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

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.

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.

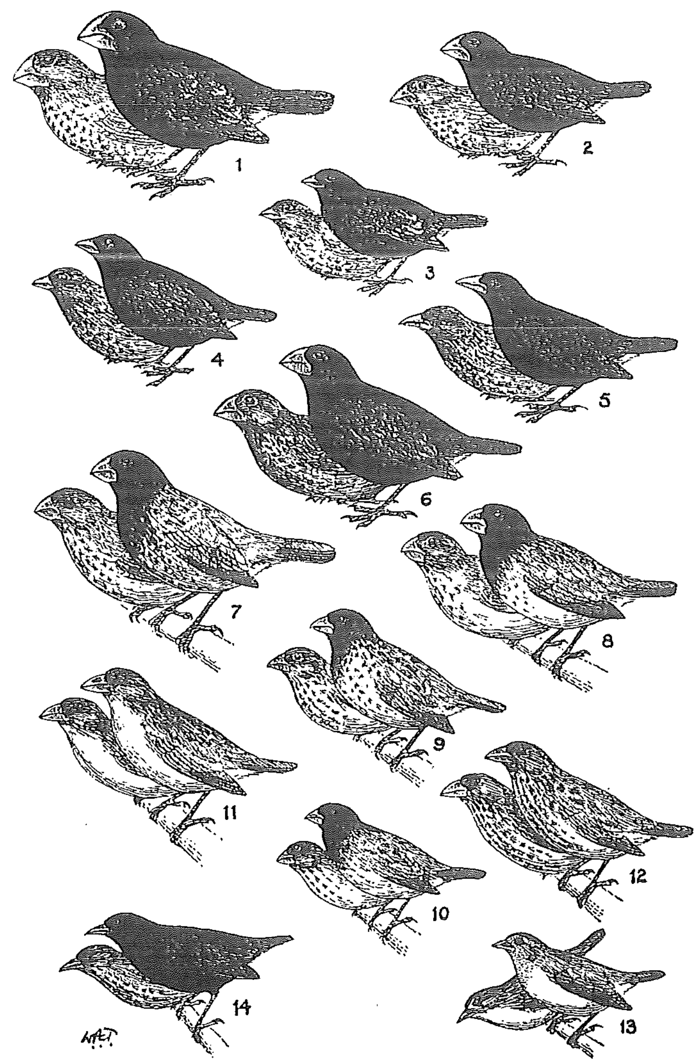

Figure 1. The 13 species of finches that Darwin observed in the Galapagos Islands, from Lack (1947). The most noticeable differences occur in the beaks, which vary due to their respective ecological niches. Reprinted with permission from Cambridge University Press.

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

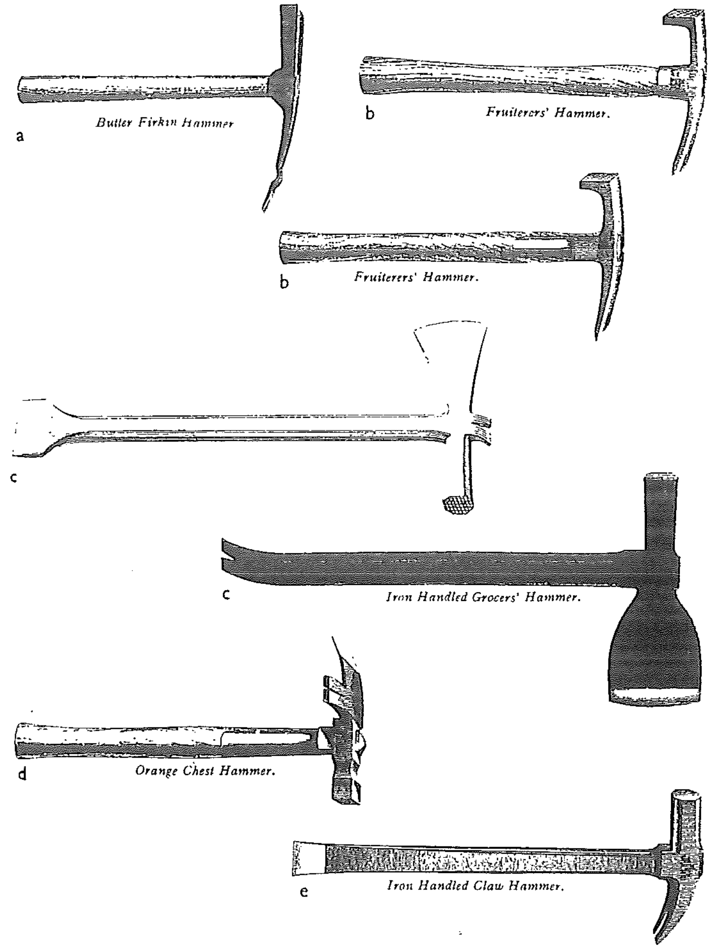

Figure 2. A selection of grocer's and warehouse hammers, taken from Salaman (1989). The most noticeable differences occur in the claws, which vary due to their respective ecological niches. Reprinted with permission from Harper Collins Publishing.

> 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)

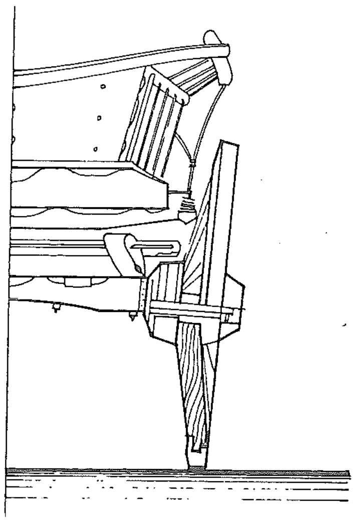

Figure 3. Sturt's dished wagon-wheel. Reprinted with permission from Eastview Editions, Inc.

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) examines the complementary crisis existing today: the technical knowledge or scientific basis for most professional design does not guide the actual practice of design as an investigation into the unique properties of each individual problem. Nor does the act of designing itself appear to be adequately described as the rigorous application of this technical knowledge, however complete, to some predetermined goals. Instead, significant other activities take place before technical knowledge can be effectively applied. Before problem *solving* can begin, claims Schön, problem *setting* must occur. In problem setting, we as designers "interactively *name* the things to which we will attend and *frame* the context in which we will attend to them." And even after the problem has been set, the designer engages in a sort of dialogue with the problem situation to uncover the most appropriate means and goals, rather than through the rigorous application of technical knowledge toward predefined ends. The model of design that Schön suggests is called "reflection-in-action":

> A designer makes things.... Typically, his making process is complex. There are more variables... than can be represented in a finite model. Because of this complexity, the designer's moves tend, happily or unhappily, to produce consequences other than those intended. When this happens, the designer may take account of the unintended changes he has made in the situation by forming new appreciations and understandings and by making new moves. He shapes the situation, in accordance with his initial appreciation of it, the situation "talks back," and he responds to the situation's back-talk.... In answer to the situation's back-talk, the designer reflects-in-action on the construction of the problem, the strategies of action, or the model of the phenomena, which have been implicit in his moves. (Schön, 1983, p. 78)

Similarly, Rittel and Webber (1973) introduced the notion of *wicked problems*, that is, problems whose formulation is neccesarily vague and whose optimal solution cannot practically be found or measured. Design is clearly one of these wicked problems. Rittel and Webber claim that each wicked problem is "essentially unique": similar problems and their solutions can guide the solution of a wicked problem but cannot guarantee it, for a critical difference may always be discovered that invalidates the analogy between the old problem and the new.

We are left with a model of the modern designer as one who struggles to change systems whose behavioral properties are often too complex to fully understand, who must resort to *in situ* experimentation merely to uncover those properties, and whose store of technical knowledge is useful only when the playing pieces have been named, the problem has been framed, and the major design decisions have been made. This model is, after all, not so different from the craftsman of old.

B. Software Construction Viewed as a Craft

In this section, I will make the case that software construction is sometimes practiced as a craft. I will discuss evidence that the design and manufacture of software is, at least sometimes, unified. Also, I will make the case that the categorical boundaries between tools and materials completely disappear during the practice of software construction.

Until recently, it has been a tacitly accepted goal of software engineering research to drive the craft out of software development to make it as predictable, schedulable, and manageable as possible. The resulting models were based on an analogy to the manufacture of physical goods and were intended primarily to benefit software management, not as an accurate model of how such efforts usually proceed (Curtis, Krasner, Shen, and Iscoe, 1987). In fact, the meaning of the word "manufacture" requires a nontrivial mapping into the construction of software, since the work of producing the program is done in programming, and the duplication of the media for distribution is trivial in comparison.

Not surprisingly, recent work has brought this goal into question. Guindon and Curtis (1988) summarize the results of several studies of software designers by describing such activities as being an opportunistic discovery process. They stress that the parameters of the problem emerge during attempts to build solutions; some attempts fail but reveal previously unrecognized constraints in the solution space.

There is an abundance of anecdotal evidence that some software construction efforts are craft-like. A telling comment comes from Jones (1988):

> The more I see of software designing the more I notice resemblance not to design in other fields but to craftsmanship. In each the designing, if such it can be called, is done by the maker, and there is much fitting, adjusting, adapting of existing designs, and much collaboration, with little chance of a bird's eye view, such as the drawing board affords, of how the whole thing is organized, though, in craft evolution, if not in software, the results have the appearance of natural organisms or of exceptionally well integrated designs. But there is an important difference: software is increasingly made by modifying the actual material of previous pieces of software, as a building may be altered for a new use, whereas a waggon-maker, for instance, modifies the form, but does not reuse the material, in making each small step in the gradual evolution of his product. (Jones, 1988, p. 219)

The reality is that building software has been successfully practiced both as a craft and as an engineering endeavor. When a reliable specification of the problem is available, the problem can be tackled by conventional programming languages and methodologies. When the specification is difficult to state or unreliable, however, only more craft-like efforts have succeeded (Sheil, 1984). When problems do crop up in larger, apparently well-specified projects, it is the individual "superconceptualizer" who steps in to repair the design (Curtis, Krasner, Shen, and Iscoe, 1987).

The categorical breakdown between tools and materials is complete in the crafting of software. As Jones points out, programs have the unique property of serving as the raw materials for new programs. In our lab, for instance, we routinely make new programs by copying and editing old ones. Subroutine libraries (either tools or materials, depending on your perspective) often arise from the systematic extraction of code developed for a particular application, whose generality and value to other programmers is recognized. Additionally, new software tools arise when building or testing an application. For instance, imagine building a simple program to monitor network traffic to help debug latency and collision problems for a distributed computing application. The intended result is the distributed computing application, but the network traffic analyzer might be noticed by someone else who then uses it for entirely different purposes. The software craftsman works in a virtual toolsmith's shop, where all materials can become tools, and all tools are raw materials.

C. *Human-Computer Interface Construction Viewed as a Craft*

In the construction of human-computer interfaces, a craft perspective is not only in evidence but inevitable. Several recent attempts to make psychological theory more relevant to HCI practice suggest the same model of the HCI design problem: to change, by the introduction of a new artifact or the modification of an existing one, an ecology of tasks and artifacts whose behavioral properties are so complex as to be difficult or impossible to predict with accuracy or detail (Carroll, 1990; Landauer, 1990; Suchman, 1987; Winograd and Flores, 1986; Wixon, Holtzblatt, and Knox, 1990).

This problem is often experienced by the designer as a problem of "details." As Carroll (1989) notes, every interface is a theory of how a task ought to be supported. A major feature in the interface is a claim that such functionality is central to completing the task. The absence of a feature is also a claim that the corresponding subtask is incidental enough to be considered a detail. Sometimes a strange switch in figure and ground happens when the tool is fielded for evaluation. The details begin to appear more central to the task than originally thought while the major features are ignored, misused, or cursed by the user. For instance, Erikson (1989) is devoted almost entirely to the discussion of such unexpected problems in several CSCW systems, including difficulties with seating arrangements, the unexpected desirability of ambiguity in office communication, and the inadvertent usurping of a corporate status hierarchy. Most or all the systems considered had strong theoretical underpinnings and were built by accomplished interface designers. Nevertheless, when placed in realistic settings, the designers witnessed the Necker-cube flip of details and features. This is reminiscent of Rittel and Webber's "wicked problems": similar problems do not necessarily lead to similar solutions. Details can have disproportionate effects.

Experiences such as those described in Erikson's paper, although still largely undocumented in the HCI literature, are pervasive and lead to statements like this, from Wixon, Holtzblatt, and Knox (1990):

> Principles alone are not enough in the design of artifacts. Design involves not only a broad architecture but a myriad of details which are not obvious until observed in practice. (Wixon, Holtzblatt, and Knox, 1990, p. 333)

Carroll (1990) agrees:

> An important body of work in current HCI stresses that designed artifacts cannot be understood apart from the situations in which they are used....These investigators emphasize that even small details can have substantial effects on the usefulness of an artifact, that such details emerge from the situation of use, and hence can be neither predicted outside of that context nor meaningfully abstracted from it. (Carroll, 1990, p. 322)

Schön (1983) argues that the problem is one of expectations. He has seen the same phenomenon in other design disciplines, such as architecture and urban planning. His characterization of the design process as a "reflective conversation with the materials of the situation" exactly fits the *in situ* experimentation that has been suggested in HCI to help better predict results.

However, don't misunderstand this argument: It is *not* about the presence or absence of a basis for informed design decisions in practice. On the contrary, I assume all conscientious practitioners are well-versed in the state of theory in HCI, psychology, and computer science. Instead, it is an observation that building human-computer interfaces involves applying the relevant knowledge in a complex problem-solving context to systems of tasks and artifacts too complex to be completely understood. In practice, the distinctions between design and implementation are necessarily so blurred that the construction of human-computer interfaces can surely be considered a craft.

III. Implications for HCI Research and Practice

If the construction of human-computer interfaces is a craft, then we might rethink some current strategies for researching, managing, and teaching it. Most obviously, any project management decision that separates HCI design from implementation risks the disuse or ineffective use of the resulting interface. However, we might also explore more constructive alternatives that, though they break with tradition in this field and within the software professions generally, are worth considering.

First, we can acknowledge that craftsmen value facile tools and responsive materials above all else and make the development of such tools a priority in HCI research. Facile tools free craftsmen from worrying about the mechanics of building and focus attention on the properties of the task/artifact ecology to be changed. In most current interface design systems, building human-computer interfaces is a complex, tedious task characterized by clumsy tools, catastrophic

consequences resulting from minor errors, and attention to microscopic implementation tactics rather than macroscopic design strategies. Techniques for evolving interface-building tools as a result of individual design episodes are primitive or nonexistent.

I am suggesting here that more facile tools and responsive materials *alone* can bring about the next revolution in human-computer interfaces. One advantage of this approach is that it can proceed without a comprehensive model of the design process. Facile tools are facile no matter what your theory of design is, even if it is nonexistent, just as good musical instruments don't inherently commit to a particular school of music. Its main drawback is the same, however. It is possible that we would never build a theory of interface design by concentrating exclusively on building tools and materials to support the practice.

A. *Researcher as Articulate Craftsman*

My second suggestion is to recast the researcher's role to be consistent with his membership in a community of craftsmen. What is it like to do research in HCI construction? How about glass-blowing? Pottery? Authorship? The traditional answer has been to investigate and describe the physical and logical processes at work within the work materials, or the psychological processes that guide human performance in the appreciation or use of the craft product. Unfortunately, much information generated this way is not used in practice because it is not posed in a form useful to making decisions in the context of crafting actual products.

There are alternatives. Another role we could play is that of the *articulate craftsman*. In this role, the researcher reports the forces that shape crafted artifacts as a result of undertaking the craft activities, in a form meaningful to the practice context, though not necessarily useful in traditional analytic or quantitative techniques. In particular, sets of design trade-offs, which I will call *design economies*[2], seem to arise from and strongly guide work in the practice context. Sometimes these design economies are instantiations of more general principles or conservation laws (for example, the well-known space vs. speed trade-off in computer programs), and other times they are quite particular to the design situation, and their generality is unclear (for example Wroblewski, 1987, describes a problem-specific design economy found in a graph unification algorithm.)

Such design economies do exist in human-computer interface construction. Norman (1983) poses several, and attempts a quantitative analysis of one, called "information versus time," which manifests itself in a messaging system as a

---

[2]Also called "exchanges" in (Minsky, 1987).

problem of choosing how much screen space to devote to permanently visible menus, and how much to leave for user workspace. Another trade-off, made famous in (Norman, 1988) is "knowledge in the head versus knowledge in the world." Even more recently, Grudin (1989) and Grudin and Norman (1990) have examined the trade-offs between ease of learning (consistency in the interface) and ease of use (inconsistent, but optimized interfaces.) Other research has taken the approach of studying finished artifacts in order to glean the extant design economies from them, in a process called "claims extraction" (Carroll, 1990).

B. Researcher as Craft-Methodologist

We can also undertake the role of *craft-methodologist*, by studying craft activities in a variety of domains, ranging from almost purely aesthetic crafts, such as authorship and glassblowing, to the production of more functional artifacts in such bastions of mature design as mechanical engineering, electrical engineering, and architecture. The benefits of such a course of inquiry are straightforward but profound, clarifying and refining our models of the craft process, our understanding of the nature of facility in tools, and extending our models of how design knowledge is formed and used in the practice context.

Perhaps you have a hard time imagining that, for instance, authors of fiction struggle with the same problems of craftsmanship as designers of human-computer interfaces. Consider then the following passage from John Gardner, a highly regarded writing teacher and author, discussing what he calls "the fictional process," with eerie similarily to the craft processes we have discussed so far:

> It was once a fairly common assumption among writers and literary critics that what fiction ought to do is tell the truth about things, or, as Poe says somewhere, express our intuitions about reality. Viewed in this way, fiction is a kind of instrument for coming to understanding. But we can see that there are problems to be solved if that view is to be defended. The realist says to us: "Show me, by a process of exact imitation, what it's like for a thirteen-year-old girl when she falls painfully, faintingly in love." And he folds his arms, smug in the conviction that *he* can do just that. But questions dismay us. Shall we tell the truth in short, clipped sentences, or long smooth gracefully ones? Shall we tell it using short vowels and hard consonants or long vowels and soft consonants?... A common answer at the present time is that that is the question the serious writer spends his whole life trying to work out by means of the only kind of thinking he trusts; that is, the fictional process. (Gardner, 1983, pp. 37-38)

Or another from John Steinbeck:

> Although it must be a thousand years ago that I sat in a class on story-writing at Stanford. I remember the experience very clearly. I was bright-eyed and bushy-brained and prepared to absorb the secret formula for writing good short stories, even great short stories. This illusion was canceled very quickly. The only way to write a good short story, we were told, is to write a good short story. Only after it is written can it be taken apart to see how it was done. (Steinbeck, 1989, p. 216)

In this case, for instance, the craft-methodologist can begin to see a theme of trust in the investigative properties of craftsmanship that cuts across domains of expertise. Other parallels surely hold.

C. Teaching HCI Construction: Apprenticeship Learning

In Terry Winograd's closing address at the 1990 ACM CHI conference, he asked the question "what can we teach about human-computer interaction?" It is an especially troublesome question if HCI construction is a craft, for teaching craft has been problematic. The traditional method of teaching a craft is through apprenticeship, and contemporary apprenticeship is alive and well in many fields, including medicine, architecture, and even software engineering. What varies is the degree to which the apprenticeship is a formal part of the education process and who pays for it.

There are two sets of processes involved in most apprenticeship systems. The first, and one we will not consider here, concerns the socialization and social control of the apprentice (Graves, 1989). The second is the notion that learning craft skills requires individual coaching, since significant elements of the craft cannot be well-communicated via classroom lectures, texts, or other nonparticipatory techniques. After studying Sturt's account of the wheelwright's shop, for instance, Jones concludes

> It may be necessary to read the whole of Sturt's book, and perhaps to serve as an apprentice oneself, if one is to understand more fully how the craftsman's blend of know-how and ignorance can produce works that a scientist would find hard to explain and in which the artistic eye can perceive a high level of formal organization. (Jones, 1981, p. 19)

Schön considers this issue at length. After examining examples of "professional artistry" in urban planning, architecture, and psychiatric therapy, he concludes

> Perhaps, then, learning *all* forms of professional artistry depends, at least in part, on conditions similar to those created in studios and conservatories: freedom to learn by doing in a setting relatively low in risk, with access to coaches who initiate students into the "traditions of the calling" and help them, by "the right kind of telling," to see on their own behalf and in their own way what they need most to see. We ought, then, to study the experience of learning by doing and the artistry of good coaching. We should base our study on the working assumption that both processes are intelligent and – within limits to be discovered – intelligible. And we ought to search for examples whenever we can find them – in the dual curricula of the schools, the apprenticeship and practicums that aspiring practitioners find or create for themselves, and the deviant traditions of studio and conservatory. (Schön, 1987, p. 17)

It is also instructive to consider that many human-interface designers have been making apprenticeship a part of the normal interface design process for quite some time. This idea is that to effectively design a human-computer interface, one must become an expert in the task domain. How is this to be done? Of course, by apprenticeship to those domain experts available. Several researchers have begun to formulate design methodologies that capitalize on this apprenticeship model, such as the participatory design methodology articulated by Ehn (1988), or the contextual design techniques of Wixon, Holtzblatt, and Knox (1990).

D. Teaching HCI Construction: Use of Paradigmatic Examples

Another method of teaching craftsmanship can be called "teaching by paradigmatic examples."[3] This technique rests on the assumption that access to a repertoire of interfaces could provide the learner with useful landmarks in the design space. These landmarks become the initial passes of the design process, and subsequent design iterations may take the designer arbitrarily far away from the starting point. Choosing the proper starting point is dealt with by analysis of the task domain and matching to the repertoire. If several examples apply along different dimensions, then the learner chooses the relevant properties of each example and uses it in guiding the new design where appropriate. When no example from the repertoire applies, the designer must find other means to guide the process.

In order for this technique to work, several problems must be addressed. First, one must be able to choose those members of the repertoire most applicable to the new problem. A comparison of the design economies at work within the new problem and within the examples in the repertoire could fill this gap, for an important measure of similarity must be the primary design economies at work within that problem space. Another problem with this approach is that the final form of an artifact does not capture design decisions that were tried but failed (Jones, 1981). Clearly, the absence of a design feature can sometimes be cause for concluding it doesn't work, but not always. One response is to include negative examples in the student's repertoire to provide landmarks to steer away from. Again, these landmarks require a level of analysis to emphasize the design economies at work and why they failed. Recent techniques to capture the design rationale of the artifact during the design process, such as Issue-Based Information Systems (McCall, 1986) and computer-based critics (Fischer and Morch, 1988), begin to address these issues.

[3]Thanks to Michael Williams of IntelliCorp™ for these ideas and access to his own paradigmatic example of using these ideas to teach design skills in the slides for his presentation "The Craft of Building Intelligent User Interfaces."

IV. Summary

In this chapter, I have considered the notion that building human-computer interfaces is a craft, with all the advantages and disadvantages that that entails. My main points follow:

1. A craft is any process that attempts to create a functional artifact without separating design from manufacture.
2. Significant by-products of the craft process are new tools and materials as well as the intended artifact. The distinction between tools and materials begins to dissolve when viewed this way.
3. Creating software is sometimes, but not always, a craft. The degree of variability in practice is due to the availability of a reliable specification.
4. Creating a human-computer interface is usually, and perhaps always, a craft, because of the investigative nature of each designing.
5. Productive HCI research can take the form of facile tools and responsive materials, articulate craftsmanship, or craft-methodology.
6. We can begin to codify operational design knowledge by searching for and articulating design economies at work within individual interfaces.
7. Two approaches to teaching HCI as a craft are apprenticeship and exposure to paradigmatic examples.

The future of software practice and HCI construction belongs to those who take their craft most seriously and least respect the bounds of tradition, be they craft or science. I have argued that it is instructive to study other craft and design professions, but we must also be mindful of the limits to such analogies. Fundamentally, the materials shape the craft. Computer programs are unlike any other material, and the form of craftsmanship in software will surely be unique.

V. Acknowledgments

Many members of the MCC research community reviewed, supported, and improved this paper through hours of discussion, especially Will Hill, Tim McCandless, Loren Terveen, Elaine Rich, and Jim Hollan. Jean McKendree was particularly helpful in providing insight on the issues of apprenticeship and exemplar learning. During the workshop, Alan MacLean was especially enthusiastic, and afterward, Tom Dayton and John Karat were gracious and accommodating in letting me submit drafts late, and contributed several useful suggestions and lots of encouragement. Don Norman, Gerhard Fischer, Jonathan Grudin, Terry Winograd, Mike Williams, and Tom Erickson provided useful feedback, suggestions, and encouragement. Finally, Patricia Stuhr of the Ohio State University Art Education Department helped guide me into the literature of craft and craft education.

VI. References

Agre, P. and Batali, J. (1990). "Collaborative Instruction Use in Furniture Assembly." In Nigel Gilbert (Ed.) *AAAI Workshop on Complex Systems, Ethnomethodology, and Interaction Analysis*, (pp. 41-56).

Carroll, J. M. (1989). "Artifact as Theory-Nexus: Hermeneutics meets theory-based design." *Proceedings of the CHI'89 ACM Conference on Human Factors in Computing Systems*. (pp. 7-14). New York: ACM.

Carroll, J. M. (1990). "Infinite Detail and Emulation in an Ontologically Minimized HCI." *Proceedings of the CHI'90 ACM Conference on Human Factors in Computing Systems*. (pp. 321-327). New York: ACM.

Curtis, B., Krasner, H., Shen, V., and Iscoe, N. (1987). "On Building Software Process Models Under the Lamppost." *Proceedings of the IEEE 9th International Conference on Software Engineering*, (pp. 96-103).

Dougherty, J. and Keller, C. (1982). "Taskonomy: A Practical Approach to Knowledge Structures." *American Ethnologist*, Volume 9, Number 4, 763-774.

Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts*. Stockholm: Arbetslivscentrum.

Erickson, T. (1989). "Interfaces for Cooperative Work: An Eclectic Look at CSCW '88." *SIGCHI Bulletin*, Volume 21, Number 1, 56-64.

Fischer, G. and Morch, A. (1988). "Crack: A Critiquing Approach to Cooperative Kitchen Design." *Proceedings of the International Conference on Intelligent Tutoring Systems*. (pp. 176-185). New York: ACM.

Gardner, J. (1983). *The Art of Fiction: Notes on Craft for Young Writers*. New York: Vintage Books.

Graves, B. (1989). "Informal Aspects of Apprenticeship in Selected American Occupations." In Michael W. Coy (Ed.), *Apprenticeship: From Theory to Method and Back Again*, (pp. 51-64). Albany, NY: State University of New York Press.

Grudin, J. (1989). "The Case Against User Interface Consistency." *Communications of the ACM*, Volume 32, 1164-1173.

Grudin, J. and Norman, D. A. (1990). "Lessons for System Design from the Study of Natural Language Evolution." Manuscript submitted for publication to CHI'91.

Guindon, R. and Curtis, B. (1988). "Control of Cognitive Processes During Design: What Tools Would Support Software Designers?" *Proceedings of the CHI'88 ACM Conference on Human Factors in Computing Systems*. (pp. 263-268). New York: ACM.

Jones, J. C. (1981). *Design Methods: Seeds of Human Futures*. New York: John Wiley and Sons.

Jones, J. C. (1988). "Softecnica." In John Thackara (Ed.), *Design After Modernism: Beyond the Object*, (pp. 216-226). New York: Thames and Hudson Publishing.

Lack, D. (1947). *Darwin's Finches: An Essay on the General Biological Theory of Evolution*. Cambridge, England: Cambridge University Press.

Landauer, T. (1990). "Let's Get Real: A Position Paper on the Role of Cognitive Theory in the Design of Humanly Useful and Usable Systems." Bellcore Technical Memorandum TM-ARH-016841. Also to appear in J. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge, England: Cambridge University Press.

Lawson, B. (1980). *How Designers Think*. New Jersey: Eastview Editions, Inc.

Lucie-Smith, E. (1981). *The Story of Craft*. New York: Van Nostrand Reinhold Company.

McCall, R. (1986). "Issue-Serve Systems: A Descriptive Theory of Design." *Design Methods and Theories*, Volume 20, Number 3, 443-458.

Minsky, M. (1987). "Form and Content in Computer Science." In *ACM Turing Award Lectures: The First Twenty Years, 1966-1985*. (pp. 219-242). New York: ACM. [Originally published as Minsky, M. (1969). "Form and Content in Computer Science", *Journal of the ACM*, Volume 17, Number 2, (pp. 197-215.)]

Norman, D. (1983). "Design Principles for Human-Computer Interfaces." *Proceedings of the CHI'83 ACM Conference on Human Factors in Computing Systems*. (pp. 1-10). New York: ACM.

Norman, D. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

Pirsig, R. (1975). *Zen and the Art of Motorcycle Maintenance*. New York: Bantam.

Rittel, H. and Webber, M. (1973). "Dilemmas in a General Theory of Planning." *Policy Sciences*, Volume 4, Number 2, 155-169.

Salaman, R. A. (1989). *Dictionary of Woodworking Tools*. Newport, Connecticut: The Taunton Press, Inc.

Schön, D. (1983). *The Reflective Practitioner*. New York: Basic Books.

Schön, D. (1987). *Educating the Reflective Practitioner*. San Francisco: Josey-Bass.

Sheil, B.A. (1984). "Power Tools for Programmers." In D. Barstow, H. Shrobe, and E. Sandewall (Eds.), *Interactive Programming Environments*. (pp. 19-30). New York: McGraw-Hill.

Steinbeck, J. (1989). "John Steinbeck." In B. Strickland (Ed.), *On Being a Writer*. (pp. 216-218) Ohio: Writer's Digest Books. (Originally published 1963.)

Suchman, L. (1987). *Plans and Situated Actions*. Cambridge: Cambridge University Press.

Winograd, T. (1990). "What Can We Teach About Human-Computer Interaction?" *Proceedings of the CHI'90 ACM Conference on Human Factors in Computing Systems*. (pp. 443-449). New York: ACM.

Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition*. New York: Addison-Wesley Publishing.

Wixon, D., Holtzblatt, K., and Knox, S. (1990). "Contextual Design: An Emergent View of System Design." *Proceedings of the CHI'90 ACM Conference on Human Factors in Computing Systems*. (pp. 329-336). New York: ACM.

Wroblewski, D. (1987). "Nondestructive Graph Unification." *Proceedings of the 6th National Conference on Artificial Intelligence*. (pp. 582-587). Seattle: AAAI Press.

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