Making Everything Easier!™

Pattern-Oriented Software Architecture

FOR DUMMIES®

Learn to:

- Understand software architecture basics and implement best practices
- Recognize and utilize patterns, layers, pipes, filters, and MVC
- Create patterns and build your own pattern collection
- Incorporate plans for emerging platforms and technologies into your projects

Robert Hanmer

Making Everything Easier!™

Pattern-Oriented Software Architecture

FOR DUMMIES®

Learn to:

- Understand software architecture basics and implement best practices
- Recognize and utilize patterns, layers, pipes, filters, and MVC
- Create patterns and build your own pattern collection
- Incorporate plans for emerging platforms and technologies into your projects

Robert Hanmer

Pattern-Oriented Software Architecture For Dummies®

Visit www.dummies.com/cheatsheet/patternorientedsoftwarearchitecture to view this book's cheat sheet.

Table of Contents

- Join a Mailing List
- Join a Reading Group
- Write Your Own Patterns
- Attend a Pattern Conference
- Start a Writers’ Workshop

Cheat Sheet

Pattern-Oriented Software Architecture For Dummies®

by Robert Hanmer

A John Wiley and Sons, Ltd, Publication

Pattern-Oriented Software Architecture For Dummies®

Published by
John Wiley & Sons, Ltd
The Atrium
Southern Gate
Chichester
West Sussex
PO19 8SQ
England

Email (for orders and customer service enquires): cs-books@wiley.co.uk

Visit our home page on www.wiley.com

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats. For more information about Wiley products, visit us at www.wiley.com.

British Library Cataloguing in Publication Data: A catalogue record for this book is available from the British Library.

ISBN 978-1-119-96399-8 (pbk); ISBN 978-1-119-96631-9 (ebk); ISBN 978-1-119-96632-6 (ebk); ISBN 978-1-119-96630-2 (ebk)

Printed and bound in the United States by Bind-Rite

10 9 8 7 6 5 4 3 2 1

About the Author

Robert Hanmer is a director of The Hillside Group, an organization whose mission is to improve quality of life for everyone who uses, builds, and encounters software systems. The Hillside Group also sponsors Pattern Languages of Programming (PLoP) software pattern conferences. Bob is active in the software pattern community and has been program chair at pattern conferences in the United States and overseas.

He is a consulting member of technical staff with Alcatel-Lucent near Chicago. Within Alcatel-Lucent, Lucent Technologies, and Bell Laboratories (same office, new company names), he is involved in development and architecture of embedded systems, focusing especially on the areas of reliability and

performance. Previously, he designed interactive graphics systems used by medical researchers.

Bob is the author of *Patterns for Fault Tolerant Software* (Wiley) and has written or co-written 14 journal articles and several book chapters. He is a senior member of the Association for Computing Machinery, a member of the Alcatel-Lucent Technical Academy, and a member of the IEEE Computer Society. He received his BS and MS degrees in Computer Science from Northwestern University in Evanston, Illinois.

Dedication

For Karen

Author's Acknowledgments

First, and most important, I want to acknowledge the authors of Pattern-Oriented Software Architecture: A System of Patterns (Wiley): Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. Peter also has been helpful with questions about modern C++ and the software architecture classroom.

Many other people answered questions, reviewed sections, or generally consulted with me while I was writing this book. Thanks to Ademar Aguiar, Omar Aldawud, Dan Bergen, Filipe Correia, Chuck Corwin, Jerry Dzeidzic, Christoph Fehling, Becky Fletcher, Brian Foote, Karen Hanmer, Kenji Hiranabe, Lise Hvatum, Satomi Joba, Dr. Ralph Johnson, Capt. U.S. Navy (Ret.) Will H. Jordan, Steven P. Karas, Allan Kelley, Christian Kohls, Christian Koppe, John Krallman, John Letourneau, Steffen Macke, Dennis Mancl, Jyothish Maniyath, Veena Mendiratta,

Pedro Monteiro, Karl Rehmer, Linda Rising, Hans Rudin, Eugene Wallingford, Michael Weiss, and Joe Yoder.

Thanks to the members of my writers' workshop group at PLoP 2011 who held a workshop on parts of this book: Dr. Tanya L. Crenshaw, Andre Hauge, Jiwon Kim, Alexander Nowak, Rick Rodin, YoungSu Son, and Hironori Washizaki.

The Real-World Example sidebars in the pattern chapters are based on a workshop at the 1998 OOPSLA conference. It was organized by Michael Duell, Linda Rising, Peter Sommerlad, and Michael Stal. Russ Frame, Kandi Frasier, Rik Smoody, and Jun'ichi Suzuki participated in the workshop and contributed to the examples that I've adapted here.

Thanks also to the many people at John Wiley & Sons, including Birgit Gruber, Chris Katsaropoulos, Elizabeth Kuball, Ellie Scott, Jim Siddle, Kathy Simpson, Chris Webb, and the others whose names you see on the Publisher's Acknowledgments page.

Publisher's Acknowledgments

We're proud of this book; please send us your comments at http://dummies.custhelp.com. For other comments, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

Some of the people who helped bring this book to market include the following:

Acquisitions and Editorial

Project Editor: Elizabeth Kuball

Executive Commissioning Editor: Birgit Gruber

Assistant Editor: Ellie Scott

Copy Editor: Elizabeth Kuball

Technical Editor: James Siddle

Editorial Manager: Jodi Jensen

Sr. Project Editor: Sara Shlaer

Editorial Assistant: Leslie Saxman

Cover Photo: © teekid / iStock

Cartoons: Rich Tennant (www.the5thwave.com)

Composition Services

Senior Project Coordinator: Kristie Rees

Layout and Graphics: Joyce Haughey

Proofreaders: John Greenough, Tricia Liebig

Indexer: Sharon Shock

Marketing

Associate Marketing Director: Louise Breinholt

Marketing Manager: Lorna Mein

Senior Marketing Executive: Kate Parrett

Marketing Assistant: Tash Lee

UK Tech Publishing

Michelle Leete, Vice President Consumer and Technology Publishing Director

Martin Tribe, Associate Director–Book Content Management

Chris Webb, Associate Publisher

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Kathleen Nebenhaus, Vice President and Executive Publisher

Composition Services

Debbie Stailey, Director of Composition Services

Introduction

Wouldn't it be great to never rewrite code? To always face new challenges rather than solve the same problems over and over? To always solve new and interesting problems instead of rehashing old ones? If you remember how you solved a problem before, reuse that solution. Don't reinvent the wheel!

Software patterns help you avoid reinventing the wheel, in that they help you avoid reinventing the solution to a software problem that someone else has already solved.

Patterns have been around in the software community since at least the early 1990s. Software pattern authors have been writing patterns that document their proven solutions in the hope that you — the reader — will benefit from their experience.

In particular, many people are collecting and publishing patterns that structure software architecture — the underlying structure of the software. The goal of architectural patterns is to speed your development; allow you to move forward, knowing that a particular architecture will help rather than hinder you; and ultimately give you the time you need to solve new and interesting problems.

Pattern-Oriented Software Architecture For Dummies is written to help you understand the basics of software architecture. It also helps you understand software patterns. The book brings these two concepts together and presents eight software architectures that you can use in your next software design project. It also gives you some design patterns, tips, and resources where you can find out more about software patterns.

About This Book

This book provides proven architectures and designs expressed as patterns. These patterns aren't the only ways you can structure your software architecture, though, and this book doesn't replace the other references you use for software design patterns.

As you read this book, keep in mind that you can't just plug-and-play these patterns. *Your* intelligence and taste are required to adapt these patterns to *your* design problem. This is the norm with software patterns: No respectable pattern author will tell you that you can use his or her patterns without adapting them to your situation.

In the early days, software patterns provided valuable assistance to people who were trying to get a handle on object-oriented design. The discussions of these patterns seemed to me, however, to focus on getting the structure of the object-oriented program's header files and class definitions correct at the expense of the real application. In this book, I give you an understanding of the solutions to the problems, not the detailed header files. I want you to understand the principles involved rather than get caught up in the implementation details. As a result, this book isn't language-specific or programming paradigm-specific; instead, it explains the underlying principles involved in the solutions that you will apply using your prior experience and expertise.

Finally, you don't have to read the whole book from front cover to back. Instead, use the table of contents and index to locate the information you need when you need it.

Conventions Used in This Book

Here are the conventions I use throughout this book:

- I capitalize the names of patterns. In some chapters, the name of the pattern is the same as the name of a key component of the architecture. In general, the pattern name is capitalized, and the name of the component is not capitalized.
- I abbreviate the names of many of the patterns discussed in Parts III and IV because they're quite long. Model-View-Controller, for example, becomes MVC. On the first use in a chapter, the whole name is spelled out, and the abbreviation is used thereafter.
- When I introduce a new term, I put it in *italics* and define it shortly thereafter (often in parentheses).
- I put web addresses in `monofont` so they stand out from the surrounding text. ***Note:*** When this book was printed, some web addresses may have needed to break across two lines of text. If that happened, rest assured that we haven't added extra characters (such as hyphens) to indicate the break. So, when using one of these web addresses, just type in exactly what you see in this book, pretending as though the line break doesn't exist.

What You're Not to Read

I've sprinkled a few sidebars around in the text. They show up as gray boxes. You can safely skip them. They contain information that I think you may find useful but that isn't required to understand the patterns or software architecture.

You also can skip anything marked with a Technical Stuff icon (see “Icons Used in This Book,” later in this Introduction, for more information).

Foolish Assumptions

I make some assumptions about who would read and benefit from this book. I don’t expect that you’re an expert in software architecture; in fact, I assume that you’re pretty new to it. I do assume that you know something about writing software, however, and that you’ve already written some software. In particular, I assume that you’ve written software in some sort of team setting on a project bigger than a school project. From this experience, you’ll have learned about designing with modules and components.

Because more software is changed, evolved, and maintained than written from scratch, I assume that you’ve experienced some software maintenance. Maintenance of someone else’s (or even your own) code will have given you an understanding of the importance of modularity and good structure.

I don’t assume that you’re an expert in object-oriented design or any other particular design methods. The architectures in this book can be adapted to any paradigm you work in and are familiar with. Some familiarity with at least the basic terminology of objects, classes, and methods is assumed.

How This Book Is Organized

This book has five parts. Parts I and II introduce software architecture and software patterns. The next two parts present real live patterns that you can use in your software. Finally, Part

V shows you where to turn next to explore the exciting world of software patterns.

Part I: Introducing Software Architecture and Patterns

To build a foundation for the rest of the book and to explain the basic concepts, Part I focuses on software architecture: what it is, how to create it, and how to document it. Architecture builds on the needs of the customer or client, so Part I also talks about the requirements that shape your architecture.

Architecture needs to be explained to those who will build the application. Even if you're the sole builder, an explanation will help you remember later what you did today. Part I introduces various ways of documenting your architecture, including simple Class-Responsibility-Collaboration cards, the basics of the Unified Modeling Language, and an outline of an architecture description document.

Part I ends with a chapter that describes the basics of software patterns. This chapter provides a foundation for the discussions in Part II of making the most of software patterns.

Part II: Putting Patterns to Work

You need to find patterns that address the problems you need to solve. Part II describes how patterns are organized and catalogued. It also presents a process you can use to find the patterns that can help you.

As you start using patterns, you'll find that you use the same patterns over and over. Part II has instructions for collecting the