TAKING SOFTWARE DESIGN SERIOUSLY

Practical Techniques for Human–Computer Interaction Design

TAKING SOFTWARE DESIGN SERIOUSLY

Practical Techniques for Human–Computer Interaction Design

Edited by

John Karat

IBM Thomas J. Watson Research Center
Yorktown Heights, New York

ACADEMIC PRESS
Harcourt Brace Jovanovich
Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

Permissions for Chapter 6

Figure 1. Reprinted with permission of Academic Press Limited from a forthcoming paper in the *International Journal of Man–Machine Studies.*

Figures 2 and 3. Reprinted with permission of the ACM. Chew, Jane Carrasco, and Whiteside, John (eds.), Proceedings of CHI 1990 (Seattle, Washington, April 1-5, 1990), p. 315. Copyright © 1990, Association for Computing Machinery Inc.

Cover illustration by *Chris Eveland.*

Cover design by *Pascha Gerlinger.*

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24-28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data

Taking software design seriously: practical techniques for human
—computer interaction design / edited by John Karat.
p. cm.
Includes bibliographical reference and index.
ISBN 0-12-397710-X (alk. paper)
1. Computer software—Development. 2. Human—computer interaction.
I. Karat. John.
QA76.76.D47T35 1991
005.1—dc20 91-2550
CIP

Printed in the United States of America
91 92 93 94 9 8 7 6 5 4 3 2 1

Contributors

JOHN L. BENNETT, IBM Almaden Research, 650 Harry Road, K52/803, San Jose, California 95120 (63)

ROBERT E. BRAUDES, 11504 Arrowhead Court, Fredericksburg, Virginia 22401 (195)

JAMES A. CARTER, JR., Department of Computational Sciences, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, S7N 0W0 (209)

GEORGE CASADAY, Digital Equipment Corporation, 129 Parker Street, MSPK03-2/H26, Maynard, Massachusetts 01754 (45)

BERNARD J. CATTERALL, The HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, United Kingdom LE11 1RG (315)

ANDREW M. COHILL, Systems Development 0525, 1700 Pratt Drive, Blacksburg, Virginia 24060-6361 (95)

PATRICIA A. CRAIG, Microsoft Corporation, 1 Microsoft Way, Redmond, Washington 98052 (137)

TOM DAYTON, Bellcore, RRC-1H226, 444 Hoes Lane, Piscataway, New Jersey 08854 (21)

MARGARET D. GALER, The HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, United Kingdom LE11 1RG (315)

SUSAN HARKER, Department of Human Services, Loughborough University of Technology, Loughborough, Leicestershire, United Kingdom LE11 3TU (339)

H. REX HARTSON, Department of Computer Science, Virginia Tech., Blacksburg, Virginia, 24061 (157)

DEBORAH HIX, Department of Computer Science, Virginia Tech., Blacksburg, Virginia 24061 (157)

ELLIS HOROWITZ, Computer Science Department, University of Southern California, Los Angeles, California 90089 (257)

M. GREGORY JAMES, 211 Pinegate Circle #7, Chapel Hill, North Carolina 27514 (235)

JOHN KARAT, IBM T.J. Watson Research Center, Hawthorne, P.O. Box 704, Yorktown Heights, New York 10598 (63)

THOMAS R. LANNING, GTE Laboratories Incorporated, 40 Sylvan Road, Waltham, Massachusetts 0225[illegible] (127)

JOHN F. MCGREW, Room 2E-750JJ, Pacific Bell, 2600 Camino Ramon, San Ramon, California 94583 (287)

HAROLD H. MILLER-JACOBS, TASC, 55 Walkers Brook Drive, Reading, Massachusetts 01867 (273)

CHRIS ROUFF, Computer Science Department, University of Southern California, Los Angeles, California 90089 (257)

ANTONIO C. SIOCHI, Department of Physics and Computer Science, Christopher Newport College, Newport News, Virginia 23606 (157)

JOHN C. TANG, Sun Microsystems, 2550 Garcia Avenue, Mountain View, California 94043 (115)

BRONWEN C. TAYLOR, The HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, United Kingdom LE11 1RG (315)

DAVID A. WROBLEWSKI, MCC Human Interface Laboratory, 3500 West Balcones Center Drive, Austin, Texas 78727 (1)

Preface

What tools or techniques are useful in developing an effective computer system that people will find useful and usable in some domain? If one attempts to formulate a comprehensive software design methodology that has usability as a central focus, it is easy to be overwhelmed by the range of advice offered. High-level suggestions such as "know the users and their tasks" are of course important, but how does one go about doing this? Should we model users or tasks or both, and if so how? Are different techniques more appropriate to different domains? Since we never get the design right the first time, are there techniques that can support design that is truly iterative? How can we evaluate designs in progress and provide timely information to individuals who could use such information?

This book does not answer all of these questions, but it does offer a broad perspective on user-centered software design assembled from a number of experienced designers and researchers. The view expressed here is that the design of complex software systems involves many people exercising multiple skills and carrying out a variety of activities. If we studied a large sample of the software that is available for computer systems and asked, "How was it designed?" we would find that many methodologies have been used productively in the software design process. What is missing is some sense of how the various pieces and approaches fit in an overall picture of software design. Here we try to address design techniques, tools, and tricks, not by looking to describe a single design activity, but rather by describing a number of skills and methods that might be useful in providing a framework for deciding when to use what.

This book showcases a number of views on how to design user-centered software, with an emphasis on practical application. The chapters summarize results of a workshop held in April of 1990 in Seattle as part of the Association for Computing Machinery Conference on human factors in computing systems. As organizer, my intention was to bring together researchers and practitioners who represented a variety of views on the kinds of techniques that are useful in designing software systems that meet users' needs. In response to a general call for participation, speakers representing 22 position papers were invited to attend the workshop. While the actual workshop lasted only one day, discussions between participants have both preceded and continued after the event, and many of the chapters show the strong influence of these discussions.

I have organized the chapters in the book as follows. The book begins with several integrating views of software design. Wroblewski characterizes the design activity as craft and points out important implications of this view. Dayton describes an approach in which multiple design tools and techniques can be coordinated. Chapters by Casaday and by Karat and Bennett discuss frameworks for coordinating multiple perspectives on the system under design.

The second group of chapters discusses important individual talents or perspectives for successful design. The role of an information architect is described in the chapter by Cohill. Lanning addresses the role of eventual system users in the design of systems which will impact their work. In a case study, Tang covers some of the roles for social scientists as experienced observers of behavior, and Craig points out the importance of graphics design. These chapters do not speak to all of the skills required, but they do highlight several roles that the human-computer interaction field has not treated as central to the software design process.

The third group of chapters deals in some fashion with software design activities and the tools to support them. Siochi, Hix, and Hartson describe a technique for specifying user behavior in interacting with direct manipulation systems. Braudes describes a tool for examining conceptual models of systems that allows designers to carry out analyses early in the design. A very important topic in designing for effective human-computer interaction is the integration of user-centered approaches with more traditional software engineering approaches. While many of the chapters touch on such issues, three chapters in this book (those by Carter, by James, and by Rouff and Horowitz) provide detailed discussions of the topic and offer techniques for such integrated design. Miller-Jacobs focuses on the iterative nature of design and the importance of prototyping within the process. A technique that draws on analysis of graphs is described by McGrew. Chapters by Harker and by Catterall, Taylor, and Galer provide two perspectives resulting from the European Human Factors in Information Technology (HUFIT) project.

I wish to express my thanks to all of the people who have assisted in the completion of this book. Through their willingness to put their ideas forward carefully and then to listen to those of a number of others, each of the workshop participants helped make this a rewarding exercise. Particular thanks go to Tom Dayton who acted as workshop recorder, and who made significant contributions to the assembly and the review of the contents of the book. Additionally, I would like to thank members of the workshop who were not able to take part in this book: Meredith Bricken, Gary Klein, Allan MacLean, Larry Miller, Mark Notess, and Peter Polson. We all benefited from their discussions. Finally, I would like to thank SIGCHI for providing an environment for the workshop that led to this book.

John Karat

Contents

1 The Construction of Human-Computer Interfaces Considered as a Craft

DAVID A. WROBLEWSKI

MCC Human Interface Laboratory
Austin, Texas

I. Thesis

In this chapter I begin articulating a *craft perspective* on the construction of human-computer interfaces. The process of craft is ancient; the notion of software-as-craft has existed in the hallways of software houses since building software has been a profession. It has seldom, however, been the topic of discussion in the academic literature of computer science, psychology, or human-computer interaction. It is the premise of this chapter that the study of human-computer interfaces is rightfully and centrally concerned with the modern-day practice of craft.

ISBN 0-12-397710-X

I will focus on two categorical distinctions that disappear during craft processes. The first distinction is between design and manufacture, the second between tools and materials. After exploring these issues in the realm of design generally, in software development, and in the specialized subdomain of human-computer interface construction, I will examine some ideas for research, practice, and education that arise from taking seriously the notions of craft and craftsmanship.[1]

A. Design and Manufacture Unified

Let us begin by defining the word *craft*. The meaning of *craft* has changed since it was first used in the 16th century, when it referred to a forceful act, or an act of fraud, or cunning, or magic. The Oxford English Dictionary includes eight senses of *craft* describing such related but distinct notions as skillfulness in planning or acting; a device created out of such skill; an art, trade, or profession calling for special skill and knowledge; or the collective knowledge of such skill as embodied in its practitioners.

Among scientists and engineers, *craft* carries both positive and negative connotations. Negative connotations arise from the implication that its practitioners are either unaware of or incapable of articulating the principles motivating their designs, or unable to consistently reproduce the results of applying those principles. Positive connotations of *craft* come from the desirability of one-of-a-kind articles built with attention to detail, the ideal to which much engineering aspires.

For this chapter, I offer the following definition, derived from (Lucie-Smith, 1981): *A craft is any process that attempts to create a functional artifact without separating design from manufacture*. This definition provides two significant constraints. The first is that the result be a functional artifact. The construction of purely aesthetic artifacts does not qualify. This is a matter of degree, since all artifacts have some function, however small, even if it is only to evoke an aesthetic response. Painting comes close to the purely aesthetic end of the spectrum, for instance. Jewelry-making is slightly more a craft, since jewelry balances a small amount of functionality (it must be wearable in one of several ways and enhance the wearer's appearance) with a high degree of aesthetics (it must look nice.) A water pitcher thrown on the potter's wheel has significantly more functionality than jewelry (it must contain a liquid without leaking, it must afford lifting and pouring, and it must insulate its contents) and hence, we tend to consider pottery a craft.

---

[1]Throughout this discussion I will use the terms "craftsman" and "craftsmanship" to denote practitioners of either sex. I have retained these terms due to extensive usage in quoted material.

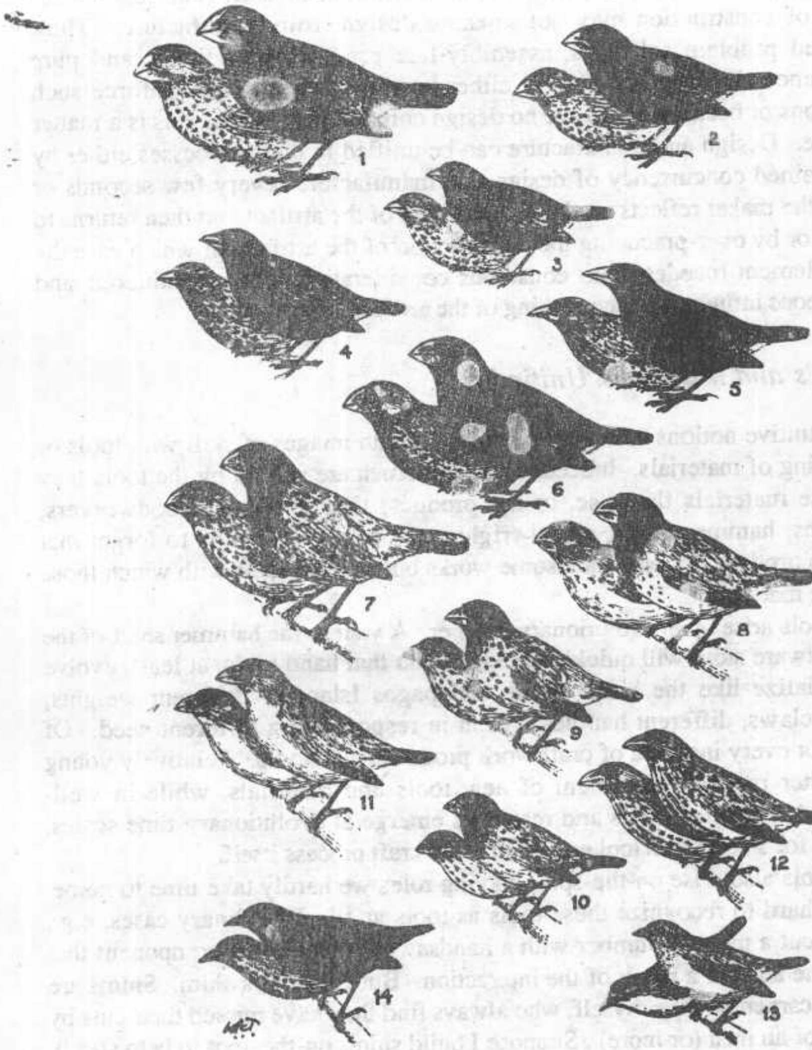

Figure 1. The 13 species of finches that Darwin observed in the Galapagos Islands, from Lack (1947). The most noticeable differences occur in the beaks, which vary due to their respective ecological niches. Reprinted with permission from Cambridge University Press.

The second constraint imposed by this definition of craft requires that the method of construction may not separate design from manufacture. Thus, routinized problem solutions, assembly-line production methods, and pure performance skills do not qualify either because they strongly enforce such distinctions or because they have no design component. Again, this is a matter of degree. Design and manufacture can be unified in craft processes either by small-grained concurrency of design and manufacture (every few seconds or minutes the maker reflects on the present form of the artifact and then returns to making) or by over-practicing the construction of the artifact, in which case the design element recedes from conscious consideration into a continuous and simultaneous influence on the making of the artifact.

B. Tools and Materials Unified

Our intuitive notions of craft are bound up with images of skill with tools or the working of materials. Indeed, most craftsmen are named by the tools they wield, the materials they use, or the products they produce: woodworkers, goldsmiths, hammersmiths, wheelwrights, and so on. We tend to forget that craftsmen produce not only handsome works but also the tools with which those works are made.

New tools arise in an evolutionary manner. A visit to the hammer shelf of the local hardware store will quickly convince you that hand tools, at least, evolve and specialize like the birds of the Galapagos Islands: different weights, different claws, different handles – each in response to a different need. Of course, not every instance of craft work produces new tools. Relatively young crafts foster rapid development of new tools and materials, while in well-established crafts new tools and materials emerge at evolutionary time scales. The agent for selection in tool evolution is the craft process itself.

New tools also arise on-the-spot, playing roles we hardly take time to name. It can be hard to recognize these tools as tools at all. In ordinary cases, e.g., when we cut a piece of lumber with a handsaw, the tool is the component that changes the least as a result of the interaction. But consider a shim. Shims are handy for carpenters like myself, who always find they have missed their cuts by an eighth of an inch (or more). Suppose I build shims on-the-spot to help steady some parts in preparation for assembly. Are the shims tools or materials? Their status is ambiguious precisely because it changes so quickly.

As you might expect, this categorical breakdown is complete in situations where the craftsman's trade involves working the very materials from which his tools are made. Dougherty and Keller (1982) discuss precisely this phenomenon in their observations of blacksmiths at work:

> Similarly, the smith is not constrained by a given inventory of tools, but is largely free to create new tools as the need arises. For example, tongs are manufactured to hold

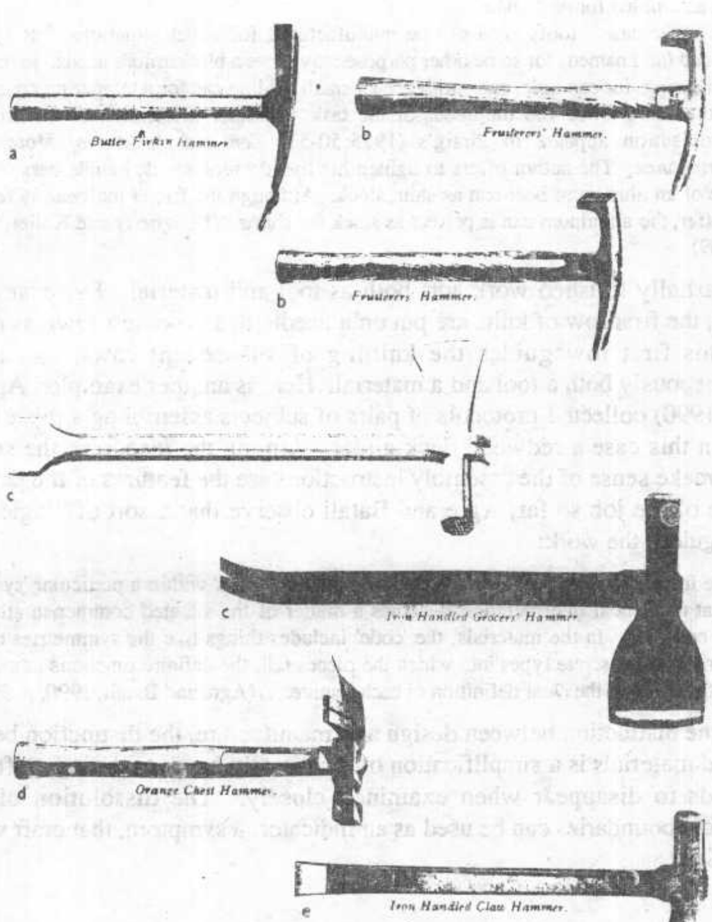

Figure 2. A selection of grocer's and warehouse hammers, taken from Salaman (1989). The most noticeable differences occur in the claws, which vary due to their respective ecological niches. Reprinted with permission from Harper Collins Publishing.

> standard stock endwise and sideways. If a special shape stock is used, or a standard size significantly modified, the blacksmith can reforge the jaws to create a new set of tongs. The discontinuities evident in a tool inventory at any given point in time are not perceived as fixed boundaries within which one must work, but as the arbitrary result of tools assembled for past tasks.
>
> In other cases, tools need not be manufactured for novel situations, but objects intended (and named) for some other purpose may serve a blacksmith's needs. In making a gun spring, for example, one smith used a small sardine can for a tempering container, and the can well served the needs of the task. Another illustration of this kind of improvisation appears in Pirsig's (1975:50-51) *Zen and the Art of Motorcycle Maintenance*. The author offers to tighten his friend's motorcycle handle bars using a piece of an aluminum beer can as shim stock. Although the friend indignantly refuses the offer, the aluminum can is perfect as stock for shims. (Dougherty and Keller, 1982, p. 769)

All partially finished work acts both as tool and material. For example, in knitting, the first row of knits are put on a needle in a process known as *casting on*. This first row guides the knitting of subsequent rows, and thus is simultaneuously both a tool and a material. Here is another example: Agre and Batali (1990) collected protocols of pairs of subjects assembling a piece of furniture, in this case a redwood deck glider. Among the resources the subjects used to make sense of the assembly instructions are the features of the parts and the state of the job so far; Agre and Batali observe that a sort of "logic of the object" guides the work:

> ...the instructions and materials both participate in 'codes' within a particular 'system', so that reciprocal interpretation becomes a matter of the situated commensuration of these codes.... In the materials, the 'code' includes things like the symmetries of the materials, the discrete types into which the pieces fall, the definite functions associated with these types, the clear definition of each feature.... (Agre and Batali, 1990, p. 53)

Like the distinction between design and manufacture, the distinction between tools and materials is a simplification of what really happens during craft work, and tends to disappear when examined closely. The dissolution of these categorical boundaries can be used as an indicator, a symptom, that craft work is taking place.

II. Evidence for a Craft Viewpoint

I have offered two categorical distinctions – design versus manufacture and tool versus material – whose boundaries dissolve during craft work. In this section I will argue in steps of increasing specificity that design in general, in the construction of computer software, and in the specialized subdomain of HCI construction, are crafts.

A. Design Viewed as a Craft

An extensive literature exists on the design process and its role in past and present society. It is generally agreed that the designers we see today can trace their professional roots to the craftsmen of antiquity. Lucie-Smith (1981) identified three stages in the history of craft, each of which has gradually led to the next. In the first stage, all objects were made through craft, whether for utilitarian, ritual, or decorative purposes. At this stage, all design was accomplished by craft evolution, in which repeated practice produced better and better artifacts. After the European Renaissance, a distinction arose between craft and fine art; artists were presumably concerned with aesthetic artifacts while craftsmen focused on the utilitarian. Finally, the Industrial Revolution removed from the realm of craft those products for which one constant design could be specified, and construction could be relegated to a machine. Where once there had been only the craftsman, there were now the artist, the craftsman, and the designer.

Mass production forced the separation of design from manufacture, but some objects were industrialized more readily that others. Coins, for example, have nearly always been mass-produced since they are useless unless there are many, virtually identical, copies. For more functional objects, the separation of design from manufacture has taken much longer, when it has happened at all. Jones (1981) discusses the account of George Sturt, who apprenticed to a wagon-maker in the late nineteenth century, and who, in 1923, published a book entitled "The Wheelwright's Shop" describing the craft of wagon-making as he learned it. The wagons he described were well-integrated into their work environment but had been designed only through craft evolution. The form of each wagon was influenced by many things: the type of material it was to haul, its dimensions, the type of soil through which the wagon would travel, and the kinds of horses that would pull it. The resulting features were nontrivial. They included dished wheels and a waist to reduce the turning radius of the cart. The dished wheels in particular have taken on an iconic quality in the literature of design methods, representing the dual qualities of exceptional design success accompanied by ignorance of the principles governing the design:

> There is probably no one "true" reason for the dishing of the cartwheels but rather a great number of interrelated advantages. This is very characteristic of the craft-based design process. After many generations of evolution the end product becomes a totally integrated response to the problem. Thus if any part is altered the complete system may fail in several ways. Such a process served extremely well when the problem remained stable over many years.... Should the problem suddenly change however the vernacular or craft process is unlikely to yield suitable results. If Sturt could not understand the principles involved in cartwheel dishing how would he have responded to the challenge of designing a wheel for a steam-engine or even a modern petrol-driven vehicle with pneumatic tires? (Lawson, 1980, p. 14)

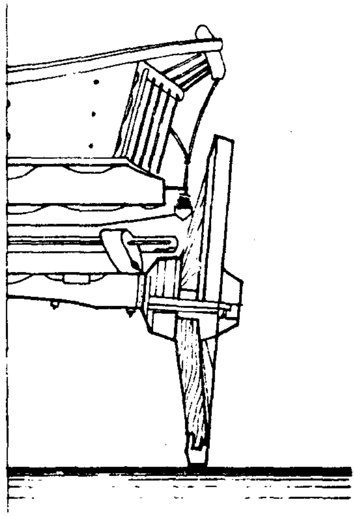

Figure 3. Sturt's dished wagon-wheel. Reprinted with permission from Eastview Editions, Inc.

How was the wagon-maker able to confidently begin a new wagon with no blueprint? There were two reasons for confidence. First, he had made similar wagons; an architectural precedent had been established and proven by experience. Second, the wagon-maker trusted the craft process as a way of thinking - situated and responsive instead of removed and analytical. Although the successful completion of the wagon could not be guaranteed, the craftsman could respond to emerging problems continuously throughout the construction process and take steps to repair them. The result could be quite effective: "Farm-waggons had been adapted, through ages, so very closely to their own environment that, to understanding eyes, they really looked almost like living organisms." (Jones, 1988, p. 220)

As Lawson points out, craft evolution begins to fail when the environment changes faster than the forces of craft evolution can cope. Tools, materials, situations of use, and performance requirements all change in the modern world much faster than in the wheelwright's shop of the 1860's. Without a deeper, explicit knowledge of the principles behind successful designs, the craftsman has no way to prioritize alternative responses to change. We are left with a model of the craftsman toiling to produce artifacts whose design he neither understands nor controls.

Some thinking on design has come full circle, however. Schön (1983) exam-