Encountering Others: Reciprocal Openings in Participatory Design and User-Centered Design

John M. Carroll
Virginia Polytechnic Institute and State University

Participatory design (PD) refers to a collection of attitudes and approaches originally codified in Europe under a variety of banners in the decade following the Second World War. Starting in the 1970s, PD was widely applied to the design of information technology in Europe. In contrast, PD became prominent in North American human–computer interaction (HCI) research only during the latter 1980s. In North America, PD is often understood as an approach to user-centered design (UCD), in which users are directly consulted throughout the system development process. Putting the relationship this way is correct, but not complete. There are important distinctions between PD and UCD.

UCD can be nonparticipatory. For example, model-based engineering approaches to UCD seek to supplement or replace user involvement through user descriptions (Card, Moran, & Newell, 1983). Empirical approaches to UCD frequently operationalize user involvement by surveys or tests of random user samples, or by detailed observations of particular user situations (Shneiderman, 1987). In these approaches, users are involved as records, subjects, or cases, but not as full participants.

However, even when UCD is participatory, it most typically justifies this on purely technical grounds. Typical examples of such justifications are that models and theories of users are inadequate, that users on the

John M. Carroll is a cognitive/computer scientist interested in learning and problem-solving in human–computer interaction contexts, and in methods and tools for instruction and design. He is Professor of Computer Science and Psychology, Director of the Center for Human–Computer Interaction, and Head of the Computer Science Department at Virginia Tech.
design team bring specific expertise regarding typical and critical scenarios of use, and that users may be able to articulate specific design requirements—particularly the emergent requirements that become visible only in the context of a specific design, and therefore cannot be gathered in a requirements study (Brooks, 1975/1995). In this view, the role of PD in UCD is that of a technical method for ensuring better design results (Nielsen, 1993).

PD in Europe has a different foundation, one anchored in notions of workplace democracy and human development. In Europe, the conflict between management and labor is regarded as an abiding structure. Acknowledging and managing this conflict is intrinsic to the European process of workplace democracy. One explicit objective of this process is the design of work activities that support human development; in this worldview, work is supposed to challenge and enrich people by design.

The European foundations of PD do not have good mappings in North America. In North America, management seeks to be invisible or to deny the premise of conflict; labor is poorly organized and concerned chiefly with employment security and wage rates. The goal of supporting human development through the design of work activity is rarely articulated in North American labor discourse. A paradigmatic example from HCI is the very concept of usability. In North America, usability is often operationalized as mere simplification at the level of operations and tasks. However, in Europe such low-level simplification is seen not as merely liberating workers from keystrokes, but as potentially de-skilling them—that is, as imperiling the opportunity and challenge that should inhere in the design of work.

Over the past decade, HCI has become much more of an international endeavor. The North American HCI community, embodied for example by the membership of the Association for Computing Machinery’s Special Interest Group on Computer–Human Interaction (ACM SIGCHI) and attendees of CHI conferences, has become more eclectic about usability and design methods in general, and about the nature and purpose of user participation in particular. This assimilation has flowed the other way too. The PD community has taken on many of the engineering goals of UCD, producing a variety of focused tools and techniques to evoke user participation more efficiently in the formative evaluation and redesign of prototypes (e.g., Kuhn & Muller, 1993).

The articles in this special issue offer a view of where PD is going now, and a chance to reflect on further possibilities for how PD and UCD can continue to interact productively.

Johannes Gärtner and Ina Wagner (1996 [this issue]) use actor networks to describe three arenas of design and participation: system design, organizational frameworks, and industrial relations. Their article illustrates the
broader scope analysis typical of PD, as well as a particular descriptive framework for couching and depicting such an analysis. The article challenges UCD practitioners and researchers to notice the bigger context, the interactions among actors in the background that constrain what can happen in design and use.

An actor network nicely captures the topology of interactions, but not the content of those interactions. For example, it cites and classifies the interaction of management and shop stewards as established and routine, but it provides no explicit framework for analyzing the specific issues that constitute this link either historically or in a given case under current analysis. Recent work in HCI on design rationale has developed and investigated a variety of approaches for quasi-formally reconstructing the argumentation that lies behind design decisions and results (Moran & Carroll, 1996). Design rationale work to date has focused most closely on Gärtner and Wagner’s “system design” arena, but perhaps both UCD and PD could move forward by extending design rationale analysis throughout all three arenas of design.

Susanne Bødker and her colleagues at Aarhus University were among the vanguard of PD as it arrived in North America a decade ago. Her article (1996 [this issue]) describes recent work in which she explores new roles for PD action researchers. The earlier work of the Aarhus group was a collaborative, sociotechnical advocacy on behalf of national labor unions, realized through custom software development (the widely cited Utopia project; Bødker, Ehn, Kammersgaard, Kyng, & Sundblad, 1987). The new work is pursued through facilitating the adoption, customization, and integration of shrink-wrapped applications. In the new work, the researchers play the roles of consultant and trainer; they cooperate with both management and labor. In part, these new roles are adaptations to contemporary changes in the contexts of computing technology and work organization.

Bødker's approach involves facilitating an autonomous learning community in the workplace. She describes how her team makes a suggestion, and then waits patiently as its consequences diffuse and develop throughout the organization. Indeed, she emphasizes the researchers' goal of not being missed when the project ends. This is a far less interventionist role than the one they played in the Utopia-era work. Perhaps such a distributed, group-initiated educational process could be supported by computer-mediated communication tools, such as newsgroups and other forums. These could take the form of relatively focused FAQ (frequently asked question) databases (e.g., Ackerman, 1994), or of more wide-ranging discussions of community history, practices, and motives (e.g., Carroll, Rosson, Cohill, & Schorger, 1995). The variety of these tools is increasing rapidly, and they offer many approaches to placing control within the
community, while still allowing participation from outside consultants. It would be interesting to explore the support of Bødker's research goals and organizational roles with these information system frameworks.

Like Bødker, Jeanette Blomberg, Lucy Suchman and Randall Trigg (1996 [this issue]) are exploring new roles for PD action researchers and new variations on PD. They describe a case study of what might be called "technology-driven" PD, in which they played a role that might be called "mediator." Unlike traditional PD practitioners, Blomberg et al. bring to the design collaboration specific, a priori commitments to technologies; they were interested in retrieval systems for text and image collections. However, they are not themselves the developers of such systems or technologies. On one hand, they tried to make visible an understanding of the workplace in order to provoke insights from the developers (and, it turned out, among various workplace constituencies). On the other hand, they tried to create a setting for introducing and evaluating the use of prototypes (see also Hughes, Randall, & Shapiro, 1993).

The mediator role has been criticized on grounds that mediators may have too little knowledge to represent users and developers adequately to one another (e.g., Kyng, 1995). Blomberg et al. respond to this in their suggestion that high-fidelity, case-based prototypes can vividly evoke technology possibilities for users and workplace needs for developers. Nevertheless, some things can get lost; in their case study, it seems that simplifying the retrieval work may also diminish the organizational role of their informant (M) as "coach" to the junior attorneys. Perhaps the case-based prototype approach can be pursued by taking the prototypes even more seriously. For example, current efforts to closely integrate HCI and software engineering incorporate tools and methods for the concurrent development of use cases (scenarios), object-oriented domain models, design rationale, and running prototypes (e.g., Carroll, 1995; Taylor & Coutaz, 1995). Such an integration could help PD mediators to "read" prototypes more insightfully for both users and developers.

Karlheinz Kautz's article (1996 [this issue]) argues that PD should be incorporated into curricula for computer science and other computing disciplines. He illustrates this by briefly describing some of his own course development work. Kautz's critique of recent curriculum change in North America suffers somewhat from a Eurocentric perspective. Seen in its own context, curriculum development in North America in the past decade has been sweeping and radical: A decade ago, HCI was peripheral in computer science. By the end of the 1980s, an ACM task force had enumerated HCI as one of nine core areas of the computer science discipline (Denning et al., 1989), and a joint-curriculum task force of the ACM and the Institute of Electrical and Electronics Engineers (IEEE) recommended the inclusion of HCI as a common requirement in computer science programs.
(Tucker & Turner, 1991). Today, computer science departments are scurrying to expand their course offerings in HCI.

Kautz, however, is correct that the treatment of PD in these North American courses is, by and large, grounded purely in technical UCD. Discussion about ethical obligations toward users and their work is generally treated briefly and in isolation from design exercises. Even in Kautz’s course syllabi, where there is good integration of ethical and social issues with design exercises, the pedagogical vehicle is role playing, in which students pretend to be users, developers, and so forth. Maybe it is time to push this a step further on both continents with an integrated and realistic approach, in which students identify and analyze social and ethical issues in the context of cooperative education projects situated in the community outside the classroom. Cooperative education is relatively well developed and pervasive in the North American curriculum model for computer science. It could provide an excellent setting within which students could explore the responsibilities and consequences of roles they will assume in society.

PD is more than a particular implementation of UCD. The two have interacted fruitfully in the past decade in part because of their somewhat complementary foundations. We should hope and expect that this dialectic will continue to develop through the next decade.

NOTES

Author’s Present Address. John M. Carroll, Department of Computer Science, 660 McBryde Hall, Virginia Tech (VPI & SU), Blacksburg, VA 24061–0106. E-mail: carroll@cs.vt.edu.

HCI Editorial Record. First manuscript received March 18, 1996. Revision received May 13, 1996. Accepted by Thomas Moran. – Editor

REFERENCES


