

[Back to Library Catalogue](#)

From Human Factors to Human Actors

The Role of Psychology and Human-Computer Interaction Studies in Systems Design

Liam J. Bannon

*Book Chapter in Greenbaum, J. & Kyng, M. (Eds.) (1991)
Design at work.: Cooperative Design of Computer Systems.
Hillsdale: Lawrence Erlbaum Associates, pp. 25-44.*

Man is one of the best general-purpose computers available and if one designs for man as a moron, one ends up with a system that requires a genius to maintain it. Thus we are not suggesting that we take man out of the system, but we are suggesting that he be properly employed in terms of both his abilities and limitations .

(E. Llewellyn Thomas, Design and Planning, 1965)

I believe that there needs to be a better understanding among researchers, and many system designers too, about the "users" of computer systems and the settings in which they work. Part of the problem resides in an implicit view of ordinary people which, if surfaced, would seem to treat people as, at worst, idiots who must be shielded from the machine, or at best, as simply sets of elementary processes or "factors" that can be studied in isolation in the laboratory. Although psychology, particularly as represented by the field of human factors (HF), or ergonomics, has had a long tradition of contributing to computer systems design and implementation it has often neglected vitally important issues such as the underlying values of the people involved and their motivation in the work setting.

Understanding people as *actors* in situations, with a set of skills and shared practices based on work experience with others, requires us to seek new ways of understanding the relationship between people, technology, work requirements and organizational constraints in work settings. Studying such a multi-faceted and multi-layered issue requires us to go beyond more traditional controlled laboratory studies that are the hallmark of experimental human-computer interaction (HCI) studies. In this Chapter I recount some experiences, give some background on the field of HCI and point out some problems. I suggest some alternative perspectives and directions for more fruitful research on, or rather with, people in work settings that may assist in the design of more usable and useful computer systems.

A True Story

I once worked in the human factors group of a large organization and was asked to build an interface to a particular system. I was given the manuals from the previous system, some reports by investigators of problems noted with this system, and some specifications of the new hardware and software capabilities and the new functional specifications. Fine as far as it went. My wish was to go out and talk with users of the current system, to get a better "feel" of the situation they were working in, and subsequently to interact with them in an iterative design process in order to make a usable interface. To my astonishment, I was informed that not only could I not proceed in this fashion, but - ultimate irony - I was not allowed, for political organizational reasons, to meet with even a single user throughout this process!! I complained, but being in a very junior position was over-ruled, so I spent some months on what to me seemed an insane task, developing a logically coherent, though naive, mapping of tasks onto a menu-based interface based solely on the paper specifications I had been given. A report was duly completed and forwarded to another level in the organization, and I believe much of it was actually implemented, to my amazement. Given my deskbound limited knowledge of the actual use situation, I am sure that many of my "logically" elegant design solutions turned out to be hopelessly complex and/or inappropriate in the real work situation.

I do not want to claim that such a situation is the norm, but I do claim that this true scenario is not nearly as far from today's reality as one would hope. Granted, the fault here does not lie solely at the feet of the HCI community, though within this organization, that group must take some of the blame for not insisting on an iterative design process that insisted on *active* user participation. Partly this is due to a rather limited perspective on what users can offer in the design process, compounded by organizational politics that sometimes makes contact between designers and users a much more difficult process than one would suppose. Readers might keep this little story in mind as they read the rest of the text, as the link between some of the attitudes and practices of designers and researchers discussed here and the above story should emerge. In the next Section I take a short tour over the landscape of human factors and user interface design and note a few pertinent examples of terminology that betrays a certain limited view of the people that we aim to design for, and suggest some alternatives.

A Question of Perspective(s)

Language does not simply symbolize a situation or object which is already there in advance - it makes possible the existence or appearance of that situation or object, for it is part of the mechanism whereby that situation or object is created.

(G.H. Mead, *Mind, Self and Society*, 1934)

Replacing Human Factors with Human Actors

The terms used in a discipline often give a clue as to how members "see" that field. Once certain distinctions are accepted, and become part of the standard vocabulary, these terms can become a barrier to the reality that lies outside. For this reason, it is a useful exercise from time to time to re-examine the language used to express our understanding of the world. I have chosen to mention the terms *human factors* and *human actors* in the title of the Chapter as I believe it highlights a difference in the perception of the person, the former connoting a passive, fragmented, de-personalised, un-motivated individual, the latter connoting an active, controlling one. I claim that traditional human factors work, although it undoubtedly has merit, and has produced many improvements to existing technological systems, is often too limited in scope with respect to its view

of the person. Within the HF approach, the human is often reduced to being another system component, with certain characteristics, such as limited attention span, faulty memory, etc. that need to be factored into the design equation for the overall human - machine system. This form of piecemeal analysis of the person as a set of components de-emphasizes important issues in work design. Individual motivation, membership in a community of workers, and the importance of the setting in determining human action are just a few of the issues that are neglected. People are more than a sum of parts, be they information-processing subsystems or physiological systems, they have a set of values, goals and beliefs about life and work. This more encompassing interpretation of the *human factors* field is not signalled by the name, nor the bulk of studies appearing under this topic in the journals so designated. By using the term "human actors" emphasis is placed on the person as an autonomous agent that has the capacity to regulate and coordinate his or her behaviour, rather than simply being a passive element in a human-machine system. This change in terminology may help in adjusting one's perspective, emphasizing the wholistic nature of the person acting in a setting, in contrast to the view of the person as a set of information processing mechanisms that can be analysed in the same manner as the information processing mechanisms of the technology .

Re-thinking the Concept of "Users"

Another term, ubiquitous in articles about the HCI field, that deserves scrutiny is that of "users". This general term refers to all people who use a particular computer system, or application. It can be distinguished from the term "operators" in that the latter implies a greater involvement with the machine or system, presumably one where the person is more uniquely assigned to the device. The focus of the system design or human - computer interaction (HCI) research group is biased towards the technology, and the view of people is often simply as "users" of this piece of technology, and "naive users" at that! This can lead to problems. People may not know the technology, but they are not "naive" as to their work, rather it is the system designers that are "work naive"! There is nothing inherently wrong in taking this stance - talking of "naive users", though I prefer the less judgmental terms "casual" or "discretionary users" - for periods when designing computer applications, but there is a danger in thinking of people as *nothing but* "users". In fact, it is often still the case that computer users still need to make some modifications to the system in various ways, tailoring the system (see the Chapter by Henderson and Kyng) before it is truly usable, so that in a very real sense *users are designers* as well. Focusing on people simply as users can also blind us to the fact that the "users" view of the technology we are developing may be very different to that of the designer's. The user is often a worker that has a set of tasks to perform, and their use of the computer may only be one element necessary to the accomplishment of their work. We can become focused in design on one particular application and forget that the user has different needs to that of simply performing the set of system operations. As an example, [Whiteside and Wixon \(1987\)](#) note one case where a word processing clerk spent a lot of time manually counting the lines she had typed as she was paid by the line, and the particular system did not have a facility to do this (relatively straightforward) task! Often the user is involved in multi-tasking, not just on the computer, but also with co-workers, perhaps with clients through the phone system, and other sundry daily happenings. Neglecting this rich and complex set of "non-essential" but truly everyday work interactions can lead to unworkable or unwieldy systems (see [Cypher, 1986](#)). It is the ability to understand the user perspective, to be able see a problem from other than the system viewpoint, to be able to empathize and work with the users of the intended system that mark the good design team.

Users are not Idiots

While viewing computer users as "naive" is bad enough, viewing them as idiots is even worse! Apart from the value questions that arise, there are clear design implications if this faulty view of users is

implicitly adopted. Just because users do not understand how the machine works, or have difficulty with the system designer's terminology does not imply that they are stupid, as some developers apparently conclude - if we are to judge from the systems that are at times designed. The system design team should start out with the understanding that workers/users are competent practitioners, people with work tasks and relationships which need to be also taken into account in the design of systems, and with whom they must collaborate, in order to develop an appropriate computer system. The idea that we must design systems so that "any idiot can use them" needs to be subjected to close scrutiny. Taking this as a serious design goal can often result in systems that necessarily produce such stupid behavior (see [Bannon, 1986a](#), for further comment). In addition, as noted in the initial Chapter quotation, a consequence of trying to make such a system is that an incredible amount of "intelligence" must go into its initial design and maintenance. Taken to the extreme, we have the prospect of artificially intelligent systems operated by morons, an absurd scenario. Thankfully, in recent years this particular problem seems to have diminished, probably as a result of designers and others finally developing a better understanding of the user's perspective. A slightly less obnoxious version of this can be seen in the over-emphasis on "easy to learn" systems in HCI research. Certainly, there are some applications where we want a minimal interface that is easy to learn, but this is often not the case when dealing with systems that will be used by people in their everyday work, over long periods. Here, we need to pay attention to the capabilities of the system, and allow users greater flexibility and expressiveness in their use of the system.

Allow for Active Users

While focussing attention on the user may be a positive step, users are not simply passive objects that others must study and design for, as some accounts would have it. People are, or can become, *active agents*. They often wish to accomplish tasks, to understand what is going on, and are willing to jump ahead and explore the computer system on their own if, for example, the tutorial material is unclear or too pedantic. If the system does not give an explanation for its behaviour, the user will often try and make one up, in order to render the doings of the system comprehensible. People are always struggling to make sense of their world. Developing instruction sequences for users that are to be followed by rote, with inadequate explanation, fail to satisfy, or to work in most real situations. For example, in discussing LisaGuide, an on-line tutorial for the Lisa system, Carroll and Mazur (1986) note one user's reaction: "I'm getting impatient. I want to do something, not learn how to do everything". Understanding the needs of "active users" has been the subject of investigation by psychologists, with Carroll and his colleagues especially prominent (see [Carroll and Rosson, 1987](#), for a synopsis of this work). Here are the words of an office worker, a "user", that I interviewed which captures her wish to understand, to learn what's going on, that we should be aware of, and support, in system design .

"People have to know how to understand, they have to be able to rationalize things out, to work things out, in a logical sequence. If they don't understand something, or how it works, then they have to go back to either just learning it by rote, or by asking, or by just making the mistake, and going back, and asking, and then correcting. But if you understand the system a little bit then sometimes you can think it outand if you can reason it out then it stays with you longer - it's easier to understand, to work with. But if you are just learning piecemeal, then you can't. If you are just learning by rote.....someone tells you, this is the way to do it, then you can memorize it, but you'll never fully understand it, and [never] be able to expand from there."

To summarize this Section, we can see on reflection that some of the concepts and terms currently common in the system design and HCI research fields contain at least an implicit, if not explicit perspective on "users" of computer systems that I would claim is inaccurate and liable to lead to

faulty design decisions. Likewise within the theory and research areas of HCI, it has led to a concentration on issues that can be studied in the laboratory at the expense of other concerns that might be more crucial in actual work settings. But exactly what is this HCI field, and how did it come about? In the next Section, I give a brief account of the origins of the field, before proceeding to discuss the current status of HCI - some results, limitations, and possible future directions.

The Field of Human Factors and Human-Computer Interaction: Some Background

This Section gives a short account of the changing relationship between people and the machines with which they work, and the contribution of human factors studies in these changes. It is evident that unless a process is totally automated there will still be a need for some human intervention. How to divide the work between the human and machine becomes an important task. Traditionally, in human factors, this "allocation of functions" task proceeds according to sets of guide-lines about general human and machine performance capacities. We also need to design an "interface" between the machine and the person - knobs, dials, displays, controls, so that they can interact at some level. In the early days of this century, the focus was on how to get a machine to perform to the required functionality. The human component was often reduced to yet another "cog" in the overall machine - one that was, at times, more expendable than the machine. The slow improvement in the conditions and nature of work for people operating with machines came about not simply for altruistic reasons, but partly also in the search for greater efficiency of operation. Poorly trained or motivated operatives and poorly designed equipment could cause breakdowns in the smooth functioning of the industrial process. Better designed controls, and tasks that reduced both mental and physical strain on the operator allowed for improved performance of the human-machine system. The attempt to fit the machine to the skills and limitations (physical and mental) of the operator developed into a new field of applied study early this century. This new field was called *human factors engineering* in North-America, and very similar efforts in Europe became known as *ergonomics*, from the Greek words *ergon*, meaning work, and *nomos*, which can mean law or knowledge. People working in this field usually had a background in either behavioral science or in industrial engineering. Physiologists and medical practitioners also contributed to the understanding of human capabilities and limitations in work settings - effects of stress, psycho-motor ability, perceptual acuity, mental processing workloads, etc. Out of this observational and occasionally experimental discipline arose a body of knowledge that could be useful in the design of complex human-machine systems (see [Van Cott & Kinkade, 1972](#)).

In the early days of this century the problem was to build machine systems to do something useful. The focus was on the machine performance, as I noted above. Machines were not being modified substantially every year, so *ease of learning* of the system was not a high priority. Training was a one-shot process. People could be trained to perform whatever operations were required, and subsequently serve as *operators* of the machine. As computing developed the separation between operating and programming developed. Focus was still however on the functionality of the software rather than its ease of use. So programmers spent years learning arcane languages to communicate with the system. The user community started to change from one that was focused either intrinsically on studying the properties of the machines themselves (computer scientists) or application programmers that mediated between user needs and the computer system. A growing number of people using computers could be classified as *discretionary* users, people who saw themselves as having a job or profession that was not primarily geared to the computing medium itself, but used it

directly, themselves, as a tool in their everyday work. However, these people were frustrated by the difficulty of learning how to program computers for their work. With the advent of the personal computer over a decade ago it was obvious that their widespread acceptance would become more dependent on their ease of learning and use. Not everyone was willing to learn something akin to IBM's notorious Job Control Language (JCL) in order to operate a personal computer!

The field of human-computer interaction (HCI) emerged in the early eighties partly as a response to these changing conditions. It was linked to, but somewhat distinct from, its human factors progenitor in both its eclectic make-up and its more theoretical bias. The older field, somewhat dismissively referred to as "knobs and dials" psychology, was seen as lacking in theoretical motivation by cognitive scientists. What was required was a better *cognitive coupling* between the human and the new universal machine, the computer, and not simply better designed surface characteristics of displays. Software engineers were also involved, as they were experimenting with the design of highly interactive interfaces, and concerned about how to conduct dialogues with users and how to present complex information effectively to users on graphic displays. Over the last decade the area of human-computer interaction has grown enormously, both within academic research environments and corporate research laboratories. This commercial concern was a major impetus for human-computer interaction studies, and "ease-of-use" and "user-friendliness" have become advertisements for particular computer systems.

Beyond current conceptions of HCI

Despite the legitimate advances that have been made in various arenas of human computer interaction (see [Shneiderman \(1987\)](#) for a collation of some of this material), there has been serious criticism of the field for its lack of relevance to practitioners in systems design. Despite the widespread interest, there is no clear set of principles that has emerged from this work. The experience of certain designers has been loosely codified, various long lists of design guide-lines are available, and a large number of evaluations of existing systems have been produced, but the attempt to place this applied science on a more rigorous footing has been difficult. [Gray & Atwood \(1988\)](#) in a review of a recent collection of papers ([Carroll, 1987](#)) noted the lack of any examples of developed systems in the papers and the general lack of contact of the work with "real world" design situations. They explicitly state that the sceptical designer will not be convinced of the relevance of the cognitive sciences to the design of better human-computer systems based on this work. Within this collection, a discussion section by [Whiteside and Wixon \(1987\)](#) makes a number of pointed observations about the limitations of cognitive theory in its application to everyday design situations.

If there is not too much in the way of theory that is directly relevant, can we utilize some of the sophisticated methods and techniques used in psychology to analyse user behavior, so that designers could find out how users perform on different versions of a system, or different prototypes, and not have to rely on intuition? Standard lab experimentation is too limited and costly and time-consuming. What is needed are quick and dirty methods that can give rapid feedback to designers about the utility and usability of their products. This is happening in practice currently, in the work on *usability* (see below). Design involves making many tradeoffs, and in many instances empirical data can assist in determining the appropriate choices. Such tasks as setting up small in workplace empirical studies, noting reactions, user preferences, taking verbal protocols from subjects to see how they view the application, etc., are all investigative methods familiar to psychologists that can be useful in an iterative design process.

Cognitive psychologists have also studied particular issues, for example, we now know a lot about how people learn to use word processors, about the kinds of errors they make on different systems, about the mental models they attempt to construct of systems, but the application of such findings to new situations is not always obvious. From the perspective of the designer, the work to date can highlight some pertinent issues, but we are still impoverished in our search for new ways of thinking about and developing systems. The hoped for contribution of HCI to the design of completely novel interfaces has not materialised to date, though some believe that it may yet happen, given a shift in priorities among psychological researchers (Carroll, 1989).

I believe that there are a number of limitations in much of the current work on HCI that needs to be remedied in order for the field to be more useful to designers in practical situations. In the following paragraphs I suggest some changes in direction within the field of HCI that might assist in bridging the gap between theory, experiment, systems design and the actual work setting.

From Product to Process in Research and Design

By this I mean that more attention needs to be paid to the *process* of design, to working with users in all stages of design, to see the iterative nature of design, and the changing conception of what one is designing as a result of the process itself. This is in contrast to a view of design that proceeds from a set of fixed requirements without iteration, and without involvement of the users. This change in orientation has been evident for some time in much of the systems design work within the Scandinavian tradition (see the Chapter by Ehn and Kyng). It is also evident in the work of [Jones \(1988\)](#) and [Floyd \(1987\)](#) coming from industrial design and software engineering backgrounds, respectively, and by [Bannon and Bødker \(1989\)](#) in the field of HCI.

From Individuals to Groups

The majority of HCI studies to date take as their focus the individual user working on a computer system. This research focus totally neglects the importance of coordination and cooperation between work processes that is necessary in many work situations. Indeed, here again the applied field has been more astute than the theoretical, for system designers have been aware of this coordinated aspect of work activity and have tried to support it from early on, albeit rather crudely. For example, much of the Office Automation work in the seventies attempted to model work flows, but in too rigid a manner, thereby not allowing the human "components" in the overall system enough flexibility to make the system work. With a better understanding of how work gets accomplished coming from research such as [Suchman and Wynn \(1984\)](#), designers have a better "model" from which to build. The word model is in quotes here because of course, what such research shows is that a strict model of human action in most work situations is not possible or appropriate, rather, human action is driven by the concrete situation that exists at any moment and is constantly changing. This implies that we should *support office workers in their activities*, rather than building *office automation systems*. Extending the focus of concern from the human-computer dyad to larger groups of people and machines engaged in collaborative tasks is an important area for research in the next period. The quick growth of this field, labelled Computer Support for Cooperative Work (CSCW), attests to its importance.

From the Laboratory to the Workplace

Much of the early research done in the HCI field was confined to rather small controlled

experiments, with the presumption that the findings could be generalised to other settings. Examples of such studies were those done on command naming conventions (see [Barnard & Grudin, 1988](#) for a review of this research). It has become increasingly apparent that such studies suffer from a variety of problems that limit their usefulness in any practical setting. Firstly, by the time these studies are done the technology often makes the original concerns outdated. Witness the disappearance of line-oriented editors from commercial systems just as a "scientific" understanding of how people used them was being developed. [Newell and Card \(1985\)](#) refer to this problem as the race between the tortoise of cumulative science and the hare of intuitive design! Also important contextual cues for the accomplishment of tasks were often omitted in this transfer from the real world to the laboratory, and so the results of the lab studies became difficult to apply elsewhere. Increasingly, attention is shifting to in situ studies, in an effort to "hold in" the complexity of the real world situations, and a variety of observational techniques are being employed to capture activities, especially video. We can see an increasing focus on the concept of "usability" among the research community - whether people can and do actually use the resulting systems designed for them. From a design perspective, this means that we need to have a prototype or test system for users to experience in order to get information on the usability of the resulting system. There are many examples from recent large development projects that point out the importance of empirical methods for getting feedback on design decisions from users. For example, on the path-breaking Xerox Star system, both more formal experiments ([Bewley, Roberts, Schroit, & Verplank, 1983](#)) and more informal studies were used extensively. On the Apple Lisa, the designers note ([BYTE Interview, 1983](#)) "Another thing we have done is user tests - taking our ideas and bringing in naive users and sitting them down and seeing what their impressions are. This has caused some changes, and I think that's all shown in the quality." Having psychologists on the design team can thus assist in the planning, conduct and evaluation of such studies, whether or not detailed manipulations and data analyses are planned.

From Novices to Experts

The majority of experimental studies in HCI focus on first time learners of computer systems or applications. Typically performance is monitored for the first hour or two on the system. Exceptionally, perhaps use of an application is observed over a few days, but rarely for periods longer, such as weeks, never mind years! This has been partly due to the ease of obtaining "naive" subjects for experiments from subject-pools at Universities and Employment Centers. While granting that there is some need for studying such users, the paucity of studies that are concerned with the process of development of expertise with a computer application is remarkable. The issue is not simply that expert performance needs further examination, but that we need to pay greater attention to how users become skilled in the use of the computer application. What obstacles or incentives are there within the system to encourage the growth of competence? Additional issues relate to the difference in system learning and use between freshman University students and particular work groups with their own already established set of work practices which may hinder or support learning and development of competence on the computer system.

From Analysis to Design

Early human factors work tended to focus on evaluation of existing systems, and analysis of features that had been found in the use situation to be good or bad, from the point of view of the user. However the concern of people in HCI now is how to build better artifacts, so we don't just want to know about systems after they have been built, we want to know how we should build them in the first place, and even what we should build. HCI should be a design science: "design is where the

action is", to quote a memorable phrase of Allen Newell. So the question is how can HCI contribute to the design of more usable and hopefully useful artifacts? [Newell and Card \(1985\)](#) argue for the importance of giving designers approximate calculational models of the person performing a task for use in making design decisions about the human-computer interface. Many groups are currently active in such "user modelling", looking at the structure, content and dynamics of individual user cognition at the interface. Much work in the area continues the GOMS (Goals, Operators, Methods, and Selection rules) model tradition of [Card, Moran, and Newell \(1983\)](#), extending it in various ways. The practicality and utility of such low-level calculational models in actual design has been the subject of some debate ([Carroll & Campbell, 1986](#)). While some reject this approach as too narrow and time-consuming, others have pursued even more generalized user models as design support aids. For example, the concept of Programmable User Models (PUMs) ([Young, Green & Simon, 1989](#)) is based on a generalized architecture of human cognition. This approach to assisting system designers in understanding users needs appears to be unduly narrow. Rather than moving designers closer to actual users, such a device, if it existed, would seem to support the view that actual contact with real users was unnecessary. The designer could just program the PUM in order to understand the constraints on usability and potentially not have to observe actual users at all! The very vision of a PUM seems a rather abstract view of human activity in the world, and to imply a rather strange relationship (or rather lack of one!) between designers and users. So I would claim that both the GOMS work and that on PUMs support a human factor, rather than a human actor perspective discussed earlier. Such work can never replace prototyping and actual empirical user testing, although it might have a role at a certain stage in the design of a new system.

From User-Centered to User-Involved Design

As a first step towards focusing the attention of the designer on the needs of the user and away from a concentration on the hardware, there has been an increasing emphasis on taking a "user-centered" approach to design ([Norman & Draper, 1986](#)). Exactly what the term "user-centered system design" means, or how it can be achieved, is far from clear. In some cases it dissolves into platitudes such as "Know the User". Such kinds of general guidelines are of little use in practical situations of design, due to their lack of specificity. Gould (1988) discusses how important it is to have an early and continuous focus on users, to develop iterative designs, and to have early and continuous user testing. This is a large step in the right direction. Users here are being given a larger role in the design process, but it is still a relatively passive role. Although actual participation by users on the design team is mentioned, it does not figure prominently in this user-centered approach. A more radical departure from much current thinking within the mainstream HCI world is to look on users not simply as objects of study, but as active agents within the design process itself. This involvement of users in design is both a means for promoting democratization in the organizational change process, and is also a way to ensure that the resulting computer system adequately meets the needs of the users. It is this approach that has been the hallmark of many of the Scandinavian studies mentioned in this book and evident in the Chapters here.

From User Requirements Specifications to Iterative Prototyping

Over the years, it has been acknowledged that the standard way of representing user requirements in the functional requirements specification document is often inadequate, and the question has begun to be asked whether this is because of some problems in the way of doing the studies locally, or whether there is a fundamental problem with the very assumption that we can map out users needs and requirements successfully in advance through simple techniques of observation and interviewing. In Part II of this book, it is argued that users need to have the experience of being in the future use situation, or at least an approximation of it, in order to be able to give comments as to

the advantages and disadvantages of the proposed system. So, some form of mock-up or prototype, needs to be built in order to let users know what the future use situation might be like. In the construction and evaluation of such prototypes, the skills of the psychologist can be useful. This particular issue is a part of the first mentioned shift - from a product to a process orientation in both research and development of systems.

Conclusion: HCI in Systems Design

Some distinctions can be drawn between different kinds of systems development projects. In the development of products the need to have a user-friendly interface is very high, as it can determine the success of the product in the open marketplace. At the same time, the exact user group for the newly designed product is not clearly specified in advance. This can have implications for user-involvement in design as discussed throughout this book. However, it does not invalidate the need to work with future users, it just means that we cannot be as sure about satisfying their needs as fully as on a systems development project oriented towards a single application for a particular, specific group. We should be able to find representative users to at least allow us to iteratively prototype the design. Much of this book is dealing with another form of project, specific development projects, where the involvement of users is the least problematic, at least in theory - we know who they are! In some projects requirements specifications may be drawn up by third parties, other than the users and developers, and this can make it difficult, though not impossible to utilize some of the user-involvement techniques outlined in the second part of this book.

Understanding the user's needs, and the tasks performed by the user is basic to the system development process. However, it is a mistake to think that simply having a human factors person on the design team is by itself sufficient to ensure that the "human factor" has been adequately taken into account, in the sense that is being discussed here. Even in companies where there exist groups specifically targeted to give "human factors" advice on projects, one often finds that they have little influence over the design process, often regarded as "add-ons" by the engineering staff. This state of affairs has sometimes been unfortunately encouraged by the human factors personnel themselves, who often seem unwilling to really understand the complete project, or product, but focus on the narrow aspects that are adjudged to require human factors input. Such an attitude can also be forced on human factors people as a result of the structure and functioning of the organization, with project support from the HF unit only allowed at certain periods, and the physical separation of the HF department from that of the software engineering group.

Human factors, or ergonomics considerations often are incorporated into the design process simply as a set of specifications that the delivered system must adhere to. The actual work of the human factors personnel is seen as operator task analyses to be fed into these specifications and perhaps some interface re-touching near the end of the development cycle, when the system design has already been fixed. In general the role of these personnel has been seen as ancillary to the main task of building the system. The role of HF or HCI in systems design today should be a more fluid, pragmatic one, whose input is vital in discussing the initial capabilities of the system and required functionality, and persists in the development and evaluation of prototypes, and in final screen layout considerations.

What is being advocated here is an approach that, while acknowledging the contribution that different disciplines can make to the design process, ultimately depends upon the users themselves to articulate their requirements, with the system design team, composed of a variety of specialists,

acting in the capacity of consultants to the project. Design teams and users must be prepared to acknowledge each others competencies and to realize that effort must be made by both parties to develop a mutually agreed upon vocabulary of concepts that can be shared across the different groups that comprise the project. It is no easy task for different disciplines and work activities to accomplish this, and it is in this area that additional research would be valuable.

Some of the efforts in Scandinavia (see for example the papers in [Bjerknes et al, 1987](#)) on involving users in design provide a promising start towards the alternative systems design paradigm advocated here, at least for project systems development. Within such an approach, the starting point, and the end point of the design process is with the users themselves, from what they require, to how they evaluate the prototype, and the iterations that follow. Along the way, the services of a variety of disciplines may be required, not just the software engineer and the ergonomist, but perhaps also architects, sociologists and anthropologists. These disciplines should come together in the overall design process as required, and not dictated by some arbitrary flow model by which the system design gets handed around sequentially from one discipline to another. It is in the mutual interaction of these different perspectives, including that of the end users, focused on a particular design project, that good design may emanate.

Acknowledgements

Thanks to Niels Bjørn-Andersen, Susanne Bødker, Jonathan Grudin, Randy Trigg and the editors for helpful comments on earlier drafts.

References

- Bannon, L. (1986a) Issues in Design: Some Notes. In Norman, D. & Draper, S. (Eds.) User Centered System Design. London: Erlbaum, 1986.
- Bannon, L. (1986b) Helping Users Help Each Other. In Norman, D. & Draper, S. (Eds.) User Centered System Design. London: Erlbaum, 1986.
- Bannon, L. & Bødker, S. (1989) Beyond the Interface: Encountering Artifacts in Use. Dept. of Computer Science, Aarhus University, DAIMI PB-288, Oct. 1989. (To appear as a book chapter in J. Carroll (Ed.) Designing Interaction: Psychological Theory at the Human-Computer Interface).
- Bannon, L. & Schmidt, K. (1989) CSCW: Four Characters in Search of a Context. In Proceedings of First European Conference on Computer Support for Cooperative Work, London.
- Barnard, P. & Grudin, J. (1988) Command Names. In Helander, M. (Ed.) Handbook of Human-Computer Interaction. (Amsterdam: North-Holland, 1988).
- Bewley, W., Roberts, T., Schroit, D & Verplank, W. (1983) Human factors testing in the design of Xerox's 8010 "Star" office workstation. In Proceedings CHI 83 Conference on Human Factors in Computer Systems, ACM Press, 72-77.

- Bjerknes, G., Ehn, P. & Kyng, M. (1987) *Computers and Democracy: A Scandinavian Challenge*. Aldershot, UK: Avebury.
- Bjørn-Andersen, N. (1988) Are "human factors" human? *The Computer Journal*, 31, 5, 1988, 386-390.
- Brown, J.S. (1986) From Cognitive to Social Ergonomics and Beyond. In Norman, D. & Draper, S. (Eds.) *User Centered System Design*. London: Erlbaum, 1986.
- Bødker, S. (1989) A human activity approach to user interfaces. *Human Computer Interaction*, 4, 3, 171-195.
- BYTE Interview (1983) An Interview with Wayne Rosing, Bruce Daniels, and Larry Tesler. *BYTE*, Feb. 1983.
- Card, S., Moran, T. & Newell, A. (1983) *The Psychology of Human-Computer Interaction*. New Jersey: Erlbaum, 1983.
- Carroll, J.M. (Ed.) (1987) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. (Cambridge: Bradford/ MIT Press, 1987).
- Carroll, J.M. & Campbell, R.L. (1986) Softening Up Hard Science: Reply to Newell and Card. *Human Computer Interaction*, 1986, Vol. 2, 227-249.
- Carroll, J.M. & Rosson, M.B. (1987) The paradox of the active user. In Carroll, J.M. (Ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. (Cambridge: Bradford/ MIT Press, 1987).
- Cypher, A. (1986) The Structure of User's Activities. In Norman, D. & Draper, S. (Eds.) *User Centered System Design*. London: Erlbaum, 1986.
- Floyd, C. (1987) Outline of a paradigm change in software engineering. In Bjerknes et al., 1987 op. cit.
- Giedion, S. (1969) *Mechanization Takes Command*. (New York: W.W. Norton, 1969).
- Gray, W.D & Atwood, M.E. (1988) Review of *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction* by Carroll, J.M. (Ed.) (Cambridge: Bradford/ MIT Press, 1987) in *SIGCHI Bulletin*, October 1988, 88-91.
- Greif, I. (Ed.) (1988) *Computer-Supported Cooperative Work: A Book of Readings*. San Mateo, CA: Morgan Kaufmann, 1988.
- Helander, M. (Ed.) *Handbook of Human-Computer Interaction*. (Amsterdam: North-Holland, 1988).
- Jones, J.C. (1988) Softecnica. In Thackara, J. (Ed.) *Design After Modernism*. (London: Thames & Hudson) 216-226.
- Landauer, T. (1987) Relations between cognitive psychology and computer systems design. In Carroll, J.M. (Ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. (Cambridge: Bradford/ MIT Press, 1987).
- Newell, A. & Card, S.K. (1985) *The Prospects for Psychological Science in Human-Computer*

Interaction. *Human Computer Interaction*, Vol. 1, 209-242.

Newell, A. & Card, S.K. (1986) Straightening out softening up: Response to Carroll and Campbell. *Human Computer Interaction*, 2, 251-267.

Norman, D. & Draper, S. (Eds.) (1986) *User Centered System Design: New Perspectives on Human - Computer Interaction*. London: Erlbaum, 1986.

Reisner, P. (1987) Discussion:HCI - what it is and what research is needed. In Carroll, J.M. (Ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. (Cambridge: Bradford/MIT Press, 1987).

Shneiderman, B. (1987) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, Mass.: Addison-Wesley, 1987.

Suchman, L. & Wynn, E. (1984) Procedures and problems in the office. *Office: Technology and People*, 2, 2, 1984, 133-154.

Thomas, J. & Kellogg, W. (1989) Minimizing Ecological Gaps in User Interface Design. *IEEE Software*, January, 78-86.

Van Cott, H. & Kinkade, R. (Eds.) (1972) *Human Engineering Guide to Equipment Design* (rev. ed.) Washington: American Institute for Research.

Whiteside, J. & Wixon, D. (1987) Improving human-computer interaction - a quest for cognitive science (Discussion). In Carroll, J.M. (Ed.) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge: Bradford Press, 1987, pp. 337-352.

Young, R.M., Green, T. & Simon, T. (1989) Programmable User Models for predictive evaluation of user interface designs. In *Proceedings of CHI '89 Conference on Human Factors in Computing Systems*, New York: ACM Press.

[Back to Library Catalogue](#)