book reviews

ELIZABETH ZWICKY, WITH MORTEN WERNER FORSBRING, TROND HASLE AMUNDSEN, STÅLE ASKEROD JOHANSEN, BRANDON CHING, JASON DUSEK, AND SAM F. STOVER

DESIGN PATTERNS EXPLAINED: A NEW PERSPECTIVE ON OBJECT-ORIENTED DESIGN

Alan Shalloway and James R. Trott

Addison-Wesley, 2005. 418 pages.
ISBN 0-321-24714-0

Here's a telling fact: to write the review of this book, I had to make my husband give it back to me, because I suggested it when he needed to solve a problem, and he had squirreled it away so he could read more of it. He did lend it back to me for the duration of the review, but not without regret.

I learned my object-oriented programming on the street, and my patterns mostly by tripping over them, so I've been looking for a good book explaining what patterns really, properly are, and how real programmers use them. This is a lucid, practical explanation of patterns and how you can use them to design programs.

It works through a bunch of basic patterns, motivating the use of the pattern by presenting a problem that is well solved by the pattern. It discusses issues in integrating pattern-based designing with other systems and provides an outline of how you can figure out what patterns to use and how to string them together, starting from a customer specification, and including suggestions on identifying what the customer left out of the specification.

This is a good starter book on patterns, offering a refreshingly concrete take on a very abstract subject. It will make the most sense to somebody with programming and program maintenance experience, and it helps to know something about Agile/XP practices as well (at least, you should be aware that they exist). If you don't have a basic familiarity with patterns and the pattern literature, start at the beginning. There's no reason one might believe otherwise, but my poor husband got thrown in at page 300-odd, and while he did figure out that it was in fact the answer he was looking for, he had some trouble with the references to "the Gang of Four," which he associated with politics (not Chinese politics, either, so my attempts to disambiguate merely dragged us in irrelevant directions). Starting at the beginning would have clarified this as a shorthand for the authors of the seminal book on patterns in programming.

BEYOND BARBIE & MORTAL KOMBAT: NEW PERSPECTIVES ON GENDER AND GAMING

Yasmin B. Kafai, Carrie Heeter, Jill Denner, and Jennifer Y. Sun

MIT Press, 2008. 350 pages.
ISBN 978-0-262-11319-9

If you're interested in gender issues in computing, this is a nicely nuanced collection. The issues for games are slightly different from the issues for other kinds of computing, but not massively so. Mostly, they're just more visible. The book also points out that gaming is explicitly linked to entry into other computing fields, particularly IT. (To play a networked game, you've got to get the computer on the network.)

This book explores several themes dear to my heart. For instance, appealing to women doesn't mean adding pink bows, and doing it well may give you a product that's more appealing to everybody. Women are, after all, not a different species. The social pressures that restrict women's access to computers are not all overt; they include subtle and unsubtle differences in where people can go when. Often it's not obvious to men that a situation may be unsafe or socially unacceptable for women and that "lack of interest" reflects an effective lack of access.

PROCESSING: A PROGRAMMING HANDBOOK FOR VISUAL DESIGNERS AND ARTISTS

Casey Reas and Ben Fry

MIT Press, 2008. 698 pages.
ISBN 978-0-262-18262-1

I love processing; I've said it before, and no doubt I'll say it again. I already reviewed (and loved) Ben Fry's *Visualizing Data*, which includes a basic introduction to processing suitable for people who're comfortable programming in something or other. This book is aimed, as it says, at people who know more about pictures than about programs. Part of what it does is proselytize for the joy of making art with programs, and it assumes the reader knows nothing about programming. (Seriously, nothing.

Like, variables and loops are explained from the ground up.)

From there, it gets to complicated programs, and on the way it throws in handy tips I learned the hard way (such as the fact that sine and cosine functions are endlessly useful to give smooth variation). It covers the basics of programming, but the artistic bent is visible in any number of ways. The clearest are the sections about computer artworks, which are luscious, but the topics covered are also different from the ones you might expect. Anybody could figure out there'd be more about color handling, but if you haven't thought about computer-based art a lot, you might be surprised to note the early and extensive handling of sources of randomness and noise.

This is a good introduction to a good language for visual artists. It may also provide a good window into the world of art programming for programmers, although anybody with a basic knowledge of any object-oriented programming language is going to find themselves skipping a lot. For people who already do art programming, it may be worth it for the inspiration and the coverage of extensions, but if you're at all confident with programming, you'd do just fine with the Web site.

DAVID POGUE'S DIGITAL PHOTOGRAPHY: THE MISSING MANUAL

David Pogue

O'Reilly, 2009. 284 pages.
ISBN 978-0-596-15403-5

I recently bought a new camera, and my husband asked if he should read the manual. I said that was a terrible idea, as I had barely survived the experience, and I have a lot more background than he does. But he wanted something that would help him understand more camera geekery. Normally, for somebody who doesn't know anything about cameras and wants to take better pictures, I recommend Nick Kelsh's books—*How to Photograph Your Life* is a good all-purpose choice. And he's read it, but he wants more: something with more technical details, but still reasonable for somebody who never can remember what an f-stop is and whether you want to increase or decrease exposure when you're trying to take a picture of something white.

As I had hoped, *David Pogue's Digital Photography* is the right choice here. It's a beginner's book, for sure, but it talks about both technique and what buttons to push. It puts everything in the context of photos you might want and how to get them, so you don't have to try to understand f-stops without first deciding that you care about how much of

your picture is blurry and what's in focus. It gives a swift picture of the whole camera-buying and -using process, starting from selecting a camera, continuing through using it, dealing with the pictures, editing them, and showing them off.

That's a lot to cover in 284 pages, but he does a good job of hitting the high points, for the most part. When there's no right answer to a dilemma, he's willing to say so. Should you delete photos on the camera or after downloading? It depends; do you have space left? Is looking at them in glorious detail going to help you decide between them or merely depress you?

I disagree with him on some points, of course; I used to love lens-cap tethers, but they disappointed me too often, either by failing suddenly or by letting a lens cap or tether dangle into my photo. The front cap I just leave the way it comes. But I have to modify the back cap with a dot of contrasting paint, because one of my lenses is terribly finicky about alignment.

This is a good introduction to digital cameras to people who are not intimidated by technology but don't know the nuts and bolts of photography. It's not going to satisfy a serious film photographer looking to switch to digital, and it's not going to give you much geek cred. It is going to help the average person take better photographs.

LEARNING NAGIOS 3.0

Wojciech Kocjan

Packt Publishing, 2008. 301 pages.
ISBN 978-1-847195-18-0

REVIEWED BY MORTEN WERNER FORSBRING, TROND HASLE AMUNDSEN, AND STÅLE ASKEROD JOHANSEN

The University of Oslo has been using Nagios as the primary monitoring tool for UNIX servers and their services for two years. We switched to version 3.0 only some months after its release. We monitor about 750 hosts and, in Nagios terms, approximately 11k services, in a diverse and heterogeneous academic installation with ~60k users. We have spent much time tuning and adapting Nagios to our needs, and we know some of its strengths and weaknesses, although we do not use or know in detail all of its features.

This book aims to be a general guide to Nagios as well as to the new features in version 3.0. Our impression is quite good. The focus is said to be on giving a good introduction to system administrators who are new to Nagios and want to learn more about it in general. In our opinion, the author does indeed provide a good overview of Nagios as an ad-

ministration tool and offers a good start for people who are curious about trying it out.

The book does not venture into all of the challenges we have faced with Nagios, so people who are looking for more hardcore information might need to look deeper in Nagios's own documentation or on the various mailing lists.

That being said, in its 11 chapters the book covers basically all the steps necessary to get a more-than-basic Nagios rig up and running. Everything from installation to using more complex but essential features such as NRPE and NSCA is covered. In between you'll find explanations on what monitoring really is all about and the role of Nagios, as well as configuration and explanations on how to monitor specific typical things (e.g., email, processes, how ports answer).

There is also a focus on scalability and tuning, with good examples of "best practice." We also like the fact that the book mentions using Nagios in larger environments—for example, discussing the challenges of handling SSH checks. We agree that, for many, NRPE will be a better choice, but you have to pay attention to its security issues.

What do we miss? Considering the title, the book does its job. We would have liked more info on tuning the GUI, but many use it less than we do, and there are projects to improve it.

There are also several other things the book does not include, but it feels unfair to expect too much detail from a book targeted at beginning Nagios users.

What did we learn? We had several "wakeups," where we got reminded about things we want to look into—for instance, features we haven't started using yet. In this sense we think this book could be a useful read for many people who use this great software already. The book also covers many of the new features in version 3.0, which are very useful for those who are already into Nagios to be aware of.

Regarding the features of Nagios we don't use, we naturally cannot evaluate the content. This includes: monitoring Windows; extensive use of dependencies; adaptive monitoring; and running multiple Nagios servers.

This book is a good read for all system administrators who want to learn more about Nagios or want to start using it. Although some large installations may require more configuration and adaptation than this book goes into, it does cover all the important topics to get you up and running, and then some.

WEB 2.0: A STRATEGY GUIDE

Amy Shuen

REVIEWED BY BRANDON CHING

Some say that imitation is the sincerest form of flattery. If that is indeed the case, then Amy Shuen has written an informative guideline for all of us to flatter the likes of Google, Flickr, Amazon, and Facebook. In *Web 2.0: A Strategy Guide*, Shuen analyzes the Web 2.0 movement by deconstructing the strategies used by successful Web companies large and small; then she explains the how and why of their success through Web 2.0 principles.

Shuen details, in six chapters of highly informative analysis, how companies can adopt Web 2.0 ideas to enhance profitability and expand their businesses. Shuen does this through a detailed analysis of many successful Web 2.0 companies and many of the strategies they used to add value and profitability to the services they offer. According to Shuen, the strategies of primary importance to Web 2.0 businesses are to build on collective user value, activate network effects, work through social networks, dynamically syndicate competence, and recombine innovations.

Shuen begins by analyzing the collective user value concept by scrutinizing the strategies of Flickr and Netflix. The user value concept is the idea that users, not the companies themselves, are the main contributors of value. By allowing users to add, share, organize, and promote their own content, businesses need only provide a context through which users can interact. Although this may sound easy, balancing user demands while ensuring profits and adding value to services is, as Shuen points out, difficult to attain.

Building upon the collective user value concept is the network effect. "Positive network effects increase the value of a good or service as more people use or adopt it" (p. 41). In this chapter, Shuen breaks down how Google's pay-per-click keyword advertising has used a variety of positive network effects to generate sustainable profitability ($8 billion per quarter, p. 44). With coverage of five different types of positive network effects, Shuen provides a solid overview of ways your business can capitalize on this core Web 2.0 principle.

Perhaps one of the most visible Web 2.0 strategies is that of social networking. Chapter three delves into the understanding of this effect and covers topics such as Malcolm Gladwell's tipping point theory, the Rogers adoption curve, and the Bass

diffusion curve. These more theoretical explanations for social networking effects Shuen follows by an analysis of the rise in membership, participation, and success of Facebook and LinkedIn.

Every business has a set of core competencies. Chapter four analyzes IBM, SalesForce.com, and Amazon, showing how each company used their core competencies in dynamic ways to expand service offerings and increase profits. In the case of Amazon, the decision to open up their online book-selling service to third-party sellers in 2001 resulted in additional billions of dollars in sales. Later, capitalizing on their experience of running a large, multi-million-user system, Amazon began offering back-office competencies through Web services such as the Simple Storage Service (S3) and the Elastic Compute Cloud (EC2).

Finally, Shuen covers the recombining innovation aspect of the Web 2.0 movement by addressing how businesses can expand upon existing services utilizing different modes of innovation including democratized, crowd-sourcing, ecosystem, platform, and recombinant innovation. The most prominent example Shuen addresses here is that of Apple's iPod and iTunes application capitalizing on platform innovation through iPod assembly, creative software, accessories, and user-provided music data.

The final chapter is a short how-to consisting of five steps on how to incorporate Web 2.0 strategies into your business. Essentially a summary of the previous five chapters, it also provides general business ideas and strategies on getting started with your newly acquired Web 2.0 knowledge. Weighing in at only 15 pages, this chapter won't give you all your answers, but it does give you a good push in the right direction.

Shuen's book is an exceptional analysis of many processes and strategies used by successful Web 2.0 companies. Each chapter ends with a valuable series of strategic and tactical questions that can be asked by business owners and executives which directly relate the principles in the chapter to Web and IT-based organizations.

Web 2.0: A Strategy Guide should be considered required reading for business executives and students, entrepreneurs, and information technology investors. Although approachable, the book does contain a good amount of business and economic principles and terminology that may leave some just skimming a few pages here and there. Yet, overall this was a very good, and relatively concise, source of valuable information on the busi-

ness strategies and principles of successful Web 2.0 businesses.

REAL WORLD HASKELL

Bryan O'Sullivan, John Goerzen, and Don Stewart

O'Reilly, 2008. 710 pages.
ISBN 978-0-596-51498-3

REVIEWED BY JASON DUSEK

Haskell has long lacked a book with both up-to-date worked examples and an overview of language fundamentals. *Real World Haskell* introduces pure functional programming with the Haskell language and covers a wide array of libraries that are in active use.

Most programmers coming to Haskell will find their greatest difficulty lies in learning purely functional programming and type inference. These two features allow a rich combination of optimization, code verification, and script-like brevity. *Real World Haskell* is veined with explanatory material, highlighting the utility of Haskell's novel features; for example, Chapter 25, on profiling and optimization, illustrates how functional purity and static types allow for a remarkable speedup without damage to clarity or code size.

At first, many common idioms appear impossible or nonsensical in the functional setting. Among them are error handling with exceptions, pointer arithmetic, and multithreaded programming with shared values. Nothing could be further from the truth, and *Real World Haskell* does an excellent job of clarifying both how these idioms fit in with purely functional programming and how they are executed in practice in Haskell.

Material in the text runs the gamut from introductory to practical to sophisticated. The first four chapters are devoted to introducing the language and pure functional programming; remaining chapters are divided between the case studies—syslog, sockets, databases, parsing /etc/passwd—and discussion of functional programming idioms. The famed science of monadology is made approachable without dumbing down.

Type inference, pattern matching, and the Haskell type checker are powerful aids to static verification. These unfamiliar tools are demonstrated throughout the text with appropriate comparison to idioms in other languages.

Some figures and footnotes are notably in error; this is generally obvious (e.g., a footnote labeled "a" but referenced by an asterisk). Most of the errors I encountered have found their way onto the errata

page (at http://oreilly.com/catalog/9780596514983/
errata/). There are some places where the text re-
peats itself (e.g., on page 574). These mistakes in
presentation are a small blemish on fine material.

In referencing online documentation and API
search tools, the book upholds the standard of all
O'Reilly books. *Real World Haskell* is itself available
online, at http://book.realworldhaskell.org/. The site
supports per-paragraph comments and has been
collecting notes since January of 2008.

ALGORITHMS IN A NUTSHELL

George T. Heineman, Gary Pollice, and Stanley Selkow

O'Reilly, 2008. 362 pages.
ISBN 978-0-596-51624-6

REVIEWED BY JASON DUSEK

Algorithms in a Nutshell offers an informal, compact
reference to basic algorithms for sorting, searching,
and graph traversal as well as more specialized ma-
terial for computational geometry, pathfinding, and
network flow.

The text touches upon a number of programming
languages but the vast majority of examples are in
Java, C, or C++. A chapter tends to use all one or
another: the reader is assumed (not unreasonably)
to have a reading knowledge of all three.

Deliberately light on math, the text's presenta-
tion is inspired by Christopher Alexander's *A Pat-
tern Language,* which is introduced in the chapter
"Patterns & Domains." After a short visit with the
philosophers, this chapter introduces the finest or-
ganizational aspect of this text: the algorithm trad-
ing cards, short fact sheets with well-thought-out
iconography to allow you, at a glance, to assess an
algorithm's performance, operating principles, and
relation to fundamental data structures. In addi-
tion to the trading cards, the book offers plentiful
source code, pseudo-code, and diagrammatic ex-
planations of the algorithms.

In keeping with the practical emphasis of the text,
the authors generally benchmark the algorithm
under study on two or three computer systems in
addition to presenting its "big O" characteristics.

A few small printing errors detract from the text.
On page 137, there is either an error or there is
an airline with service from one's desktop to one's
router.

Algorithms in a Nutshell is readable and informative;
it is both an excellent refresher and a fine introduc-
tion to a number of important algorithms.

OS X EXPLOITS AND DEFENSE

*Kevin Finisterre, Larry H., David Harley, and
Gary Porteous*

Syngress, 2008. 400 pages.
ISBN 978-1-59749-254-6

REVIEWED BY SAM F. STOVER

I walked away from this book with a burning ques-
tion in my mind: Just who exactly is the target au-
dience? In my opinion, if you are a reverse engineer
who can walk though the stack but need screen-
shots to help you install KisMAC on a Mac and you
are interested in a lot of OS 9 historical facts, this
is the book for you. Instead of a book for newbies
and veterans alike, this book feels unbalanced in
its content.

The first chapter starts out with an incredibly brief
description of the boot processes for OS 9 and OS
X. The rest of the chapter discusses forensic tool-
kits designed for the Mac, which is really just a
bunch of GUIs wrapped around *NIX utilities
we've all been using for years. Chapter 2 then goes
into an incredible amount of detail (e.g., discussing
memory register contents) of several bugs in OS 9,
early versions of OS X, and even Windows running
via Cross Over/WINE. Very deep stuff.

Chapter 3 then takes a look at Mac-focused mal-
ware since the beginning of time. There is a lot of
background and historical information; in fact the
chapter contains more OS 9 information than I
ever knew existed. Chapter 4 outlines the different
tools, both past and present, to detect malware on
the Mac platform. Again, there's lots of historical
info, but maybe more than you would find interest-
ing—definitely more than I found interesting.

Chapter 5, "Mac OS X for Pen Testers," was the
chapter I was honestly looking forward to but was
ultimately disappointed in. If you need help install-
ing the Developer Tools, X11, CPAN, Wireshark,
and nmap, then read this chapter. If you can figure
that stuff out on your own, don't bother. The same
goes for the next chapter, which addresses war-
driving and wireless pen testing by walking you
through installing and using KisMAC.

Chapter 7, "Security and OS X," is very reminis-
cent of Chapter 2 in that you get a very in-depth
description of buffer overflows, library injection,
and address space layout randomization. I thought
this chapter was the highlight of the book: The in-
formation is current, valid, and interesting, and the
explanations were solid.

The final chapter, on encryption implementations,
starts with an overview of encryption in OS 9, then
follows changes and additions in OS X such as File-

Vault, SSHD, WEP, and WPA. Not a bad chapter, if you are interested in learning the evolution of encryption in the Mac world.

Overall, I'd have to say that this book was a bit of a disappointment. I really liked Chapter 7, and even Chapter 2 held my interest, but beyond that, I felt as though I was being inundated with either OS 9 information I don't care about or installation instructions for open source tools. If you really want to learn about pen testing, on a Mac or otherwise, I'd recommend the *Penetration Tester's Open Source Toolkit*. If you are interested in learning more about your Mac, I'd recommend *OS X for Hackers*. But that's just me.

USER FRIENDLY by J.D. "Illiad" Frazer

OH, HA HA! SIR, IT'S JUST A SECURITY QUESTION TO ENSURE YOU'RE THE AUTHORIZED PARTY. SO WHO DO YOU BANK WITH?

UM. I DON'T HAVE A BANK.

BUT...YOU SAID YOU BANKED WITH US?

THAT'S BECAUSE YOU SAID YOU WERE CALLING FROM MY BANK. SO HOW MUCH DO I HAVE?