

经 典 原 版 书 库

Software for Use

(英文版)

WEB
POWERED

Larry L.
Constantine

& Lucy A.D.
Lockwood



SOFTWARE FOR USE

A Practical Guide
to the Models
and Methods of
Usage-Centered
Design

(美) Larry L. Constantine
Lucy A.D. Lockwood



机械工业出版社
China Machine Press

Software for Use

版

机械工业出版社

经典原版书库

Software for Use

(英文版)

Software for Use

A Practical Guide to the Models
and Methods of Usage-Centered Design

(美) Larry L. Constantine 著
Lucy A.D. Lockwood



机械工业出版社
China Machine Press



English reprint edition copyright © 2002 by PEARSON EDUCATION
NORTH ASIA LIMITED and CHINA MACHINE PRESS.

Software for Use: A Practical Guide to the Models and Methods of
Usage-Centered Design by Larry L. Constantine & Lucy A. D. Lockwood.
ISBN 0-201-92478-1. Copyright © 1999.

All rights reserved.

Published by arrangement with Pearson Education, Inc.

本书英文影印版由美国Addison Wesley公司授权机械工业出版社
在中国大陆境内独家出版。未经出版者书面许可,不得以任何方式抄
袭、复制或节录本书中的任何内容。

本书版权登记号: 图字: 01-2002-1876

图书在版编目(CIP)数据

Software for Use (英文版) / (美) 康斯坦丁 (Constantine, L. L.),
(美) 洛克伍德 (Lockwood, L. A. D.) 著. - 北京: 机械工业出版社,
2003.1

(经典原版书库)

书名原文: Software for Use

ISBN 7-111-10581-8

I. S... II. ①康... ②洛... III. 软件设计-英文 IV. TP311.5

中国版本图书馆CIP数据核字(2002)第051894号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 华 章

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年1月第1版第1次印刷

880mm × 1230mm/32 · 18.875 印张

印数: 0 001-3 000册

定价: 39.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭集了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔除外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们的服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

Acknowledgments

This book has been a long time coming. Too long, many would say, and we would have to include ourselves in the chorus. The concepts and techniques of usage-centered design evolved slowly over the entire six years we have been working together. They have flowed from many fonts, emerging from our own work on user interface design problems, from the work of our many clients, from our attempts as teachers to communicate ideas, and from the efforts of our countless students to learn what we were communicating.

The book itself has taken about twice as long to write as it should have. We apologize to our clients and students and colleagues who have waited so long, but for all the frustrations there is also compensation: It is a much better book for having had the extra review and input and revision and refinement.

Along the way, there have been many contributors, believers, and supporters whom we would like to acknowledge. It began with Ivar Jacobson, who not only contributed a cornerstone of the foundation on which our work is built but also has been a supportive colleague throughout the evolution of our models and methods.

It has been said that consulting is a form of paid education. Every good consulting engagement is a learning experience, and as we have helped clients we have also learned from them. Among our many clients have been visionaries who recognized usability as a critical problem in delivering quality service and products and who saw in usage-centered design a practicable solution. It has been our privilege to work with people like Werner Hoefler, Franz Petz, and Helmut Windl of Siemens AG in Germany and Bryan Luke at Westpac Bank in Australia and to help them turn visions into realities.

We are grateful to numerous colleagues who have engaged in a dialogue with us that helped shape and refine our thinking, among them Hugh Beyer, Ian Graham, Karen Holtzblatt, Hermann Kaindl, Meilir Page-Jones, and Rebecca

Wirfs-Brock. Special thanks go also to Jared Spool, whose affectionate antagonism has often led us to tighten our concepts and streamline our methods.

We are particularly grateful to James Noble, who joined with us to teach usage-centered design and brought new energy and insight at a stage when everything seemed settled. In insisting we practice what we preach and calling on us to simplify and generalize, he helped make fundamental improvements in both the approach and how it is taught.

We also want to thank Roy Lockwood, who at the age of 87 started learning to use computers and in the process gave us new insights into the shortcomings of modern graphical user interfaces.

The book would not be in your hands now were it not for the faith and fortitude of our long-suffering editor, Peter Gordon, who said at the outset that he wanted a work of lasting value, even if it took longer to deliver. We look forward to the long-promised celebratory dinner with him. Our appreciation also goes to associate editor Helen Goldstein and production coordinator Genevieve Rajewski, who alternately cajoled, pestered, and pleaded with us in an effort to bring this work to print. And thanks also to Jean Peck, our vigilant, thorough, and unfailingly correct copy editor, who saved us from numerous small embarrassments and helped polish the final work to an acceptable patina.

The greatest debt, not surprisingly, is owed to our students. Those thousands of students in the seminars, workshops, lectures, and tutorials we have taught have also taught us. Every time we teach, we discover something new that causes us to refine the concepts, polish the presentation, and revise our thinking. Thank you, all of you. We will try to keep up with you.

Preface

Software for Use is a book for professionals under pressure, for those who work under the dual demands for high-quality software-based systems on the one hand and faster delivery with fewer resources on the other. This book shows how to use streamlined techniques to make software-based systems significantly more usable and useful regardless of whether these systems are installed on desktop computers, ensconced on large mainframe servers, embedded in industrial electronics, or accessed via the World Wide Web. It is written primarily for computer software and applications developers—the analysts, designers, engineers, and programmers who are responsible for producing software systems. In concept, perhaps, usability ought to be the bailiwick of usability specialists. In practice, however, developers of various stripes make the vast majority of the myriad decisions that shape and determine the ultimate usability of software-based products; it is for these frontline troops of the industry that this book was conceived.

Usage-centered design, the approach taken in this book, originated with our own frustrating experiences as users of software and our recurring dissatisfaction with software that was unnecessarily awkward and difficult to use. The wealth of established knowledge about human–computer interaction was somehow not being fully translated into more usable products. As we studied how software-based products were being developed, we began to understand where things were going wrong and what needed to be changed in order to deliver better systems.

Usage-centered design is a streamlined but systematic approach for devising software closely fitted to the genuine needs of users—software that is not only more useful and easier to use but also simpler and easier to construct. Usage-centered design is tailored to the high-pressure realities of modern software development. Using a few simple but powerful models, it provides a scheme for quickly understanding users in relation to systems, their working intentions in carrying out tasks, and the support they need from the system to perform those tasks. Its

methods and models can be applied within almost any software development life cycle (SDLC) model and incorporated into almost any modern development practices, including various object-oriented (OO) approaches such as the Unified Modeling Language (UML). Usage-centered design is not partial to any one language or platform, being equally effective with rapid iterative development using the latest integrated visual development environments or with character-based control systems running on specialized hardware.

A major part of our message to designers and developers of every ilk is simple: Usability is not rocket science. It is often hard work and invariably requires great attention to detail, but, given a few basic conceptual tools, you, too, can learn how to recognize problems in usability and how to improve the usability of the systems you develop. Professor Woody Flowers of the Massachusetts Institute of Technology has given cameras to middle-school students, who then must photograph hard-to-use things and explain the problems. If untrained teenagers can manage the basics of usability, competent adults ought to be able to master the essentials.

Quite honestly, we think the model-driven approaches explained in this book can be of value to almost any professional with responsibility for product design and development, and that includes a full range of design specialists, from user interface and interaction designers to ergonomists and human factors engineers, from graphic artists and industrial designers to human-computer interaction specialists and usability testers. In our experience, the most important prerequisite is not a particular academic degree or level of applied experience, but the willingness and ability to approach the material with an open mind.

In our own design work and teaching, our emphasis has always been on simple, powerful techniques that can be quickly learned and applied. For example, a free-lance designer specializing in Web-based applications learned about usage-centered design techniques in a short conference presentation and was able to apply them immediately in her work. After attending a one-hour lecture, a project leader at a computer peripherals company taught his group how to improve the user interfaces of their software through use case models. Around the world are companies successfully applying these techniques to problems as diverse as banking applications, industrial automation controls, and commercial software development tools.

We should not give the impression that usage-centered design is for everyone or for all problems. We use models as a kind of vaulting pole, to gain leverage and height in hurdling the problems of user interface design. In our experience, using simple models to understand a problem and potential solutions speeds up the process, but some newcomers to such techniques will find it difficult not to just skip up to the bar and leap before they look.

Although we both have backgrounds heavy in the human sciences, we draw a sharp distinction between interesting research issues and the realities of everyday decision making. Among academics and professionals are those who seem to

believe that human-machine interaction can be understood only through cognitive and perceptual psychology, that a thorough grounding in the academic literature is a prerequisite for doing effective interaction design. Our own views are that many of the complexities and subtleties that loom so large for researchers and research laboratories are of lesser consequence in everyday decision making.

We consider the creative urge a vital force in good design, but we base it on a platform of sound engineering. Some professionals who think of user interfaces as a form of theater or consider design to be based primarily on artistic inspiration have had to struggle with the methodical nature of usage-centered design. On the other hand, the enthusiastic converts to our model-driven approach include designers and artists who have found that abstract models can inspire greater creativity.

In organizing the book, we have striven to live up to our subtitle, creating a practical guide that is more than an introduction even if perhaps less than a handbook. To this end, the book is organized into distinct sections. Of course, they are written with the expectation that most readers who want to understand the material thoroughly will take them in sequence. In particular, most of the later chapters depend to a substantial degree on the conceptual foundation laid in the first six chapters. In fact, if all you seek is a general understanding of usage-centered design, Sections I and II cover the core material. In Sections III and IV, we shift to the sundry and sometimes messy details of practical application. This is likely to be of particularly keen interest to developers, especially those with limited background in user interface design, but we would not recommend beginning with these sections.

In keeping with the practical, applied focus of the book, numerous small specific examples are used throughout the text. In addition, we have included two completely worked-out applications of usage-centered design. One, introduced in Chapter 4, is used for illustration throughout the middle chapters and concludes in Chapter 10; the other forms the whole of Chapter 15. If you are one of those “get-down-and-dirty” detail-oriented developers who cannot resist seeing how it all works out in the end, you might begin with the applied example in Chapter 15 before going back to build the necessary background by working through Chapters 4 through 14.

Because we see on-line documentation to be an integral feature of the user interface and a major factor in usability, we have devoted an entire chapter to help systems. Once familiar with the basics in Chapters 2 and 3 plus Section II, technical writers and documentation specialists should find much of interest in Chapter 11.

Professionals who already have a thorough background in human-computer interaction and user interface design and who are most interested in what makes usage-centered design unique may want to skip directly to the core material on the models and modeling techniques in Section II. Other material that has proved to

be of particular interest to such professionals can be found in Chapter 12 on the progressive usage model and in Chapter 17 on usability metrics.

Managers, project leaders, and others who might be more interested in the management and organizational implications of usage-centered design than in its technical details might start with the first two chapters, then skim Section V on assessment and improvement, and finish off with Section VI on organizing and managing the process.

Despite its long gestation period, the material herein is still a work in progress. Just as it has been enhanced and extended through the contributions of our many students and clients, we anticipate it will also be advanced by you, our readers. As we tell every class we have ever taught, we want to hear from you. So, please tell us about your experiences in putting usage-centered design into practice, and we will keep improving on the usability of the process.

Contents

Acknowledgments *xii*

Preface *xiv*

SECTION I	TOWARD MORE USABLE SOFTWARE	1
1	SOFTWARE FOR USE: Usage, Usability, and User Interfaces	3
	Upgrading Usability	3
	Approaching Usability	12
	Changing Contexts	18
2	BUILT-IN USABILITY: A Usage-Centered Design Approach	21
	Interfacing with Users	21
	Elements of a Usage-Centered Approach	23
	Driving Models	25
	Coordinated Activity	33
3	IN PRINCIPLE: Rules and Principles of Usage-Centered Design	41
	Design as Dialogue	41
	Rules and Principles	45
	Usability Rules	46
	User Interface Design Principles	51
	Other Rules	63
	Details, Details, Details	65

SECTION II ESSENTIAL MODELS FOR USABILITY 67

4 USERS AND RELATED SPECIES: Understanding Users and User Roles 69

- Of Use and Users 69
- Real Users and Others 70
- User Role Models 78
- User Role Maps 84
- User Roles in Action 86
- Structured Role Models 89

5 WORKING STRUCTURES: Task Modeling with Essential Use Cases 97

- Work, Work, Work 97
- Task Modeling 99
- The Use Case Map 109
- Building Essential Use Case Models 115
- Application 119

6 INTERFACE ARCHITECTURE: Interface Contents and Navigation 125

- Workplaces 125
- Interface Contents 127
- The Context Navigation Map 135
- Application 141

SECTION III CREATING THE VISUAL DESIGN 147

7 DESIGNING THE DIALOGUE: Layout and Communication 149

- From Abstraction to Expression 149
- Communication Channels 150
- Screen Real Estate 158

8 PRACTICAL WIDGETRY: Choosing and Designing Visual Components 165

- Buy or Build 165

Iconic Communication	166
Menus	170
Selecting Selection Widgets	179

9 INNOVATIVE INTERFACES: Creative Interface Engineering and Custom Components **185**

Creative Engineering	185
The Process of Innovation	189
Instructive Interfaces	194
Applied Innovation	205

SECTION IV COMPLETING THE DESIGN **209**

10 EXPRESSING SOLUTIONS: Implementation Modeling and Prototypes **211**

Fun Stuff	211
Prototypes and Prototyping	212
Mapping the Models	218
Implementation Modeling Illustrated	225

11 HELP ME IF YOU CAN: Designing Help and Helpful Messages **231**

Even Experts Need a Lift	231
Use Cases for Help	232
Access and Presentation Techniques	237
Special Techniques and Modalities	243
Helpful Writing	248
Helpful Messages	253

12 ONCE A BEGINNER: Supporting Evolving Usage Patterns **265**

Beyond Beginners	265
Skiing the Interface	267
Progressive Usage	268
Supportive Interfaces	272
Designing for Progressive Usage	280
Progressive Usage Applied	293

13	IN PLACE: Fitting the Operational Context	297
	Unsound Context	297
	Operational Modeling	298
	Environmental Adaptation	299
	Binding Context	300
	Environment Profile	308
	Putting Context in Place	313
14	SAME GAME, DIFFERENT FIELDS: Special Applications, Special Issues	317
	Theme and Variation, Again	317
	Web Design for Use	317
	Web Wisdom Applied	330
	Embedded Systems Applications	338
	Other Special Interfaces	354
15	USAGE-CENTERED DESIGN APPLIED: The TeleGuida Case	357
	Scaling Up	357
	Telephone Tag	358
	Gathering Requirements	359
	TeleGuida Users and Uses	366
	Toward a TeleGuida Prototype	378
SECTION V	ASSESSMENT AND IMPROVEMENT	389
16	BETTER NEXT TIME: Improvement by Inspection and Review	391
	Assessing Usability	391
	Inspection Methods	399
	Collaborative Usability Inspections	401
	Focused Inspections	413

17	BY THE NUMBERS: Measuring Usability in Practice	417
	Comparison Shopping	417
	Measured Quality	418
	User Interface Design Metrics	423
	Essential Usability Metrics Suite	426
	Metrics in Practice	441
18	TEST SCORES: Laboratory and Field Testing of Usability	443
	History Testing	443
	Testing, One, Two	444
	Test Protocol	450
	Testing Tactics	456
	Why Test, Why Not	460
SECTION VI ORGANIZING AND MANAGING THE PROCESS		463
19	CODE AND YOU'RE DONE: Implementing Interfaces	465
	Objects and Interfaces	465
	Accelerated Development	472
	Visual Development of Visual Designs	475
20	USING YOUR USERS: Users in the Development Process	481
	Use or Abuse of Users	481
	Requirements Dialogue, Requirements Dance	485
	Going to the Source	489
	Using Users	494
	Joint Essential Modeling	499
21	GETTING ORGANIZED: Usability in the Larger Context	511
	Organizational Units	511
	Standards and Style Guides	513
	Competing Constituencies	517
	Experts and Expertise	522
	Cultural Fit	524