heiro, F., & Goguen, J. (1996). An object-oriented tool for tracing requirements. *IEEE Software*, 52–64.
anyi, L. (1978). So what's the point? *Semiotica, 25*(3–4), 208–224.
anyi, L. (1989). *Telling the American story.* Cambridge, MA: MIT Press.
anyi, M. (1967). *The tacit dimension.* London: Routledge & Kegan Paul.
inson, M., & Bannon, L. (1991). Questioning representations. In M. Robinson, L. Bannon, & K. Schmidt (Eds.), *Proceedings, Second European Conference on Computer-Supported Cooperative Work.* Dordrecht, The Netherlands: Kluwer.
ks, H. (1972). On the analyzability of stories by children. In J. Gumpertz & D. Hymes (Eds.), *Directions in sociolinguistics* (pp. 325–345). New York: Holt, Rinehart & Winston.
ks, H. (1974). An analysis of the course of a joke's telling in conversation. In R. Baumann & J. Scherzer (Eds.), *Explorations in the ethnography of speaking* (pp. 337–353). Cambridge, England: Cambridge University Press.
ks, H. (1992). *Lectures on conversation* (G. Jefferson, Ed.). Oxford: Basil Blackwell.
ks, H., Schegloff, E., & Jefferson, G. (1974). A simplest systematics of the organization of turn-taking in conversation. *Language, 504*, 696–735.
iller, D. (1994). From culture to information and back again: Commoditization as a route to knowledge. *Critical Studies in Mass Communication, 11*, 92–115.
nnon, C., & Weaver, W. (1964). *The mathematical theory of communication.* Urbana, IL: University of Illinois.
nmerville, I., Rodden, T., Sawyer, P., Bentley, R., & Twidale, M. (1993). Integrating ethnography into the requirements engineering process. In S. Fickas & A. Finkelstein (Eds.), *Requirements engineering '93* (pp. 165–173). New York: IEEE.
r, S. L. (1988). The structure of ill-structured solutions: Heterogeneous problem-solving, boundary objects and distributed artificial intelligence. In M. Huhns & L. Gasser (Eds.), *Distributed artificial intelligence* (Vol. 3, pp. 37–54). San Mateo, CA: Kauffmann.
hman, L. (1987). *Plans and situated actions: The problem of human–machine communication.* New York: Cambridge University Press.
hman, L., & Jordan, B. (1990). Interactional troubles in face-to-face survey interviews. *Journal of the American Statistical Association, 85*(409), 232–241.

3

Reconfiguring the Social Scientist: Shifting From Telling Designers What to Do to Getting More Involved

Yvonne Rogers
University of Sussex, England

The development of computer systems may be viewed as a struggle to configure the user. For example, Woolgar (1993) described an ethnographic study of a computer company, where the architects of a new computer system (i.e., the hardware and product engineers, project managers, salespersons, technical support, and others) spent considerable time discussing and arguing over who the new user of the system would be and what kinds of functionality would be appropriate for them. Moreover, Woolgar pointed out that it was the decisions made about the hypothetical user that subsequently became embodied in the new system. In this sense, the user has a *configured* relationship with the system, whereby only certain types of interaction are possible.

This chapter aims to extend the notion of the struggle to configure the user by presenting a critique of the struggle by the social scientist to contribute to the process of system design and use. In contrast to the various company architects, who are directly involved in developing computer systems that configure the user, social scientists (and others[1]) have largely channeled their efforts through more indirect means. In particular, a vast body of prescriptive knowledge has been constructed that is intended to equip designers more appropriately so they can better configure the user than through their existing repertoire of means. The main kinds of knowledge that have been offered include a melee of

[1]The term *social scientist* is used loosely here, and refers to academic researchers, ranging from cognitive psychologists to ethnomethodologists, who are involved in applying theories and methods, from their respective fields, to system design.

evaluation and design methods, user models, conceptual frameworks, design toolkits, design principles, and guidelines. The source of such knowledge has largely been drawn from the theoretical and methodological bases of the contributing disciplines, including cognitive psychology, management science, and sociology.

Implicit in the translation of such academic knowledge is an altruistic conviction that current system development is inadequate and can be helped out through input from the social sciences. An underlying assumption is that system designers have all too often overlooked the needs and capabilities of "the user"; the social scientist, on the other hand, having a background in psychology or sociology, is in a much better position of understanding and articulating the user's concerns.

Endemic to the social scientist's approach of indirectly seeking to improve the design of technical systems by focusing more on the user is a further assumption that this can be achieved through informing designers of what needs to be done. For example, in any guidebook on interface design, designers are told at the onset that they must develop systems that are easy to learn and use. Whereas the intentions of those advocating what, when, and how they should design is often fueled by a genuine faith in their ability to provide insightful knowledge for others, it may seem rather presumptuous to the others. Such an authoritative position is particularly problematic in situations in which those in the targeted group have developed their own craft knowledge, skills, and domain expertise to inform their work practice. For example, programmers, graphic designers, and software engineers have all evolved their own set of methods, tools, and skills. Why should they, therefore, take up alternative methods and tools that another set of researchers have developed for them, which may not, in themselves, map onto the problems of the specific domain being designed for? Besides undermining their own expertise and practice, there is often a lack of appreciation of what their work entails; for example, the many constraints and complexities they have to juggle and manage in their day-to-day work activities. Alternatively, a rather stereotyped view of software designers has percolated; namely, they are too narrow and technically minded and hence need to be educated about social or psychological concerns.

The prescriptive approach to bridging the gap between academic research and "better" system design is clearly problematic. The next section explores why this is so in the context of developments in the sociotechnical movement, human–computer interaction (HCI) and more recently computer-supported cooperative work (CSCW). Following this, an alternative approach is proposed that promotes more proactive involvement of the social scientist in the processes of design, deployment, and use of technical systems. One such role is as an intermediary; negotiating, providing

feedback, and coordinating between the different parties involved (Rogers, 1994). The emphasis here is more on getting involved rather than prescribing, by helping the users and designers of new technology *reconfigure* their working practices.

THE SOCIOTECHNICAL MOVEMENT

The sociotechnical movement came about largely through a concern by social scientists with the effects that automation was having on the well being of the work force. Most notable was the mechanization of the U.K. coal mining industry. Trist and Bamforth (1951) showed that the new technology installed for cutting and removing coal did not lead to the expected increased productivity gains. Instead, it brought many social problems, such as absenteeism and stress-related illnesses, among the workers. A main problem identified was that the resultant organizational changes considerably upset the existing practices. Prior to mechanization, the miners had worked in small independent units (typically pairs) deciding for themselves whom they worked with and setting their own goals and pace of working. Postmechanization replaced these working practices with the longwall method, whereby the small groups were reorganized into much larger groups, resulting in the miners' having less autonomy to organize their work. Furthermore, the tasks were fractionated (i.e., broken down into simplified and segregated tasks), making it much more difficult for the miners to maintain good working relationships.

Based on the findings of this and other studies, in which automation was introduced into the workplace, the sociotechnical movement proposed that changes in both technical and social systems were needed. Focusing only on the technical system, as happened in the coal mining industry, would lead to suboptimization of the whole system. Accordingly, the goal of the sociotechnical approach was to obtain the right balance between the social and technical aspects of the total system. To achieve this harmony, various sets of principles, tools, and techniques were proposed. These included a set of design principles (Cherns, 1976) and a nine-step program for determining the organizational requirements (Mumford, 1987). Examples of the former include "the process of design must be compatible with its objectives," "people should not be given fractionated tasks," and "variances must be controlled as close to their point of origin as possible." Examples of the latter include assessing whether the "workers believe that their roles meet (their) psychological needs" and "all the hypotheses and proposals considered during the process of analysis must be gathered together, considered, and turned into an action program."

Although Mumford (1993) claimed to have had positive effects in changing various organizations through using her particular program (known as ETHICS), the sociotechnical approach—as it stands—has proved to be difficult for others to put into practice. In particular, Blacker and Brown (1986) discussed the problems of attempting to operationalize such general principles, as those already illustrated, in any applicable way. Even though it may be relatively easy for workers to identify their putative psychological needs (e.g., the need to talk with each other), the difficult problem is turning this knowledge into a course of action. Thus, one of the main weaknesses of the sociotechnical approach is the paucity of advice it provides on how to turn requirements into design specification.

Mumford (1993) attempted to bridge this gap by providing more specific recommendations for job and organizational design. Examples include "each job should not be so routine as to cause boredom nor so demanding as to cause stress." However, whereas the content of such commonsensical guidelines may have escaped the attention of some system designers, they are still not particularly helpful in specifying how to design actual jobs or, indeed, how to prevent boredom and stress in a particular work context.

Furthermore, the sociotechnical movement has never really given much advice on how to design new technology. This is somewhat surprising given that the main thrust of the movement was to optimize both the technical and social systems. Instead, the focus has been, ironically, on privileging the social; taking the technology as a given from which to reorganize work around (Clegg & Symon, 1989). For example, an extensive review by Pasmore, Francis, Haldman, and Shani (1982) found that the most typical solution initiated in over 100 sociotechnical studies was the introduction of autonomous working groups. Only 16% of the studies claimed to have accomplished any technological changes.

One of the main problems with the sociotechnical movement, therefore, is its limited characterization of the process of organizational change as, primarily, a set of generalized job design prescriptions. Besides those who conceived of the prescriptions, others have found them very difficult to use and, in particular, to know how to translate them into concrete specifications that optimize both the social and the technical.

THE RISE AND FALL OF HCI PRESCRIPTION

A major concern of researchers who moved into the field of HCI in the early 1980s was that most computer systems being designed were difficult to learn, difficult to use, and did not enable the users to carry out their tasks in the way they wanted. Such was the extent of "bad" interface

design, that the formation of what amounted to the user's charter emerged, in which *user-centered design* became the central philosophy (Norman & Draper, 1986). Aptly described as the rhetoric of compassion (Cooper, 1991), the user-centered approach came up with a range of prescriptive goals for designers, some of which have now become classic slogans (e.g., "know your user!").

The first wave of HCI researchers, mainly cognitive psychologists, attempted to help designers achieve the objectives of user-centered design, by translating their theoretically based knowledge of human capabilities and performance into a bricolage of design support tools. The development of user models, interface design guidelines, design principles, and analytic and empirical methods proliferated. Some researchers aspired to rigorous engineering standards and benchmark tests (e.g., usability engineering; Tyldesley, 1988), whereas others devised more user-friendly techniques that could be performed rapidly (e.g., cognitive jogthrough; Rowley & Rhoades, 1992) with minimum costs and minimal training (e.g., cooperative evaluation; Monk, Wright, Haber, & Davenport, 1993).

Task Analysis

The most pervasive technique to evolve from the translation of psychological knowledge was task analysis. An underlying assumption was that knowing how users cognitively perform their tasks was essential to good system design and so a suitable means of gathering and representing this knowledge was needed. The basic approach was to describe the way users carry out their tasks by breaking them down into smaller and smaller components. Having reduced human–computer interactions into simple unit or atomic tasks, it was assumed that designers would then be able to analyze, predict, and explain the performance of users with different interface and system designs (Brooks, 1991). To distinguish these analyses from the less theoretical approaches that had been established in human factors research, and to show they were based on a cognitive analysis of the user, the methods were typically described as *cognitive task analyses*, each adopting their own reductionist acronym (e.g., CLG, GOMS, CCT, ETIT, TAG, ETAG, DTAG, TAKD; see Diaper, 1989, for an overview).

Although there have been claims for the success of such cognitive task analytic methods in informing design, they have, like Mumford's ETHICS, tended to be for studies carried out by those who have been involved in the development of the methods or trained in the same school (e.g., Gray, John, & Atwood, 1992). In contrast, there have been few, if any, reported studies in which designers or software engineers have taken on board the task analytic methods provided by their HCI colleagues and found them to be truly useful. Besides being intimidating, difficult to learn, and very time-consuming to put into practice, they lack scalability to real-world problems (e.g., see Bellotti, 1988). Above all, they have never really helped the designer design interfaces. Landauer (1991) argued that "the amount of insight into process offered by the keystroke model could be easily exceeded in any real design problem by the empirical evidence of a crude prototype test with a handful of users."

Implications: Design Principles, Rules, and Guidelines

Another approach to helping designers develop more user-centered systems was to transform established knowledge from cognitive psychology, in the form of theories, models, and laws, into "implications" for design. A rationale was to make such knowledge more accessible and usable by designers, by re-presenting it as explicit prescriptions that could be used as a set of criteria from which to evaluate design decisions. Three main forms of implications were developed: principles, rules, and guidelines. The main difference between them is their degree of generalization; the former being the most general and the latter being the most specific. An example of the former is "design for consistency" and an example of the latter is "always place the exit button in the bottom right-hand corner of the screen."

There are two related problems with this re-representation of knowledge: first, finding ways of generating design principles, rules, or guidelines from psychological theory and second, knowing whether and how they can be interpreted in a way that can inform system design. Most attempts have sought to translate empirical findings derived from a theoretically driven context. One of the main problems of taking this approach is that the knowledge base is largely inappropriate. For example, there does not appear to be much in common between the questions addressed by psychologists on how memory is structured and the questions that need to be answered in relation to user interface design. Moreover, many basic theories are too low level, unable to be specified beyond the laboratory settings in which they were determined (Barnard, 1991).

In spite of this dilemma, several researchers have managed to develop extensive sets of interface design guidelines (e.g., Gardiner & Christie, 1987; Smith & Mosier, 1986). Typically, included alongside a guideline, is a brief discussion referring to the psychological theory from which it was derived. An example is the implications of findings from long-term and working memory for interface design. The various limitations and characteristics of human memory are discussed in conjunction with the implications, for example: "As working memory is the bottleneck through which all information has to pass, . . . it implies that complex computer

messages may be digested only if they relate to prior knowledge" (Waern, 1989, p. 43).

One of the major problems with this approach is that the meaning of a piece of empirical research can get distorted in the process of translating it into design guidance. A classic example is the misrepresentation of the well-known psychological finding that the capacity of short-term memory (STM) is limited to 7 ± 2 items (Miller, 1956). In some sets of guidelines this has been translated into a design principle prescribing the maximum number of items that should be displayed at any one time. For example, in a set of recommendations for using color displays, Durrett and Trezona (1982, p. 83) suggested: "The average user should not be expected to remember (the meaning of) more than 5 to 7 colours. This is the 'magic number' usually associated with STM. . . . Novel displays should have no more than 4 colours since this is well below the average limit of STM."

Although it is true that overuse of colors at the interface is undesirable, the reason for this is not because individuals can remember the meaning of only a few colors at any time (this of course will depend on the nature of the information being represented by the colors), but because having a multitude of colors on any screen causes problems of distraction and confusion. Individuals may well recognize 250 different animal names (or remember the meaning of 250 different animal names), but the extent to which they can recall a set of unfamiliar animal names that are briefly presented is limited to roughly 7 ± 2 of those (although this can vary depending on various factors such as individual differences, context, and content). The guideline, therefore, is misleading and has led some system designers to think in terms of categorizing and displaying items on the screen in terms of no more than 7, irrespective of the context of use.

Empirical studies of the use of sets of guidelines have also shown them to be difficult to follow. The main problems encountered have to do with interpretation (de Souza & Bevan, 1990; Tetzlaff & Schwartz, 1991). Mosier and Smith (1986) also reported how designers found it difficult to translate "generally worded guidelines into specific design rules" (p. 39). Löwgren and Laurén (1993) also found that in a study of designers who were required to develop a prototype user interface, half did not use the guidelines with which they had been provided. Simply stated, "you never have time to do it under normal circumstances, so why do it now?"

Current Developments in HCI

The previous discussion has shown how both the cognitive task analytic and implications approaches within HCI, for prescribing better system design, have suffered from trying to translate one discipline's knowledge into practical tools for another. Such difficulties, however, have not gone unnoticed within the HCI community. For example, Long and Dowell (1996) commented on how such "one stream" attempts to pull pure science into an applied one, have shown little evidence of significantly contributing to the design of more effective technological systems. Bannon (1991) and Landauer (1987, 1991) also published influential critiques of these kinds of cognitive approach to interface design.

More recently, several HCI researchers have begun rethinking how to best inform system design using alternative theoretical perspectives (see Rogers, Bannon, & Button, 1993). These include adapting concepts from Activity Theory (Bødker, 1991; Nardi, 1996) and using the framework of Distributed Cognition (Flor & Hutchins, 1992; Halverson, 1992; Hutchins, 1995; Hutchins & Klausen, 1992; Rogers, 1993a) to analyze *computer-mediated activities.* Long and Dowell (1996) also argued for a more design-oriented approach to HCI, which is based on more domain-specific theories. They proposed that a new discipline of cognitive engineering needs to evolve that is separate from that of cognitive psychology, that has its own theories that specifically address the concerns of "users interacting with computers to perform work effectively."

Other researchers within HCI have also sought to develop alternative methods that explicitly support the design process more. These include:

1. *Design rationale.* This aims to document and make explicit the implicit aspects of the design process. The idea is that through explicating implicit knowledge more design alternatives can be explored while also providing a set of reusable designs to be readily available (e.g., see Carroll, 1995; MacLean, Young, & Moran, 1989).
2. *Participatory design.* Designers and users both play an active role in redesigning technology and its use in the workplace (e.g., see Greenbaum & Kyng, 1991; Schuler & Namioka, 1993).

It is too early to say whether or not the new methods will prove to be more effective, useful, and usable by designers, than the user-centered ones derived from cognitive theory. It is interesting to note how the rhetoric of compassion has shifted its focus from being about *empowering* the (single) user, to *supporting* groups of users (or actors) through a "turn to design" (Bowers & Pycock, 1994) and a "turn to the social" (Button, 1993). In so doing, a new set of social and organizational phenomena are being brought to the attention of designers. "Interaction," "the organization of work," "context," and the like are now being offered to designers as the important social science concepts that need to inform design (just as the likes of "mental models," "user modeling," and "information processing" were by psychologists in the 1980s).

DESCRIBING AND INFORMING THE DESIGN OF COMPUTER-SUPPORTED COOPERATIVE WORK (CSCW)

A new research area was established, calling itself Computer-Supported Cooperative Work (CSCW), which set itself apart from HCI and other fields concerned with computer design (e.g., management information systems). Instead of just focusing on the needs of one user sitting at one terminal (as has been the traditional framework within HCI), system designers and social scientists alike became concerned with how to support groups of people in their work settings through and with computers. Advances in enabling technologies, in particular distributed computing and networking, made system developers aware of new opportunities for developing groupware and other technological systems that could support groups working and communicating together. Social scientists, especially sociologists and social psychologists, saw their potential for making a contribution to the new field, by providing accounts of the nature of cooperative work and the way technology is used in work settings.

It is interesting to note that the same dilemma facing psychologists, who moved into the HCI field with the objective of applying their knowledge, is now confronting sociologists and other social scientists in the field of CSCW, namely, how to inform the design of technical systems that can more effectively support groups working together. The problem is further exacerbated, however, by virtue of the fact that designing distributed systems for multiple interacting users is much more complex than designing interfaces for single users (Grudin, 1994). Besides having to translate the social needs of individual users as system requirements, there is also the problem of how to re-represent social and organizational knowledge about the interactions between groups of people and their use of technical systems.

In the early days of CSCW, several social scientists followed the same user modeling approach, as the applied psychologists had done in HCI, but extending the modeling activity to group processes. Whereas individual user models were intended to form the basis of individual interfaces, group models were built for systems that would support group activities (i.e., groupware). However, as pointed out by Bannon (1993), many of the early models of group communication and group coordination had little evidence of any practical relevance to the design task at hand. Just as user models had been abstracted in HCI with little regard of the context of use for the systems they were intended to configure, so too were group user models beginning to appear that were unrealistic and divorced from any context of use. For example, many of the first generation of collaborative drawing and writing tools were based on implicit assumptions about the coordinating mechanisms needed, without taking into account the respective user's awareness and informational needs in a given context (see Dourish & Bellotti, 1992, for a review).

Other social scientists, notably ethnomethodologists,[2] took a very different approach and sought to understand existing work practices by carrying out a number of illuminating ethnographic studies of the organization of work and the interactions and actions that are accomplished in different work settings. For example, ethnographic studies of control centers in the London Underground (Heath & Luff, 1991), American airports (C. Goodwin & M. J. Goodwin, 1996; Suchman & Trigg, 1991), and air traffic control (Bentley et al., 1992) have revealed how informal working practices, like inadvertent overhearing and flexible division of labor, are instrumental to the coordination of work and the co-management of unexpected events. Ethnomethodologists have also turned to the design process itself as a social phenomenon in need of ethnographic study. For example, Anderson et al. (1993) focused on the problem of organizational priorities that designers have to contend with when making design decisions, noting how often such organizational concerns tend to override more technical concerns.

Building a body of knowledge about the use of technologies in work settings, where clearly there is a need for a better understanding, is considered in itself a significant contribution to the field of CSCW (see Plowman, Ramage, & Rogers, 1995). Bentley et al. (1992) suggested that the "information provided by ethnography is essentially background information which has provided a deeper understanding of the application domain," whereas Sommerville, Rodden, Sawyer, Bentley, and Twidale (1993) stressed that "ethnographic studies generate nuggets of useful information at unpredictable intervals." Others, too, have emphasized the valuable insight gained from such studies (e.g., Bowers & Pycock, 1994; Cooper, Hine, Low, & Woolgar, 1993; Rogers & Ellis, 1994).

The reported accounts provided from these studies may also play an indirect role in shaping system design. In particular, they can provide researchers and designers with a means of conceptualizing and framing

[2]Ethnomethodologists are referred to here as a group of sociologists who reject many of the fundamental tenets of traditional sociology, especially the theorizing and structures of the discipline, and who instead insist on a rigorously descriptive program, through doing ethnographic studies, that accounts for members' (sic) working practices (Button & Dourish, 1996; Shapiro, 1994). Researchers who do field studies and call themselves ethnographers, on the otherhand, can come from a variety of social science backgrounds, including anthropology and cognitive science. They do not adhere to the same agenda as ethnomethodologists (although sometimes the social phenomena they observe, interpret, and describe can overlap).

the concerns, issues, and key questions of CSCW. One example of such a work is by Suchman (1987), who provided both a critique of cognitive approaches to system design and an ethnomethodologically informed study of pairs of people using a photocopier machine. Although it is difficult to demonstrate explicitly how this book and her other writings have influenced the design of actual systems, it seems certain that some designers have become enlightened after reading them, and in so doing, much more attuned to the significance of "everyday working practices" through being exposed to her work (Button & Dourish, 1996; Grudin, personal communication).

On the other hand, many designers and software engineers do not have the background, time, or proclivity to digest the typically rich, poetic, and somewhat rambling accounts delivered by ethnomethodologists. Ethnomethodologists are only too aware of this, having encountered software engineers and system designers who find their descriptions of sociality alien and difficult to comprehend (Hughes, Randall, & Shapiro, 1992). At the same time, there is considerable pressure from the CSCW community, at large, for ethnomethodologists to demonstrate more directly how their ethnographic studies can inform design. This creates a dilemma for ethnomethodologists (Grudin & Grinter, 1995), whose research involves analyzing practice rather than "inventing the future" (Button & Dourish, 1996). Moreover, the conventional route of generalizing and theorizing adopted in the social sciences for applying research findings to design is counter to the atheoretical stance taken by ethnomethodologists, which instead tries to remain faithful to the descriptions of members' accomplishments in their work practices. Hence, unlike cognitive psychologists and other social scientists, who have been able to offer designers all sorts of prescriptive advice and design support tools (leaving aside the usefulness of them at the moment), ethnomethodologists have had a real struggle finding ways of translating their accounts that can be accessible to designers.

Some *applied* ethnomethodologists have tried to circumvent the "purist" requirement of rigorously adhering to the data in their accounts by resorting to a form of "covert theorizing" (Shapiro, 1994). This usually takes the form of adding a short implications for design section at the end of a publication that is often in sharp contrast to the discursive style of the account preceding it. Whereas the implications approach is common practice for other social scientists, when discussing their findings in relation to system design, it seems somewhat ill-suited for ethnomethodological accounts. For example, Anderson, Button, and Sharrock (1993) outlined four "bullet point" implications, from their study of an organization's practice, which seem so commonsensical that it leaves the reader wondering why such a detailed study needed to be carried out in the first place. Arguably, anyone could have come up with such advice. An example of their bullets is that designers need support tools that take up a minimal amount of their time and such tools should be adaptive to the exigencies of changing priorities.[3] Most designers know the former only too well and desire the latter only too much.[4]

A problem with this kind of summarizing, therefore, is that it both belittles the ethnomethodologists' rich descriptive accounts while appearing very tokenistic to designers and the rest of the CSCW community. Another approach to getting round the ethnomethodologist's dilemma has been to follow in the footsteps of applied psychologists by offering designers detailed prescriptive advice with respect to the design of a specific system. For example, Heath and Luff (1991) suggested various detailed recommendations for screen design on the basis of their ethnographic study of how an interface was used. Following the prescriptive route, however, makes them prone to the same set of problems applied psychologists encountered when trying to construct guidelines for interface design (see previous section). Moreover, replacing cognitively oriented sets of design guidelines with sociologically oriented ones may prove to be even more problematic, given that there appears to be little relevance between the descriptive language and sociologically generated analytic categories constructed in ethnographic studies and the practical problems of actually designing computer systems (Button, 1993). Hence, there may be even more susceptibility for misinterpretation of meaning. Furthermore, although the conceptual frameworks and analytic tools used in cognitive psychology and cognitive science have something in common with those used by software engineers and system developers (e.g., formal and semiformal notations for describing system requirements and user–system behavior) the terms used in "ethno-talk" and "techno-talk" would appear to be very different. Button and Dourish (1996), however, argued otherwise; they suggested that much of ethnomethodology is in fact concerned with explicating "generally operative social processes," which can, if operationalized appropriately, provide abstractions for design.[5] How such abstractions can be operationalized (or, indeed, have been by the authors) in system design, however, is not made clear.

Another approach to overcoming the ethnomethodologist's dilemma has been to move more toward a designers' perspective and consider alternative solutions for the redesign of a specific system. For example, Heath, Jirotka, Luff, and Hindmarsh (1993) speculated how to preserve

[3]Interestingly, Bellotti (1988) raised the same issue 5 years earlier, with respect to the nonuse of HCI methods for single-user systems.

[4]My apologies to Wes Sharrock for using this study again to make the point.

[5]The example they used to illustrate what they mean by this expression is turn-taking in conversations.

existing work practices in dealing rooms by envisioning how novel technological input devices could be used. Hughes et al. (1992) also constructed interface metaphors as a vehicle for exploring the implications of their fieldwork for the redesign of a flight strip for air traffic control. Taking it even one step further, some researchers have proposed new agendas for integrating ethnographic methods with system design. These range from a prerequisite that informal ethnographic records must be written in a way that they can be translated into more formal system requirements (Sommerville et al., 1993) to practical how-to-do-it guides intended for designers (or in conjunction with ethnographers) to carry out field studies (e.g., Blomberg, Giacomi, Mosher, & Swenton-Wall, 1993).

THE WAY FORWARD: PROACTIVE RESEARCH

The value of "packaged prescriptive" advice provided by social scientists—be they cognitive psychologists, ethnomethodologists, or otherwise—for the purpose of improving system design through applying their theories and methodologies has been questioned. It is argued that approaches putting the onus on designers to translate their advice will remain largely unused and, hence, have little impact.

It is time for a change. Rather than always take a backseat role, researchers need to become more proactive in their involvement with the people and objects of their study. This means engaging more in an ongoing dialogue with the various groups of people working or designing together (i.e., the users, the managers, and the designers). Researchers should stop shying away from being involved. On the contrary, they should be seeking ways of taking a more active role in the design and implementation process, even becoming "change agents" (cf. Blomberg et al., 1993) where appropriate. In so doing, ideas can be fed back, discussed, and negotiated as part of the ongoing practice of research (Cooper et al., 1993). Through adopting a more reflexive orientation to design the articulation of system requirements also can be improved (Bowers & Pycock, 1994). It could also act as a learning experience, enabling researchers and designers to develop an understanding of what, when, and why methods work in practice. In turn, this could lead to some truly useful generalizations.

Emergent Involvement in the Implementation and Use of Technology

It is encouraging to see that a number of researchers are becoming more involved in the design and implementation processes. One trend has been to explore and explicate the design requirements and development of computer products with designers through the use of various prototypes and scenarios (cf. Blomberg et al., this volume; Bowers & Pycock, 1994; Carroll, 1995). Another approach has been to play more of a role in facilitating the implementation of new computer systems in organizations in conjunction with coadapting their work practices.

One experience while carrying out some research at a travel company, illustrates how the latter can be achieved (Rogers, 1993b, 1994). A study was begun under the pretext of trying to understand the problems surrounding the introduction of a new distributed computer system into an established workplace, which provided an opportunity to play an active role in implementing change. Specifically, the potential was there to help the company overcome some of the problems that had arisen through the introduction of the new system.

The company had decided to change from its existing multi-user booking and ticketing system to another one for various reasons, including the benefits of more functionality, a better maintenance package, and more security. None of the reasons, however, were concerned with the usability of the system. It was not surprising, therefore, that when the new system was first introduced in the company, there was widespread discontent among the consultants on the shopfloor. Their main complaint was that the system was cumbersome and unwieldy to use in contrast with the existing system. Moreover, they felt the new system was far more constraining, preventing them from carrying out their work in the flexible manner they had become accustomed to with the old system. On discovering these usability problems, the directors assumed they were simply teething problems that would dissipate once the consultants had become more familiar with the system.

Following discussions with the various consultants using the new system and listening to their conversations, a very different picture emerged from the shopfloor. In particular, their dislike of the system continued to grow. Several consultants complained bitterly that the new system was seriously disrupting their ability to carry out their work.

Gradually, management became more aware that all was not well. Informal feedback from myself and the consultants made them change their minds about attributing the discontent to teething problems and, alternatively, decide to do something. Initially, they felt that priority should be given to getting the software improved. Accordingly, they agreed on an arrangement with the system suppliers, whereby the suppliers would develop enhancements to the software that the travel company felt would substantially improve the usability of the system. Any agreed changes, however, would have to be negotiated with regard to their cost. It was important for the travel company, therefore, to be able to assess the additional costs (determined by the amount of new lines of

code required to be written to make the changes) in terms of perceived benefits. To facilitate this process of negotiating required initially collecting a list of all the problems the consultants were experiencing. This required that someone go round to the various departments, collate the information, classify it, and then write it up as a list of requirements in terms that were understandable to the suppliers.

But no one wanted to do this task because it was considered to be additional work load. Originally, one of the directors was going to do it but became too busy; so one of the consultants was given the responsibility. The consultant, likewise, was unable to carry out the task having become too busy coping with the continuous stream of technical problems from the new system. An opportunity arose to help them overcome this impasse.

During weekly visits to the company, the director would see me, which would remind him of the need to deal with the new system "problem" and consequently talk to the consultant in my presence, asking whether the list had been made. The consultant would explain that he never had enough time now and that when he had asked the other consultants they had not volunteered the information. However, as the weeks of procrastination progressed, the director became increasingly concerned. He specifically requested that I join him and the consultant to visit all the different departments and collect the gripes there and then. We collected written lists from the various departments and got the other consultants to explain in more detail what they meant by what they had written down. There was even a joke at the end of this process, that the company could usefully employ me as a secretary to type up the lists.

It was possible to accomplish the task because there were three individuals working together. Hence, in one sense, I acted as a mediator, helping to instigate participation from the consultants to provide input into the redesign of the system to improve its usability. Of course, the consultant responsible for the information-gathering task would have eventually created the list of requirements, but it could be argued that the process was accelerated and legitimized through me being there. Moreover, the explicit discussions that ensued between myself, the consultants, and the director while doing the rounds enabled both the director and the consultant to gain a better picture of the difficulties the consultants were having. I also contributed my own researcher's views on why I thought some interface changes were necessary, which were based on observations and understanding of the problems the consultants were experiencing. In this sense, I was acting very much as a facilitator, using academic knowledge/expertise in CSCW and HCI in a largely implicit manner.

There were also several other instances during the study, where I became implicated in mediating between the consultants, management, and the suppliers. For example, I was involved in the decision-making activities concerning the possibility of introducing a new software package that would have radical effects on the current ways of working (Rogers, 1993c). Each occasion where I became involved was largely opportunistic. In doing so, I moved with the ebb and flow of the tide of obstacles, problems, and developments as they surfaced in the company, providing feedback and facilitating discussion as and when appropriate.

The advantage of being an outside researcher meant that I could avoid the situation of having to be aligned with either management, the consultants, or the software suppliers. I could move freely between the different parties. Trying to remain neutral, however, was sometimes difficult. For example, there were occasions when management attempted to exploit the situation of me being present on the shopfloor, by asking me sensitive questions about the working practices and activities of certain consultants. In such circumstances I used my discretion. I would either give noncommittal answers (e.g., "they seem to be OK") or point out that they could always ask the consultant in question, themselves.

This form of proactive research does not lend itself to any formalization. As such it cannot be regarded as a particular approach to design, but, alternatively, should be viewed as research that both enables an ethnographic study to be reported in the CSCW community of the changes that occur when a new system is introduced into the workplace (Rogers, 1994), while also enabling me as a researcher to play a more active role in helping the particular company to better configure its technical systems and working practices.

Social Scientists as In-House or Corporate Researchers and Consultants

Getting involved with the people who are being studied is of course nothing new; participatory design methods such as those espoused by Mumford (1993) have always stressed the value of researchers as facilitators in helping users specify their own requirements. Likewise, several in-house/corporate HCI groups (and other socially based research teams) together with independent consultancies have sought in vain to be more directly involved in dialogues with the primary architects of design (e.g., see Blomberg et al., this volume; Rousseau, Candy, & Edmonds, 1993). The difficulty that such groups frequently experience, however, is getting their voices heard and being able to have some influence in the design process. Often, the contributions proposed by such researchers and external consultants are not incorporated, because it is too late for the necessary changes to be made in the design process (see Grudin, 1993, for a historical overview of why such obstacles have evolved). A further problem is the degree of relevance and value attributed to HCI/CSCW specialists and other socially oriented researchers by the architects of design. For example, Blomberg, Suchman, and Trigg (chap. 8, this volume) discuss the pigeonholing of their ethnographic work by others in the organization, who regard their findings as being only relevant to product development for particular markets. In contrast, they believe their research has much more universal application.

A difficult problem facing corporate researchers employed in large organizations, therefore, is how to overcome competing demands. There are many other stakeholders involved in the development of systems, whose vested interests often take precedence over the social scientist. In trying to make a case, the corporate social scientist may be forced to justify their contribution through the provision of concrete and tangible results. The danger here is that such visible outcomes will be equated with the creation of prescriptive sets of guidelines and design tools. In contrast, proactive research entails much "articulation work" (Strauss, Fagerhaugh, Suczek, & Wiener, 1985), where informal talking, negotiating, and reflecting with in-house designers are central. The corporate social scientist, therefore, is in the difficult position of trying to convince the other stakeholders of the importance of this form of reflexive activity.

CONCLUSIONS

Previous attempts in the field of HCI to configure the user indirectly through prescribing ways of designing technical systems to be more user-centered have been largely ineffective. Part of the problem stems from the use of an inappropriate knowledge base from which to derive the prescriptive advice. The production of paper and computer-based tools by one community for another, with little consideration of how that community does its work, was also seen as problematic. Alternative means of informing system design that have since emerged within HCI and CSCW were outlined. These include reconceptualizing the area of study, using alternative theoretical and analytic frameworks, and a shift toward understanding and supporting the design processes and the implementation of technology in organizations.

Ethnomethodological and other kinds of ethnographic studies of the workplace and design were seen as playing an important role in shaping the emerging field of CSCW. It was stressed, however, that such research does not have to follow the same misguided path of informing system design through the provision of prescriptive advice, taken by the applied psychologists in HCI and sociotechnical systems. Instead, social scientists could try considering how to be more proactive in their research by engaging in continuous dialogue with the people being studied. In doing