the bookworm

by Peter H. Salus
<peter@pedant.com>

Peter H. Salus is a member of the ACM, the Early English Text Society, and the Trollope Society, and is a life member of the American Oriental Society. He is Editorial Director at Matrix.Net. He owns neither a dog nor a cat.

BOOKS REVIEWED IN THIS COLUMN

TRUST AND RISK IN INTERNET COMMERCE
L. Jean Camp
Cambridge, MA: MIT Press, 2000. Pp. 279. ISBN 0-262-03271-6.

EFFECTIVE TCP/IP PROGRAMMING
Jon C. Snader
Boston, MA: Addison-Wesley, 2000. Pp. 299. ISBN 0-201-61589-4.

ESSENTIALS OF THE JAVA PROGRAMMING LANGUAGE
Monica Pawlan
Boston, MA: Addison-Wesley, 2000. Pp. 301. ISBN 0-201-70720-9.

JAVA SERVER & SERVLETS
Peter Rossbach & Hendrik Schreiber
Reading, MA: Addison-Wesley, 2000. Pp. 429. ISBN 0-201-67491-2.

JAVASERVER PAGES
Larne Pekowsky
Reading, MA: Addison-Wesley, 2000. Pp. 282 + CD-ROM. ISBN 0-201-70421-8.

THE HUMANE INTERFACE: NEW DIRECTIONS FOR DESIGNING INTERACTIVE SYSTEMS
Jeff Raskin
Reading, MA: Addison-Wesley, 2000. Pp. 256. ISBN 0-201-137937-6.

Every time I open a newspaper or click on the TV, some pundit is telling me about eBusiness or eCommerce and its importance. Most of what I read and hear is absolutely worthless. (This may indeed be true of most of what the media purvey, but I just don't know enough about most things to be able to tell. Unfortunately, my guess is that the pundits know far less than a random cabbie in a European or North American city.)

To be frank, I've only got a half-dozen keepers on electronic commerce over the past few years.

So it's really nice to get another.

L. Jean Camp's view of commerce in general is that every transaction involves assumptions of trust and risk. These, in turn, relate to security, privacy, and reliability. Years ago, I would tear up the carbons of my charge slips, so that phony cards couldn't be created from them and the imprint of my signature. Card companies' elimination of those carbons made me feel better about my security and my privacy.

In any Internet transaction, the questions of who trusts whom and what the risks are arise. Who pays when trust is misplaced (both in terms of hard cash — virtual cash? — and in terms of data)? When there are failures, who is at risk? When a third party is involved, who is liable?

Lots of questions. Jean Camp may not have all the answers, but what she's done is extremely valuable in terms of putting the questions and in delivering a jargon-free presentation of the issues.

I use eCommerce a lot: I buy things, I sell my writings, and I deliver "product" via the Internet. As time goes on, more and more of us will transact more and more business via the Internet. This book does a fine job in presenting the risks and also the security features that go to ensure our trust.

TCP/IP

By and large, I admit to relying on the (many) tomes of Doug Comer and of the late Rich Stevens when it comes to understanding TCP/IP. So I'm generally wary when yet another book on TCP/IP thuds onto my desk.

But Snader's little book is good and useful.

He used Stevens's `groff` macros, so the look-and-feel is a familiar one. There are a lot of good tips and some neat code, but the last four "tips" may easily be the most important ones:

- 41. Read Stevens
- 42. Read Code
- 43. Visit the RFC Editor's Page
- 44. Frequent the News Groups

Yep.

So, if you do networking, buy Snader, too.

Where IPv6 is concerned, there are now seven volumes of RFCs edited by Pete Loshin. In June, I mentioned this "Big Book . . ." series. Once again, my compliments to him and to Morgan Kaufmann for these.

Too Much Java Keeps You Awake

It's nearly four years since I first complained about the number of Java books I had received. Well, I need to admit that there are a few new ones that I found worthwhile.

Monica Pawlan's volume is perfect for someone who's had a first course in programming — a real beginner may have some difficulties. But Pawlan has done a fine job with her explanations, and many chapters end with sections on "further information." There is a brief but good bibliography.

Rossbach and Schreiber came out last year in German; no translator is noted, but whoever did it deserves a pat on the back. If you want to build portable Web apps, you'll want this. The bibliography is thorough, but Rich Stevens is under "R," not "S."

JavaServer Pages is a new technology that is intended to facilitate page development. JSP is included in Sun's Java 2 Enterprise Edition. The book seems good, though the compulsory history sections (pp. 2—6) are fairly vapid. The brief chapter summaries are quite handy.

Some Apologies

A while back I complained that there was no book on `grep`. Several readers pointed out that Friedl & Oram, *Mastering Regular Expressions* (O'Reilly, 1997) has a lot on both `grep` and `egrep`. That's true, and Friedl and Oram have a fine book — but it goes into Perl, Python, Emacs, etc., as well as `grep`.

Second, I wrote that Ted Dolotta created the `-mm` macros. Ted wrote me:

> What you say there is not quite correct: it is true that I decided to have the `-mm` package written (there was some feeling at the time that Mike Lesk's `-me` macros could be improved/enlarged upon). But most of the design and implementation was done by John Mashey and Dale Smith. I provided the overall management, critiqued the design and the documentation (surprise!), and served as alpha tester. I also designed and implemented the whole footnote mechanism — a very complicated piece of `troff` code which was subsequently broken and never repaired when someone (who did not understand it) tried to add a feature to automatically print legal notices at the bottom of every page below the footnotes; turnover footnotes have never worked since.

I do try to get things right . . .

Jef Raskin
The Humane Interface

Reviewed by Steve Johnson

Jef Raskin is best known as the designer of the Macintosh interface. In *The Humane Interface*, he has written a very entertaining and, in the best sense, radical book about how we deal with computers and applications.

Probably half the book is spent putting together a model that describes how people interact with programs. Fairly high-level concepts ("you can only focus on one thing at a time") were mixed artfully with much more mundane ideas (such as a model that estimates how long the average user will take to, for example, click in a dialog box and type 10 characters of text). I found myself saying "of course" over and over again, as he pointed out obvious things that I'd just never brought into consciousness.

As one example, Fitt's law predicts how long it will take to move the cursor and click a button, based on the size of the button and how far you need to move the cursor. Halving the dimensions of a button can add nearly 150 msec. to the time it takes to press it (and this doesn't count the increased error rate). I recently "upgraded" a program I use frequently, and found the upgrade significantly harder to use. After reading this book, I now see that the buttons' being smaller is a lot of the reason.

He also has a discussion of information theory, and the uses and abuses of it in interface design. His goal for an interface is that it become monotonous — like a perfect waiter, it serves you without calling attention to itself. He loathes error boxes where the only thing you can do is say "OK." His critiques of the Windows interface are bullseye accurate — it becomes hard to use any Windows application after reading this book, because you keep being aware of how much harder you are working than you need to.

Finally, he discusses a number of radical ideas for interfaces. One is based on a two-dimensional infinite field where data can be piled, and you can zoom in and out to find the information you need. Another is a truly radical notion of eliminating file names from the user interface. As a long-time UNIX user, this idea almost sent me into shock, but it's good to have your world view upset every couple of decades!

Finally, the book is very engaging to read. It is well laid out, written in a light, somewhat ironic style, and peppered with amusing quotes. It would be an excellent read for anyone involved in generating user interfaces (and who isn't?).