

## Volume 13, Number 3, December 2006

---

*Volume 13, Number 3, December 2006*

---

Technology Electronic Reviews (TER) is a publication of the Library and Information Technology Association.

Technology Electronic Reviews (ISSN: 1533-9165) is a periodical copyright © 2006 by the American Library Association. Documents in this issue, subject to copyright by the American Library Association or by the authors of the documents, may be reproduced for noncommercial, educational, or scientific purposes granted by Sections 107 and 108 of the Copyright Revision Act of 1976, provided that the copyright statement and source for that material are clearly acknowledged and that the material is reproduced without alteration. None of these documents may be reproduced or adapted for commercial distribution without the prior written permission of the designated copyright holder for the specific documents.

---

### **News from the *TER* Editorial Board by Martin R. Kalfatovic**

This issue of TER represents the first under my editorship. I would like to thank outgoing editor Sharon Rankin for all her work the past few years on TER. Thanks are also due the outgoing TER Editorial Board members, especially Florence Tang. New board members for this issue include Terry Reese, Cataloger for Networked Resources & Digital Production Unit Head, Oregon State University Libraries, Corvallis, and Brian L. Pytlik, Digital Initiatives Librarian, University of Nebraska-Lincoln Libraries. With the publication of this issue in December 2006, TER will close out volume 13. Look forward to some new and interesting changes to TER in 2007. Please feel free to contact me directly ( [kalfatovicm@si.edu](mailto:kalfatovicm@si.edu) (<mailto:kalfatovicm@si.edu>)) with any suggestions or changes that will make TER even better.

---

### **Contents:**

- REVIEW OF: Doug Addison. (2006). *Web Site Cookbook*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN 0-59610-109-0). 261 pp. \$39.99.
- REVIEW OF: Jacob Babbin, et.al. (2006). *Security Log Management: Identifying Patterns in the Chaos*. Rockland, MA: Syngress Publishing, Inc. (ISBN 1-59749-042-3). 350 pp. \$49.95
- REVIEW OF: Dave Crane and Eric Pascarello, with Darren James. (2006). *Ajax in Action*. Greenwich, CT: Manning. (ISBN 1-932394-61-3). 650 pp. \$44.95

- REVIEW OF: Leander Kahney. (2005). *The Cult of iPod*. San Francisco: No Starch Press. (ISBN: 1-59327-066-6). 160 pp. \$24.95.
- REVIEW OF: Bruce W. Perry. (2006). *Ajax Hacks: Tips & Tools for Creating Responsive Web Sites*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN: 0-596-10169-4). 414 pp. \$29.99.
- REVIEW OF: Chris Pine. (2005). *Learn to Program*. Raleigh, NC: The Pragmatic Programmers, LLC. (ISBN: 0-9766940-4-2). 149 pp. \$19.95.
- REVIEW OF: Eric Pulier and Hugh Taylor. (2006). *Understanding Enterprise SOA*. Greenwich, Conn.: Manning Publications. (ISBN: 1-932394-59-1). 242 pp. \$39.95.
- REVIEW OF Rich Shupe and Robert Hoekman, Jr. (2006). *Flash 8: Projects for Learning Animation and Interactivity*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN: 0-596-10169-4). 340 pp. \$34.99

### Reviewers in this issue:

- Jim Blansett is Information Resources Librarian in the McLure Education Library of the University of Alabama
- Mark Cyzyk, formerly Head of Information Technology in the Albert S. Cook Library at Towson University, is currently the Scholarly Communications Specialist in the Library Digital Programs group, The Sheridan Libraries, Johns Hopkins University, Baltimore, Maryland
- Paul Kittle is the Distance Learning/Electronic Reference Librarian, at Mt. San Antonio College in Walnut, California
- Holly Mercer is the Coordinator of Digital Content Development at the University of Kansas
- Scott Rice is Networked Information Services Librarian at the University of North Carolina, Greensboro
- Christine Noonan received her MLIS from the University of Wisconsin-Milwaukee in 2003 and is currently an Electronic Resources Librarian in the Pacific Northwest
- Martin R. Kalfatovic is the Head, New Media Office and Preservation Services, Smithsonian Institution Libraries

---

### **REVIEW OF: Doug Addison. (2006). *Web Site Cookbook*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN 0-59610-109-0). 261 pp. \$39.99.**

By Jim Blansett

Reading Doug Addison's *Web Site Cookbook* is an interesting experience. It is not a book to be read straight through. Addison states that his book is "intended to be a ready reference for the site builder who needs a quick solution to an immediate problem." This is, indeed, the case. If the reader is expecting a "how-to" book, for the most part, he/she will be disappointed. Mr. Addison strives to provide answers to "real-life problems" that come up while producing and managing a growing web site. Although, certain parts of this book can

be approached by the site builder with a working understanding of HTML (a basic tool that Addison suggests is needed to successfully navigate this book), much of the content demands greater command of overall web site development and open source software.

As you leaf through the Table of Contents, you quickly discern that this book is abundantly supplied with topics to readily focus in on specific problems involving designing, coding, and marketing web sites. Chapters include 1) Web Service Set-up, 2) Site Planning and Setup, 3) Page Design and Navigation, 4) Formatting Text and Code; 5) Formatting Graphics, 6) Displaying and Delivering Information, 7) Interacting with Visitors, 8) Promotion and E-Commerce, and 9) Maintenance and Troubleshooting. These chapters are organized in a Problem -> Solution -> Discussion, format with a "See Also" section that refers the reader to online resources that address each problem. An "Introduction" to each chapter gives a brief overview of its contents. A problem statement directly follows. The problem statement is succinct, easy to read, and clearly descriptive. After the problem statement, the Solution, Discussion, and See Also sections are presented. These last three sections can vary in intelligibility and length, depending on the complexity of the solution. To help clarify, or add dimension to the text previously presented in each section, pages throughout the book are strewn with graphical representations of paw prints that trail into an indented block of text that displays a tip, suggestion, or general note.

As an example of what the reader can expect, in Chapter 3, Page Design and Navigation, Addison briefly introduces the chapter's contents in a short paragraph that explains that he will explore choices that will help to "balance aesthetics with usability." From this point, section 3.1, Choosing Between a Flexible and Fixed Layout, addresses the problem of deciding which design format (flexible or fixed) presents web contents better to specific audiences. Code for a CSS-styled (cascading style sheet-styled) content block with three-columns (fixed), and a flexible layout are given. Addison continues to briefly explain how to combine these options for a hybrid layout (both fixed and flexible). Paw prints lead to an explanation of why widths and margins total less than 100 percent in the preceding code. It is in the "Discussion" section, however, that the solution to the problems begins to take on life. Addison provides additional hints (paw prints), figures that represent examples of fixed and flexible designs, and clearly defined sections that, when combined, add reason to his solution. He explains to the reader under which circumstance flexible and/or fixed layouts shine. Finally, the section ends with "See Also," where the reader is given references to an article and a web site that will provide further information.

Following this first section in Chapter 3, Addison continues the chapter focus, addressing additional problems faced when creating a color scheme, designing web pages for advertisements, expanding your web site, and adding a background. In total, there are nine sections to this chapter, each following the same format described for section 3.1. Many solutions are more complex than described above, and are not nearly as straightforward. The more complex solutions require greater understanding, or, at least, the time to run down a

clearer understanding through reading the suggested article/s or becoming engaged in the web site/s referenced in the "See Also" section. Additionally, while reading the discussion section generally provides a brief overview that, in many cases, seems to elicit an academic interest, this overview will not allow the reader to forego further research. Addison may provide the basic ingredients for his recipes; however, it is up to the chef to go out shopping to become more familiar with each of the ingredients. In this regard, I would not describe this book as a cookbook for a novice. While Addison suggests that the successful user of this book "should have a working knowledge" of HTML code, in my opinion, more is required, sometimes, much more. The *Web Site Cookbook* favors an open-source, or LAMP approach (Linux/Unix server platform, Apache web server, MySQL as database server, and PHP scripting language). Addison readily provides complex coding that works within this format. While he provides references in his "See Also" subheading to supplement his offering, this requires readers to chase after needed information to glean the essence of his recipes. In this regard, it would be helpful to have a good understanding of the LAMP approach, too. That being said, if you have a problem that is addressed in this book, and enough initiative, much can be learned. Addison has ample experience to ascertain the kinds of problems that can be encountered when developing, maintaining, and growing a web site.

Addison's *Web Site Cookbook* is far-sighted enough to include some of the more obscure problems that can be unimaginable to a beginning, or intermediate, web developer. This author offers the reader an opportunity to display foreign and special characters (4.2), add discretionary hyphens to long words (4.5), create a printer-friendly version of their site (6.10); as well as offer their site on mobile devices (6.12). These are only a very few of the problem sections that are addressed. In total, there are 89 problem sections. That is a lot to chew on! Addison states that he has set out to "lead the way" in showing the reader how to publish a web site that is "not only useful and attractive" but also "easy to build, maintain, and update." Easy may be a matter of viewpoint. However, as a proper cookbook should, Addison's *Web Site Cookbook* provides interesting recipes and abundant food for thought. While there are many recipes in this book, as most cooks know, the most useful function of a cookbook is to provide a specific recipe that you need. That is when the cookbook shines. Bon appetit!

*Jim Blansett is Information Resources Librarian in the McLure Education Library of the University of Alabama*

Copyright ? 2006 by Jim Blansett. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at jim.blansett@ua.edu. (mailto:jim.blansett@ua.edu)

**REVIEW OF: Jacob Babbin, et.al. (2006). *Security Log Management: Identifying Patterns in the Chaos*. Rockland, MA: Syngress Publishing, Inc. (ISBN 1-59749-042-3). 350 pp. \$49.95**

By Paul Kittle

Many of us use system or proxy log files for checking database usage around budget times for tasks such as attempting to justify maintaining particular online databases. This work is NOT about that type of system log. This work takes on the now real challenge of defending our public online systems against hostile access.

Jacob Babbin is listed as the "Lead Author" with separate technical and contributing authors that run a wide range of electronic security experience and expertise from private and public sectors. As such, the reader should be prepared for a fairly thorough and "hands-on" work - which this certainly is. If you have little experience with network security, be aware this book requires an intermediate or advance level of knowledge of Unix/Windows O/S and log files.

The work starts out with an overview of the system log files, why examining them regularly is important, what challenges/attacks may be found, and some of the basic tools to utilize in this endeavor. Babbin, et.al., make extensive use of code examples, both as reproduction of system log files and perl/php code, to parse possible and real attacks. *Security Log Management* is wonderfully thorough for those who are ready to advance in the defense of their networks. Each chapter concludes with summaries, quick solutions, and FAQs. The authors also include public domain software (important for those of us with limited or no budgets) that can be utilized to guard the fort. Attacks, such as denial-of-service, firewall breaches, html browser attacks, malware, and more are given great overviews. Examples are presented in detail, however, be certain you have your "geek" glasses on!

Most of this book deals within Unix/Linux servers, but there are chapters dedicated to Microsoft's servers as well. Along with the purchase of the book comes free access to Syngress' website and additional e-booklets germane to security.

Security Log Management is not for everyone. It is intended for those tasked with protecting the network. If you want to stay awake a night - take a hard look through this book - and then be thankful if you already have someone protecting you from the dangers laid out in this work. This book is highly recommended as a purchase for all libraries and network security IT departments.

*Paul Kittle is the Distance Learning/Electronic Reference Librarian, at Mt. San Antonio College in Walnut, California*

Copyright ? 2006 by Paul Kittle. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission

to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at [pkittle@mtsac.edu](mailto:pkittle@mtsac.edu) (<mailto:pkittle@mtsac.edu>).

**REVIEW OF: Dave Crane and Eric Pascarello, with Darren James. (2006). *Ajax in Action*. Greenwich, CT: Manning. (ISBN 1-932394-61-3). 650 pp. \$44.95**

By Mark Cyzyk

The sheer size of this book is daunting. It's been my sense that discussions of the technologies falling under the heading "Ajax" has hitherto been short and sweet, always with the ubiquitous "Drop-Down-Box-of-Zip-Codes-Mapped-to-State-Abbreviations" example at the ready. Not so this book!

But first, what is Ajax?

Ajax is an acronym for "asynchronous Javascript and XML". Simply stated: Ajax is the label we use to refer to that set of technologies embedded in all modern Web browsers whereby client-side, programmatic access via Javascript to server-side data in the form of XML is made available, specifically through the modern browser's support for a Javascript object: XMLHttpRequest. In short: Ajax is both a technology and a method for retrieving server-side data, then displaying it via client-side Javascript and Cascading Stylesheets seamlessly, on a single page, without the sometimes annoying (to the user) "Hold everything while I do something here" page refreshes required of traditional Web application programming.

This book, however, goes so far beyond this that it's difficult to know where to start.

The book is divided into four broad sections: "Rethinking the Web Application"; "Core Techniques"; "Professional Ajax"; and "Ajax by Example". And the chapter headings within these sections provide a good description of the content of the book and of how its narrative proceeds:

A new design for the Web

First steps with Ajax

Introducing order to Ajax

The page as an application

The role of the server

The user experience

Security and Ajax

Performance

Dynamic double combo

Type-ahead suggest

The enhanced Ajax web portal

Live search using XSLT

Building stand-alone applications with Ajax

The book itself, of course, does start with an explanation of what Ajax is, why it's needed, and how an Ajax-based application differs from the architecture of a traditional Web application. It begins with a nice overview of the elements of client-side scripting: Javascript, Cascading Stylesheets (CSS), and the Document Object Model (DOM). In particular, it provides a very nice, understandable explanation of the DOM and of how individual nodes in the DOM can be addressed and thus programmatically manipulated. The combination of these three technologies provide the foundation for purely client-side scripting.

Almost immediately then, in Chapter Three of a thirteen-chapter book, the authors make the bold proposal, which forms the basis for the rest of the book, that traditional Web applications be rearchitected and rewritten so that they execute almost exclusively in the client, making only the occasional call back to the Web server to retrieve data, and this, always behind the scenes. This is, in sum, a description of the new and widely publicized "Web 2.0" rubric. The purpose of this book is therefore not to just introduce Ajax-based techniques for creating some nifty user interface (UI) widgets. The purpose of the book is to propose that the traditional way of architecting Web applications, with the application logic and HTML-based UI generation residing server-side, be replaced with a client-focused, Ajax-based architecture. And so the main thrust of the book begins.

In particular, Chapters Four and Thirteen - "The page as an application" and "Building stand-alone applications with Ajax" - flesh out this new application architecture. Here one finds not only a comprehensive discussion and illustration of how an Ajax-based application as a whole could be written to adhere to the Model-View-Controller (MVC) design pattern, but also how the native browser's event handling mechanism can be massaged and complemented so that it more properly functions within the Observer design pattern. The authors, throughout the book, are all about adherence to design patterns and so-called best practices, so in addition to providing clear instruction on specific Ajax tips and techniques, the book also serves as a sort of primer in software engineering. The second of these two chapters details how data is dealt with in an Ajax environment, including how one might construct an Ajax-based application without the foundational presence of a backend server.

The rest of the book provides examples of Ajax in action including optimizing the user experience using Ajax techniques, securing Ajax-based applications, optimizing performance of Ajax/Javascript code, and detailed illustration and discussion of an Ajax-based double combo box (two HTML select boxes linked, such that a change in the contents of one changes the contents of its linked relation) and a "type-ahead suggest" widget, similar to what is currently found in the widely implemented Google toolbar.

While Ajax is certainly a hot topic these days, with Service Oriented Architecture (SOA), "enterprise mashups", and Ajax as the glue holding it all together being the rage in the trade publications, it is my personal opinion that the jury is still out on whether it is advisable to write entire applications exclusively using Ajax technologies and techniques. However, if doing so is something you decide to do, perhaps to push computationally-intensive tasks out

to otherwise idle client workstations, because server-side middleware is unavailable or unappealing, or maybe even because in the middle of the night you suddenly decided to write an entire Office Productivity Suite to run exclusively within a Web browser (as a few companies are now attempting to do), then this book should prove to be indispensable!

*Mark Cyzyk, formerly Head of Information Technology in the Albert S. Cook Library at Towson University, is currently the Scholarly Communications Specialist in the Library Digital Programs group, The Sheridan Libraries, Johns Hopkins University, Baltimore, Maryland.*

Copyright ? 2006 by Mark Cyzyk. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at [mcyzyk@jhu.edu](mailto:mcyzyk@jhu.edu).

**REVIEW OF: Leander Kahney. (2005). *The Cult of iPod*. San Francisco: No Starch Press. (ISBN: 1-59327-066-6). 160 pp. \$24.95.**

By Holly Mercer

*The Cult of iPod* is a cleverly designed, attractive book with cover art that evokes the iPod shape and display and a layout for the table of contents similar to Apple's iTunes menu. Each chapter shows an iPod screen with the chapter title as the "now playing" selection, and the iPod's battery icon, reproduced on each page, drains as one nears the end of the book. The color illustrations are rich and many, and delightfully illustrate the points made in the text.

An entertaining and informative book, especially for iPod owners and enthusiasts; it reads a bit like a self-congratulatory paean for those already in the iPod know. The author, Leander Kahney, is knowledgeable of Apple and its Mac brand. He is an editor at Wired News, and author of the book *The Cult of Mac* (2006).

*The Cult of iPod* examines the impact of Apple's portable digital musicplayer on society, and it convincingly argues that the iPod has indeed changed popular culture, specifically, the way people listen and interact with recorded music. Digital music players give people access to thousands of songs, and the iPod's shuffle feature can create unexpected pairings and surprising playlists. Rather than listen to an album from beginning to end, iPod owners create their own playlists, which Kahney compares to creating soundtracks for everyday events such as the commute to work or a workout at the gym. The book includes an interview with James Bull, University of Sussex lecturer in media and culture and author of *Sound Moves: iPod Culture and Urban Experience*. Additionally, he talked to hundreds of iPod owners about how and when they listen to their iPods.

Chapter 3, "A Star Is Born: The Making of the iPod" begins with an unofficial history of the iPod, including the concept, design, hardware, and software. The remaining chapters continue the examination into how people use, promote, and accessorize their iPods. iPod

fans have also turned their enthusiasm into successful businesses, such as iPod My Photo. Customers can upload digital photos to the business and receive back an "iPodified" photo using the familiar white iPod and white earbuds and silhouette forms of Apple's iPod advertising campaign. The chapter then describes the love affair some owners have with their iPods, and the lengths to which they go to accessorize them. For example, the chapter reports on designer Karl Lagerfeld's collection of seventy iPods and the suitcase he designed to carry them. Lagerfeld designed a similar case, the Juke Box, for Fendi; it retails for \$1,500. Cottage industries have grown up around the iPod. Dance clubs sponsor playlist parties, where people can sign up to be DJ for a portion of the evening, and then plug their digital music players into the clubs' sound systems.

Kahney reports in chapter six, "Inspiring the Imagination," that Apple is not interested in manufacturing iPods with video or personal digital assistant (PDA) features. Yet one year after the book's publication, Apple's web site features "The New iPod," capable of storing 150 hours of video [1]. Like any publication that reports on new technology devices, some of the information is out-of-date by the time it makes it to print.

The book provides good coverage of the impact of the iPod on popular culture and defines the new vocabulary associated with the iPod: iPod jacking, playlistism, playlist parties, and mobile dance parties are all concepts associated with the rise in popularity of the iPod. Much of the book is devoted to the social aspects of technology, exploring how ownership of an iPod can actually support human interaction. *The Cult of the iPod* focuses on the positive impact of the iPod on society, and is an enlightening and fun look at the growth in the popularity of the iPod.

This book is recommended for public libraries and universities with programs in popular culture.

#### References

[1] "Apple - iPod." Retrieved April 27, 2006 from <http://www.apple.com/ipod/ipod.html> (<http://www.apple.com/ipod/ipod.html>).

*Holly Mercer is the Coordinator of Digital Content Development at the University of Kansas.*

Copyright ? 2006 by Holly Mercer. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at [hmercerc@KU.EDU](mailto:hmercerc@KU.EDU) (<mailto:hmercerc@ku.edu>).

**REVIEW OF: Bruce W. Perry. (2006). *Ajax Hacks: Tips & Tools for Creating Responsive Web Sites*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN: 0-596-10169-4). 414 pp. \$29.99.**

By Scott Rice

*Ajax Hacks* is one of many popular books in the "Hacks" series. Written by Bruce Perry, a software developer and author of a book on Java and JSP, it features a number of 'hacks' or small projects that can be accomplished with Ajax. For those unfamiliar with Ajax, it stands for Asynchronous Javascript And XML, and this technology, (or rather, set of techniques) stands behind many of the most popular web applications today, such as Google Maps. Ajax uses a little known feature of browsers, the XMLHttpRequest object, in order to perform its work. The object allows a browser to query the server and get a response, all without refreshing the current web page. Ajax is asynchronous because this query and response take place entirely behind the scenes. Thus a user is able to stay engaged with a web application that doesn't need to refresh itself, go to a new page, submit a form and wait for a response from the server, all those processes that make many web applications time-consuming and cumbersome to use.

In Chapter 1, "Ajax Basics" Perry explains the nuts and bolts of the components of an Ajax transaction. This chapter alone is enough to get an individual started with some very basic Ajax functionality in their web pages. Some of the hacks listed include detecting browser capability, posting data, receiving data, handling request errors, and generating user messages. Chapter 2, "Web Forms" applies the set of Ajax techniques to streamlining form usage for the user. In ordinary web applications, the loop from form to server to updating the form generally involves reloading the page or moving to a new web page. Using Ajax, a form can be created that can post data and refresh data without making that trip to the server. The convenience and time saved with using Ajax in this manner with web forms is one of the main contributors to its current popularity. Validation is the topic of Chapter 3, and here Perry covers a number of types of validations that can be done using Ajax. These include validating e-mail addresses, usernames, credit card numbers and security codes, and postal codes. Chapter 4 is entitled "Power Hacks for Web Developers" and has some of the most interesting applications in the book. Here, Perry shows off some of the possibilities of Ajax web applications, including an eclectic assortment of different hacks. There are a number of hacks that involve interactions with other web applications such as Google Maps, Yahoo! Maps, or a Weather.com XML feed. In addition, there are hacks detailing how to use Ajax to create large bookmarklets, how to control browser history, and how to create an RSS feed reader.

Where Chapters 1-3 provide scripts that are generally small (though very useful) parts of a larger application, Chapter 4 provides scripts with much more functionality that might be the main focus of a web application. Chapter 5 covers Direct Web Remoting, or DWR, which involves using Java in conjunction with Ajax to create even more interesting and robust web applications. The hacks in this section, as with a later chapter involving Ruby on Rails, are not as much use to the person looking to learn more Ajax tricks, because these chapters involve knowledge of other languages that the reader may not be familiar with. Chapter 6, "Hack Ajax with the Prototype and Rico Libraries" details how to use these two libraries to extend the capabilities of Ajax and also make programming easier. One of the most

interesting hacks in this section is the last one, which explains how to create an online bookstore with drag-and-drop capabilities. In Chapter 7, Perry again shows how to use Ajax in conjunction with another programming language, in this case, Ruby on Rails. Ruby on Rails is becoming a popular programming language because of its ease-of-use and robustness. The hacks in this section involve getting setup with Ruby, getting Rails working, getting Javascript working with Rails, and finally, a few small projects with everything working together. The script.aculo.us Javascript library is built on Prototype (which was introduced in Chapter 6, along with the Rico library). Chapter 8 introduces this library and includes a few hacks to do some interesting effects, like a login box that shakes if an incorrect login is used, or an auto-complete form field. Chapter 9, "Options and Efficiencies" is for those who are very comfortable with programming, or already have an application that they are working on and are seeking to optimize it. A number of the hacks here are very useful for taking an existing web application and making it operate more efficiently. This book, like many in the Hacks series, is not for the casual user or beginner.

While the explanations are clear and render much of the difficulties of programming in Ajax less confusing, it would still be very difficult for someone not already familiar with programming to get much use out of the book. Also, *Ajax Hacks* is not a book most users will read straight through from start to finish. It is much more useful (and fun) to pick and choose among the projects those that seem interesting, or easy to do, or related to other projects you may be working on. The setup of the book makes this possible to a limited extent, as projects within chapters do build somewhat on previous skills and information learned in that chapter. However, moving between chapters (and sets of hacks), will generally introduce the reader to a new range of skills. The book does include a table of contents that lists each hack, which is invaluable in navigating to the projects the reader is interested in. An index is also helpfully provided, which is an amenity not often found in programming books. These finding aids are essential in a book of this sort, where the vast majority of users will want to take parts of scripts from many different places to assemble in to their own web applications. The only questionable part of the book lies in the inclusion of the chapters on Java and Ruby on Rails, which detract somewhat from the focus of the book. Many users who pick up this book are not also interested in learning more about Java and Ruby on Rails, but want to know as many tips and techniques as they can get about Ajax. However, for the most part, this is a very well written and informative book and the hacks chosen to implement are generally very useful and can be integrated into an application quite easily. This is a book that should definitely be included in any well-rounded computer science collection.

*Scott Rice is Networked Information Services Librarian at the University of North Carolina, Greensboro.*

Copyright © 2006 by Scott Rice. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission

to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at serice2@UNCG.EDU (mailto:serice2@UNCG.EDU).

**REVIEW OF: Chris Pine. (2005). *Learn to Program*. Raleigh, NC: The Pragmatic Programmers, LLC. (ISBN: 0-9766940-4-2). 149 pp. \$19.95.**

By Scott Rice

*Learn to Program* by Chris Pine is part of The Facets of Ruby Series, published by The Pragmatic Programmers. Chris Pine, a game programmer living in Norway, developed this book as a result of his evolving online tutorial about learning to program using the Ruby language. He added three chapters to the online material [1] and modified the other content to create the book. Pine's writing style is very casual, but engaging, reflecting the book's origins as an online tutorial. With an emphasis on humor and the inclusion of emoticons is a dead giveaway that the writer is used to writing on the Internet!

Pine takes very little for granted in his approach to the subject, and starts with the basics. He starts with explaining how Ruby treats text, numbers, and variables and builds from there to talk about methods, arrays, iterators, and flow control. Each topic builds on the previous information in a very logical progression. With each chapter, Pine includes several code examples and provides sample programs illustrating key concepts. In addition, he provides multiple exercises to give the reader practice in trying out new skills and concepts.

In Chapter 1, "Getting Started," Pine explains how to install Ruby for the Windows, Mac, and Linux operating systems. Here, screenshots and explanations of the basics of navigating in each operating system make it possible for the greenest of beginners to install the language. He also offers recommendations on the text editing program that should be used to create programs in Ruby. In chapters 2-4, Pine explains about how Ruby treats numbers and text. He covers how Ruby differentiates the two, and a few simple operations that can be performed on text and numbers. He also includes a discussion of variables and how values are assigned to them.

Chapter 5, "Mixing It Up" talks about methods, which are the ways in which the programming language can manipulate numbers and text, in other words, get the actual computing accomplished. Chapter 6 has more information about methods and also explains the use of several methods, such as random number generation and determining the length of a string. Chapter 7, on flow control, begins a discussion about if/then statements, while loops, and a brief introduction to Boolean logic. With these concepts, we are able to create more complex programs and the programming examples become more complicated to take advantage of these new capabilities. The next chapter, about arrays and iterators continues the introduction of more complex concepts.

Pine returns to methods in Chapter 9 and shows how to write your own methods. He also covers the concept of return values and local variables. The discussion of local variables needed to go into further detail, as there is no mention of their opposite, global variables. It is also not made clear when or why you would use a local over a global variable. In Chapter 10, entitled "There's Nothing New to Learn in Chapter 10" Pine takes the concepts and skills already introduced and shows how a program can be made more powerful by implementing recursion. Recursion is a very powerful technique, but a beginning programmer would need a little more explanation to fully utilize it.

Chapter 11 gives a very quick overview of how to read, write, and save files. In this chapter Pike introduces YAML, a string representation format. His explanation of YAML is cursory and he quickly drops the subject, not referring to it again. This leaves the reader wondering how much is being left out of the explanation and how important it may be to know YAML. In Chapters 12 and 13, Pine covers classes of objects and explains how to create new classes and modify existing classes. Chapter 14 covers procs which are blocks of code that can be treated as objects. Finally, in Chapter 15, Pine talks about some places you can go to get more information about Ruby, including a book, a program and a mailing list. Here, as in other places, you get a sense that you are missing a lot of depth and detail not only in Ruby, but in programming itself. Pine refers to "unless" and "yield" statements in passing and talks about how he introduced only one of six methods of creating a string.

This reflects one of the major drawbacks of the book; it can't seem to decide whether it is a book about programming or a book about Ruby. In many places in the text, Ruby takes a backseat to basic programming concepts. Pine could be writing about any programming language. In other places, however, such as the quick side trip into YAML, Pine goes into unneeded (and slightly confusing) detail. The book would have also benefited from the inclusion of many more resources for further reading, whether print or online, where the reader could develop his skills and learn more about programming and Ruby. Although I think the book mostly succeeds on its own terms, providing a basic introduction to programming concepts using the Ruby language, there are a number of other drawbacks. Each chapter is very short, and does not go far beyond the cursory in its explanation of concepts. There is also perhaps too much code included in the book. A great deal of space is devoted to repetitive blocks of code that do not add anything to the explanations. The book suffers from the lack of an index, which would certainly help someone new to programming to refer back to key points in the text. There is a somewhat detailed table of contents, but it is not always clear from the subheadings what is covered in each section.

All in all, the book is a good introduction, but further reading is necessary to get a more well-rounded understanding of programming or using Ruby. On the plus side, you definitely get a sense of the enthusiasm Pine brings to the subject and this may provide encouragement to beginners to experiment and attempt to learn more. The book goes a long way to making the

subject unthreatening which is great for beginners. More advanced readers may want to start elsewhere, or seek a book that goes farther into the nuts and bolts of either programming or Ruby.

[1] The early version of the book can still be found online at <http://pine.fm/LearnToProgram> (<http://pine.fm/LearnToProgram>) (accessed 18 November 2006)

*Scott Rice is Networked Information Services Librarian at the University of North Carolina Greensboro*

Copyright ? 2006 by Scott Rice. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at [serice2@UNCG.EDU](mailto:serice2@UNCG.EDU) (<mailto:serice2@UNCG.EDU>).

**REVIEW OF: Eric Pulier and Hugh Taylor. (2006). *Understanding Enterprise SOA*. Greenwich, Conn.: Manning Publications. (ISBN: 1-932394-59-1). 242 pp. \$39.95.**

By Christine Noonan

For those new to web services and service oriented architecture (SOA), this is a great starting point. That said, the book is written primarily for business professionals who need to understand how web services and SOA can effectively be implemented in a business setting. Using a case study of an insurance company pulling through the "IT aftermath" of a troubled merger, Pulier and Taylor make a compelling case for any medium to large sized company which may be toying with the ideas presented.

Organizationally the book is flawless. Each chapter naturally builds upon the previous and is written in such a manner that anyone with a cursory knowledge of networking and database architecture will be able to follow along. In other words, the concepts presented coupled with the case study and graphical representations produce a highly readable and usable text. There are two main sections to the book. Part one, consisting of chapters one through twelve, lays the foundation for the reader to understand the technology behind web and SOA. Part two, chapters thirteen through eighteen, explores the people and the business processes involved in implementing SOA.

So what the heck are web services anyway and why would you want to use them? Services are essentially software components that are capable of communicating with one another over multiple networks using open standards based on XML and three interrelated XML-based software standards: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description Discovery and Integration). Web services make it a cinch to open communication between systems regardless of the

hardware platform or the programming language. In this sense there is a natural discovery process whereby systems can investigate each other's capabilities without human intervention. This kind of environment is known as SOA.

The main purpose of implementing web services is to enable organizations to leverage their pre-existing, and in some cases legacy systems. The data contained within them needs to be exposed to other applications for repurposing. And, as Pulier and Taylor emphasize, this is a cost-effective approach which results in a positive return on investment; there is a reduced reliance on proprietary and custom interfaces and a model of "vendor neutrality" can be introduced. Now that said, web services are not without limitations. They can expose legacy systems but do not replace them, which depending upon a business need, may not be the best solution for a given situation. Services are also inherently insecure as they are merely a set of standards and not comprehensive software packages so they require you to either buy or develop a security solution. Performance is also slower which makes them unsuitable for certain tasks. These issues aside, the beauty of SOA is that it is a flexible and adaptive approach to enterprise architecture and allows an organization to focus on their business processes without regard for the underlying IT systems. Each element can be moved around, replaced, and modified. In other words, it is a dynamic solution.

Technology aside, what about the people and business processes? Any major change requires good leadership. And, to achieve consensus, key stakeholders at all levels of the organization need to be kept informed and clued in to every vital step in the implementation process. Four "best practices" or 4-Ps are identified by the authors to accomplish this - people, pilot, plan, and proceed.

In order for web services to be a success, all affected staff need to learn about XML, web services and the principles of an SOA. They should be encouraged to work together; continuous training and learning can help facilitate this as can the creation of smaller groups of individuals from different functional areas in the organization. The most important aspect of the "people" best practice is establishing a governing body for the SOA development process. This group can serve as a steering committee for the development effort and can guide the discussion of any issues which may arise in the implementation process. To train people and to demonstrate the idea of web services, a small pilot project should next be undertaken and executed. Working on one piece of the larger puzzle will enable staff to figure out how to handle the unexpected and learn about how different options and alternative use scenarios can be carried out.

After training and getting a little bit of exposure to web services by developing a pilot project, the planning stage can commence. As with any project plan, a migration path should be drawn up, budget and staffing needs should be identified and an implementation schedule drafted. This is the most flexible component of implementing web services using SOA. A

project plan should in essence become a "living document," one that can be modified depending upon funding cycles, staff changes, and organizational requirements. Finally, you can proceed and begin working on development and implementation of SOA.

Plain and simple - web services and SOA offer a promise of increased efficiency and improved IT management. For libraries, this means that as we grow more partnerships and develop cooperative projects with our founding institutions, whether they are the local city government, the University, or the corporation, we will be better suited to offer more integrated services bridging multiple platforms and technological infrastructures. This book is a must for anyone looking for a general and practical introduction to the business processes behind SOA.

*Christine Noonan received her MLIS from the University of Wisconsin-Milwaukee in 2003 and is currently an Electronic Resources Librarian in the Pacific Northwest.*

Copyright ? 2006 by Christine Noonan. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at Christine.Noonan@PNL.GOV (mailto:Christine.Noonan@PNL.GOV).

**REVIEW OF: Rich Shupe and Robert Hoekman, Jr. (2006). *Flash 8: Projects for Learning Animation and Interactivity*. Sebastopol, CA: O'Reilly Media, Inc. (ISBN: 0-596-10169-4). 340 pp. \$34.99.**

By Martin R. Kalfatovic

Flash long ago, at least in web terms, surpassed clunkier methods of providing animation and interactivity to websites. Javascript, animated GIFs, and other related kludges are now easily replaced with simple Flash movies that even the novice can generate after going through the Flash tutorial. Without going into the particulars of the how's and why's web designers should consider when contemplating a web design project that will integrate Flash into the overall architecture of the site, creators of websites should assume that their users will expect more from a site than simple black text on a white screen. Even with the important mandate to create accessible sites, web designers should not allow accessibility become an excuse for poor architecture and even poorer graphic design, this is especially true with the capabilities that Flash now offers additional features for users with visual, aural or mobility impairments.

For Flash first timers, Shupe and Hoekman begin with the basics: drawing, coloring, importing graphics, simple animations, etc. Users should, however, bring certain skill sets with them before opening the book. Basic HTML (or related web-design coding) and graphics tools (PhotoShop, Illustrator, Freehand, etc.) will make for a shallow Flash learning curve.

A well-designed Flash movie can bring more than just, er, "flashy-ness" to a website. As Shupe and Hoekman describe and demonstrate, a well-done Flash animation can create a useful and productive interactive experience for the user. Flash allows designers to move beyond the simple animations of a logo or a method simpler than Javascript to create rollover buttons. The authors include a number of projects and examples that show the novice Flash designer how to create more complicated movies that incorporate increased levels of user interactivity, integrating dynamic content within Flash animations, and similar activities. Additionally, recognizing the importance of mobile computing devices, the authors include a section on Flash design for handheld devices (and CDROMs).

Flash 8: Projects for Learning Animation and Creativity is not designed as resource for the development of ActionScript (the Flash programming language). A number of excellent ActionScript books are available that will supplement Flash 8: Projects for Learning Animation and Creativity [1]. Neither will it serve as a replacement for instruction in overall interactive control design. It will, however, provide beginning and intermediate Flash writers increased knowledge and tools for their work.

Though the Shupe and Hoekman are too kind - or fearful of losing a book sale - to point it out, it should be noted that knowledge of what keys to press and which clicks to make, will not automagically create a killer Flash application. Knowledge of site architecture design, graphic skills and the other wild cards of web design are still the inspirational 1% factor that goes into a great website. Until Adobe (which recently purchased the Macromedia suite of web/graphic tools) can package a "Creativity Module" the majority of us web hackers will be forced to rely on those with access to the genie of inspiration to supplement our 99% knowledge of software use.

Do the authors offer anything for the library web designer? Do library websites need the type of gee-whiz interactive Flash portals that one sees for contemporary culture sites such as (picking a site off my Firefox recently visited list) such as Beck.com? That depends on your library and who you want to visit your site.

The authors follow the tried and true methodology of O'Reilly books. Sidebar information, "learning by doing" examples, grouping material in related digestible parts, and an easy to read, clear page layout, and other design tools familiar to O'Reilly readers will make for comfortable reading. The table of contents and index are well-done and allow for quick browsing of topics. The book includes a CD-ROM containing a trial version of the Flash 8 software, all lessons and examples. Sample files and related material is also available for downloading from the book's support website. Recommended for novice and intermediate Flash developers.

[1]. See for instance *ActionScript 3.0 Cookbook* by Joey Lott (O'Reilly Media, 2006) or *Flash 8 ActionScript Bible* by Joey Lott and Robert Reinhardt (Wiley, 2006).

*Martin R. Kalfatovic is the Head, New Media Office and Preservation Services, Smithsonian Institution Libraries.*

Copyright ? 2006 by Martin R. Kalfatovic. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at [kalfatovicm@si.edu](mailto:kalfatovicm@si.edu).

[table of contents](/ala/lita/litapublications/ter/volume10no2.htm#toc) (/ala/lita/litapublications/ter/volume10no2.htm#toc)

[ter issues](/ala/lita/litapublications/ter/tertechnology.htm) (/ala/lita/litapublications/ter/tertechnology.htm)

---

## **About TER**

Editor is Martin R. Kalfatovic, Smithsonian Institution Libraries ( [kalfatovicm@si.edu](mailto:kalfatovicm@si.edu)).

Editorial Board:

- Frank Cervone, Northwestern University ( [f-cervone@northwestern.edu](mailto:f-cervone@northwestern.edu) (<mailto:f-cervone@northwestern.edu>))
  - Tierney Morse McGill, Colorado State University ( [tmcgill@manta.colostate.edu](mailto:tmcgill@manta.colostate.edu) (<mailto:tmcgill@manta.colostate.edu>))
  - Brian L. Pytlik Zillig, University of Nebraska-Lincoln Libraries ( [bpytlikz@unlnotes.unl.edu](mailto:bpytlikz@unlnotes.unl.edu))
  - Terry Paul Reese, Jr, Oregon State University ( [terry.reese@oregonstate.edu](mailto:terry.reese@oregonstate.edu) (<mailto:terry.reese@oregonstate.edu>))
  - Laura Wrubel, University of Maryland ( [lwrubel@umd.edu](mailto:lwrubel@umd.edu) (<mailto:lwrubel@umd.edu>))
  - Michael Yunkin, University of Nevada, Las Vegas ( [michael.yunkin@ccmail.nevada.edu](mailto:michael.yunkin@ccmail.nevada.edu) (<mailto:michael.yunkin@ccmail.nevada.edu>))
-