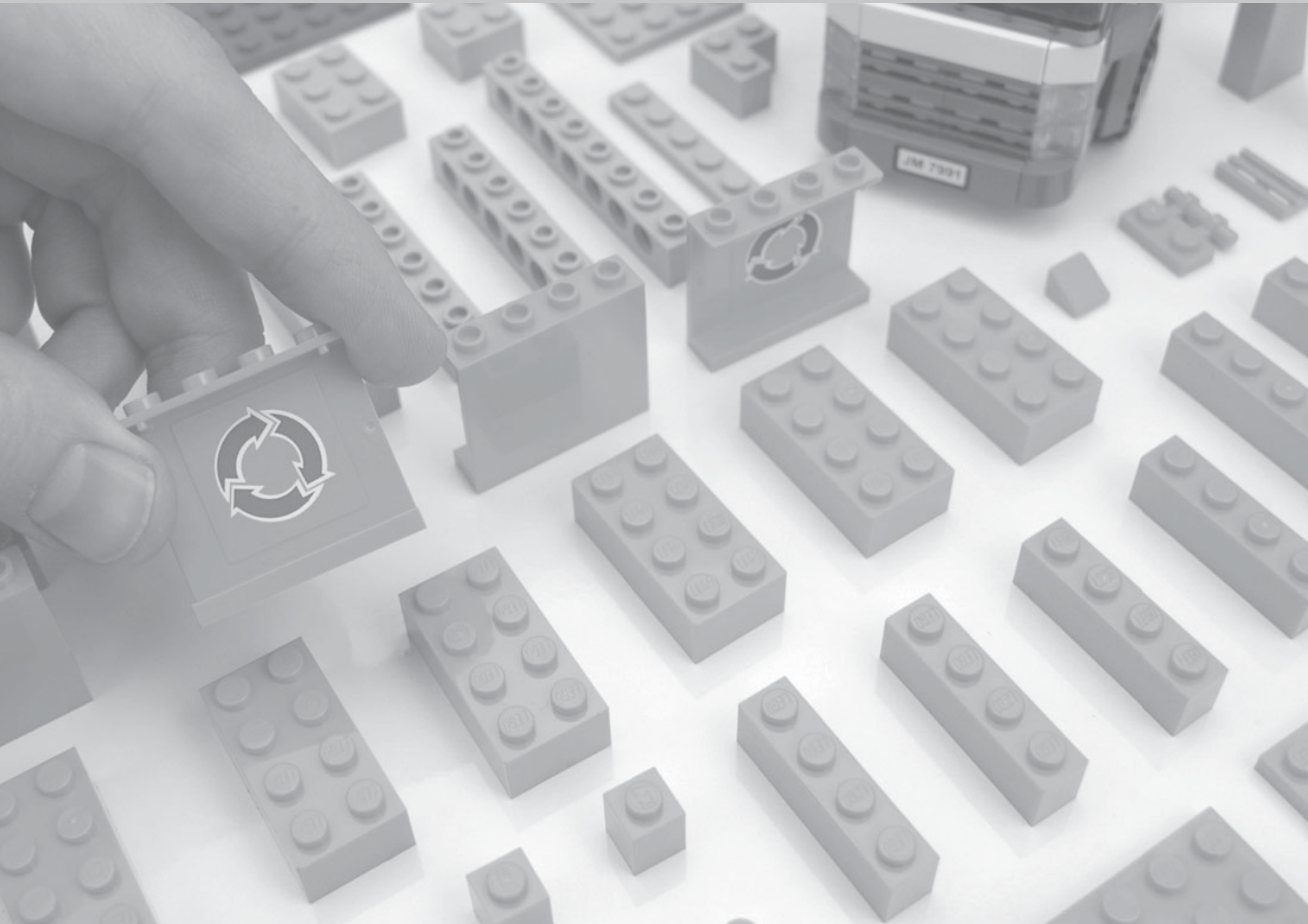
Creating Reusable Components for User Experience Design & Documentation

MODULAR WEB DESIGN

NATHAN CURTIS

MODULAR WEB DESIGN

Creating Reusable Components for User Experience Design



This page intentionally left blank

MODULAR WEB DESIGN

Creating Reusable Components for User Experience Design

Nathan Curtis

Peachpit
Press

Modular Web Design: Creating Reusable Components for User Experience Design and Documentation

Nathan Curtis

New Riders
1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)

Find us on the Web at www.newriders.com
To report errors, please send a note to errata@peachpit.com
New Riders is an imprint of Peachpit, a division of Pearson Education

Acquisitions Editor: MICHAEL NOLAN
Project Editor: VALERIE WITTE
Technical Editor: AUSTIN GOVELLA
Copyeditor: DOUG ADRIANSON
Proofreader: SCOUT FESTA
Production Editor: CORY BORMAN
Composition and Interior Design: MAUREEN FORYS
Indexer: VALERIE HAYNES PERRY
Cover Design: TERRI BOGAARDS

Notice of Rights

Notice of Liability

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

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 Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN-13: 978-0-321-60135-3

ISBN–10: 0-321-60135-1

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

To my son Liam,
with whom I build.

To my daughter Caroline,
who tears down with glee.

And to my wife Tina,
who makes it so we can all play.

Foreword

THE OTHER MORNING I got called into an emergency online meeting to look at a new design that one of our marketing groups had created and needed to roll out worldwide on a one-day deadline. Frankly, I feared the worst. Cisco is a pretty big company, and in a typical week I may see between 10 and 20 different designs, some of which in their first incarnations start with visual comps of things our Web team can't quickly implement, or otherwise take liberties with the brand identity or style and therefore require a course correction. (If you work in a design group anywhere, I am sure you will empathize.)

As I opened the PDF from the marketing group, I had a delightful surprise: The group had recently adopted our standard wireframing templates, pioneered by Nathan Curtis, and everything they had put together on this fast-turn project was completely standard. In discussing the design with them, I made one note on a component that they'd placed in an unusual location, and that was that. Design approved, deadline met. And I knew that there would be a good-looking, user-tested visual design and HTML code and style sheets available from our off-the-shelf library to back up what they had designed.

At Cisco, we have a component system that includes all of the commonly used capabilities on our sites. Increasingly, this system extends down to the level of HTML code and CSS. We put this system in place not because a central group has tight control of every page, but because we don't. Our Web teams are distributed and work collaboratively, and need to work from standard templates and documentation so that all of the pieces on our millions of Web pages work together.

Our employees who support our designs across our many Web areas are distributed worldwide, and in addition we have dozens of vendors worldwide who produce new creative. The fact that we have templates—that work down to the component level—is a big win for global consistency.

In the years I have worked with him, Nathan Curtis has been a passionate and tireless advocate for the component-driven design approach to design and design systems that ultimately made the above possible, including that pleasant interaction with that marketing group's design. Nathan's natural, module-driven philosophy can benefit almost any size of Web or mobile experience, and is essential for large sites that often manage hundreds of key experiences and present them via millions of pages that need to stay consistent and clear.

I am looking forward to having this book within easy reach at work and at home.

—Martin Hardee
Director, Web Experience Design, Cisco

Preface

BY THE FALL OF 2007, I'd been working on—and become fascinated by—the component libraries that teams use to efficiently and consistently create user experience. I'd been "thinking components" for much longer. So when my family traveled back to my boyhood home in northeastern Ohio for Thanksgiving, I wasn't prepared for the giant revelation that awaited me.

On the holiday morning, we were all sitting around the living room, and my mom dumped out a huge container of LEGO bricks in front of my 2-year-old son. He was awestruck and dove right in, putting all sorts of pieces together. As he built "ships" and other things, I uncovered the directions for building an actual, formal spaceship. He was intrigued, so I took advantage of a "teachable moment" to show him how to build LEGO constructions.

Whoa. As I slowly sifted through the massive LEGO pile, retrieving each individual piece, my son quickly lost interest and began building his own stuff again. So I set about finding all the pieces we needed, one by one, for over an hour. The set wasn't that big, but it took forever to source the collection. As I did so, I laid them out on the floor in an organized way, arranging pieces in rows and columns based on shape, size, color, and other facets. When done, my son and I launched into building the ship, and got it done in no time at all. And he was ecstatic.

Partway through that process, I realized that "this is what I do for my job."

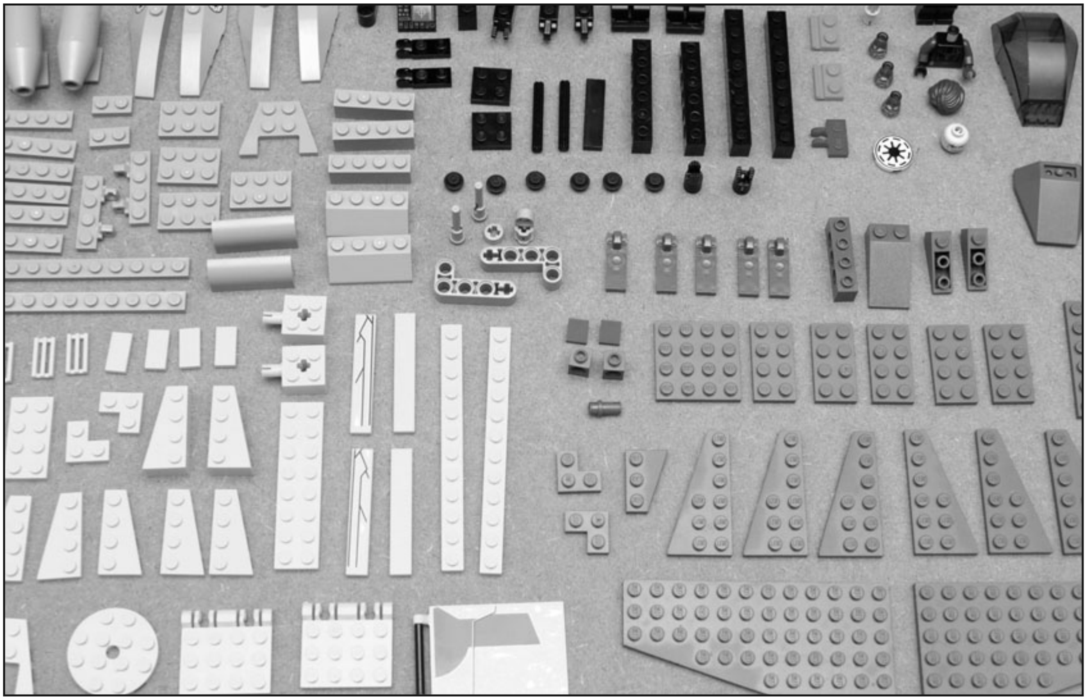
As a designer, each project begins with stakeholders requesting that we design and build something for them, much like some sort of spaceship. They may even bring a picture of what they want: a shiny fighter with all the latest bells and whistles.

Our initial reaction, knowing that we've built lots of ships before, is to respond with, "Well, here's how we build ships. It's not exactly like your picture, but it flies, goes really fast, has adjustable wings, engines, a cockpit, missiles, landing gear—the works." Collaborative discussions yield a common understanding of what they'll get, and then we go to the proverbial big bucket of LEGO to get started.

What do we find? An unorganized morass of all sorts of pieces. We start with the best intentions, but we must build it fast. We dump out all the LEGO on the floor. We grab pieces fast, and assemble a design. We have the best intentions, but we make compromises and uninformed decisions. Over many projects, we end up with a fleet of ships that sorta, kinda, look and feel the same. Or, more likely, they don't.

LEGO even goes so far as to clearly instruct us to start with two critical steps before putting any pieces together. First, make sure that you've got the right environment set up so that it's easy to snap each piece into place. For designers, that means using templates with grids, asset libraries, and more. Second, organize your pieces into piles, first by color but then those other "special" and miscellaneous piles. For designers, that means creating categorized, reusable chunks that are easy to find and use.

But what if, by sheer magic, when you dumped out your box, pieces magically aligned into helpful rows and columns based on color, shape, size, and special use? You could just glance at the table and grab the piece you need for each step. That's LEGO nirvana!



A table of near-perfectly organized LEGO pieces, arranged so that you can scan, choose, and use each piece to quickly build your ship.

That's usually how I feel when I'm assembling layouts based on reusable page components, even if they only solve 70 percent or 80 percent of the overall page design I need. I'm excited about creating the custom work needed for the last, focused innovation that's needed. But I get off to a great start, don't have to reinvent any wheels, and have a framework that's consistent with the overall design system. Your teammates, your engineers, and—most importantly—your users appreciate this.

This book covers two concepts: designing with components and standardizing an experience with a component library.

Part I explores design principles and techniques for chunking your design into components, and using those components to effectively design rich interactions and communicate them to others. Over the first six chapters, we'll cover the following:

1. Define. Understand components and how they fit in the design process.
2. Divide. Break down your page designs into meaningful chunks.
3. Vary. Communicate how a component changes under different conditions.
4. Combine. Assemble components together to form page designs.
5. Reuse. Apply principles for embedding and linking component instances in your artwork.
6. Document. Create useful deliverables to illustrate component-based designs.

With a solid foundation in component-based design principles and reuse, Part II teaches you how to build (chapters 7 through 11) and manage (chapters 12 through 15) a library of reusable component assets:

7. Appraise. Ask all the right questions to make sure you're ready to invest in a library.
8. Discover. Figure out what goes in—and what stays out of—your library.
9. Organize. Define categories, variations, names, keywords, and more.
10. Setup. Select your software tools, create templates, and decide conventions.
11. Build. Create each reusable chunk and package them up for everyone else.
12. Administer. Know your role as librarian, and be ready to curate the collection.
13. Guide. Document the role each component plays in your experience.
14. Adopt. Execute a planned series of activities so that your library takes hold.
15. Integrate. Transform how your team gets work done using components.

With that in mind, let's break it down.