Machine-, Human-, or Culture-centered Computing? A View from the Trenches

David Hakken
Associate Professor of Anthropology
Acting Director, Technology Policy Center
State University of New York Institute of Technology
PO Box 3050
Utica, NY 13504

Introduction

However practical or impractical, elegant or awkward, computer-based information systems are human and therefore cultural. In this paper I wish to argue that failing to appreciate just how cultural computing is is the source of many of the problems encountered by those who develop computer artifacts and implement their use. Indeed, this blindness to culture is so extensive that many system developers may in essence be trying to do the wrong thing. I will make this argument by reflecting on my experience as a social scientist who has tried to work with computer scientists and engineers to open up the system development process. While I see reasons to be optimistic about the ultimate success of a project to center system development culturally, and while I will make some suggestions about how to make system development more culturally informed, I also believe that such attempts will accomplish little until basic issues about information systems are clarified.

I am an anthropologist, and I articulate these goals aware that they will strike some computer specialists as either nonsense or heresy, while they will strike others as self-evident. I ask you to give my ideas a hearing in either case. I am an "experienced user," having been a social scientist moderately "into computers" for the last twenty-five years. Further, I have spent the last twelve years teaching sociological and anthropological perspectives on work and technology to upper division college students, and the last eight or so years studying computerization as a cultural process and teaching computer literacy. Moreover, my research increasingly revolves around a search for more socially-aware methods for information system development. For example, the Disability Technology Project (a pseudonym), of which I was the principle investigator and about which more below, aims to involve people with disabilities in the development of an advanced information system for accessing assistive technology. Finally, I ask for a hearing because I am trying to address the misgivings about the unsatisfactory and sometimes inhuman ways current systems operate expressed by computer professionals.

The paper begins with a brief illustration of the problems involved just in specifying appropriately the task of the information system developer, an illustration intended to show why these tasks must be seen as having important cultural dimensions. I then state what I mean by a culture-centered approach to computing and give some examples to indicate how it is feasible and can be economical. After another example of the ambiguous, even schizophrenic character of technical computer thinking about the role of people in system development, I discuss what has happened on the Disability Technology Project. After considering alternative ways to account for the problems described, I argue that they are rooted in the very basic ways in which system development is conceptualized. I conclude by sketching the philosophical basis of an alternative design philosophy and offer an initial set of guidelines for design practice based on the alternative. Because I realize the programmatic nature of these guidelines, I conclude with a request for more collaborative research to evaluate these observations and fill out the alternative methodology for system development.

A Useful but Ultimately Flawed Statement of the Problem with Current System Development Methodologies

A critical article in a recent issue of the Communications of the ACM is a good place to begin an examination of how the conception of the nature of information systems affects ideas about the proper way to approach their design. In a "Commentary," Donald Norman addresses "Human Error and the Design of Computer Systems." After considering a well-known example of system failure, Norman labels lack of data on how to integrate "the human operator into the system analysis" as the major gap in knowledge of information systems. Arguing that this gap is linked to basic presumptions and stances, Norman calls for a change in basic attitude. In language very congenial to the social scientist sensitive to the prevalence of "blaming the victim," he calls for elimination of the term "human error" from the system design vocabulary: "[I]nstead of blaming the human who happens to be involved, it would be better to try to identify the system characteristics that led to the incident..." He concludes by calling on the ACM "...to elevate the human side of computing to a level of concern and respectability equal to that of the physical and symbolic side" (1990:4,7).

Norman's plea is salutary, and his commentary goes a long way toward establishing points I wish to take as given: that information activity is not merely technical

machine activity, and that if we are to develop effective information systems, we must think very differently about them. However, an approach to information practice based on the perspective he articulates would still have major flaws. For one thing, Norman conceives of information systems dualisticly--e.g., in terms of human vs. system: "The behavior of an information processing system is...a product of the interaction between the human and the system" (p. 4; my emphasis). It is not inevitable that information systems be conceived as interactions between separate entities rather than single entities. Rather, the analysis presented below presents this dual vision as an artifact of the very approaches to system development which Norman critiques.

Further, in a related manner, Norman conceives of human action individualisticly--in terms of single persons--rather than culturally, as the shared activity of collectives. How different is the approach of Niels Bjorn-Andersen, who calls for an "anthropological" approach to information system development. For example, Bjorn-Andersen offers a richer model of the system development process, one which is framed by the relationship between the entire organization and its technological infrastructure, not just the user and his/her machine. His approach is more cultural than Norman's, acknowledging for example how important it is to support users during the implementation process, but that this support must be directed to groups and collectives rather than individuals (1986:65).

In making such points, authors like Bjorn-Andersen draw on an extensive body of social science literature which examines computerization empirically. (Among the most important examples of this empirical research, in addition to those cited elsewhere in this paper, is the work of Attewell and Rule (1983), Wilkinson (1983), and Williams (1984).) This work shows concretely how the consequences of computerization initiatives are highly mediated by their contexts. My own fieldwork with Barbara Andrews on new information technology and change in working class culture in Sheffield, England, for example, justifies giving less attention to what information technologies do than to what they are perceived as doing (Hakken 1990a, 1990b; this work was supported by the U.S. National Science Foundation Anthropology Program).

Culture-centered computing

Bjorn-Andersen points toward what I like to call a "culture-centered" approach to information system development. System development is "culture-centered" when it is appropriately contextualized culturally--that is, when explicit attention is given systematically to the

socio-cultural as well as techno-psychological dimensions of creating or transforming an information system. A fully culture-centered development process, for example, integrates understanding of the specific users' notions about computers, as well as more general knowledge regarding the social correlates of computing, into each stage, including design, of a new system.

Just how innovative a culture-centered approach to computing is, at least conceptually, emerges when it is contrasted to standard "machine-centered" and even developing "human-centered" approaches. In conventional approaches, system development is approached as a closed, technical process. The process is "machine-centered" in that the most basic decisions are about hardware and software; they have to do with which artifacts are, in the opinion of the designer, best able to complete information processing tasks. The tasks as well as the artifacts are treated as given.

It is by now standard practice among many computer professionals to acknowledge that some attention should be given to non-machine or "human" factors in information system development. However, the implications of broadening how to think about the context of computerization are worked out only narrowly in human factors or so-called "human-centered" approaches. In general, such approaches are conceived within the framework of an individualistic psychology, and/or the human dimension is seen, as in Norman's argument, as relevant primarily at the level of the physical individual rather than that of the social group. This latter perspective is particularly characteristic of many ergonomic studies in the U.S. Alternately, human factors are presented as relevant to only a part of the system development process--e.g. implementation trials--rather than to the entire process. In such so-called human-centered approaches, human information systems are in essence still treated as closed systems. It is precisely the attempt to be systematically open to all the human aspects of information which distinguishes culture-centered from machine- and most existing human-centered approaches to computing.

Culture-centered computing is also to be distinguished from the "socio-technical" orientation of Mumford (e.g., 1983) and her colleagues as well. While this approach does integrate awareness of organizational dynamics into the development process, it too easily accepts the validity of management views of information processes and ignores the broader class cultures in which workforces participate. (However, the same term, a "socio-technical approach," is used by Fenno-Scandians like Ehn as a label for their own approach, which is much closer to what I am

calling "culture-centered.")

The term "culture-centered computing" is a bit of an oxymoron or contradiction in terms. The contextualization characteristic of the kinds of cultural approaches taken by anthropologists implies a diffusion of thought, a de-centering of focus, away from the thing itself or things basically similar to it and toward its context. For example, our Sheffield research documents how the class culture of the workers using computers is an important mediator of what happens "when the computers arrive." However, we judge the elegance and simplicity of "culture-centered," as well as the communicative value of maintaining a parallelism with "machine-centered" and "human-centered," as being more important than the potential communicative value of any non-oxymoronic alternative we have encountered to date.

"Culture-centered computing" thus refers both to a way of developing specific computer-based information systems and to a general way of thinking about computers. In the specific sense, the term is a useful label for various new approaches which, like those in Sheffield discussed below, place a broad cultural perspective at the center, rather than the periphery, of conceptualizing the system development process. Computing activity is culture-centered, then, when the technological activity is clearly and consciously located within the broader stream of collective human life--that is, the culture. A understanding of the culture-centeredness of computing <u>in general</u> is an important part of developing any system culturally, but equally important is an understanding of the <u>particular</u> human activity--be it office, workplace, or community action--into which computing is being introduced. A important source of particular information is the regular participants in the process, the eventual prime operators and users. The information perceptions of those with power in organizations tends to be strongly ideological, resistant to any data suggesting that the way things are deviates significantly from the way things are supposed to be. Consequently, it is particularly important to understand the view of those lower down the hierarchy who actually run the system.

Thus, the culture-centered computing perspective includes both a statement about <u>what computing is</u>--a cultural process--and a statement about <u>how it should be done</u>--e.g., by collectively involving users and consumers, especially those not normally involved in formal decision-making.

<u>The Feasibility and the Economics of Centering Computing on Culture</u>

Numerous projects demonstrate that more culture-centered approaches to computing are feasible. In the Nordic nations, where these notions have often been given legislative support, and in regions like Sheffield in England, where progressive computerization has been seen as an important part of local political strategy, such projects have become institutionalized, even if they are not necessarily the dominant trend in computerization.

The Human-centered Office Systems Group at Sheffield Polytechnic, for example, has developed a series of initiatives which use the "study circle" approach to office automation (Owen 1990). In this approach, clerical workers facing new information technology are periodically released from their normal duties to become involved in the system development process. In addition to more formal examination of the actual work practice, these workers are provided with additional training to enable them to understand at a deeper level the implications of alternative information hardware and software. The approach draws heavily on the consciousness raising procedures developed by feminists. Members of the initial Sheffield Libraries' Study Circle have gone on to participate fully in Departmental discussion of what information systems are for as well as purchasing decisions. They are themselves now running additional study circles.

Another Sheffield example of culture-centered computing is SPRITE, Sheffield People's Resource in Information TEchnology. The SPRITE Project deliberately rejects the workplace orientation of most computerization in favor of a community perspective (Darwin, Freyer, Fitter, and Smith 1985). The Project aims to put computing training, hardware, and software into the hands of community organizations and the unwaged generally, to accomplish whatever purposes they themselves articulate. Heavy emphasis is placed on sharing of skills and interests among groups. SPRITE has opened progressive computing as a new terrain of community politics; it has also provided an important alternative sense of identity for working class people. For our purposes, however, the experience of projects like SPRITE is most relevant in that it suggest alternative ways of thinking about what computing is for.

The economic viability of a culture-centered approach is also demonstrable by example. Traffic Systems Co-op is an expanding Sheffield enterprise which services traffic signals. The Co-op was formed to provide an alternative to the expensive system maintenance contracts offered by multinational electronics firms; its services are now

sought eagerly by local governments throughout the Yorkshire region. Co-op members have a definite strategy for using computers to aid the work of signal engineers by providing them with easy access to technical and historically-relevant information regarding a particular signal system. Members are adamant about not using computers to monitor workers' performance or break up the engineers tasks; they see the point of the coop as preserving the quality of the work for both employee and service user. The result is a very creative extension of computer applications and an eager receptivity to real technical innovation (Hakken 1990b; Hakken and Andrews 1991).

Indeed, a strong case can be made that a culture-centered approach is essential to organizational survival in the changing economic terrain of the 1990s. Frank Dubinskas (1988), for example, argues against implementing computerization in a Tayloristic fashion, by using them to deskill workers and simplify production. Firms that computerize in this manner will be less flexible and lack the skills of adaptation which will be so much at a premium in the future.

The Ambiguities of Current Development Practice

Of course, a convincing demonstration of the practical and economic superiority of more open approaches to the system development process is beyond the scope of this paper. The intent of these examples has been to indicate not only that such approaches are viable but also that they provide food for thought with regard to the reconceptualization of system development which is necessary. The ambiguities of the current conceptualizations among design professionals, ambiguities which necessitate such a reconceptualization, are well reflected in Shackel's "Ergonomics in Design for Usability" (1986). Shackel initially casts the concern of the system developer for the human dimension in mechanical and individualistic language: "[T]he first process...is to define the system aims and the various functions needed to achieve those aims and then to examine and decide which functions within the whole system should be assigned to human elements and which to machine elements...[S]uch factors as cost, weight, size, reliability, safety, and efficiency must be assessed and compared" (p.45) in order to find "the optimum fit of the people to the jobs" as part of "...setting limits within which humans can be used" (p. 51).

Despite initially casting the problem so mechanisticly, Shackel goes on to use very different language to discuss the features of the "Design for

Usability" which he advocates. Among these features are: 1) User-centered design, based on study of who the users are and their tasks, "...requiring direct contact with users at their place of work and learning their tasks..."; 2) participative design: "A panel of users should work closely with the design team, especially during the early formulation stages"; 3) experimental design, where users actually use prototypes of the system for real work; 4) iterative design, where difficulties identified by experimental users are corrected through as many redesign steps as are necessary, and 5) user-supportive design, in which user supports are included all through the system development process, not just at the end, through help-screens, etc. (pp. 57-8).

Shackel's approach in these later sections is very consonant with Bjorn-Andersen's admonition to "be anthropological." There is thus a clear disjunction between the language within which the development problem is posed and the language of solution. Two very different conceptualizations of the information system underlie the two parts of Shackel's argument, yet his article does not express any awareness that this is the case.

How Such Ambiguities Hamper Design Practice: The Disability Technology Project

In general, the current practice of computer professionals illustrates a broad consensus that an system development which "integrates the human side" is desirable. However, there is an absence of consensus over what such integration actually means and how to go about it. I also would argue that the very notion of "integration" is inappropriate.

Why the current situation is unsatisfactory was brought home to me sharply in the Disability Technology Project (DTP). The purpose of the project was to develop a computer system for people whose severe disabilities limit their ability to access assistive technology. DPT aimed to use a new generation of technology developed for the military to ease such access. The project was culture-centered two ways. One was the inclusion of an advisory panel of consumers with disabilities from a local Independent Living center; indeed, consumer involvement in the project was an important consideration for the funding agent, a State Technology Foundation. A second form of culture-centered practice was the placement of an anthropologist, myself, as principle investigator on the project. The idea here was that I would be able to perform a traditional role in applied anthropology, that of a "broker" between the worlds of the consumers, the funding agency, the academic staff of the project--which

included a professor of electrical engineering technology and a graduate student and professor in computer science, the staff of the Independent Living center, the federal agency which was supporting the project through access to facilities, the vendors supplying the hardware, and potential system vendors in the future. In performing this role, I would not only help translate from one world to another, but I would also be in a position to increase the influence of the consumers on system development.

Recently, I found it necessary to resign from the role of principle investigator. As in any project, particular circumstances--in this case, gross errors in public relations by my academic institute, the limits of my own and others' abilities in various areas, and the fact that the project was very complex and under-budgeted--make it difficult to generalize. In retrospect, however, the difficulties on the project seem entirely predictable. Despite my best efforts, the academic computer science/engineering staff conceptualized the project in strictly technical terms. For example, they conceived of the panelists primarily as sources of inspiration rather than as full participants. A long discussion at a panel meeting of various ways to approach the system was dismissed by one academic staff member with the following query: "What is this project about, anyway; just talk, or are we serious about a system?" Some panel members saw their role in complementary terms. Their job was to develop a "wish list" for what the system should do, while the technical staff was to tell them what was possible. These panel members expressed a lack of interest in the "technical details," despite the fact that they themselves were experienced computer users with considerable practical knowledge of computer-based disability technologies.

The untenability of my own role became evident in a controversy over an interim report. Out of frustration, I had drafted the report more or less on my own, because I was unable to get enough cooperation from the other academic staff. My draft, based on brief conversations and numerous assurances that we were effectively meeting project objectives, was denounced as "worthless." One staff person refused to use it as even a working draft whose "blatant errors" were to be corrected, because such a document "should not be written by someone from Arts and Sciences." Discussion over whether the technical objectives of the project were being met--talk which seemed to involve disagreement among the technical staff--was cut short, implying that such discussion was inappropriate within the presence of someone not technically qualified. Ultimately, two interim reports were submitted, one "technical" and one "non-technical."

Accounting for Such Problems

Informal discussion with other social scientists involved in information development projects suggests that my experiences on the Disability Technology Project were not unique. One way to account for them is in terms of the limitations of narrow professionalism; for example, the claim that the writing of an interim report which involves technical matters, irrespective of the substantive issues involved, is necessarily the job of a credentialled technical person. From this perspective, the problem on the project is seen as comparable to a trade union dispute over job demarcation, complicated of course by the colonizing of knowledge consequent to the history of academic professionalization. Such difficulties can be overcome at least in theory by selecting personnel who are more confident of their position and motivated by factors broader than narrow professional markers of prestige.

An alternative explanation draws on a linguistic reading of C.P. Snow's famous "two cultures" distinction (1969). In this view, problems like those encountered on the DTP are essentially failures of communication which follow from differences in the way people use words. Such problems can be overcome through move effective translation between "technical" and "humanistic" language, the restating of equivalent propositions in appropriate technical jargon.

Problems in Culture-centered Computing as Philosophical Problems

While there is much of truth in both of them, I find neither of these analytic perspectives, even taken together, sufficient. We on the DTP were highly motivated and put considerable effort into the project, in the face of numerous difficulties. We were clearly not just motivated by narrow professional concerns. Further, the problems we encountered were not purely differences in jargon; attempts to find equivalent language, of which there were several, were generally unsatisfying.

In my attempts to present an argument for culture-centered computing to my technological institute colleagues, I generally encounter two linked responses. One is, "Oh, we already do that, by setting up user groups," etc. The other is to see my demand for culture-centered computing as a thinly veiled political agenda, a sociological (read socialist) attack on the power structure, military domination of R&D spending, etc. Most of the time, I feel like my colleagues just don't see

my point; we're talking about different things, not just using different words.

The problems I have encountered trying to center computing culturally are more philosophical than professional and linguistic. I suggest that culture-centered practice may be incompatible with current design practice, because thinking about this practice systematically misconceives it. Further, this misconception is so basic that it renders whole aspects of the computing process inaccessible to examination.

I read Lucy Suchman's Plans and Situated Action (1987) as illustrating my point nicely. Suchman traces many difficulties in system development to a basic misunderstanding of the character of human action, especially its presumption that humans generally act by setting goals, developing plans to achieve them, and executing the plans. In her view, humans do not, generally speaking, proceed this way; indeed, the "plan" conception is characteristic only of certain activities (e.g., the theorizing of cognitive scientists) in Western society. Rather, most people move through life from situation to situation, developing understandings of situations as they enter them and acting primarily out of these understandings. Our action is situated rather than planned; as a consequence, our action has much more to do with how we interpret situations at the moment than with the prior articulation of abstract, "logical" intentions.

As a college professor, of course, I am often in the position of encouraging my students to plan, but I make a big mistake if I assume that this is what they regularly do. Similarly, to the extent that computer programs presume planning as normal, they are likely to fail. Software environments like those developed by the Computer-Supported Cooperative Work network aim to incorporate a situational rather than planning model of human information activity.

Pelle Ehn's Work-oriented Design of Computer Artifacts (1988) outlines a critique of standard information system design theory which parallels Suchman's critique of design practice. On his reading, standard design is marked by a Cartesian dualism, a one-sided realism which similarly produces a distorted model of human action. Ehn would have us replace an overly rationalistic approach to design, one which takes as its method the development of linked sets of logical propositions, with a practice-based approach, one which takes into account the social character of the construction of reality. In place of Cartesian rationalism, Ehn would have us conceive of design in the manner of Wittgenstein, as a form of "word play":

> The Language-games played in design-by-doing can be viewed both from the point of view of the users and of the designers. This kind of design becomes a language-game in which the users learn about possibilities and constraints of new computer artifacts that may become part of their ordinary language-games. The designers become the teachers that teach the users how to participate in the particular language-game of design. In order to set up these kind [sic] of language games, the designers have to learn from the users (p. 118).

Ehn's argument is too complex to be encapsulated in a short paper, but it immediately suggests some alternative perspectives relevant to Norman's commentary. I was critical of Norman's dualistic view of the "human error" problem. The problem is not to "integrate" the user into "the system." Rather, the user and the "system" are one to begin with; in the philosopher's language, they are ontologically unitary. The misconception that the problem is one of "integration" is based on a Cartesian dualism.

The argument that such misconceptions are not just matters of technique but rather reflect an entire world view is developed more fully in philosopher Stephen Toulmin's Cosmopolis: The Hidden Agenda of Modernity (1990). Toulmin aims to show how the conception of scientific/technical activity promulgated by Seventeenth Century thinkers like Descartes, who invented the modern world view, is a false abstraction. Responding to their chaotic social environment, these thinkers wished for a transcendental truth. Their conception of science appears to be technical but is in fact theological, its roots hidden in a presumed correspondence between the natural and the social worlds. The ordered cosmopolis which is to reflect this built-in correspondence is to be accomplished by an emergent human reason, which is to mediate the divine order revealed in nature and the Satanic disorder of human emotion. This human reason is presumed to correspond to divine reason.

To prepare for a recent faculty seminar on artificial intelligence, my colleagues and I were asked to read recent articles in Scientific American by Searle and others. Doing this reading in the context of my difficulties on the Disability Technology Project, and while reading Suchman, Ehn, and Toulmin, I realized one of the problems I have always had with the idea of artificial intelligence--that it, too, is close to being an oxymoron. As Searle argues, whatever intelligence is, it is human, whereas the artificial is by definition unnatural, and therefore inhuman. Toulmin might add that the problems of

thinking about what AI is derive directly from the dualism of Seventeenth Century thought.

On occasion, I try to put such arguments to my technological colleagues. Running through many of their responses is a profound ambiguity about human activity. Sometimes, they presume that human activity is anti-rational, (i.e., emotional) and therefore by definition unpredictable and incapable of being taken into account in system design, except in some "para-normal" fashion. On other occasions, they wish to treat human activity as part of the natural world, a Newtonian mechanical process, and therefore highly systematizible (i.e., a "human factor"). I suggest that most of the time, however, their conception of human action falls on neither side of the Cartesian divide; it is the "rational mediator" of the Cartesian abstract "cogito." Unlike Descartes, however, they cannot accept rational activity as essentially moral and therefore explicable in terms of doctrine. Human activity, including design activity, once no longer conceived as corresponding to divine intelligence, is essentially unknowable.

As a social scientist, I am committed to a view that human activity and most of its forms (e.g., intelligent activity) is explicable, although not in the impoverished terms of a scientistic Cartesian methodology. While Suchman's, Ehn's, and Toulmin's perspectives force us to question whether information "systems" as presently constituted are all that systemic, they also suggest alternative ways to conceptualize the system development process. Development, as human activity, is situated. While it is useful during moments of the design process to abstract from the real activity and pretend that it is systemic, rational, and planned, we must never lose sight of the fact that such abstractions are distortions. Our design/development tasks are never completed until we move from abstraction back into the concrete of situated action.

Conclusion: Real Collaboration in Information Practice

My experience with really existing, more culturally-centered computerization projects like those in Sheffield suggests the outlines of an alternative methodology for system development. Such a methodology would likely include the following steps:

1) System development should begin with a thorough analysis of the broad range of existing activities within the organization. Fully culture-centered computing is highly self-conscious, which implies approaching any information task critically. For

example, if the organization is a for-profit enterprise, it is essential to understand the specific nature of the labor process and the way in which this process is shaped by broader social forces. The frank involvement in this examination of all participants in the work process, especially direct producers, is essential.

2) The next essential step is a similar participatory, critical analysis of the existing information "system" of the organization, especially how current information flows have developed historically. This analysis should attempt to separate out information which is truly needed from information "needs" which are an artifact of the organizations history or the reproduction patterns encouraged by social groups beyond the organization. Because of the danger of presuming that the best solutions are technological--i.e., the technicist "dazzle" effect--it may be necessary, at least initially, to formulate the model of real information needs independent of consideration of computer options.

3) Rigorous consideration of computer options should only begin after all groups involved have access to the requisite technical knowledge. Culture-centered computing often requires education and hands-on experience with new systems. It is important that those providing "computer literacy" training have no self-interest in the particular options from which choices may eventually be made.

4) Finally, and most generally, culture-centered computing depends upon development of a collaborative perspective on the nature of the organization and the interests of those within it. This perspective depends upon an understanding of the broader social dynamics within which individuals and organizations develop particular patterns and choose goals and the strategies for achieving them. Particular attention needs to be given to the class cultures within which individuals participate and the existing class strategies among which they can choose, since these have great influence over individual choices and eventual group dynamics.

I am aware that these guidelines are very abstract and programmatic. I hope to have communicated enough about the perspective of Suchman, Ehn, and Toulmin to indicate that they have something important to say to system developers who hope truly to "informate" (Zuboff 1987) human activities. I cannot be more concrete about how their perspectives should be applied because I lack enough positive experience of how to do it right from beginning

to end.

I therefore seek real collaboration with information system developers who wish to work with me in projects which aim to illuminate these perspectives in practice. I think such collaboration holds out real promise for developing more effective as well as more humane information practice. Through real collaboration, we can make a positive contribution to both developing more effective information practice and more robust, less dualistic theory.

References Cited

Attewell, Paul and J. Rule
1984 "Computing and Organizations: What We Know and What We Don't": Communications of the ACM 27:1184-92.

Bjorn-Andersen, Niels
1986 "Understanding the Nature of the Office for the Design of Third Wave Office Systems," in Harrison, M.D., and A.E. Monk, eds., People and Computers: Design for Usability, pp. 65-77. Cambridge: Cambridge University Press.

Darwin, John, Mike Fitter, David Fryer, and Leigh Smith
1985 "Developing Information Technology in the Community with Unwaged Groups." Presented to the Aarhus Conference on the Development and use of Computer-Based Systems and Tools.

Dubinskas, Frank
1988 Making Time: Ethnography of High-Technology Organizations. Philadelphia: Temple University Press.

Ehn, Pelle
1988 Work Oriented Design of Computer Artifacts. Stockholm, Sweden: Almqvist and Wicksell.

Hakken, David
1990a "Has There Been a Computer Revolution? An Anthropological View": The Journal of Computing and Society, 1(1):11-28.
1990b "Culture-Centered Computing: Social Policy and Development of New Information Technology in England and the United States." Under revision for Human Organization.

Hakken, David, and Barbara Andrews
1991 Computers and Culture: New Information Technology and Social Change in Sheffield, South Yorkshire, England. Boulder: Westview Press.

Mumford, Enid
1983 Designing Secretaries: The Participative Design of a Word Processing System. Manchester: Manchester Business School.

Norman, Donald
1990 "Commentary": Communications of the ACM

33(1):4,7.
Owen, Jenny
1990 "Women Office Workers Developing "Human-centered" Systems: Initiatives and Opportunities in a Local Authority Context." Presented to the annual meeting of the Society for Applied Anthropology, York, England.
Shackel, B.
1986 "Ergonomics in Design Usability," in Harrison, M.D., and A.E. Monk, eds., People and Computers: Design for Usability, pp 44-64. Cambridge: Cambridge University Press.
Suchman, Lucy A.
1987 Plans and Situated Actions: The Problem of Human/Machine Communication. Cambridge: Cambridge University Press.
Snow, C.P.
1969 The Two Cultures, and a Second Look. Cambridge: Cambridge University Press.
Toulmin, Stephen
1990 Cosmopolis: The Hidden Agenda of Modernity. New York: The Free Press.
Wilkinson, Frank
1981 ed. The Dynamics of Labor Market Segmentation. London: Academic.
Williams, Virginia
1984 "Employment Implications of New Technology." The Employment Gazette, May, 1984:210-215.
Zuboff, Shoshana
1987 In the Age of the Smart Machine. New York: Basic Books.

DIAC-90

DIRECTIONS AND IMPLICATIONS OF ADVANCED COMPUTING

July 28, 1990
Boston, Massachusetts

Sponsored by Computer Professionals for Social Responsibility

Edited by Douglas Schuler