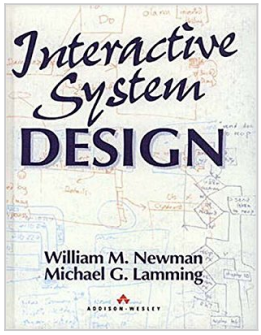

Interactive System Design

By Michael G. Lamming, William M. Newman

Interactive System Design By Michael G. Lamming, William M. Newman

"The human-computer interface of the future will not be perceived as the interface to a computer, but as a pervasive part of the environment we all inhabit. Our challenge is ...to convey to today's students of software design the underlying concepts and competencies that will not be bound either to the current GUI or to the specific details of its successors. Newman and Lamming offer a significant new step in that direction." --From the Foreword by Terry Winograd, Stanford University From multimedia workstations to hand-held PDAs, from VR headsets to networked PCs --the modern computer is predominantly interactive. Today's designers and software engineers need to adopt a user-centred approach to system design. Newman and Lamming present a comprehensive guide to modern design techniques using proven methods and realistic applications. Highlights of the book include: * A coherent framework for user-oriented design, covering all stages of development from requirements analysis to implementation and evaluation. * A rich set of examples based on real-world data from studies of air traffic control, police detective work, medical practice, telephone operators--and more.* Two in-depth case studies illustrating methods of usability analysis and an exercise in innovative design. * The authors' support pages for this book , including Instructor's Guide (new 2/27/96) System designers and software professionals will welcome the book's focus on methods which have been tried and tested in real applications. Students of human-computer interaction and software engineering will benefit from the practical guidelines it offers for effective interface design.

Interactive System Design

By Michael G. Lamming, William M. Newman

Interactive System Design By Michael G. Lamming, William M. Newman

"The human-computer interface of the future will not be perceived as the interface to a computer, but as a pervasive part of the environment we all inhabit. Our challenge is ...to convey to today's students of software design the underlying concepts and competencies that will not be bound either to the current GUI or to the specific details of its successors. Newman and Lamming offer a significant new step in that direction." -- From the Foreword by Terry Winograd, Stanford University From multimedia workstations to hand-held PDAs, from VR headsets to networked PCs --the modern computer is predominantly interactive. Today's designers and software engineers need to adopt a user-centred approach to system design. Newman and Lamming present a comprehensive guide to modern design techniques using proven methods and realistic applications. Highlights of the book include: * A coherent framework for user-oriented design, covering all stages of development from requirements analysis to implementation and evaluation. * A rich set of examples based on real-world data from studies of air traffic control, police detective work, medical practice, telephone operators--and more.* Two in-depth case studies illustrating methods of usability analysis and an exercise in innovative design. * The authors' support pages for this book , including Instructor's Guide (new 2/27/96) System designers and software professionals will welcome the book's focus on methods which have been tried and tested in real applications. Students of human-computer interaction and software engineering will benefit from the practical guidelines it offers for effective interface design.

Interactive System Design By Michael G. Lamming, William M. Newman Bibliography

- Sales Rank: #2459387 in Books
- Brand: Brand: Addison-Wesley
- Published on: 1995-05-10
- Ingredients: Example Ingredients
- Original language: English
- Number of items: 1
- Dimensions: 9.63" h x 1.25" w x 7.13" l, 2.10 pounds
- Binding: Hardcover
- 504 pages

Editorial Review

From the Inside Flap

Interacting with computer technology has become an essential part of everyday life. Computer-based interactive systems provide support to an ever widening range of human activities, some as simple as checking one's bank balance, others as complex as flying a passenger aircraft. It is becoming increasingly important that these systems should be usable, reliable and cost-effective, and this means ensuring that they are well designed.

We have written this book for those involved in the design of interactive computer systems and for those who are learning to become designers. In it we present a range of methods essential to solving the problems encountered in designing systems that are interactive by nature. Principal topics covered include problem definition, user study methods, systems analysis and design, requirements definition, prototyping, and a range of methods for designing and evaluating user interfaces.

A major objective of ours has been to provide readers with a framework that can link together all aspects of designing computer-based systems for human use. In other words, we are offering a means of integrating methods specific to interactive systems (such as user interface design) with methods of a more general nature such as systems analysis and requirements definition--methods normally treated as a part of software engineering. We do this to make sure that the sets of methods are not treated each in isolation, but are collectively given the attention they deserve.

We are trying here to address a serious problem facing the computer industry and the institutions that train the industry's future designers. Basically, system design methods have changed much less rapidly in the last two decades than the systems themselves and the ways these systems are used. It is still common for software to be designed with little or no attention to the human activity it supports. There is an overall tendency to treat interactive system design as a 'special case' of software design. These practices are inappropriate in an era when interactive systems have become so widespread, and when virtually every software system built has an interactive user interface of some kind. It would be better, we think, if interactive systems were treated as the norm, and so we offer here a way of treating them as such.Challenges

There are many challenges in writing a book of this kind. We will highlight three in particular, concerning the designer's need for self-sufficiency, underlying concepts and access to real-world data.

We want our readers to gain familiarity with a full range of methods, so that they can tackle entire problems in interactive system design on their own if necessary. Unless they can achieve this degree of self-sufficiency they will rely on the help of experts to deal with interactive aspects of system design. This is risky, because experts are not always around to help when interactive systems are designed. Furthermore, when experts are brought into the team the designer is always better off knowing something of their area of expertise. This is why we include topics that some might consider specialized, such as Chapter 8's usability analysis methods and Chapter 10's statistical techniques for checking the validity of evaluations.

We are also concerned to make sure that the reader has an adequate foundation of underlying concepts to carry out design. The challenge here has been to extend the theoretical base of system design. The largely mathematical body of theory supporting computer science is simply inadequate to deal with interactive situations. The solution to this problem is usually to draw on psychological theories, but these are of limited

use in collaborative situations where people interact with other people as well as with computer systems. We have therefore taken a multidisciplinary approach, offering a model of human performance based primarily on cognitive theory, but supplemented with concepts drawn from recent sociological and anthropological research.

Our third concern is to place system design in a context of real-world human activity. Success in design involves understanding the context into which the designed artefact will be introduced--the 'outer environment' of design identified by Herbert Simon (1981). By tradition, the context for which software is designed is simply another software system. An operating system, for example, is designed to be 'used' by application programs. When interactive systems are designed, however, the context of use is the human activity that the system will support, and this is much more difficult to understand. Indeed it is hard to find well-documented examples of real-life activity. We have nevertheless managed to collect together a number of examples, covering a broad spectrum from air traffic control to police work and collecting payment on unpaid bills. We use these examples to help the reader gain a broader and deeper understanding of the kinds of activities that interactive systems can support.

Our whole approach is thus centred around the need to design systems that support human activity. This is not greatly different from the 'user centred' approach advocated by other writers. The point we emphasize is the need to understand and support the tasks and processes that users perform; this is the way to ensure understanding and support of the users themselves.

Readership

The book's overall purpose is to assist readers in becoming effective system designers. It should be of value, therefore, to practitioners and students alike. As regards practitioners, our primary readership includes designers themselves, software engineers, system analysts, researchers and human factors engineers. We would expect parts of the book to appeal to a wider set of practitioners, however, including product managers, technical writers, marketeers, and indeed anyone closely associated with the development of interactive systems. The first seven chapters are oriented particularly towards this wider readership.

In the teaching context the book can serve a number of functions. Taken as a whole, it can serve as a text for courses on user interface design and human-computer interaction (HCI), for it covers all of the major topics generally included in such courses. The full set of topics, shown below in Figure 0.1, will typically be spread over an entire semester.

The first six or seven chapters can be used more widely, as shown in the darker shaded areas of the diagram. They can be used, for example, to introduce a course on software engineering, or (with Chapters 8 and 9) to teach a short course on interactive system design to non-majors. None of the material in the book is of a highly technical nature. The later chapters go into design methods in greater detail, however, and are likely to appeal more to students who are majoring in computer science or information science.Structure

The book is divided into four parts. It starts with a set of four chapters on foundational topics. These are followed by three chapters on system design and three more on evaluation. By this route the book covers the basic framework of design before turning to user interface design, which is covered in the final five chapters.

Each part of the book is organized around a set of methods and concepts. We think it best to recommend methods only if they have worked well in practice. Therefore we include methods that we personally have found useful in our design work. For the most part these methods have been developed elsewhere and have been thoroughly tried and tested by teachers and practitioners. Some of the concepts and representations are our own, however, because we could find nothing else capable of doing the job.

In Part I, for example, we begin with an introductory chapter and then focus on Problem definition, the topic of Chapter 2. We emphasize here the need to identify the design problem, and to distinguish this from the situation of concern that gives rise to designing the system. We introduce a number of key concepts, including usability factors and representations of human activities, and we suggest the use of a single-sentence problem statement to bring together the components of the design problem.

Also in Part I, Chapter 3 (the Human Virtual Machine) introduces a number of important theoretical concepts. Our primary purpose here is to offer a model of the workings of the 'machine' that performs human activities. Many aspects of human behaviour can be explained by the psychological theory of human information processing, but in order to cover behaviour in social settings this theory needs to be supplemented with findings drawn from anthropological and sociological research.

The remaining chapter in Part I, Chapter 4, provides an explanation of how the methods in the book link together and how the various design representations form the essential links.

Part II is concerned with system design, and in its initial Chapter 5 we therefore explain how to gather data about human activity in preparation for design. This leads into the important material of Chapter 6, systems analysis and design, which explains how observations of human activity are transformed into outline designs. It explains the nature of the analysis that takes place here, a topic that has rarely received much attention. This chapter is essential reading for anyone wanting to understand how analysis contributes to system design, and how interactive system design relates to software engineering. In the following Chapter 7 we cover requirements definition, and forge the link with software engineering more strongly. We explain the need to define, in a set of requirements, a usable functional form; we also explain why this can sometimes lead to writing voluminous sets of requirements.

Part III complements Part II; it explains how, in the course of design, interactive systems are evaluated in terms of their usability. Methods for checking usability are presented under two headings. Chapter 8 covers methods for evaluation 'on paper', before the design has been implemented. These methods are important because they can be applied early in the design and can sometimes save a great deal of unnecessary prototyping effort.

Chapter 9, prototyping and evaluation, plays a central role in Part III. It covers a number of methods for evaluating working prototypes, all of them following a general six-step evaluative process. This chapter introduces the concept of documenting the evaluation in terms of the performance requirements that the system is to meet, and it suggests a 'pro forma' notation for doing this. Chapters 8 and 9 may be considered for inclusion in a course organized around Parts I and II of the book. In Chapter 10 we explain how the results of prototype evaluation are checked for statistical significance, and offer a few simple methods.

In Part IV we cover user interface design. At the outset, Chapter 11 offers a number of different notations for describing user interfaces, and Chapter 12 explains the role of interaction styles and illustrates several different styles by means of a worked example. We place particular emphasis on conceptual design, by which an appropriate mental model is chosen for adoption by the user. We devote two chapters, 13 and 14, to this topic, presenting a range of design methods in Chapter 14. Chapter 15 covers the use of guidelines, explaining how to make use of sources of design knowledge and how to apply guidelines to answering questions of detailed design.

A number of case studies are included in the book. Some are used as linking material between the chapters; others illustrate methods, e.g., of evaluation and of choice of interaction style. Two detailed case studies are included at the ends of Parts III and IV respectively. The first describes a major exercise in analysing and evaluating the usability of a telephone operator's workstation, while the second covers a programme of

research leading to the development of a portable memory aid.Acknowledgements

We have been fortunate to receive a great deal of support and assistance during the four-year gestation period of this book. Our colleagues at the Rank Xerox Research Centre, Cambridge, have been a constant source of insight and learning, and the management of the Research Centre have remained patient and supportive despite our repeated inability to complete the writing task within the time allotted. We are particularly indebted to Bob Anderson, Manager of RXRC Cambridge, without whose support the book could not have been written. We are also most grateful to Richard Harper, who helped us to fashion the book's coverage of sociological aspects of system design.

The book benefited greatly from the reviews conducted along the way, and we would like to thank those who reviewed the book in its various stages of development: Bob Anderson, John Bates, Dik Bentley, Marge Eldridge, Stu Card, George Coulouris, Jean Dollimore, Paul Dourish, Jonathan Grudin, Richard Harper, Bill Hefley, Brian Inwood, Allan MacLean, Gary Perlman, Richard Rubinstein, Abigail Sellen and Ian Sommerville. We would like to thank Mike Flynn and Maureen Stone for their help in providing illustrations. We are most grateful to Mike Atwood, Wayne Gray and Bonnie John for their assistance with Case Study A. We were also greatly helped by the feedback we received from the students we taught from early versions of the book, at Cambridge University, Queen Mary Westfield College, the Joint Research Centre, Ispra, and ACM SIGCHI '92.

Many people assisted us with the mammoth task of getting the manuscript into its final form. We are especially grateful to Christine King at the Rank Xerox Research Centre, and to the Centre's administrative staff. We would also like to thank Simon Plumtree for his constant advice, and all other staff of Addison-Wesley who helped shepherd the book through editing and production. Finally, we express our deep gratitude to our families, whose tolerance, forebearance and support has meant so much to us while this book was taking shape.

--William Newman
--Mik Lamming

Cambridge, 1995 0201631628P04062001

From the Back Cover

0201631628B04062001

About the Author

About William Newman

William Newman is a Principal Scientist at the Rank Xerox Research Centre in Cambridge,England. As a member of research staff at Xerox PARC during the 1970s, he was closely involved in the early development of personal computing and desktop publishing. He co-authored, with Robert Sproull, the pioneering text on **Principles of Interactive Computer Graphics**. Since 1980 he has been a Visiting Professor at Queen Mary & Westfield College, London.

0201631628AB04062001

Users Review

From reader reviews:

Jasmine Myers:

Book is to be different for every grade. Book for children until eventually adult are different content. As it is known to us that book is very important usually. The book Interactive System Design had been making you to know about other understanding and of course you can take more information. It doesn't matter what advantages for you. The book Interactive System Design is not only giving you much more new information but also to become your friend when you sense bored. You can spend your current spend time to read your reserve. Try to make relationship with all the book Interactive System Design. You never experience lose out for everything if you read some books.

Kristen Self:

Hey guys, do you wants to finds a new book to see? May be the book with the title Interactive System Design suitable to you? The particular book was written by popular writer in this era. The particular book untitled Interactive System Designis one of several books that everyone read now. This particular book was inspired many men and women in the world. When you read this publication you will enter the new dimension that you ever know just before. The author explained their concept in the simple way, thus all of people can easily to comprehend the core of this book. This book will give you a lots of information about this world now. So you can see the represented of the world with this book.

James Furlow:

Are you kind of hectic person, only have 10 or perhaps 15 minute in your day time to upgrading your mind skill or thinking skill perhaps analytical thinking? Then you are receiving problem with the book as compared to can satisfy your small amount of time to read it because this all time you only find guide that need more time to be learn. Interactive System Design can be your answer given it can be read by a person who have those short spare time problems.

Gregory Kile:

On this era which is the greater man or woman or who has ability to do something more are more valuable than other. Do you want to become considered one of it? It is just simple approach to have that. What you need to do is just spending your time little but quite enough to get a look at some books. One of many books in the top list in your reading list will be Interactive System Design. This book that is certainly qualified as The Hungry Hills can get you closer in becoming precious person. By looking up and review this reserve you can get many advantages.

Read Interactive System Design By Michael G. Lamming, William M. Newman for online ebook

Interactive System Design By Michael G. Lamming, William M. Newman Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Interactive System Design By Michael G. Lamming, William M. Newman books to read online.

Online Interactive System Design By Michael G. Lamming, William M. Newman ebook PDF download

Interactive System Design By Michael G. Lamming, William M. Newman Doc

Interactive System Design By Michael G. Lamming, William M. Newman Mobipocket

Interactive System Design By Michael G. Lamming, William M. Newman EPub

FPHCX6GV3O1: Interactive System Design By Michael G. Lamming, William M. Newman