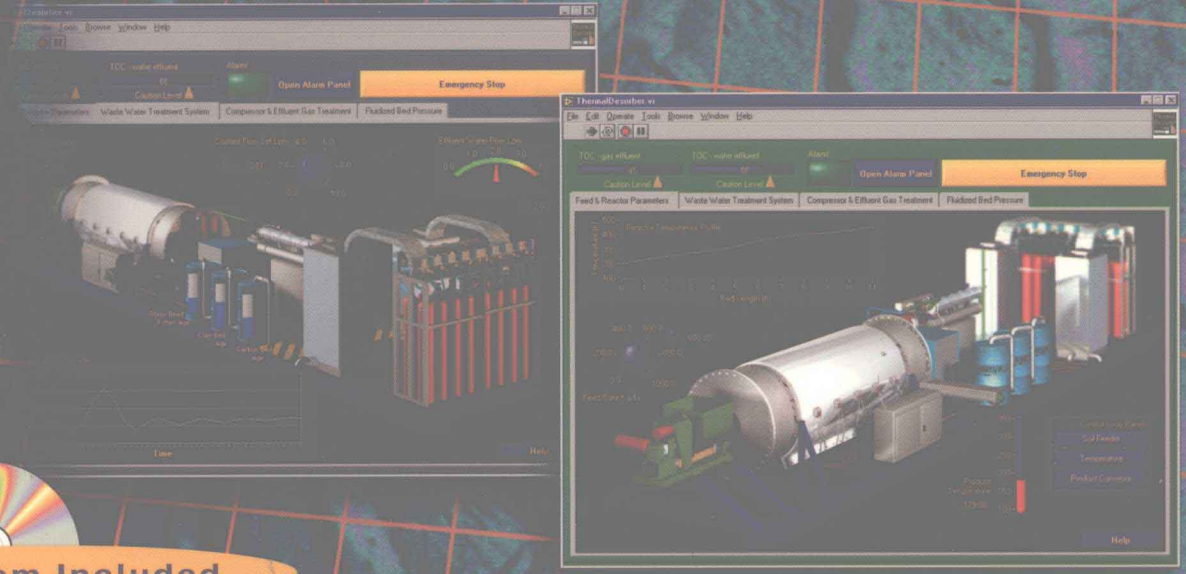
David J. Ritter

LabVIEW® Essential Techniques GUI



CD-Rom Included

LabVIEW GUI

Essential Techniques

David J. Ritter

McGraw-Hill
New York Chicago San Francisco Lisbon London
Madrid Mexico City Milan New Delhi San Juan
Seoul Singapore Sydney Toronto

Library of Congress Cataloging-in-Publication Data

McGraw-Hill

A Division of The McGraw-Hill Companies

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 9 8 7 6 5 4 3 2 1

P/N 139093-6
PART OF
ISBN 0-07-136493-5

The sponsoring editor for this book was Stephen S. Chapman and the production supervisor was Pamela A. Pelton. It was set in Times Roman by MacAllister Publishing Services, LLC.

Printed and bound by R.R. Donnelley and Sons Co.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, McGraw-Hill Professional, Two Penn Plaza, New York, NY, 10121-2298. Or contact your local bookstore.

This book is printed on recycled acid-free stock.

To the users of our applications—those who must endeavor to understand the intent of our designs.

Foreword

Every so often, you meet someone with an exceptional aptitude and unique combination of skills—one of those people who will most assuredly make a difference in his or her line of work. Dave Ritter is such a person.

When I first encountered his articles in *The LabVIEW Technical Resource* several years ago, it was clear that Dave has an unusual and intuitive approach to graphical programming and its stylistic demands. Later, I saw some of his LabVIEW code in action at National Instruments Week, and there was no doubt: This guy needs to tell the world how he works his GUI magic. So I challenged him to swallow the elephant, just like I had done, and write a book. And he did it!

Most of us technical folks get bogged down in the algorithms, ones and zeros, and logic of a problem and leave the user interface as an afterthought. Dave approaches a problem from the opposite and probably more correct direction, making his software user-centric. The original Macintosh team adopted that philosophy when they created that machine's pioneering GUI. "Deliver the most pleasurable user experience possible," they decreed. By golly, I *like* that, but I must admit that it's much easier said than done. Worse, the existing guidelines and texts on the subject of GUI development seem to be written for software quality assurance inspectors rather than developers. So we go about our work in an ad hoc manner and hope for the best.

Now at long last you have in your hands a comprehensive volume devoted to the craft of GUI development à la LabVIEW. Dave covers all the bases in this book, beginning with fundamental aspects of philosophy and psychology that drive human-computer interactions, and ending with advanced programming techniques. He stresses a methodical approach, where you start by understanding your user and defining the tasks your program must accomplish *from the user's point of view*. You then lay out the panel according to some general rules and guidelines. Only after all this up-front conceptual work is done do you select a likely software architecture and begin programming. Dave goes beyond good software engineering practice by having the artist's (or should I say humanist's?) influence circumscribe all efforts. This GUI-centric, soft-science approach is a unique and refreshing way to look at the software development process.

This is really an unusual book as software titles go, and I think that it adds a great deal to the body of GUI design literature. Take the time to absorb what Dave has to say. Adapt his concepts to your applications, and try some of the programming techniques. I bet that you, too, can design effective and enjoyable user interfaces that will make people smile.

Gary W. Johnson
Livermore, CA
August 2001

Acknowledgments

In retrospect, I suppose it began with a phone call—a fateful phone call from well-known author and preeminent LabVIEW guru Gary Johnson; this was the catalyst that set the big wheel in motion. His jovial words of caution now echo in my mind's ear: something about how "writing a book of this size can feel like kicking a beached whale—you keep hammering away, but the monster just doesn't seem to move." Being no stranger to large projects, I simply laughed off this lighthearted warning. Later, I began to wonder whether this warning was indeed intended as a joke—for although I had come up against orcas, humpbacks, belugas, and even the odd narwhal in the past, as Captain Ahab attests in *Moby Dick*, nothing can adequately prepare you for a battle with "the Great White One."

Luckily, as I made the daily pilgrimage to the water's edge to begin the daily kicking ritual, I often found I wasn't alone. In this section I would like to acknowledge all of those who also lined up to take a shot at the beast on my behalf.

First of all, I would like to thank my wonderful wife Kathy, young daughter Jaqueline, and newest arrival Julian for providing inspiration, love, and support throughout the process. Although I wasn't away on an extended sea voyage (as implied by at least one-half of the previous mixed metaphor), at times I might as well have been. Thank you all for your patience and understanding during my virtual absence, and for providing a constant source of support and inspiration. I would also like to thank my parents for instilling in me a fascination with both technology and the arts. Without a deep respect for all types of human creativity, this book could not have been possible.

Next, I would like to thank the following people for improving the book by reading early drafts and supplying corrections and critical advice:

Thanks to R.E. "Dick" Berry, distinguished engineer from the Ease of Use Architecture and Design Group at IBM. Dick's informative and useful clarifications did much to enhance the sections on GUI psychology and the GUI design process. Thanks also to Jason Dunham from San Francisco Industrial Software for making several constructive and thought-provoking suggestions about both style and content. Thanks also to Gary Johnson and Richard Jennings, who, while revising Gary's book, managed to find time to read a couple of chapters each and make constructive suggestions. And to Russell Davoli for the useful feedback he provided

while reviewing two of my *LTR* articles that were later updated and reworked for inclusion here. Thanks also to my brother Paul, for not only reading a few chapters, but also for spending several hours modeling and responding the 3-D Thermal Desensor Process Graphic features in several figures throughout the book. And again, to my wife Kathy, whose comments and critiques inspired me to inject some life into several otherwise dull sections of the text.

I would also like to thank those special people who contributed material in the form of articles or VIs, thereby enhancing the text and the CD-ROM offerings:

George Wells from JPL who, when asked for a couple of screen captures and a short article, contributed a complete chapter on sophisticated *xy* graph techniques; Dr. Don Roth, from the NASA Glen Research Center, who offered several interesting insights during the writing process and the article in Appendix A; Jean-Pierre Drolet from Scientech R&D Inc., who submitted several in-depth articles and sample VIs demonstrating very creative LabVIEW GUI techniques (for a period of several weeks, every time I checked my e-mail, I discussed a new article or VI from Jean-Pierre); Christophe Salzmann from EPFL, who, despite the demands of a looming Ph.D. and the birth of his son Tristan, still contributed several example VIs, including the amazing 100 percent G Movie Player VI featured in Chapter 9, "Exploring LabVIEW GUI Customization." Thanks also to Christophe and his colleagues Denis Gillet and Pierre Huguenin for permitting me to include an electronic copy of their enlightening article "Remote Experimentation: Improving User Perception Using Augmented Reality" on the CD-ROM; Jean Paul Osborne from CARDIAC in Norway for his article introducing LabVIEW Voice Recognition using ActiveX, and for the VI Localization Toolkit and supporting materials included on the CD-ROM; Danny Lauwers for his LVToolBox VIs—an important enabling component for a couple of the example VIs featured in the text; Sam Michael for his slick Picture Control Toolbar VI; Stepan Riha for his innovative Codeless Radio Buttons; Rus Belikov for his entertaining and enlightening Piano VI; Konstantin Shifershtain for his CalcExpress Demo; Greg Swanson and TimeSlice for the OverVIEW Demo; and Doug Wilson for the RaceVIEW screen capture.

Due to space, time, and, in some cases, copyright restrictions, some of the material submitted didn't make it into the final package. Still, I would like to thank Dominic Lavoie, Dirk Nuyts, Robert Huntley, Willem Blokland, Frank Speers, Peter Prinzen from XOn, Dirk De Mol from Honeywell-Measurex, and Bryan Webb from Microsys Technologies for their valuable, if overlooked, contributions to this project.

Kyle Gupton at National Instruments helped immeasurably by answering the questions only he could answer, and Gurshan Sidhu, also from National Instruments, merits special mention for raising awareness for the BetterVIEW message. Special thanks must also go to Karen Pape and everyone affiliated with *LTR* Publishing, for permitting me to borrow a few items from their excellent publication, *The LabVIEW Technical Resource*. While reviewing *LTR* back issues in preparation for this book, I was struck by the influence *LTR* has had in the evolution

of my own LabVIEW programming style. As well, thanks to Tom Coradeschi and the contributors to the Info-LabVIEW mailing list; Ed Baroth from JPL, Jean Peccoud from e-NoteBooks, Inc.; Brian Paquette from SensArray, and countless other members of the LabVIEW community who have indirectly helped to make this book a reality.

Special thanks to Steve Chapman, Jessica Hornick, and the editorial staff at McGraw-Hill for providing gentle but constant pressure throughout the process, along with Beth Brown and the team at MacAllister Publishing Services who transformed my manuscript into a real book. Thanks also to Ben Robledo, Susan Doering, and Joyce Fowler at Adobe Systems Inc. for helping to make the Photoshop section of the book a reality, and to Chris Brandkamp from Cyan for providing the Riven screen captures. Thanks also to author and GUI pioneer Alan Cooper of Cooper Interaction Design for writing that amazing GUI design book, *About Face*, and for allowing me to modify one of the figures for inclusion here. I continue to be overwhelmed at the number of highly skilled professionals who took time out of their busy schedules to enhance the value of my book. Thank you all!

Special thanks also to National Instruments, Jeff Kodosky, and the entire LabVIEW team for building one of the most flexible, powerful, easy to use—not to mention fun—cross-platform software development environments on the planet! The transparency, accessibility, and logic of the LabVIEW GUI continue to provide a limitless source of GUI design inspiration.

And finally, I would like to thank Gary Johnson for his encouragement and enthusiasm, and for providing the gentle nudge that got me in "up to the eyeballs" before I fully comprehended the size of the undertaking. Without Gary's input in the early stages, this book may never have come into being. Of course, knowing what I know now, would I every write another LabVIEW book?

Just give me a minute or two to locate my harpoon!

Introduction

> User interface design begins well below the surface of our systems and applications. Imagining that we can create a good user interface for our programs after the program's internals have been constructed is like saying that a good coat of paint will turn a cave into a mansion.
>
> Alan Cooper from *About Face* p. 51

Ask yourself this: When friends, colleagues, customers, your boss—and, most importantly, end-users—peer into your LabVIEW *graphical user interfaces* (GUIs), what do they see? Is the function and intent of your GUI design immediately apparent to end-users, or are your best GUI efforts often greeted with a bewildered stare? And what about the visual presentation of your panels? Are your aesthetic sensibilities praised by all, or are you frequently subjected to a stream of unsolicited suggestions about how you could improve the appearance of your panels? Sure, below the surface your software efficiently does everything it is supposed to do, but how much time and effort do you typically devote to making your GUI easy to use and pleasing to the eye?

With the explosive success and widespread adoption of graphically based operating systems and applications, it is not surprising that more than 50 percent of all laboratory and control applications now employ some form of GUI. Unfortunately, scientists, engineers, and technicians are not typically trained in either the technical or aesthetic aspects of effective user interface design. Few have any background in cognitive psychology, environmental design, or the other disciplines typically associated with human factors engineering. Fewer still have attended art school, studied composition, graphic design, or color theory. Yet given these limitations, legions of LabVIEW developers regularly venture totally unprepared into the multidisciplinary, right-brain-dominated world of user interface design—often with disappointing results.

Unquestionably, the GUI design process is more complex than it first appears. Still, the success of any software project hinges on the effectiveness of the GUI, and how quickly, easily, and efficiently users can achieve their intended goals. This book addresses the complexity of the GUI design process on several levels, first by examining fundamental GUI design principles and elementary graphic design considerations, and then by presenting this theoretical material in several practical contexts. Moreover, it clearly demonstrates how to implement modern GUI

methodologies during the LabVIEW application development process. Through a generous number of graphic depictions, sample programs, and practical, step-by-step explanations, this book explores not only aesthetic issues, but also the architectural considerations of designing clean, reusable LabVIEW code for GUI applications. It is the intent of this book to help even the most artistically challenged technician to design coherent, modular, and aesthetically pleasing GUIs.

According to the popular book by Michael J. Gelb entitled *How to Think Like Leonardo da Vinci*, one of Maestro da Vinci's seven guiding principles is "dimostrazione." Roughly translated, this means "learning by doing." Learning by doing provides understanding, context, and organization, and therefore is much more effective than learning by rote. Similarly, this book strives to provide understanding, context, and organization to the material presented by providing a big-picture view of the subject matter. In addition to providing specific how-to techniques, this book also attempts to build a solid foundation in the underlying psychological and aesthetic concepts of GUI design, encouraging you to complete the circle through your own experiments and dimostrazione. It is hoped this approach will offer a greater degree of depth than a conventional style guide approach and foster enhanced comprehension, storage, and eventual recall of the subject matter presented. With a conceptual understanding of the entire GUI design process, you can begin to experiment and discover the solutions that work best for your application and its intended end-users.

Extending the Model

When National Instruments began developing LabVIEW in the mid-1980s, the original vision was to design a virtual instrument, a flexible, general purpose laboratory device based on standard PC hardware, with a robust, yet simple-to-use software user interface. Designed primarily for engineers and scientists, the intended scope of this early system was limited to laboratory workbench implementations. Over the years, LabVIEW has surpassed the modest early expectations of the original design team, and the power and flexibility of the LabVIEW model has opened the door to an almost limitless range of potential application areas—many of which have sophisticated GUI requirements. In addition, continuous improvements to the underlying PC technology and to the LabVIEW runtime architecture have made it possible for LabVIEW to compete with so-called general-purpose programming languages in a multitude of application areas. Extending far beyond the intended realms of test and measurement, LabVIEW is now used for general-purpose system utilities, computer simulations, and even arcade-style video games and embedded systems!

As the LabVIEW universe continues to expand, user interface methodologies necessarily must evolve to support this change. The original laboratory virtual instrument engineering-workbench GUI metaphor has been stretched to its limits

and beyond. As we prepare for the next generation of LabVIEW applications, the time is right for expanded LabVIEW GUI awareness!

Overview of Chapter Contents

Chapter 1, "The Case for a Superior GUI," discusses the critical importance of a well-designed GUI. As the end-user's only contact with the software, a well-designed GUI reflects the care, planning, and attention that went into the building of the application. Areas of discussion include how high-quality GUIs improve usability, end-user perception, marketability, and the overall design of any LabVIEW application. This chapter ends with an invitation for you to subjectively evaluate your own GUI designs and prepare some brief notes outlining the perceived strengths and weaknesses of your current approach.

Chapter 2, "The Psychology of Human-Computer Interactions," provides background into the psychological aspects of GUI design. GUI design models, including the User's mental model, the Designer's model, and the Programmer's model, are introduced and discussed in some detail. In addition to introducing key GUI design principles, this chapter also defines the look and feel elements of the GUI and explains why these are only a small fraction of the complete GUI design equation.

Like all complex endeavors, a successful GUI implementation depends on a well-developed plan. Chapter 3, "Targeting Your Destination," discusses the importance of clearly defining the objective before the implementation phase begins and touches on how standard software engineering practices can be applied to the LabVIEW GUI design process. LabVIEW application planning is examined from both a general design perspective and the GUI design perspective. Software life cycle models are applied to the GUI design process, and the importance of usability targets is highlighted. The advantages of evaluating user requirements from the perspective of acquisition, analysis, and presentation is explored, and finally, the chapter closes with a section devoted to planning the visual presentation of your GUI.

The focus of any GUI design is the end-user; therefore, Chapter 4, "Understanding Users," is devoted entirely to gaining insight into the needs, strengths, weaknesses, and individualities of users. Common human attributes such as cognitive information-processing systems—visual and audio receptors, perception, thought process, and memory systems—are discussed in some detail. The practical GUI implications of visual and sound perception are highlighted along the way. In addition to common human attributes, differences between users are also discussed. When designing a GUI, it is important to consider the physical, psychological, professional, and personal differences between users and design accordingly. This chapter is intended to keep you focused on the primary purpose of any computer application—helping users achieve their intended goals.

Before you can build an effective GUI, you must clearly define user goals and task sequences. Chapter 5, "From Task Definition to GUI Design," discusses information-gathering strategies and task analysis techniques—practical tools to help you identify the user action sequences and the discrete interaction objects necessary to provide a complete and efficient end-user experience. Next, several important design considerations such as the three-clicks rule are introduced and developed, along with the importance of usability testing to verify GUI design decisions. This chapter concludes with an exploration of both the advantages and disadvantages of visual metaphor GUI designs, and some guidelines for incorporating visual metaphor designs into LabVIEW applications.

Chapter 6, "Graphic Design for Engineers 101—A Crash Course in Layout and Design," serves as a practical introduction to graphic design fundamentals. Topics include the importance of GUI aesthetics, developing an appropriate design concept and image, basic graphic composition and layout, and even color theory and usage guidelines (including a VI designed to automate the task of color palette selection). The chapter concludes with basic typography and notes on font usage. A generous number of GUI screen captures are included to graphically underscore the concepts introduced in the text.

Whereas Chapter 6 was concerned entirely with the front panel presentation, Chapter 7, "Building GUI VIs," is devoted completely to VI diagrams. This chapter opens with an examination of standard LabVIEW application architectures and the GUI implications of each. The GUI advantages of multiple loop architectures and parallel independent processes are highlighted, along with the wide variety of data transfer options available to developers who choose to step beyond the direct wire connections of pure data flow programming. Following this, all aspects of custom menu integration are covered—from the Menu Editor and .rtm files to programmatic menu generation. Menu design recommendations are also included to help you logically organize your menus and squeeze optimum usability levels out of these important GUI workhorses. The menu programming section culminates with two practical examples: a multilevel, programmatic Undo function and a Recently Accessed item list—both powerful features you can add to your application's custom menu. The final section of this chapter is devoted to advanced GUI architectures. Topics of discussion include the popular queued state machine and other event-driven GUI strategies.

One of the most powerful and versatile aspects of the LabVIEW environment is VI Server, and in Chapter 8, "VI Server GUI Techniques," a multitude of GUI-enabling VI Server techniques are introduced. Beginning with simple panel control tricks, such as opening, closing, and resizing front panels programmatically, this chapter traverses the gamut of VI Server's GUI possibilities. VI Server's Call by Reference capability is used to demonstrate plug-in architectures, and an interesting new VI Server technique I like to call Selective GUI Panel Control is used to construct completely portable GUI code modules. The power of this technique is fully revealed through several examples, including a modular progress display VI, a fully integrated Operator Security System, and a network-enabled error man-

ager framework. The chapter closes with a discussion of VI reference manager strategies and a simulation of the Windows taskbar—enabled, of course, using a VI reference manager scheme.

Chapter 9, "Exploring LabVIEW GUI Customization," spotlights the look and feel aspects of the GUI, and focuses on the point where front panel techniques and VI wiring converge. This chapter is chock-full of practical GUI examples and explores a variety of topics, including control references, custom graphic integration, control editor techniques, ActiveX integration, and more. Countless practical examples are presented and developed, ranging from GUI special effects like smooth text scrolling and drag-and-drop implementations to LabVIEW Draw, a simple painting application based entirely on LabVIEW picture control functions. The challenge of building a user-defined toolbar is approached from several perspectives, and a LabVIEW-based voice recognition VI is enabled with the help of ActiveX. Step-by-step control editor instructions are provided for constructing a three-position toggle switch, codeless radio buttons, a codeless toolbar, and other useful custom controls. The chapter closes with a brief discussion of tab control guidelines, including a couple of innovative tab control strategies. From start to finish, the practical GUI examples in this chapter are intended to fuel your imagination and inspire fresh, new solutions to typical GUI design challenges.

The most dramatic way to differentiate your LabVIEW GUI designs from the competition is through custom graphics integration. In Chapter 10, "Preparing Graphics Elements for LabVIEW GUI Integration," all pertinent aspects of the custom graphics integration process are explored. Beginning with a number of compelling arguments in favor of custom graphics integration, this chapter also discusses the relative advantages and disadvantages of popular LabVIEW-compatible image formats. Useful online sources for graphic elements are identified, and image optimization procedures designed to minimize the performance impact of custom graphics integration are also discussed in detail. A trial version of Adobe Photoshop 6.0 (included on the CD-ROM) and a number of supplementary hands-on exercises reveal simple techniques for generating and optimizing custom bitmap images destined for LabVIEW GUI integration.

In the first half of Chapter 11, "A Case Study: The VI Facelift," GUI requirements are defined for a real-world application and a typical LabVIEW GUI panel is proposed to meet these requirements. In the second half of the chapter, this original GUI design is subjected to a complete GUI makeover. Starting as a typical gray-on-gray LabVIEW panel with the usual assortment of controls, indicators, and GUI labels, the panel in question is transformed into a slick, professional-looking GUI using the principles and techniques presented throughout the preceding chapters. In essence, this chapter effectively combines the material presented throughout the book into a single, detailed demonstration.

The final chapter, "Advanced Graph Techniques," was contributed by George Wells from the Measurement Technology Center of NASA's Jet Propulsion Laboratory. In this chapter, George enlightens us with several advanced *xy* graph strategies. Starting with an overview of basic graph skills, Mr. Wells demonstrates

several valuable tricks and techniques, such as how complex numbers can be used to simplify the translation, rotation, and resizing of arbitrarily shaped graph objects. Using the techniques developed in the first part of the chapter, a detailed animation project is presented—a variation of the familiar Robot Arm VI, in this case implemented using an *xy* graph. Next, examples are introduced to demonstrate discontinuous or segmented plots, and how to simulate intensity graphs and waveform charts using *xy* graphs. The chapter ends with discussions and screen captures revealing some of the ways these techniques have been applied in real-world NASA applications. Unquestionably, the advanced material in this chapter is guaranteed to alter your perception of the humble *xy* graph forever.

Appendices

Dr. Don Roth from the NASA Glen Research Center at Lewis Field has contributed the content for Appendix A, "LabVIEW Interface Concepts Used in NASA Research." Dr. Roth's article discusses the GUI design strategies adopted for two recent LabVIEW projects at NASA—an Ultrasonic Measurement System and an aircraft-mounted Cloud Absorption Radiometer. A brief overview of the application requirements is provided for each project, along with screen captures of the final GUI solutions. Example VIs included on the CD-ROM accompany Roth's article, so you can experience these GUI solutions directly while reading the text.

A GUI Design Worksheet has been provided in Appendix B. This worksheet is in the form of a questionnaire and can be used to help you evaluate key design attributes of your LabVIEW GUI designs.

Appendix C outlines the contents of the CD-ROM included with this book. In addition to example VIs referenced throughout the text, the CD-ROM also contains supplemental text material, how-to Quicktime movies, interesting third-party articles, and trial versions of Adobe Photoshop for both Windows and Macintosh operating systems. For more information, refer directly to Appendix C.

In "A Brief History of Human Computer Interaction Technology," Brad A. Myers predicts that "user interfaces are likely to be one of the main value-added competitive advantages of the future, as both hardware and basic software become commodities." This book is designed to help you prepare for this future by fostering a deeper understanding of the motivations for—and the relative impact of—various GUI design choices. By combining numerous practical examples along with entire chapters devoted to GUI concepts and foundations, I sincerely believe this book will help you become better prepared for change in the fast-paced world of technical software design.

David J. Ritter
Vancouver, BC

Contents