.	
91	borders that echo fonts	300
	When drawing borders and other lines, use the same color, thickness, and curves used by one of the design’s major fonts.	
92	hairlines	303
	Use one-pixel-wide lines in borders, horizontal rules, and textures.	
93	contrasting font weights	306
	Use two contrasting fonts—one thin and lightweight, and the other heavier and darker—to separate different levels of information and add visual interest.	
94	skins	308
	Open up the look-and-feel architecture of your application so users can design their own graphics and styles.	

REFERENCES	312
------------	-----

INDEX	318
-------	-----