NeXTSTEP: No Longer the Difference Between Black and White

A review of a comprehensive introduction to the NeXTSTEP programming environment.

Michael J. Mezzino, Jr.

NeXTSTEP Programming Step One: Object-Oriented Applications, by Simson L. Garfinkel and Michael K. Mahoney. TELOS/Springer-Verlag, 1993.

IF YOU KNOW C AND WANT TO WRITE sophisticated application software for computers running NeXTSTEP—including custom interfaces for *Mathematica*—then Garfinkel and Mahoney's *NeXTSTEP Programming* should be in your library. NeXTSTEP is the object-oriented environment that runs on NeXT and 80486-based computers. A radical departure from conventional programming environments, NeXTSTEP uses object-oriented programming concepts throughout its structure. Writing for it is, therefore, fundamentally different from writing for other computers. Although Garfinkel and Mahoney don't attempt to teach object-oriented programming, the programming examples around which the book is developed serve as an excellent overview of object-oriented design and programming in Objective-C (the object-oriented extension to C used by virtually all NeXTSTEP programmers). *NeXTSTEP Programming* covers NeXTSTEP 3.0, including new development tools; since version 3.1 promises only minor improvements and bug fixes, this book should be current for a while.

NeXTSTEP Programming is so thorough that it can be used as a reference book. It is also a good resource for the experienced NeXTSTEP programmer, containing brief but helpful discussions on debugging with *gdb*, *lex*, *yacc*, PostScript, Microsoft's Rich Text Format (RTF), and other utilities. Although I consider myself an experienced NeXTSTEP programmer, I learned several tricks reading through the book. Included with the book is a floppy disk with source code disk for the programming examples, which lets you copy and paste objects directly into an application currently under development. A tutorial on building graphical user interfaces with Interface Builder is also included on the disk. This is the first book published by TELOS, Springer-Verlag's "The Electronic Library of Science."

The book is divided into five major parts comprising 21 chapters. Each chapter typically covers two concepts, which are developed into code and used to enhance the current application. The book's infrastructure, in contrast, consists of three applications to develop as you read through the text. These applications, called Calculator, MathPaper, and GraphPaper, weave the chapters together, giving a coherent view of the complete development environment. They provide a carefully considered set of tasks for which various objects are designed, built, and integrated into an application. The real virtue here is that although the final applications may not be particularly useful to readers, many of the objects to construct along the way will likely be helpful in real situations. The excellent design and implementation of the three applications make this book a refreshing departure from the usual learn-by-example literature.

The first section of the book, Part 1, is a general introduction to NeXTSTEP, NeXTSTEP programming, and the Objective-C language, beginning with an introduction to the NeXTSTEP Graphical User Interface. A discussion of the principal development tools follows. Before completing this section, you will create one simple application with Interface Builder and another, from the ground up, in the native UNIX environment. The latter exercise is very important because it removes the mystery behind NeXT's powerful new objects, instilling a deeper understanding and higher level of confidence during the early stages of work. Later, when you drag off a Window or Menu item from Interface Builder's palette of graphical objects, you understand what Interface Builder has produced for you.

In Part 2, you will build Calculator, the simplest of the three major applications. Using Interface Builder, you will design a user interface based on graphical objects obtained from a palette, and subclass Object to serve as the evaluator or calculator engine. By the end of Chapter 5, you have a working four-function calculator. You are then shown how to enhance the interface with icons and additional windows and after a discussion about delegation, you extend the calculator to handle multi-radix input. Finally, a [illegible]

Michael J. Mezzino, Jr. received his Ph.D. in mathematics from the University of Texas at Austin in 1969. He is the outgoing chair of the department of mathematics at the University of Houston, Clear Lake, where he is now spending more time with Mathematica and with his friends on the golf course.

ACM SIGPLAN NOTICES
Vol. 28 Number 11 November 1993

Book Review

NeXTSTEP Programming: Step One, Object-Oriented Applications, by Simon L. Garfinkel and Michael K. Mahoney. Springer-Verlag, New York, 1993, ISBN 0-387-97884-4.

NeXTSTEP is an object-oriented programming environment supporting rapid prototyping and development of information systems, especially the graphical user interfaces (GUIs) of these systems. The authors take a problem-solving approach in introducing the reader to NeXTSTEP and the Objective-C object-oriented programming language. That is, in addition to step-by-step instructions on NeXTSTEP-based application development, topics such as program design, organization, and style are thoroughly covered.

The subject matter revolves around four examples illustrating NeXTSTEP environment concepts and development. The only pre-requisite to being able to follow the material in the book is a familiarity with GUIs, high-level programming languages, and object-oriented design methodologies. The authors provide a cursory but sufficient introduction to related topics such as the Mach operating system, Unix system tools such as lex and yacc, and so on such that the reader should have little or no trouble understanding and working through the example applications. The complete source code listings for the example applications appear in the appendices, as well as on an accompanying diskette.

After reading the book as well as working through three of the four detailed applications, the reviewer was able to build a GUI for one of his own application programs using the NeXTSTEP environment. The reviewer had no previous experience designing and implementing applications within the NeXTSTEP environment.

This is a well-written and illustrated book, and serves both as a primer for novice NeXTSTEP users as well as a handy reference for already initiated NeXTSTEP application developers. Although the subject matter is for the most part self-contained, the book is not intended to be a replacement for but rather a companion to NeXT's NeXTSTEP product documentation.

J. Bret Michael
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439-4832
michael@eid.anl.gov

BOOK REVIEWS ON NeXTSTEP PROGRAMMING AND NEURAL NETWORKS

Department Editors: A. John Mallinckrodt
ajmallinckro@csupomona.edu
Susan McKay
rps352@maine.bitnet

NeXTSTEP Programming, Step One: Object-Oriented Applications

Simson L. Garfinkel and Michael K. Mahoney

Springer-Verlag, New York, NY, 1993;
ISBN 0-387-97884-4, 3-540-97884-4, 631 pp. plus 3.5" MS-DOS diskette, hardcover, $39.95.

Reviewed by Richard R. Silbar

NeXT Computer Inc. began by selling innovative "personal workstations" with a very innovative (but proprietary) operating system and programming environment, now called NeXTSTEP. As the years went by, the company's hardware line became more conventional (and better) but less and less price-competitive. Meanwhile, the object-oriented NeXTSTEP graphical user interface (GUI) grew more powerful and attractive. In February 1993, NeXT decided to drop its hardware line altogether and concentrate on developing NeXTSTEP for a wide variety of computer platforms, and not just computers made by NeXT.

This broadening of the customer base begins with a potentially huge new market for NeXTSTEP: the market represented by high-end PCs based on Intel chips. NeXTSTEP 486 is due to begin shipping in May 1993, and NeXTSTEP Pentium will follow soon. By the time this review appears, we will probably have also heard of several agreements between NeXT and hardware manufacturers to port NeXTSTEP to other platforms such as workstations. NeXT obviously hopes that this move to market NeXTSTEP more vigorously will transform it into an industry standard for an object-oriented GUI long before Microsoft's Cairo and Apple/IBM's Taligent become available (in 1995 or so).

Just as NeXT decided to become a software company ("go virtual"), the first really good book on NeXTSTEP programming came into print. Simson Garfinkel and Mike Mahoney, working at opposite ends of the continent (but well connected by the Internet), have written a very nice "how to" book on writing NeXTSTEP applications. The authors proceed in a sensible way, namely, by working out nontrivial example programs and explaining what is happening as they go along.

Perhaps that sounds like a book for a software developer who wants to create (and sell) a "killer" CAD or desktop-publishing program. Why should an ordinary physicist who computes be interested in this book? After all, getting into NeXTSTEP programming may mean having to learn not only an object-oriented programming language, Objective-C, but perhaps even C and some Unix. What does NeXTSTEP bring to the table that Fortran 77 doesn't already serve well?

The answer is, in my opinion, the ease with which someone who does not have the background of a computer-science major can create, using NeXTSTEP, scientific computing applications that are much enhanced by a standard, graphical, user-friendly interface. I have written such applications myself (using, in fact, a lot of Fortran), and it is easy to do, once you get the hang of it. However, it would have been much easier to "get that hang," if this book had been available to me then. Until now, getting going in NeXTSTEP programming usually involved attending the "NeXT Developer's Camp," an intensive (and expensive) one-week course at NeXT headquarters. One could also read the on-line documentation (eight volumes) that comes with NeXTSTEP. This is fine for looking up the syntax of a function call, but I personally find it difficult to absorb concepts from a computer screen. In this regard, the Garfinkel and Mahoney book fits, I suspect, most people's "cognitive requirements" to a tee.

"NeXTSTEP Programming" has 21 chapters, extensively illustrated with screen "snapshots." An appendix contains the (final) source code for the three major example applications developed (in palatable pieces) in the chapters of the book. These applications are:

- a four-function Calculator. (You will probably decide at once to add a square-root button.)
- MathPaper, a sort of "word processor" for numerically evaluating mathematical expressions. (Sorry, it

Richard R. Silbar is a theoretical physicist at Los Alamos National Laboratory (Los Alamos, NM), specializing in medium-energy nuclear reactions. He has been using NeXT computers since the days of NeXTSTEP 0.8a, i.e., April 1989. e-mail: silbar@lampf.lanl.gov.

does not integrate or solve differential equations.)

- GraphPaper, a more complex (multithreaded) application, which draws a two-dimensional graph for functions that the user inputs. This application reuses some of the code developed for MathPaper.

Along the way, one learns Objective-C (the book does assume some prior knowledge of C), some PostScript, how to use the Interface Builder and other development tools, and something of the underlying Mach kernel and window server. By the end of the book one has even learned how easy it is to include advanced NeXTSTEP features, such as the pasteboard and services menu, that make a really polished application. The book is written with the latest NeXTSTEP version in mind (3.0), but it also gives pointers for people who are still using NeXTSTEP 2.x.

On the downside, the index is woefully thin; it does not even reference "bitmap." One longs for the day when a book like this can be put online and searched using the Digital Librarian. There are also a number of typos (mostly innocuous) and some buggy pieces of code. [There was a claim on the Net that "a fairly large chunk" of code in *grammar.y* of Ch. 10 is missing (recoverable from the diskette), but I did not see any difference between my book's version and the diskette file.] Actually, considering that Garfinkel and Mahoney essentially did their own typesetting, the book "looks and feels" very good.

To summarize, I believe "NeXTSTEP Programming, Step One" will rapidly become *the* classic textbook for learning to program in what may soon become the standard GUI and object-oriented environment for the 1990s.

Neural Networks in Pattern Recognition and their Applications

C.H. Chen, editor

World Scientific, Singapore, 1991;
ISBN 981-02-0766-2; 159 pp.,
hardcover, $46.00.

Reviewed by Seth Wolpert

Machine-based recognition of sensory patterns, whether visual, auditory, tactile, or chemical, continues to be a technical issue with substantial challenges and rewards. Recent advances in available computing power, and improved understanding of the parallel and simultaneous nature of artificial neural networks (ANNs), have opened a number of new inroads into this field. In the book *Neural Networks in Pattern Recognition and their Applications,* C. H. Chen, Professor of Electrical Engineering at the University of Massachusetts, has assembled ten original articles that illustrate the use of ANNs in various image contexts and pattern-recognition strategies.

In the first of these articles, "Combined Neural-Net/Knowledge-Based Adaptive Systems for Large Scale Dynamic Control," Holden and Suddarth describe the combination of multilayered ANNs with a knowledge-based expert system to learn a complex control task. Rules for control are embedded in the structure of the network, and its training is initially accomplished at accelerated rates with the aid of "hints"—pointers to monotonic regions in solution space. This network is applied as an aid to navigation, being trained to land a spacecraft in rough terrain. In a separate paper, "A Connectionist Incremental Expert System Combining Production Systems and Associative Memory", Yin and Liang describe the use of a similar network to manage classification of animals based on a set of fifteen physical and behavioral parameters.

In "Optimal Hidden Units for Two-Layer Nonlinear Feedforward Neural Networks," Sanger investigates the process by which images inputted to an ANN are best disseminated into the optimal number of distinct features. This information is crucial, as it dictates the dimension of a network's hidden layer, the nodes of which act as individual feature detectors. The system is applied to two examples—a one-dimensional feature extractor and a controller for a robotic arm operating in an *x-y* plane.

The fourth paper, "An Incremental Fine Adjustment Algorithm for the Design of Optimal Interpolating Neural Networks" by Sin and De Figueiredo, also deals with configuring ANNs to the set of input and output parameters which they must accommodate. The resulting network configures itself to contain as many nodes as are required for the classification task at hand. Results are applied to generalized decision boundaries in two dimensions.

In "On the Asymptotic Properties of Recurrent Neural Networks for Optimization", Wang explores how ANNs approach optimal solutions with time-varying penalty parameters. An optimization problem is rigorously formulated, and an example of a network employing a generalized two-dimensional decision boundary to solve a linear equation is given.

In "A Real-Time Image Segmentation System Using a Connectionist Classifier Architecture", Blanz and Gish employ a VLSI-based back-propagation neural network for real-time segmentation for subsequent recognition of two types of visual image: wavefronts of gases in a combustion chamber and traces on printed-circuit boards.

The seventh paper, "Segmentation of Ultrasonic Images with Neural Networks" by Silverman, describes the use of ANNs for enhance-

Seth Wolpert is an Assistant Professor of Electrical Engineering at the University of Maine, Orono, ME 04469.

Programming book a must for all developers

By Brian Bias

Spackling the gap between /NeXTDeveloper/Examples/AppKit/*.m and developer's camp, the recent Springer-Verlag imprint, "NeXTSTEP Programming — Step One: Object-Oriented Applications" is the first serious NeXTSTEP programming text.

Let's face it, you just can't read a book on Objective-C and crank out Improv.app the following week. It takes time to learn how to program a NeXT well. But once you're there — nirvana!

NeXTSTEP programming is a new field requiring a certain amount of acclimation. Since the NeXT computer was introduced, developers have been forced to go to developer's camp, comb through the examples that ship with the machine, traverse the online documentation, and ask fellow developers questions on the net. Now there is an alternative.

The 21-chapter book is divided into five sections: Introductory, Building A Simple Application, Building A Less Simple Application, All About Views and Miscellany.

The first portion of the book is a boon for NeXT neophytes. I wish I would have had this information collected into one spot when I purchased my slab a couple years ago.

Users will learn about the Workspace, short cuts, establishing preferences, etc. Also laid out are the subtle stylistic things that make the NeXTSTEP user interface so appealing.

The authors, Simson Garfinkel and Michael Mahoney, present an interesting exercise of building an application without Interface Builder. This will develop an appreciation for all the tedious work that programming visually with Interface Builder eliminates.

Even without Interface Builder, it is amazing what a page of code can do in this environment.

As the reader builds an aptitude for operating a NeXT computer, he also gains experience in the general practices of object-oriented programming, as the book title suggests.

Although this text should not be considered a replacement for learning the nuts and bolts of Objective-C and object-oriented design techniques, it certainly gets one's feet wet, sometimes before you realize it.

Readers also get a crack at letting Interface Builder do its job. Using the omnipresent calculator example, the authors show how the Project Builder, Interface Builder and Icon Builder developer tools are used to produce a professional-looking application.

Delegation and multiple nibs, some of the potential stumbling blocks for newcomers, are covered. Also covered in this portion of the book is the otherwise hidden main event loop.

Although programmers from other platforms and environments will find that the engineers at NeXT have hidden much of the tedium of decoding and handling events such as mouse clicks and drags, keyboard actions, etc., there still are clean hooks into this loop for handling or overriding these things.

The authors use these techniques to map the computer's keyboard to the keypad of the on-screen calculator. Oddly enough, this portion of the book also contains information on Mach (the underlying operating system) and the Window Server (the low-level drawing and event handler).

The third portion of the book is dedicated to developing a more robust application called MathPaper.

This application uses typical techniques that a programmer would be required to use in real-world scenarios including multiple windows, submenus, autosizing, multiple processes, manipulating the text object, rich text format, scrollviews and more.

Executive Summary

"NeXTSTEP Programming
— Step One"
Springer-Verlag
Santa Clara, Calif.
$39.95
Excellent
Response No.136

The pace picks up here and even seasoned NeXTSTEP developers will find sunken treasures within these pages.

Display-oriented topics are covered in Chapters 14 through 18. Although Display PostScript is in itself a powerful language, NeXTSTEP programmers don't usually have to do it by hand.

The authors show how the sophisticated, if not unwieldy, View class is used to handle simple drawing, light duty animation, color and so forth.

The numerical evaluation engine built up in previous chapters is re-used for the core of a new application called GraphPaper, which plots the output of math functions on-screen.

Lastly, the book covers some ancillary topics necessary to make an application behave in a well-integrated fashion with the environment. These topics include pasteboarding, services and preferences.

A listing of source code used throughout the book also is included. For the impatient, a diskette with source code for all the projects in the book is included. The diskette also contains an Interface Builder tutorial.

One other item worthy of mention is the fact that the authors went to obvious lengths to make the text applicable to both 2.x and 3.x versions of NeXTSTEP.

Garfinkel and Mahoney have done a wonderful job on this book. The material provided by NeXT when the units were shipped is fantastic, but it is for reference. This book will dramatically reduce the learning curve to a pleasant slope.

Moreover, it should be obligatory reading for any newcomer to the world of NeXTSTEP. Everything builds as the book progresses, so don't jump into the middle, read the whole book.

Completing the book leaves the reader wondering what "Step Two:" will be about. My guess is more detailed information on various AppKit brethren such as the DBKit or 3DKit. We'll have to wait and see.

In the meantime, read this book and heed the advice given in the final sentence: "Now go out and write a killer app!" ◆

Brian Bias is the director of the Kentucky NeXT User Group and co-founder of Whetstone Inc., a NeXT-only development firm located in Lexington, Ky. Brian can be reached at <rainman!brian@ms.uky.edu>.

The Future Of Client–Server Database

Making Book

Two volumes get you up to speed with NEXTSTEP programming

by GREGORY H. ANDERSON

Before almost anyone had a NeXT computer, thousands had purchased *The NeXT Book* by Bruce Webster. Quickly followed by Doug Clapp's *The NeXT Bible*, these served as good general introductions to NeXT and NEXTSTEP. But until recently, hard technical information was limited to NeXT's own technical documentation (now published by Addison-Wesley). Critical mass in the market has finally spawned a whole round of in-depth books about NEXTSTEP. This month, *NeXTWORLD* reviews two introductions to NEXTSTEP programming. We'll sample other titles in future issues.

The acid test for an introductory programming book is whether you can give it to a new employee, sit her in front of a machine for a few weeks, and watch her bloom, with minimal assistance, into a competent programmer. By that standard, *NEXTSTEP Programming Step One: Object Oriented Applications*, by Simson L. Garfinkel and Michael K. Mahoney, is an unqualified success.

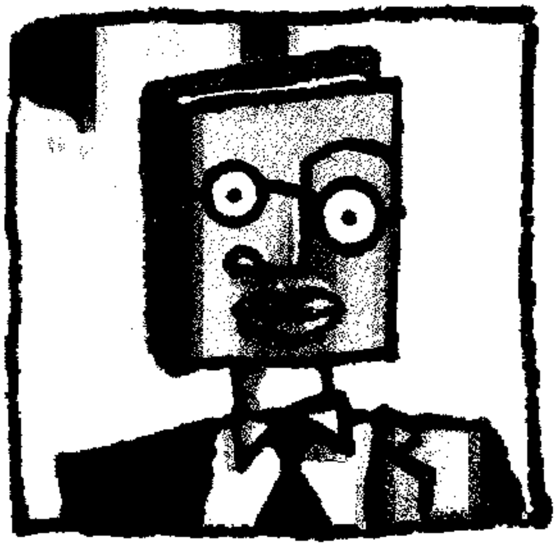
Using simple, progressive, confidence-building examples, Garfinkel and Mahoney quickly cover all the basics: Windows, Menus, Views, delegates, events, responders, nibs, and Projects. Newcomers to NEXTSTEP will quickly learn why seasoned programmers find the OS so exciting – and why a copy of this book is included in every Developer Release box.

Both the sample code and the narrative emphasize good programming habits. For example, the discussion of coherence between an interface and its controller code –

mers' first exposure to object-oriented programming, a short background chapter on object-oriented design would be valuable.

Object-oriented design is covered in *NEXTSTEP Programming: Concepts and Applications*, by Alex Duong Nghiem, but that is the only area in which this book is superior to Garfinkel and Mahoney's. Much of the conceptual material is presented without context, and the examples neglect many NEXTSTEP features.

InterfaceBuilder (IB) is not all there is to NEXTSTEP programming, but it certainly helps get programmers hooked. With that in mind, it is unreasonable to force readers through several hundred lines of



programmatic objects and a private event-processing loop before introducing IB. Even the common user-interface objects – TextFields, Buttons, Sliders – are explained with program code instead of IB examples.

By the end of chapter four, you get the feeling that if Nghiem sold cars, he would rather show you the circuit diagrams for the antilock braking system than take you for a test drive, run the speedometer up to 60, and stand hard on the brakes. Microsoft Windows and Macintosh programmers could be excused for bailing out at this point, wondering what the NEXTSTEP fuss is all about.

The sample programs are not well developed and include some questionable code. The Money example, which implements a foreign-exchange calculator, assumes how many rows the form will contain and allocates static storage to support it. It would be more instructive and more coherent to set up a secondary matrix for the exchange rates, tie it to the primary form, and eliminate the hard coding.

Nghiem skips many fundamental topics that are covered in the Garfinkel and Mahoney book, such as automatic menu-cell updating, Services provisioning, icon building, setting timed-entry events, using categories as an alternative to subclassing, and understanding the distinction between windows and panels. Given his obsession with writing code in instances when InterfaceBuilder could do the work, it is surprising that the book does not include a sample disk.

Both programming books exclude information that would address growing complaints about the lack of extensibility in NEXTSTEP applications: custom palettes, dynamic code bundles, the run-time system, and distributed objects. Ironically, a good model already exists: InterfaceBuilder. An informed discussion of how IB works internally would cover most of these topics and help programmers appreciate how open-ended design provides user empow-

[illegible]

eral introductions to NeXT and NEXTSTEP. But until recently, hard technical information was limited to NeXT's own technical documentation (now published by Addison-Wesley). Critical mass in the market has finally spawned a whole round of in-depth books about NEXTSTEP. This month, *NeXTWORLD* reviews two introductions to NEXTSTEP programming. We'll sample other titles in future issues.

The acid test for an introductory programming book is whether you can give it to a new employee, sit her in front of a machine for a few weeks, and watch her bloom, with minimal assistance, into a competent programmer. By that standard, *NEXTSTEP Programming Step One: Object Oriented Applications*, by Simson L. Garfinkel and Michael K. Mahoney, is an unqualified success.

Using simple, progressive, confidence-building examples, Garfinkel and Mahoney quickly cover all the basics: Windows, Menus, Views, delegates, events, responders, nibs, and Projects. Newcomers to NEXTSTEP will quickly learn why seasoned programmers find the OS so exciting – and why a copy of this book is included in every Developer Release box.

Both the sample code and the narrative emphasize good programming habits. For example, the discussion of coherence between an interface and its controller code – making decisions on the basis of runtime states, rather than preprogrammed information – is vital to maintaining large object-oriented systems. To reinforce this concept, the authors write a poorly constructed method to set the radix of a calculator, then show how the implementation can be improved with run-time object messaging.

But even a good programming book isn't perfect. It's difficult to justify an entire chapter on threads when several more useful topics are omitted (see below). Also, since this book will provide many program-

NEXTSTEP Programming: Concepts and Applications

Alex Duong Nghiem

PTR Prentice-Hall, 1993, 604 pages, ISBN: 0-13-605916-3

◆ ◆

Accomplishes the seemingly impossible: makes NEXTSTEP programming seem boring. Missing fundamental information and uses uninspired examples.

$36.00

Prentice-Hall Professional Technical Reference Division, P.O. Box 11073, Des Moines, IA 50381-1073. 515/284-6751.

is unreasonable to force readers through several hundred lines of programmatic objects and a private event-processing loop before introducing IB. Even the common user-interface objects – TextFields, Buttons, Sliders – are explained with program code instead of IB examples.

By the end of chapter four, you get the feeling that if Nghiem sold

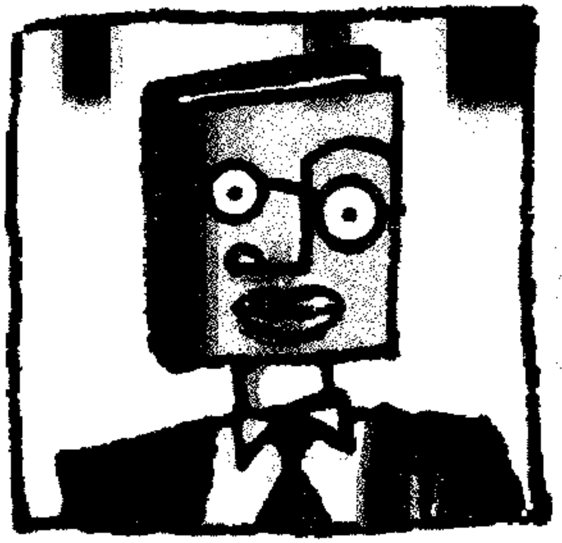


ILLUSTRATION BY J. SCOTT CAMPBELL

NEXTSTEP Programming Step One: Object-Oriented Applications

Simson L. Garfinkel and Michael K. Mahoney

Springer Verlag New York, 1993, 631 pages (with diskette), ISBN: 0-387-97884-4

◆ ◆ ◆ ◆

A superior tutorial for beginning NEXTSTEP programmers. Crisp writing, clean organization, excellent examples. Includes diskette with source code.

$44.95

Springer-Verlag New York, P.O. Box 2485, Secaucus, NJ 07096. 201/348-4033, 800/777-4643.

and more coherent to set up a secondary matrix for the exchange rates, tie it to the primary form, and eliminate the hard coding.

Nghiem skips many fundamental topics that are covered in the Garfinkel and Mahoney book, such as automatic menu-cell updating, Services provisioning, icon building, setting timed-entry events, using categories as an alternative to subclassing, and understanding the distinction between windows and panels. Given his obsession with writing code in instances when InterfaceBuilder could do the work, it is surprising that the book does not include a sample disk.

Both programming books exclude information that would address growing complaints about the lack of extensibility in NEXTSTEP applications: custom palettes, dynamic code bundles, the run-time system, and distributed objects. Ironically, a good model already exists: InterfaceBuilder. An informed discussion of how IB works internally would cover most of these topics and help programmers appreciate how open-ended design provides user empowerment. ◆

GREGORY H. ANDERSON *is president of Anderson Financial Systems, a Springhouse, Pennsylvania developer of custom Wall Street trading applications.*

Since one of the books reviewed was co-authored by a NeXTWORLD *editor, this review was written without imput from the magazine staff.*