Scandinavian Journal of Information Systems

Volume 10 | Issue 1 Article 11

1998

Users and computers: A contextual approach to design of computer artifacts

Morten Kyng
University ofAarhus, mkyng@daimi.aau.dk

Recommended Citation

Kyng, Morten (1998) "Users and computers: A contextual approach to design of computer artifacts," *Scandinavian Journal of Information Systems*: Vol. 10 : Iss. 1 , Article 11.
Available at: http://aisel.aisnet.org/sjis/vol10/iss1/11

Users and computers: A contextual approach to design of computer artifacts

Morten Kyng

Department of Computer Science, University of Aarhus

Defended for the degree of dr.scient. 16th August 1996

Preface

This article contains a presentation and overview of the papers that I have submitted for the degree doctor scientiarum (dr. scient.).

The introduction relates the subject matter of the submitted papers to current discussions in computer science. Section two gives a brief account of the research area in question, how it has developed over the last 25 years, and its current status.

Section three presents and discusses the results structured according to the frame of reference given in section two. For each sub-area the central issues are introduced. Then the results obtained are presented; the practical as well as the theoretical. Finally, a short comparison and evaluation in relation to relevant literature is made.

Section four contains a short discussion of the methods used, and section five presents ideas for future research.

1. Introduction

The efficiency of computers has increased dramatically over the last decades, as have our technical skills. However, increased technical proficiency has not been able to meet the challenges of our profession as new groups of people get involved and the use of computers proliferates. Today, we, as computer professionals, usually find ourselves in complex organizational settings, where diverse and often conflicting interests coexist. While we work, the problems our designs are supposed to handle are evolving, as are the views of the people involved. As practitioners, we no longer

© Scandinavian Journal of Information Systems, 1998, 10(1&2):7–44

face problems so well-defined that technical experts working alone can find the right solution. To meet the current challenges of our profession we need new ways of understanding our profession and our relations to other parts of society, new ways of working and new ways of cooperating with others.

This development is mainly driven by the *use* of computers, i.e. from the outside, not the inside, of computer science.[1] It is thus not surprising that general recognition of this change is slow, particularly in the scientific community, and that some of its first mainstream reflections are to be found in areas close to practice, such as the ACM Code of Ethics and discourses on the future of our profession, e.g. curricula proposals (Anderson *et al.* 1993, Denning 1992, Hartmanis 1992, Turner 1991). These writings call for a broadening of the concerns of computing education and research. Denning, for example, points to areas such as communication and collaboration (p. 88) and the Code of Ethics states that "When designing and implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs and will avoid harmful effects to health and welfare." (imperative 1.1). And "(As an ACM member and an organizational leader, I will) Manage personnel and resources to design and build information systems that enhance the quality of working life." (imperative 3.2).

These texts clearly indicate that the work of computing professionals is not merely concerned with developing solutions to given specifications. However, the concrete suggestions for re-orientation of research and education seem more modest than the changed situation—and the texts themselves—calls for (Dahlbom & Mathiassen 1994).

When we turn from North-America to Europe, the call to address issues traditionally categorized as value-laden is also present. Thus in the address from the University of Hamburg to the recent 13th IFIP World Congress the speaker pointed to the need to cater for both democratic values and ecologically sound developments. Again, the way this call is answered by the papers in the proceedings leaves much to be desired (Pehrson *et al.* 1994).

The changes and the need for reorientation are just beginning to attract wider attention in the scientific community and the kind of re-orientation called for is not something that happens overnight. However, a body of research already exists that as part of its very base incorporates a number of the concerns raised above. An influential example from the United States is the book "Understanding Computers and Cognition: A New Foundation for Design, by Winograd and Flores" (1986)—from Europe I mention the edited book "Software Development and Reality Construction" (Floyd *et al.* 1992). Two examples from my own work are the DUE [1] and the UTOPIA projects [2, 3, 4][2]. DUE supporting democratic influence on development and use of computers and UTOPIA with its dual focus on enhancing the quality of work and of products. Such work represents an opportunity to take a look at where these concerns over a period of more than two decades have taken their proponents.

I encourage the reader to view the work presented here in this light: as a search for professional and scientific re-orientation to meet the challenges emerging from the use of computers.

2. The subject area

A first characterization of the subject area is given by the two words in the main title: "users" and "computers". At the most general level, the area concerns people using computers and their relations to development and use of computer systems. Compared to a traditional approach to computer science my research acknowledges users as those who make computing meaningful. The first distinction is then between a traditional/ mechanistic/functionalistic approach on the one hand and a contextual/romantic/ "non-functionalistic" on the other—an issue I will return to below.

A contextual/romantic/non-functionalistic approach entails the notion of conflicting interests, of the absence of the one best solution. The outcome of a design effort is determined by the context—what interests are supported—and by the effort itself—which participants are supported. Furthermore, design efforts are grounded in time and space: conditions varying from one country to another and changing over time are crucial in shaping design efforts. In my own research this simple insight has had a profound influence: in the 1970ies existing contexts supported managerial interests only, and I worked on developing supplementary contexts supporting users/workers interests. Later, in the mid 1980ies, we had severe difficulties in supporting user/worker participation in design activities proper, and I came to see the development of such support as the most needed task in our research.

This latter theme, supporting user/ worker participation in design activities proper, now characterizes a research area. The label most often attached to the area is currently "Participatory Design" (PD), (Kuhn *et al.* 1992, Muller & Kuhn 1993). From a constructive point of view the question is how to develop tools, techniques and theories to support major aspects of different roles of users, including cooperation with professional designers, in system development projects, i.e. users in the role of "contributors to design". The first of the two themes above, the context—what interests are supported, often seems to be lost in current PD research.

In the following, I have chosen to present my research primarily in relation to PD although the above characterization as contextual/romantic/nonfunctionalistic does not apply to the entire the PD area. Over the years, the specific tradition, the shaping of which I myself have contributed to, has been given different names. In the mid 1980ies Pelle Ehn and I introduced the name "The Collective Resource Approach to Systems Design" (CRA) [3] and, at the same time, Bansler has named it "The Critical Tradition" (1989). With our book "Design at Work: Cooperative Design of Computer Systems" (Greenbaum & Kyng 1991), we introduced the of concept cooperative design to emphasize the increased focus on design activities proper as well as the inclusion of research contribution from the humanities and the social sciences. Personally, I find both labels suitable: "The Collective Resource Approach", CRA, denoting the Scandinavian research tradition that I have been a part of, since the mid 1970ies, and "Cooperative Design" denoting that part of our research within CRA which is directed towards design activities proper. Finally, as a supplementary characterization of my own research within CRA I have chosen the subtitle "A contextual approach to design of computer artifacts", as a way of emphasizing the central role of design in my research and, at the same time, the crucial importance that I attach to the context of design.

2.1. Players in the field

The area of PD is relatively new and has enjoyed a surge of interest recently (Muller & Kuhn 1993, Shapiro 1993). A core of work—the Scandinavian Collective Resource Approach, of which my own research is a part—has a 25-year-old history with strong emphasis on workplace democracy. Other work, travelling under the banner of participatory design or early user involvement, is appearing in a number of different disciplines. Two examples are the disciplines Information Systems and Human Computer Interaction where users traditionally enter the process only at the end of the day: When the system is to be installed and the "users-to-be" thus need to be instructed on how to use the new system; Or when the interface—not the functionality—goes through the final usability testing before release of the system. During the initial phases, such as feasibility studies and analysis, the insight needed into the work of an organization developing or buying a system is provided by the managers of the "users-to-be". In fact, this trend has been so powerful that these days we often have to retreat to such labels as "end-users" to be sure that they, the real users, are not confused with their managers.[3]

But times are changing. These days Microsoft wants users involved early in their design efforts—not managers, and not IS people. Nokia develops and evaluates e.g. remote controls by studying television users in their homes and by means of mock-ups. Lotus and IBM do usability testing "up-front" based on paper mock-ups of the interface—before functionality is frozen. And Microsoft tells us that they base their design work on a Hegelian notion of conflict.[4]

Between these extremes—the Scandinavian Collective Resource Approach and the early user involvement practised by some of the large companies—a rich and varied spectrum unfolds. Over the last decade, Scandinavian IS research has produced a huge volume of work emphasizing early user involvement in system development, cf. the proceedings from the annual Scandinavian IRIS seminars and (Floyd *et al.* 1989). Within Software Engineering, researchers from Germany have developed a comprehensive approach to system development that includes a partnership view of users, cf. (Floyd 1987, Floyd 1992). In Britain, researchers in the socio-technical tradition have since the late 1960'ies moved from an instrumental and even manipulative view of users (Mumford & Ward 1968) to a position that can be labelled as PD (Mumford & Weir 1979). In the US, early work based on a labour process perspective and focusing on the (negative) effects of computerization (Greenbaum 1979) developed into PD approaches (Greenbaum 1990). Also in the US, researchers—some with a background in ethnography—developed supplementary approaches to the investiga-

TABLE 1. Three theoretical schools according to Bansler.

	Systems theoretical tradition	*Socio-technical tradition*	*Critical tradition*
Knowledge interest	profit maximizing	job satisfaction, participation	industrial democracy
Notion of the organization	cybernetic system	socio-technical system	framework for conflicts
Notion of the labour force	objects ("system components")	subjects (individuals)	subjects (groups)
Notion of capital/ labour relations	common interests	common interests	opposing interests

tion of work and artifacts (Suchman 1987, Suchman & Trigg 1991, Trigg *et al.* 1994). These ethnomethodologically inspired approaches played an important role in placing *use* at the center of PD concerns.

2.2. *Some characterizations and delimitations*

If we take a step back to look for ways to understand the developments in PD in relation to other areas of computer and information science, several important contributions could be mentioned, however, for the purpose of this overview, I restrict the presentation to the following three: Bansler (1989), Hirschheim & Klein (1989), and Dahlbom & Mathiassen (1993).[5]

In his paper, "Systems Development Research in Scandinavia" (1989), Bansler identifies three theoretical schools: the systems theoretical, the socio-technical and the critical (the last one being identical to the Collective Resource Approach described below). Bansler's analysis is on the level of characterizing research traditions, and he succeeds in presenting the reader with useful basic distinctions between the three schools, summarized in Table 1.

Bansler's work is based on a literature study spanning the proceedings of the Nordic computer conferences NordSAM/NordDATA in the years 1960-1985. Since then, much has happened with respect to "users and computers". First, the critical school itself has developed beyond the position described by Bansler, most notably in relation to supporting user/worker participation in design activities proper and development efforts initiated by management. Secondly, participatory techniques are developed within schools that do not fit his characterization of the critical or the socio-technical school.[6] In short, characterizations like Bansler's are useful in getting a first grip on a set of research traditions, but too general and high-level—and not intended to—explain the inner workings of a tradition.

In their CACM paper (1989), Hirschheim & Klein present four Information System development paradigms based on the two pairs order/conflict and objectivism/subjectivism, see Table 2.

TABLE 2. Four paradigms

	order	*conflict*
objectivism	functionalism	radical structuralism
subjectivism	social relativism	neohumanism

In this categorization, functionalism is very similar to Bansler's information theoretical school whereas social relativism to some degree captures the socio-technical school. However, while the category functionalism is well-founded, the other three turn out to be more blurred. Indeed, characterizing functionalism vs. the other three "non-functionalist" categories seems to be the major contribution of the paper. This is done through a characterization of functionalism vs. first radical structuralism and then vs. social relativism/neohumanism.

Finally, in their book "Computers in Context" (1993) discuss two idealised world views: the mechanistic and the romantic. The mechanistic world view, as developed in the 17th century by Descartes and others, fits reasonably well with the functionalism of Hirschheim and Klein, and the romantic world view has much in common with neohumanism, and more broadly with the "non-functionalism" of Hirschheim and Klein.

In summary, contributions as the last two above describe a traditional approach vs. its negation in a number of dimensions. For the purpose of this overview, I present a version that focuses on dimensions that are crucial for a presentation of my own research. It is inspired by many of the same sources as Dahlbom and Mathiassen and developed by Joan Greenbaum and myself as an overview of the focus shifts in system development in going from a traditional approach to an approach based on cooperation with users [7, p.6; 11, p.412]:

TABLE 3. Focus shifts

Traditional systems approach –focus is on	*Cooperative approach –focus is on*
problems	situations and breakdowns
automation	support
information flow	social relationships
formal procedures	situated work
describable skills	tacit skills
expert rules	human expertize
individuals communicating	group interaction
rule-based procedures	experience-based work

As evidenced by the work of the authors mentioned above, there is an articulate critique of traditional computer science that calls for qualitatively new and different concepts as we enlarge our scope from the technicalities of computers to include the people using computers. And there is a growing body of work, particularly in the PD area, developing tools and techniques that focus on non-mechanical aspects, such as those listed in the right column above, e.g. tools and techniques to support users in bringing their tacit skills to bear in design. In this work, however, the aim of research such as my own is not to eliminate the concerns or insights of traditional approaches, but to place them in a new context, a context where different concerns are primary.

This kind of work has the design process as its object of study, not societal forces or organizations. Indeed, some researchers claim that PD, while developing improved tools and techniques for system development, has lost sight of the bigger issues of power, resources and conflicts and that PD mainly helps managers to organize development work more effectively without giving users more influence (Kraft & Bansler 1994a, Kraft & Bansler 1994b, Noble 1994).

In order to bring out, present and discuss these issues I use the distinction between "design in context" and "contexts for design". Under the first heading I look at contributions to developing non-functionalistic/non-mechanistic/non-traditional tools and techniques for design. Under the second heading, I discuss the embedding of design in the larger organizational and societal context, and in particular strategies for supporting users' democratic influence through design related activities.

However, as stated above, design efforts are grounded in time and space, and conditions varying from one country to another and changing over time are crucial in shaping design efforts. Thus, in order to explicate the rationale behind my research and, in particular, the development over time in the dual focus on "contexts for design" and "design in context" I begin with a short historical presentation of the Scandinavian PD tradition, the Collective Resource Approach, which is recognised as the common inspiration for most current PD work (Muller & Kuhn 1993) and to which my own research belongs.

2.3. Scandinavian PD: the Collective Resource Approach

Three decades ago, PD did not exist and thus work in the seventies and eighties had a strong formative trait and entailed numerous disputes over what to consider as central issues, how to understand them etc., see e.g. the book edited by Å. Sandberg (1979). Different schools or approaches in Scandinavian system development research have been identified (Bansler 1989) and exploring differences between these has been an important part in understanding each approach itself [3].

The first important step in creating the Collective Resource Approach, CRA, was taken in the early nineteen seventies with the Norwegian NJMF project (Muller & Kuhn 1993, Sandberg 1979, Shapiro 1993), which established workers as a major, active interest group in relation to computers. Based on Scandinavian ideas on workplace democracy, the NJMF project (Nygaard 1979) and related projects, such as the Swedish Demos (Ehn & Sandberg 1983) and the Danish DUE [1], set out to improve the possibilities for workers to influence the way their workplace was affected by management controlled introduction of computers. Mainstream research at that time either paid no attention to "non-managers", beyond the need for instructing them prior to the introduction of a new system, cf. e.g. (Andersen *et al.* 1972), or considered them a resource to be utilised in the fulfilment of goals defined by management, cf. e.g. (Mumford & Ward 1968). Against this background it turned out that it was not feasible to develop useful paths of influence for workers based on adjustments of existing development methods. Results from working at the level of the design process would be curtailed by the context of those processes, since this context did not contain the means to promote worker interests. It was thus necessary to work at a level that could contribute to the creation of such new contexts. As this formative work developed, it became clear that direct participation by workers and their trade unions played a key role. The results of these first generation activities relate to the context of design and they can be characterized as follows:

- A new framework for worker influence on development and use of computer-based systems, focusing on worker controlled resources, independent worker activities and on negotiation with management as a basis for influence.
- A number of concrete examples demonstrating the "how" as well as "results" in terms of differences from the outcome of traditional, managerially controlled activities without such a "workers corrective."

And finally the technology strategy was summarized as:

- Local action based on central support.

Basically, this work viewed design from the outside. The focus was on supplementary—worker controlled—activities needed to develop a "worker's point of view" in relation to a new system. And usually these activities took place more or less in parallel with a traditional system development project, which they aimed at influencing.

We took the next major step in fleshing out CRA in the early nineteen eighties with the Utopia project [3, 4].[7] Originally, our focus in the project was on developing an example of an alternative system: A system that supported goals of the workers, such as increased possibilities for developing skills at work. However, it turned out once more, "that it was not feasible to develop useful paths of influence for workers based on adjustments of existing development methods". This time. the problem was not the lack of a supportive context for the design work but rather that traditional methods were so unsuited to support user influence in the design process that real alternatives were needed. Thus, new cooperative techniques and tools for design became a major contribution of the Utopia project. Our results from this second generation project relate more to design in context and they can be characterized as follows:

- A "demonstration example"—from the graphics industry—showing that it is possible to design a credible alternative to existing systems. An alternative based on a tool perspective, that supports good working conditions including the development of skills at work as well as supports high quality products.
- A revised framework for trade union influence on the supply of technology, based on a new conception of central union design activities, and further developing the framework for local activities from the first generation activities listed above.
- A revised framework for worker/ designer cooperation in design, adding a new emphasis on the need for both worker and designer competencies in the design process itself, and adding.
- New design techniques and tools, such as mock-ups, supporting creative contributions by the workers.

With respect to technology strategy, the original "local action based on central support" was supplemented with:

- Expanding local choice through centrally developed alternatives.

Subsequent to the UTOPIA project, our CRA work has continued to focus on issues within design in context: on developing tools and techniques for cooperation in design. In addition, there has been work on creating a "whole organization" approach to development, i.e. an approach which involves all groups in an organization and thus supplements the notions of a worker's corrective and alternative systems. With respect to technology strategy, the "local action based on central support" and "expanding local choice through centrally developed alternatives" has been supplemented with:

- Local co-development based on cooperative tools and techniques.

2.4. Current Issues in Participatory Design

As described in the introduction, there has been a rapid increase in the interest in involving users early in the design process. Most of this work focuses on tools and techniques for user participation in managerially initiated projects, and there is very little emphasis on the context for design. In particular, the notion of worker controlled resources and independent worker activities in combination with negotiations with management, as a strategy for influence, has almost totally disappeared. However, there are a few exceptions, most notably the Australian Union Research Centre on Organisation and Technology, URCOT (Snelling & Jolly 1994). This initiative has been launched by the Public Sector Union and the Australian Taxation Office to support the workers in influencing the introduction of new computer systems and changes in work organization in the Australian Taxation Office over a ten year period. Apart from URCOT, there are currently no examples of "large scale & focused vision"-PD projects as was the case for Scandinavia in the nineteen seventies and eighties with NJMF, DEMOS, DUE, Utopia and Florence (Trigg *et al.* 1994). On the other hand, the rapid growth of "context-neglectant" PD and the "less-favorable" results obtained seem to refuel the interest within PD in the relations between design and context (Trigg *et al.* 1994).

2.4.1. Contexts for design

Current work in this area is mainly concerned with understanding the relations between organizational context and design: through the collection and analysis of information on both PD and non-PD projects and through the development of theoretical frameworks (Clement & Besselar 1993, Wagner 1993). The central questions concern:

- Conditions for effectively organizing PD projects and for incorporating PD techniques and tools in "traditional" development projects,
- supporting and limiting factors on the influence of different groups in organizations; and—as an important part of this—
- the demands on the designers themselves in the face of conflict, and the role of values and ethics.

2.4.2. Design in context

Within the shared context of user participation most current work in PD is related to techniques and tools for cooperation in design: either directly by presenting new or modified techniques and tools for PD together with experiences on the use of such techniques and tools or, indirectly, by addressing theoretical and methodological issues relating to techniques and tools. The central questions in this work concern the user contribution and the user/-designer cooperation:

- Which techniques and tools are effective for users and for user/designer cooperation in PD,
  - in what stage of a project,
  - in what setting, and
  - for what kind of contribution.
- The development of specific techniques and tools.
- Ways in which designers support the use of such techniques and tools.

Finally, most PD work shares an emphasis on "real life" situations as opposed to laboratory settings. This implies that conducting PD projects plays an important role in the PD research community, an issue I will return to in section 4.

2.5. Related areas

The work described above has overlapping research interests with several research areas and draws on a number other areas as auxiliary subjects.

During the formation of the CRA, sociology of work and the work on establishing a working life science played an important role in understanding the research subject (Sandberg 1979). Sociology of work provided a theoretical framework for looking at the relations between societal conditions and technological development. Later, as the work on design itself came into the center, the ties with sociology of work have loosened.

The emphasis on open-ended real life projects as opposed to laboratory experiments led to an interest in a number of theories that see the world as socially constructed. Among the philosophers that have inspired the development of the theoretical aspects of PD are (Heidegger 1962, Polanyi 1967, Wittgenstein 1953,1963).

During the last half of the nineteen eighties another new area emerged, that of Computer Supported Cooperative Work, CSCW, cf. (Grudin 1991a) and proceedings from the ACM conferences on CSCW, bi-annually since 1986. This area shares with PD the focus on cooperation. But, in addition, the complexities of computer supported cooperation have led CSCW research also to focus on tacit knowledge and situated action—concerns at which PD arrived through its focus on work practice. As a result, CSCW has developed into an area where analytical oriented studies, e.g. based on ethnography or cultural anthropology, are confronted with construction oriented work in originating in PD.

US researchers in this area have, e.g. following the early work of Suchman and Wynn (Suchman 1987, Suchman 1983, Suchman & Wynn 1984, Wynn 1979a), developed increasingly more participatory techniques based on an ethnographically inspired approach (Trigg *et al.* 1994).

Also the area of Human Computer Interaction, HCI, has recently begun a move from the laboratory towards real life situations and, in doing so, new connections with PD are established and shared points of view emerge (Carroll 1995). The processes of human computer interaction—as opposed to the research area of HCI—have, of course, been within the sphere of interest of PD much longer. Furthermore, a growing body of work related to requirements engineering is addressing the challenge to traditional, formal specification approaches raised by the situated use of computers, see e.g. (Goguen 1992).

Participatory design of information systems can also be viewed as part of the broader area of Information Systems research and, particularly, within Scandinavia there is a considerable concern for user participation in the IS community, cf. the proceedings from the annual Scandinavian IRIS seminars and the Scandinavian Journal of Information Systems.

Finally, the PD emphasis on design as an experimental inquiring process—together with an interest in being able to involve users throughout a development project—has strengthened the relations between PD and experimental system development, cf. section five below.

In other words, a growing body of work in CSCW, HCI, Requirements Engineering, Information System research and experimental system development shares with PD a profound dissatisfaction with the shortcomings of a traditional, mechanistic approach to system development—and some of the attempts at overcoming them.

3. The results of the papers

Based on the overview above, this section presents the results of the submitted papers. The aim of the presentation is to convey to the reader an understanding of possibilities and limitations of my research. However, to understand the present, the state-of-the-art, it is necessary to know where the work came from, against what background it was shaped. In particular, the increased focus on early user involvement in industry and elsewhere, in combination with the emphasis on tools and techniques for design within CRA, increases the risk that people will look at my research and other CRA work simply as a way to modify techniques firmly rooted in a traditional, mechanistic approach. As discussed in (Kyng 1994b) I am not "against" such use of my research and other CRA work—and it would not really matter if I were. The most important factors shaping the future of our profession are not internal to our science, but external, related to the *use* of computers. When more general conditions support user influence in design, a traditional, mechanistic approach, supplemented with tools and techniques from CRA, gives more leverage to users than a traditional approach without this "supplement." On the other hand, if the CRA tools and techniques are understood as contributions to a new way of doing design in context, and if the importance of the context for design is realized, then the full potential of the research presented here may be utilised and possibilities open up for reorientation as discussed in the introduction.

In order to facilitate this kind of understanding, I use a structure resembling the relevant parts of section two:

In section 3.1, I present important "time and space" conditions for my research, how these developed over time in Scandinavia, particularly in Denmark, and some results at the level of changes in conditions.

Following this, I present the results using the two categories introduced in section two: In subsection 3.2 I present results on contexts for design; this forms a natural background for the subsequent treatment of techniques and tools for cooperative design, design in context, in subsection 3.3. In each of these two subsections I present a short list of central issues followed by a discussion of practical and theoretical results.

3.1. An overview of developments in conditions and results

The work described above took place from the beginning of the 1970ies to the mid 1990ies in close interplay with changing societal conditions in Scandinavia. Around 1970, when the first project of the Scandinavian CRA was created, we may characterize the situation in the following way:

Within the trade unions only the traditional negotiation issues, such as wages and working hours, were considered to entail potential conflict with employers. Production issues, including the use and development of technology, were considered to belong to the category of "one best solution" and were consequently left to management to decide. At the same time, there was a growing dissatisfaction at the shop floor concerning work environment and new technology. A dissatisfaction that had not been curtailed by several experiments with co-determination projects since these did not seem to provide "a way forward". In fact, they were gradually abandoned, e.g. when shop stewards wanted to go beyond the limits originally imposed by management [3, p. 23-24]. However, the trade unions had no strategy towards technology based on worker interests or a conflict perspective; and there was no model for technology related projects based on worker interests.

Within the Scandinavian research communities the mainstream view was a harmony supporting notion of science as value free. However, particularly in the student movement, there was a growing awareness of the existing management bias in the application of science, in the scientific "agenda setting" and eventually in the research itself. Within computer science this recognition was supported by the widespread use of computers as control instruments, quite literally separating planning and execution of work at the shop floor.

3.1.1. Results from the first generation of projects: NJMF, DEMOS and DUE[8]

In relation to the above conditions we may describe the results in the following way:

- Within the trade unions, production issues, including technology, were now considered to entail potential conflict with employers.
- A number of technology agreements based on this view were added to the set of existing agreements.
- One week courses for shop stewards and interested workers on local work and technology were established in Denmark.[9]
- A new model for local, factory level work with technology, backed by the above mentioned technology agreements and one week courses was established.
- A new model for research/union projects based on worker interests was developed.
- University level courses on the topics of the projects were now being taught.

This summarizes the results of the first projects. In addition to these new conditions for further work, two other issues played an important role in the reasoning of my colleagues and myself when we shaped the next round of CRA work: First, the restricting factors at the factory level, including the limitations imposed by available technology, had a stronger impact than we had originally imagined. Secondly, the computer as a tool for large number of workers was becoming a very real potential.

In other words, there was a need for work that more directly, than the first projects, was aimed at producing technological alternatives. And this work was to take place in a situation where the use of the computer, as a control instrument for the few towards the many, was being supplemented with the use of computers to support people's work.

3.1.2. Results from the second generation of projects: UTOPIA and others

These rather general concerns relating to the use of technology at the workplace, and considered as project rationale when we began the UTOPIA project, were addressed directly by the outcome of our research. Thus, following the UTOPIA project, we can say that:

- A recognition was created, both within trade unions and within research, that technological and work-organizational alternatives exist, alternatives supporting high quality products and development of skill at work.

And more broadly:

- An increased awareness and knowledge of technological and work-organizational possibilities and limitations was created.[10]
- The original "first generation" model for research/union projects was supplemented with a new model for designer/user cooperation in design projects based on worker interests.

And as before

- University level courses on the topics of the new projects were established.

This summarizes the results of the second generation of projects. It should also be noted that the Nordic employers' association considered the UTOPIA project to be such a success for the unions that they decided to mimic the project—to support their own vendor independence.

3.1.3. Status for current CRA work: Normalization

As it turned out, the trade unions—and CRA researchers, including myself—did not pursue work along the lines of the new model for designer/user cooperation in design projects based on worker interests. This was basically because the context needed to make such work a success did not come into being. In other words, we were not able to supplement the context for worker influence at the factory level, established in the first generation projects, and expanded by the second generation projects, with contexts supporting these interests at a national level.[11] Instead, since the late 1980ies our work has concentrated on developing tools and techniques for cooperation in design based in projects addressing the factory, not the national, level.

If we look at the current conditions within the trade unions, we see that they have now two decades of experience in handling technology issues in ways that include potential conflict with management. Furthermore, some technology strategies have been tried out in relation to our CRA work:

1. Local action based on central support. This strategy was developed as part of the first generation of projects and has, to a varying degree, been used since.
2. Expanding local choice through centrally developed alternative systems. This strategy was developed as part of the UTOPIA project. It basically failed in the implementation phase.[12]
3. Local co-development based on cooperative tools and techniques. This strategy is currently used in a number of projects. It has potentially a lot in common with earlier codetermination strategies, it does, however, not share their notion of common interests.

As people get used to consider technology issues as belonging to the category of potential conflict, and not to "guaranteed harmony", we see that technology issues are treated more and more like other issues of potential conflict, such as health, safety and wages.

Within computer science we see a growing set of tools and techniques for cooperative design, covering a broad spectrum of project types, including product development. In particular, CRA is now established as a valid "whole organization" approach.

This concludes the first, brief presentation of developments in conditions for and results of my research and related CRA work. The following two subsections give a more detailed account of the work with explicit pointers to the submitted papers.

3.2. *The context for design activities: Strategies and resources*

The research reported in the submitted papers is part of the area of PD as described in section two, and it has contributed to the central questions listed there. However, as described above and indicated by the subtitle: A contextual approach to design of computer artifacts, my focus has been and is different—different enough to warrant a specific set of issues.

3.2.1. Central issues

My overall concern is how to support users in influencing the development and use of computers at the workplace. The first distinction is between:

- Parallel user/worker controlled activities supplementing traditional management controlled development activities,
- Cooperative/Participatory design activities guided by user/worker interests, and
- User/worker participation in Cooperative/Participatory design activities in tradition organizational settings.

Secondly, there is the issue of

- How to organize the activities, what is the organizational basis, which groups are involved, when and how.

And, finally, there is the issue of

- How to relate PD activities to other activities in a development project, how to integrate them into the organizational basis, and how to fit the activities together on a day to day basis.

3.2.2. Results

Practical

The work has contributed to establishing users, and—in the Scandinavian setting—their trade unions, as legitimate actors [1,3]. Concrete processes, such as those described in e.g. [1, 4] and to some extent [3], provided paradigmatic cases illustrating how to organize independent trade union activities, both locally, at the factory level, and centrally, at the national and international level.

First, results emerged locally through the user controlled activities combined with negotiations, and the examples provided the concrete prototypical cases needed for local unions to initiate their own supplementary activities to influence the management controlled traditional system development projects. In Denmark in particular, trade union courses played a crucial role in supporting numerous local unions in this work ([1] and Kyng 1994b).

Secondly, the work demonstrated how trade union initiated work, on an international level, could provide major input for the technological agenda in an industry ([3, 4] and Utopia 1984).

In addition to the work described above, on the context for design itself, a number of contributions have been made to supplementary activities needed to support users, particularly workers and their organizations, in influencing development and use of computers. In [3] this is summarized in the following hypothesis:

> "The most important prerequisite for trade union participation in management's design process is a parallel and independent process of accumulation of knowledge on the part of the union." p. 40.

The discussion of this hypothesis is followed by a discussion of the need for external resources and the necessity of adapting local union strategy to the particular prerequisites.

At the same time, however, the continued applicability of this type of results depends on specific societal conditions [3], conditions that since the late nineteen eighties have deteriorated in Scandinavia and were hardly present in other countries ([3] and Shapiro 1993). The above mentioned Australian URCOT initiative constitutes a new attempt at providing external resources for worker investigations, an initiative that might rekindle the interest in this type of results and bring new input to our own work.

As the PD area developed, a number of issues emerged where further work and clarification were needed.

First, in the first and second generation of Scandinavian PD projects, such as NJMF and UTOPIA, the PD techniques used were only just emerging, and later projects had used only a limited set of PD techniques. Thus, there was a need to address:

- How one might organize projects applying a broader spectrum of PD techniques.

Secondly, researchers outside the tradition requested more reports on results as opposed to process. Thus, there was a need to address:

- What can/do interesting results—in terms of designs—look like.

Finally, different authors argued that severe limits on the applicability of PD existed. The major issue raised was:

- PD in product development.

The first issue is addressed in [12, 14, 15]. Based on two different projects, the papers describe and discuss the rationale and the experiences with applying PD techniques, such as Future Workshops, Organizational Games, Mock-up Envisionment and Cooperative Prototyping, in the one and same project.

The second issue was addressed in [14, 15]. In those two papers we present the results of the PD activities at Great Belt. These include (1) an investigation of problems and bottlenecks in daily work and cooperation resulting in a new understanding of the differences between the existing vertical information systems and the needs for horizontal support, and (2) a design of an open hypermedia system, supporting the continued use of existing applications such as word-processors and CAD systems.

The third issue was addressed as a reaction to claims about the unfeasibility of PD in product development, see e.g. (Grudin 1991b, Grudin 1991c, Järvinen 1991). The paper [15] discusses these claims and presents a case of using PD in product development. At the same time, [15] presents different degrees of embedding CRA activities in more traditional development projects.

Theoretical

As described in the submitted papers, the theoretical inspiration for our work comes from two sources: one functioning as auxiliary subject and related to a Marxist view of society and forces of change, including areas such as industrial sociology and pedagogy, and represented by writers such as (Braverman 1974, Freire 1970, Negt 1972), and another, which can be labelled social construction—as opposed to the mechanistic foundations of most computer science—including areas such as hermeneutics, and represented by writers such as (Heidegger 1962, Polanyi 1967, Wittgenstein 1953,1963) and interpretations, elaborations and supplements by (Dreyfus & Dreyfus 1986, Suchman 1987, Winograd & Flores 1986).

One of the early main insights derived from this theoretical position, is that of system development as an inquiring or learning process [1, 2, 3, 7, 8]. In [1], this insight was used to criticize the standard phase models of development for sacrificing these inquiring or learning aspects in favour of external control. In [2], this critique was expanded in relation to the system perspective, a perspective that facilitates the reduction of work to algorithmic procedures and, in general, treats humans and machines alike. As a contrast we developed the supplementary "Tool perspective", primarily as a design ideal. An ideal emphasizing the experience of the users and their possibilities for controlling the computer artifact. This design perspective was further expanded in [3] where a labour process perspective on design and use is developed and subsequently refined to a set of theses on design for democracy and skill under the label of The Collective Resource Approach to Systems Design. The theses are (p. 51ff):

- Design of computer support is design of (conditions for) labour processes.
- Labour processes cannot be reduced to information processes.
- Design use models.

- Hardware should be considered early in the design, in parallel with software, not after.
- Important aspects of labour processes—in relation to design of computer support—cannot be formally described.
- Professional experience with and knowledge of the labour process for which computer support is being designed are important in the design process.
- Professional experience with and knowledge of computers are important when designing computer support for a labour process.
- Design should be done with users, neither for nor by them.
- Mutual learning should be an important part of the work in a design group.
- Design by doing.
- Designers should restrict their activities to a few domains of application, and they should spend at least a year or two getting acquainted with a new area before doing actual design.

In [6] and particularly in [7, 8], the design theory presented in [3] was revised based on our work related to the field of CSCW (Computer Supported Cooperative Work) and emphasizing the theoretical inspiration from social construction. In [7], we summarized the position in the following design ideals[13] (p. 1f):

- Computer systems that are created for the workplace need to be designed with the *full participation* of users. Full participation, of course, requires training and active cooperation, not just token representation in meetings or on committees.
- When computer systems are brought into a workplace, they should *enhance* workplace skills rather than degrade or rationalize them. Enhancing skills means paying attention to that which is often left out of formal specifications, for example tacit knowledge. Computer systems are more than the flow of information represented in flowcharts.
- Computers systems are *tools*, and need to be designed to be under the control of the people using them. They should support developing work activities—including communication—not make them more rigid.
- Although computer systems are generally intended to increase productivity, they also need to be viewed as a means of increasing the *quality* of results. More output does not mean better output. The double emphasis on productivity and quality raises new questions for the design process.
- The design process is a political one and includes *conflicts* at almost every step of the way. Managers who purchase a system may be at odds with workers who are going to use it. Different groups of users have different needs and system designers often represent their own interests. If the inevitable conflicts are pushed to one side or ignored in the rush toward an immediately workable solution, that system may be dramatically less useful and continue to create problems.
- Finally, the design process highlights the issue of how computers are used

in the context of work organization. We see this question of focusing on how computers are used, which we call the *use situation,* as a fundamental starting point for the design process.

The work on the above framework was inspired mainly by social construction theory. As a complement, the paper [11] revisits the earlier, more trade union oriented frame of reference and discusses the recent developments presented in [7, 8, 9]. In doing so, it introduces the notion of techniques unsuited for strict external, e.g. managerial, control.

3.2.3. Related work

Most of the work on contexts for design has been done by people related to CRA. This is illustrated by the recent Participatory Design Conference in North Carolina, sponsored by Computer Professionals for Social Responsibility in cooperation with ACM. In the two sessions relating to contexts for design—"Scandinavian Participatory Design: From trade unions to organizations" and "Power relations: Structure and dynamics"—three out of three and two out of three papers, respectively, were by authors related to CRA (Trigg *et al.* 1994). The one exception was a paper by Gärtner and Wagner (1994). Like the aforementioned URCOT initiative Gärtner and Wagner are concerned with worker influence on system development and introduction based on a trade union tradition. Their work shares with our CRA work the emphasis on the contexts for design as well as on worker controlled resources.

When we look outside the PD area it is obvious that CRA, including our later developments in Cooperative design, has established close ties with both the social sciences and the humanities. Early CRA work included writings related to attempts at establishing a new Working Life Science, see e.g. (Sandberg 1979). Central questions for this type of research were how economic and social structure create possibilities and limitations for change—particularly changes in a democratic direction—and how research itself may play an active role. This type of work, as well as that of Freire (1970), Negt (1972) and Braverman (1974), played an important role in the formulation of the research strategy of the Scandinavian PD projects, particularly in the formulation of the technology strategies "local action based on central support" and "alternative systems expanding local choice".

Also in Scandinavia, "Work Research" increasingly deals with issues related to users and computers. Among the issues addressed are what organizational characteristics further transfer of knowledge from old to new technology (Sørensen 1994).

Recently, researchers with a background in ethnography have contributed with important insights into the use of computers, see e.g. (Heath & Luff 1992, Suchman 1987, Wynn 1979b), and several papers by researchers from Xerox in (Trigg *et al.* 1994). Ethnography shares with CRA the emphasis on the situatedness of knowledge—and has influenced this emphasis in our CRA work. In the US, particularly Lucy Suchman and her group at Xerox PARC have conducted a series of projects that have contributed to a professional and scientific re-orientation, and have moved field-studies in a still more participatory direction. Suchman was also instrumental in presenting the Scandinavian CRA to a US audience and in creating space for such work at US conferences. On a different level, the above mentioned organization, Computer Professionals for Social Responsibility, have played an important role as a forum for discussions of alternatives to a mechanistic view on the use and development computers.

Also in the field of HCI, recent work has stressed the need to move from the laboratory into real life (Bannon 1991, Carroll 1995) and writers with a background in CRA have contributed directly to the field of HCI (Bødker 1991).

Finally, it should be remembered that participatory design or early user involvement is becoming part of the agenda in a number of areas, including Information Systems and Software Engineering. However, it is the exception rather than the rule that this work involves contributions to the context of design. The most important exceptions are those found in Nordic IS research, such as the Finnish Knowledge and Work project (Nurminen *et al.* 1985).

The areas listed above share important aspects with CRA work on the contexts for design. However, these aspects are, when viewed from the areas ethnography and HCI themselves, mainly related to work on design itself, not on contexts.

3.3. Design in context: Techniques and tools

3.3.1. Central issues

Within the context presented in sections 3.1 and 3.2 above, the three main issues for my research on techniques and tools have been:

- how to support user contributions based on user interests,
- how to ground design activities in the work to be supported, and
- the influence from cooperation itself, i.e. the consequences of viewing design as cooperation between people with different backgrounds.

Particularly the first issue relates directly to the question of context: we want to develop tools and techniques that influence system development but are not a priori considered to be an integrated part of a traditional system development project (cf. the first distinction in subsection 3.2).

3.3.2. Results

Practical

The initial motivation for my work with these issues was a number of unsuccessful attempts to use existing description and demonstration oriented approaches in PD.

The techniques and tools presented in the submitted papers are mainly non-computer-based. Using non-computer-based techniques and tools in PD has several practical advantages:

- there are no substantial costs—and virtually no difficulties—associated with getting the necessary tools,
- technical details, e.g. of a new version of a prototyping tool, do not get in the way, and
- PD activities can draw on the users initial knowledge of and familiarity with the tools (e.g. pen, paper, scissors and cardboard).

Furthermore, the non-computer-based techniques and tools support continuing, active engagement from all participants, users and professional designers alike. Through the interest created by the hands-on activities these techniques and tools overcome some of the difficulties with rapidly decreasing user engagement and eventually lack of participation experienced in some earlier PD activities, see also (Ehn & Sjögren 1991).

Turning to the techniques and tools themselves, there is first of all the use of mock-ups. Mock-ups were introduced in [2] and later further developed in [3, 4, 5, 9, 12]. The use of mock-ups allows users in PD to experience simulated use of the computer artifacts being designed and to participate in the original construction as well as in modifications of the mock-up. As opposed to the usually unfamiliar task of reading descriptions, users engage in (simulated) work and thus make direct use of their work-related knowledge and experience. As developed in the UTOPIA project [2, 3, 4] the use of mock-ups supports both user-interface aspects and the structuring of the domain model.

The organizational tool kit is another useful outcome of the UTOPIA project. Through the use of problem domain specific icons for functions, tools and materials it supports users in describing work organization [4]. As opposed to traditional flowcharts, the basic building blocks are well-known to the participating users, and thus ease of use increases and initial learning time is reduced.

In addition to these PD techniques developed in the early eighties, two other techniques were part of our tool box at that time: "True Stories" and workplace visits. Originally, we developed them in the DUE project in the late 1970ies but they were not discussed in research papers until 1988 [5]. True stories present design relevant information, such as critique of existing artifacts, in a generally understood form, that of a story and hereby makes it directly accessible to the users. Workplace visits provide users with access to relevant experiences with computer use through dialogues with people with a similar background.

Organizational Games are another important result of Scandinavian PD. Originally developed by Ehn & Sjögren, as a self-contained technique for developing work organization to make better use of new, but already installed computer support (Ehn & Sjögren 1991), it has been applied in a number of different contexts for more "traditional" PD purposes. Thus, [12] presents how Organizational Games are used as one out of several PD techniques in the AT project, a project on a "whole organization PD approach" to the development of computer support and work organization.

The techniques and tools described above have been in use since the mid eighties. And they—as well as our understanding and their theoretical underpinnings—have undergone a continuous development based on our own use of them in a number of different projects [3, 4, 5, 9, 12].

Since the introduction of the use of mock-ups we thus have:

- broadened the scope of their use, from the original focus on production type work to e.g. supervision and administration,
- developed the technique e.g. by integrating it with initial analysis/mutual learning, Future Workshops and cooperative prototyping and by integrating the use of computer-based materials in mock-ups.

We have also developed a number of cases which illustrate how the different PD techniques and tools may be combined in a development project, and how they relate to the non-PD activities [12, 14, 15, 17].

In the last few years, we have gained increased knowledge of and experience with the use of the techniques and tools by designers outside the tradition. In particular, we have learned about the breakdowns that they experience [15]. Through this, we have seen the need for developing a much more explicit treatment of that which makes the mock-ups and prototypes useful in PD activities: the work being supported. In earlier CRA papers on mock-ups and prototypes, focus had been on the artifacts being built. In papers such as [7, 13], we had stressed the importance of what we called the use situation: the concrete, situated use of existing computer support—use in context so to speak. And when we ourselves used mock-ups and prototypes, our understanding of the so-called use situation was a crucial background for doing this. As it turned out, our presentations in papers such as [7, 13] and (Bødker & Grønbæk 1991b) were inadequate in conveying our ideas on using mock-ups and prototypes in PD workshops to others. To paraphrase [8]: our presentations were not able to bridge the gab between the understanding of non-CRA designers and the ideas of simulating work using mock-ups. Thus we supplemented our work with an explicit treatment of the "use" or "work" part of the picture. The central idea is to develop the rather vague notion of use situation into the pair: work situation and use scenario. I introduced the pair in [15] and further developed it in [16, 17]. Work situations capture relevant aspects of existing situations whereas use scenarios indicate how computer support and changes in work organization may improve upon work situations. Through the corresponding artifacts, "work situation descriptions" and particularly "use scenario descriptions", the grounding of the use of mock-ups and prototypes in the work of the users is made both concrete and visible. Finally, this pair also made the notion of "example data"—i.e., data based on the work situations that make a mock-up or prototype suit a specific use scenario or set of scenarios—more concrete and thus more understandable.

As described in e.g. [15] the techniques and tools presented above have been used successfully in the sense that major contributions have been made by users in projects applying them. However, the "new" artifacts, the work situation descriptions and the use scenario descriptions, have to be put to use the right way to make sense—just like the mock-ups and prototypes. With respect to the descriptions, the main point is that they are intended to set the stage for the use of mock-ups and prototypes for people who already know the work in question, they do not make much sense to outsiders, people with no prior knowledge of the work and organization in question [16, 17].

The tools and techniques presented above have been developed in a number of major projects where users and researchers have cooperated in action research type activities. In these projects existing, traditional techniques and tools have been applied, and to the extent that they did not work satisfactorily we have tried to develop alternatives more or less on the fly. Those alternatives that worked in the concrete setting of the projects were then later reported on, cf. e.g. [4, 5], and their theoretical foundation gradually expanded, together with continued experiences from use of modified versions.

Most of the CRA techniques and tools have stood the test of time, i.e., as we developed their theoretical underpinnings we have been able to further develop the techniques and tools so that they continue to be in the front of current PD techniques and tools, cf. e.g. (Muller & Kuhn 1993, Schuler & Namioka 1993). There are, however, some exceptions—techniques and tools that did not develop as we originally hoped. I conclude this presentation of practical results with a short discussion of these.

First of all, there are the system description techniques and tools mentioned in [1]. In retrospect the cooperative nature of the description technique provided substantial improvements over non-cooperative techniques such as the "System Description for Users" discussed in [1]—improvements, which made enough of a difference in the projects applying these techniques to justify their use. However, the later techniques and tools, primarily mock-ups and the associated scenarios, are much better suited for PD than the description based techniques and tools presented in [1].

Secondly, there are the techniques based on the derivation of demands for changes based on goals. Such techniques were developed in the first generation projects, such as DUE [1], but subsequently techniques developed from the Future Workshops of Jungk and Müllert (1987) supplanted the goal based techniques. As discussed in [5] the Future Workshop based techniques allow people to work on concrete criticism and concrete, positive visions, without the intermediate step of formulating goals.

Finally, there are the computer-based techniques and tools. With respect to techniques, important results have indeed been achieved, cf. [14, 15] (and (Bødker 1987, Bødker & Grønbæk 1989, Grønbæk 1991) for related results by some of my colleagues in Aarhus). But we are still a long way from realizing the vision outlined in [6] mainly because the development of computer-based tools for PD was more difficult than we anticipated. Computer-based tools are, however, at the center of our current research efforts and I will return to the issue of the slow progress in section 5, Future Work.

Theoretical

Now let us turn to a discussion of the concepts used and developed in the submitted papers. First of all, there is the notion of cooperation itself. In [1] the characterization of system development as an inquiring process, producing new understanding was used to ague for the need of cooperation in design, i.e. for PD and to explain the problems of traditional approaches to user involvement. In [3] cooperation was also discussed from the point of view of the competencies necessary for design of computer support, and the argument was summarized in the thesis:

> "Design should be done with users, neither for nor by them."

[3] also introduced two other concepts which are basic to the understanding and further development of our techniques and tools: the notion of family resemblance derived from viewing design as a language game, and that of "hands-on" experiences, in [3] introduced under the

label "design by doing", see also [5, 6, 9].

Within the general theoretical framework of CRA "involved action" is viewed as primary, compared to "detached reflection", and thus new insight must be based on—be grounded in—involved action. At the same time, our focus is on the development of new computer-based artifacts and new work practices using these artifacts. This constitutes a significant challenge, a challenge that is not met by most techniques and tools outside the PD area because they assume the use of techniques and tools, such as requirement specifications, in ways that do not relate to the experiences of the users, that are not grounded in involved action.

Family resemblance

The techniques and tools presented above meet the challenge by creating a family resemblance between the work experiences of the users and the design situations. In the design examples discussed, e.g. in [9], family resemblance is created between, on the one hand, the situations and the artifacts involved in these design examples and, on the other, work situations, tools and materials that are well-known to the participating users. A resemblance that is sufficient to allow the users to make sense of the design situations by drawing on their experiences with involved action in work and thus allow them to act involved in the design situations, simulating work with simulated computer support.

In addition, materials and tools used to build mock-ups—such as paper, cardboard, plywood, nails, pens, scissors, and hammers—are well-known to the users. Discussing and making changes to a mock-up is thus possible by drawing on the family resemblance with activities such as drawing your own favourite house, or building a doll house, activities that are well-known to most.

The first aspect of family resemblance in design situations, supporting users in drawing on experiences from work, in understanding and using the computer artifact being designed, has also been successfully applied using computer-based prototypes, cf. e.g. [14, 15]. However, the second aspect of family resemblance, that of supporting an understanding of the space of possibilities and limitations for change, has been more difficult to achieve with computer-based tools and techniques than we originally imagined.

Hands-on experiences

The second notion introduced in [3] was that of hands-on experiences. Inspired by Polanyi (1967), Heidegger (1962) and especially Winograd & Flores (1986) and Dreyfus & Dreyfus (1986) we developed a design approach based on involved action, on the use of artifacts as a basis for reflection on them [5, 7, 8, 9]. The main point is that fluent activity, particularly expert performance, is not based on explicit rule following, and that crucial aspects of our knowledge is not explicit. In order to find out in what ways an emerging design is effective in supporting work and in what ways it fails detached reflection is insufficient. Hands-on experiences from trying to use the computer support are needed, and using mock-ups and prototypes to simulate work with the computer support being designed is one way of getting these hands-on experiences. However, viewing involved action as primary and detached reflection as secondary does not imply that detached reflection does not play an important role in our design activities—only that the role is different. In traditional prototyping for example, where new designs are only demonstrated to the users prior to soliciting contributions from them, we may characterize their reactions in the following way:

> "As long as the users do not experience what it would be like to work with a system under development, their contributions will mainly be based on prejudice, that is on pre-judgement." [16, pp. 1f – in manuscript]

User contributions are in such cases grounded in involved experiences with other artifacts. Thus the more innovative—or the more different from these other, existing artifacts—the emerging artifacts being designed are, the less appropriate the user contributions are likely to be.

On the other hand, when detached reflection follows a breakdown in involved action—in the use of a (simulated) computer artifact—then it is possible to base discussions on that breakdown and the use that led to it [8, 9].

As mentioned above, the main quality of the mock-ups and prototypes in relation to PD is their ability to support users in bringing their work related knowledge and experience to bear in the design process. But for this to happen, it is not enough that users carry out simulations of (any kind of) work. It is necessary to support the users' specific work related knowledge and experience. As described in [15, 16, 17], this is done through the creation of mock-ups (and/or prototypes) with example data and the preparation of a set of use scenarios based on initial analysis and mutual learning. Originally, we used the concept of "use situation" [7, 13] in talking about grounding design in use. However, this is basically an analysis-oriented term—as opposed to design-oriented—and as described in [15] it was insufficient in explaining the reasons for grounding the "hands-on" activities in the work of the users. To this end, the basic distinction was made between work situations, capturing relevant aspects of existing situations, and use scenarios, setting the stage for exercising a mock-up or prototype. These concepts were then used to explain the relation between the work being studied, as part of a design project, and the hands-on activities, including how to prepare mock-ups/prototypes and example data.

In addition, to the basic distinction between existing situations and stage setting scenarios, a number of supplementary categories were introduced. With respect to "the existing" these were: reminders of initial study, work situations and work situation overviews. With respect to future use these were: use scenarios, use scenario scripts, exploration/requirements scenarios and explanation scenarios. The first five of these seven categories relate rather directly to the hands-on design activities, whereas the last two categories are closer to the way requirements and scenarios are treated in other, related approaches.

Descriptions, in the first five categories, are characterized by their open-endedness: they are in no way intended to be self contained, but to be used by people (users and designers) who know the reality they refer to. Thus, when needs arise to go beyond such descriptions, this is simply done by revisiting this reality.

Contrary to this, exploration/requirements scenarios are closed in the sense that they are intended to supply the use details needed to discuss whether or not some established requirements are met by the current technical possibilities. One of the advantages of this scenario approach to requirements is that in this way it is straightforward to keep track of the relation to the use scenarios that form the basis of the requirements.

Explanation scenarios are of a third kind. Like a number of other uses of scenarios (cf. the introduction in (Carroll 1995) and (Campbell 1992)) they are rather detailed accounts of projected future use of a system. Explanation scenarios are used to capture some of the hypothesizing involved in developing a design, but these scenarios of projected use are not intended to be the last element in a movement towards bringing use into the design reasoning. Just as with the exploration/requirements scenarios, the scenario form makes the relation between explanation and use scenarios straightforward, which in this case facilitates later evaluation of the hypotheses of the explanation scenario through hands-on exploration based on a use scenario.

In [16], the open-endedness of the artifacts representing work related understanding in the design process is also used to explain why the concept of user-proxy (Hughes *et al.* 1993) is not useful: it freezes the level of understanding to that established by the user proxy in analysing the work of the users, since the user-proxy, in the user-proxy/designer activity, has no first hand access to user experience.

The papers [17] and [10] have taken the CRA work described above and investigated two supplementary aspects.

In [17] the different design artifacts and their use are presented and discussed through viewing them as representations. In general, representational artifacts can be seen as having representational and non-representational aspects. However, for some representations, such as prototypes, there is potentially a third category, that of actual aspects. This category may be used to explain some of the difficulties involved in using computer-based design artifacts—as opposed to non-computer-based.

Finally, [10] develops the notion of tailoring as "design in use", which supplements the "use in design", the hands-on exploration described above. The paper develops a terminology of tailoring covering both the changing of artifact behaviour itself and the needs of people engaged in making such changes. The activity of changing the behaviour of a computer artifact is characterized as "choosing", "constructing" or "altering". Using this classification, it is then argued that the provision of high-level, application oriented building blocks for system construction and modification can be viewed as transforming cases of "altering" into cases from the much simpler category of "constructing".

3.3.3. Related work

Numerous researchers in the PD area are working on techniques and tools, mainly non-computer-based, similar to those described above. These include PICTIVE (Muller 1991), Storyboard Prototyping (Andriole 1989), Cooperative Interactive Storyboard Prototyping (Madsen & Aiken 1993) and CARD, (cf. Muller *et al.* in (Carroll 1995)).

PICTIVE is an example of non-computer-based techniques and tools/materials emphasizing the immediate understandability of the tools/materials to the participating users. Focus is on the design of interfaces, and the process is basically a design discussion, where the interface is gradually designed in the form of concrete screens or windows. Techniques like PICTIVE do not support the kind of hands-on experience possible with mock-ups or prototypes, but the cooperative nature of the process combined with the familiarity of the tools/materials make PICTIVE and similar approaches well-suited for the design of computer support that is not intended to radically change the work in question.

The Storyboard Prototyping technique takes a task view on the use of an emerging system. Through a sequence of screens support of a specific task is demonstrated to the users as a basis for commenting on the design.

Cooperative Interactive Storyboard Prototyping, CISP, is a development of Storyboard Prototyping based on the ideas from CRA presented above, cf. (Madsen & Aiken 1993). The modification of Storyboard Prototyping is intended to solve two problems: first, it adds *use* of the prototype—i.e. hands-on experience—to broaden the basis of user contributions; secondly, it uses a computer-medium for the storyboard prototype, in a way intended to support modifications to a storyboard in a design session. This is done through developing domain specific building blocks, which are then manipulated in the design sessions. I return briefly to this issue below in section five on Future Work.

The CARD technique combines ideas from Storyboard Prototyping and Organizational Games as developed by Ehn and Sjögren. In the CARD technique, cards are used to represent elements in a work flow, such as screens, and then used for a cooperative design of task flows to be supported by the system being developed.

Scenarios are also attracting increased attention in recent years. When comparing our use of scenarios with other uses, as listed in e.g. (Campbell 1992, Carroll 1995), the major difference is that our scenarios are used mainly to set the stage for cooperative design workshops. They are not used as direct sources for evaluation or design, but contribute *via* workshops. Furthermore, our scenarios are grounded in existing work situations in real organizations.

The grounding in real work situations at specific user organizations is also central in a number of techniques from ethnography. Indeed, our own cooperation with people working in ethnomethodology influenced our views on the issue. However, ethnographic techniques, as presented in e.g. (Suchman & Trigg 1991, Wynn 1991), are still rather expert-based. The techniques emphasize the complexity and situatedness of work but the way this is brought to bear on design is through studies done by experts, not through cooperative techniques.

Finally, it should be noted that although the four techniques listed in the beginning of this subsection share a number of "technique and tool aspects" with those of the submitted papers, a philosophical rationale of the kind mentioned above is not present. Examples on related "philosophical rationales" may be found in (Naur 1985) where Naur uses Ryle to argue for the primacy of the conceptual framework building or in his words theory building, in a group of programmers, during the development of a program over the "external" design or program itself, and the indispensability of the humans having the theory in the continued use, modification etc. of the program.

4. On methods

Methods is a low key subject in our science, to say the least. As an example of a positive view of such a state of affairs I quote C. Wright Mills, who wrote in 1952:

> "Method and theory are like the language of the country you live in: it is nothing to brag about if you can speak it, but it is a disgrace, as well as an inconvenience, if you cannot", (1980, p. 64).

On the other hand, in times of emerging reorientation, and that is what I have argued for holds for computing science, we might conclude that we face a task somewhat like developing a new language, speaking it, and at the same time try to communicate with people speaking different languages. Indeed, the approach of my colleagues and myself can be presented using this analogy. However, the major point is that creating or contributing to the creation of new situations, situations that from our point of view were relevant and interesting, usually preceded our work on developing a new "language", i.e. parts of new methods. In this way our confrontation with and needs in these situations directed our work on developing new ways of working, new methods. From the outset, this work too was low key, rather similar in flavour to the following quote from Mills:

> "In brief, "methods" are simply ways of asking and answering questions, with some assurance that the answers are more or less durable. "Theory" is simply paying close attention to the words one uses, especially their degree of generality and their interrelations." (1980, p. 63).

On the other hand, it was at the core of our methodological concerns that different "ways of asking and answering questions" really made a difference.

In our work we introduced a close and manifest coupling between practical and theoretical work, together with a questioning and reinvestigation of the basis for and consequences of the prevailing notion of value-freedom. In the submitted papers, a basic methodological theme can be described as an insistence on concrete experience as the basis for theoretical work—both within research and system development, see [3, 7].

On a practical level this has led us to organize major parts of our research as projects involving close cooperation with users—the people who together with us can "produce" such concrete experience [1, 3, 4, 14, 15]. Furthermore,

- when the formation of new research areas, such as PD and CRA, is taking place, and
- when the existing framework—both with respect to research and system development—does not accommodate major concerns, such as user interests,

then such insistence on concrete experience implies a major effort directed towards creating activities at different levels that may eventually produce that experience—in the case of CRA, experience at the level of contexts for design as well as design itself. Thus initially, during the time when CRA was being shaped, large action research projects played a dominant role in our work, cf. [1, 3, 4] (as well as (Bjerknes & Bratteteig 1988, Ehn & Sandberg 1983, Nygaard 1979)). Gradually, these projects created experience with a range of possibilities for workers and their trade unions to influence the development and use of computers. Concrete experience that in turn informed theoretical investigations—and subsequently created a context with new possibilities for the development of techniques and tools supporting user influence in design. During the following years of research along these lines, projects providing concrete experience continued to play a major role but now often in the form of experimental system design projects as opposed to the more spectacular action research projects. As the body of experience grew and as the trends in society, on which much of the initial work had revolved, weakened the pressure for continued work producing concrete experience lessened and at the same time the need for theoretical work up increased. Thus theoretical work such as (Bødker 1991, Ehn 1989, Grønbæk 1991, Mogensen 1994) played an increasing role in CRA since the mid nineteen eighties. Finally, in the current stage of my research—and that of our group in Aarhus—we have come to a point where a major effort is directed towards realizing a computer-based development environment (Kyng 1994a). Viewed from the outside it might seem possible for us to have conducted this experimental work much earlier. However, without the context being in place, without the experiential basis of the earlier work, this would have been a different endeavour.

Looking back and caracterizing the work of my colleagues and myself it is obvious that we have moved outside the boundaries of what is traditionally considered methods with a solid science foundation. But I write “considered”, because according to the view presented here a “solid science foundation” is more myth than reality and, consequently, no method rests on such foundations because no science does as beautifully argued by Peter Naur in (1992). What we have done is an attempt to develop new methods suited to our scientific pursuit in computing science. We have not followed traditional science methods but in important respects we are even further removed from social science and humanistic methods: our basic interest is constructive, not “just” to understand and explain. Our basic approach is experimental, not laboratory based, but interventionistic.

5. Future work

Since the formation of CRA, beginning in the early nineteen seventies, the original vision of supporting a trend towards increased industrial democracy with the trade unions as a major player has lost its unifying role. Outside Scandinavia trade unions have never been thoroughly involved in a PD based strategy until the recent establishment of URCOT in Australia, and within Scandinavia the situation proved to be more complex and less favorable than we, as proponents of the “new Scandinavian model” for technological research and development, had hoped for—although not less favorable than we had feared [3]. We never considered CRA to be the driving force in these developments. CRA was—and is—seen as supporting existing trends in society, and to the degree that these trends change so do the conditions for how CRA may support and interact with actors outside the world of CRA. Thus, while the original concern for democracy is still there, new, more diverse and more intricate partnerships are currently being sought and the outcome of this is still uncertain. Looking at current research issues and themes one may characterize the situation as one of re-evaluating several of the original strategic premises without having found the same kind of clear direction as in the nineteen seventies.

This is illustrated by the call for participation from the Third Biannual Conference on Participatory Design (Chapel Hill, North Carolina, October 1994, see also (Trigg *et al.* 1994)), which—among others—lists the following topics:

- the ethics of participation, e.g. obligations to management versus workers, and designers' responsibility for what happens "down stream",
- the politics of participatory design, e.g. identifying "stakeholders" over the course of a project,
- relations of participatory design approaches to the labor movement, e.g. to labor unions' own technology development and analysis efforts,
- the theoretical roots of participatory approaches to design, e.g. connections to Action Research.

As mentioned, these topics are on the international agenda of PD research and they reflect a renewed focus on contextual issues. At the same time, they are high up on my own research agenda and formed an important part of my opening keynote at the above mentioned PD conference. However, my current research on these topics is emergent and not yet in a form ready to publish.

When we turn from the context, the strategies and resources, to the techniques and tools the situation is simpler—and the major determining factors are more within the realm of participatory design and traditional scientific concerns.

From the formation of CRA and until today most work in this area focuses on low tech tools, e.g. (Muller *et al.* in Carroll 1995, Ehn & Sjögren 1991, [1, 2, 9, 16] and Kensing & Madsen 1991). The work that involves computer-based tools either exclude major modifications from the participatory activities themselves, cf. (Bødker & Grønbæk 1989, Bødker & Grønbæk 1991a, Grønbæk 1989, and [14, 15]) or it focuses on very limited areas such as interface design for video cassette recorders (Madsen & Aiken 1993). This situation is unsatisfactory for three reasons.

First, PD, without computer-based tools, will prevent increasing integration of 1) direct user interaction with the computer artifact being designed and 2) the activities modifying the artifact. Particularly, when new forms of computer support are being designed such direct interaction is needed in order to engage the tacit knowledge of the users, and the non-participatory loop constitutes an unnecessary time-delay.

Secondly, the understanding of possibilities for change that the users have when low tech tools are used, is an understanding of the possibilities exactly with low tech tools and materials, i.e., in the domain of the representation. It is not an understanding in the domain of the computer artifact itself.

Finally, computer-based tools have the potential of improving the efficiency in modifying a mock-up/prototype compared to the current process of changing dozens of screen-designs on paper or slides, see also [9].

It is not the case that attempts have not been made at developing such support. There are, however, four major obstacles:

First, the tools must provide adequate support for the users in understanding the space of possibilities and limitations for change of the computer artifact being designed.

Secondly, the tools must provide adequate support for the users in understanding the different aspects of the current representation of the computer artifact being designed: what is representational, what is non-representational and what is actual.

Thirdly, it requires a combination of people skilled in PD and people skilled in the development of advanced computer-based tools.

Finally, the tools should support design as a cooperative activity.

The first issue has been on the Scandinavian PD agenda for a long time. Kristen Nygaard, in his keynote speech (1984), introduced the notion of profession oriented information processing languages, and indeed the pioneering object-oriented programming language Simula 67 introduced the idea of a general purpose language serving as a substrate for special application languages (Dahl *et al.* 1971, p. 2f). As an example, Simula realised the discrete event simulation capabilities as a special class. However, with the exception of tools for user interface design, such as (Grønbæk *et al.* 1991, Madsen & Aiken 1993), there has not been much progress since SIMSET and SIMULATION of Simula 67. It is our hypothesis that the reasons for the very slow progress are a combination of lack of attention to the three other points listed above, including inadequate understanding of the issue of developing computer-based tools that are understandable to users.

Our current work—organized in the DEVISE centre—is aimed at creating an environment for experimental system development (Grønbæk & Knudsen 1992, Kyng 1991). In doing so, we draw on and develop ideas from computer supported cooperative work in order to support design as a cooperative activity. Secondly, our group includes people with a background in PD as well as people with a background in creating object-oriented programming environments. Finally, we are developing our original approach to user understandability by combining ideas from object-oriented domain modelling with work on application frameworks (Kyng & Nielsen 1993) and hypermedia (Grønbæk & Trigg 1994). In this endeavor it seems reasonable to expect that we can benefit from a number of other efforts directed at developing alternatives to the traditional, functionalistic approaches, e.g. the work of Goguen and colleagues at the Oxford Centre for Requirements and Foundations (Goguen 1992), as well as work on object-orientation, e.g. that of Rosson and Alpert on cognitive consequences of object-oriented design (1990).

Acknowledgements

In the early 1970ies I had suspended my computer science education to study psychology and to work in the student movement. At the Institute of Psychology I met Kristen Nygaard and he convinced me that there was more to computer science than computers. In 1974 Kristen began a year long association with the Department of Computer Science, and my discussions with Kristen as well as his support, which included participation in the steering committee of the DUE project, were instrumental for my work in that period.

In the first ten years after I graduated most of my time was spent working on the DUE project and then UTOPIA. Several people were crucial in carrying through this work. I am particularly indebted to Lars Mathiassen and Pelle Ehn. Each of them has that combination of research curiosity, perseverance and project management skills, which is fundamental for the success of such novel types of projects. In addition, Lars played the crucial role of discussant and critic during my write up of this overview.

When the UTOPIA project ended in 1984, for the first time in ten years, I was in a situation where writing for a worker/ trade union audience was not my top priority and thus, by trying, I learned that writing for a scientific audience—in English—is something different. In the following ten years I practised, and I owe a lot to my co-authors for their support in that process—language- and content-wise. Joan Greenbaum, who suggested the writing of the book Design at Work, plays a special role in this: she brought a new, non-Scandinavian perspective to our work and through our co-authorship of three chapters she showed me what natives can do with the American language. Subsequently, Jonathan Grudin has struggled untiringly to improve my attempts at writing in American.

Since 1990, my work has been part of the DEVISE centre. The group, which now consists of more than 30 people, continues to make my work environment so exciting, stressing, interesting and challenging that taking out time for writing overviews like this is virtually impossible.

Thus, I thank my wife, Solvej, who thinks writing overviews like this is more important than yet another research proposal and therefore made it happen.

Notes

[1]In fact part of this debate is reflected in the choice between "computer" and "computing" science: the term computer places focus on the machine, whereas computing emphasizes the use of the machine.

[2]Throughout this overview paper, references to the papers that I have submitted for the degree doctor scientiarum are given in square brackets, i.e. "[...]". They are listed immediately after this notes section. The other papers referenced are listed in the end of this paper.

[3]Referring to the managers of people using computers as "users" is misleading and has not been done in Scandinavian work related to PD. Thus my choice is really between 1) not using the term "users" at all, and only talk about "end-users," or 2) using the term "users" to denote "people using computers." In this overview paper I do the latter.

[4]All examples are from paper and panel sessions at CHI '94, Boston 1994.

[5]Other important contributions in this area are those of Markku Nurminen (1988), Christiane Floyd (1987), Winograd and Flores (1986), and Dreyfus and Dreyfus (1986). However, for the purpose of this overview paper it suffices to say that Nurminen like Bansler presents a tricotomy whereas the other three contrast a traditional view

with its "negation" in a number of dimensions.

[6] Recently, his characterization of the information theoretical school has been criticized, but not in ways that involve the characterization of the other two schools (Dahlbom 1993, pp. 21ff).

[7] The later, Norwegian Florence project 1984-87, is another example on moving from the "workers corrective" to a managerial system development effort, to developing alternative systems (Bjerknes & Bratteteig 1988).

[8] In this subsection, 3.1, I do not attribute results to specific papers. This is done in the following two subsections, 3.2 and 3.3.

[9] Over a period of 13 years, 300-500 people per year took a one week course on local union work in relation to computer use (Kyng 1994b).

[10] The increase in awareness and knowledge was partly due to a number of Nordic conferences organized by the graphical workers unions and their decision to produce 70.000 copies of the final report from the UTOPIA project (Kyng 1994b).

[11] One first indication of this was the failure of the Swedish state-owned company Liber/TIPS to produce a commercially viable system based on their cooperation with the UTOPIA project (Ehn 1993, p. 58).

[12] However, as listed in the first bullit above, the work in the UTOPIA project itself managed to demonstrate that "technological and work-organizational alternatives exist, alternatives that support high quality products and development of skill at work".

[13] Theses and design ideals like these do not make sense in themselves, they are not self-contained, closed entities but need to be grounded in the experience of the reader. Thus, in the papers presenting them concrete examples play an important role. For that grounding, I refer the reader to the papers.

Submitted papers

[1] M. Kyng & L. Mathiassen (1982). Systems development and trade union activities. In Bjørn-Andersen, N. (ed.), Information society, for richer, for poorer. Amsterdam North Holland, pp. 247-260.

[2] P. Ehn & M. Kyng (1984). A tool perspective on design of interactive computer support for skilled workers, in M. Sääksjärvi, (Ed.), Proceedings from the Seventh Scandinavian Research Seminar on Systemeering, Helsinki, pp. 211-230 + app.: pp. 231-242.

[3] P. Ehn & M. Kyng (1987). The Collective Resource Approach to Systems Design. In G. Bjerknes, P. Ehn and M. Kyng (Eds.): Computers and Democracy - a Scandinavian Challenge, Avebury, Gower Publishing Company Ltd., Aldershot, England, pp. 17-57.

[4] S. Bødker, P. Ehn, J. Kammersgaard, M. Kyng, & Y. Sundblad (1987). A Utopian experience. In G. Bjerknes, P. Ehn and M. Kyng (Eds.): Computers and Democracy - a Scandinavian Challenge, Avebury, Gower Publishing Company Ltd., Aldershot, England, pp. 251-278.

[5] M. Kyng (1989). Designing for a Dollar a Day. In Office: Technology and People, 4, pp. 157-170.

[6] S. Bødker, P. Ehn, J. L. Knudsen, M. Kyng, & K. H. Madsen (1988). Computer Support for Cooperative Design (Invited Paper). In Proceedings of the Second Conference on Computer-Supported Cooperative Work, Sept. 26-29, Portland, Oregon, USA, pp. 377-394.

[7] J. Greenbaum & M. Kyng (1991). Situated Design. In J. Greenbaum & M. Kyng (Eds.). Design at Work: Cooperative Design of Computer Systems, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 1-24.

[8] S. Bødker, J. Greenbaum & M. Kyng (1991). Setting the Stage for Design as Action. In J. Greenbaum & M. Kyng (Eds.). Design at Work: Cooperative Design of Computer Systems, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 139-154.

[9] P. Ehn & M. Kyng (1991). Cardboard Computers: Mocking-it-up or Hands-on the Future. In J. Greenbaum & M. Kyng (Eds.). Design at Work: Cooperative Design of Computer Systems, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 169-195.

[10] A. Henderson & M. Kyng (1991). There is no place like home - continuing design in use. In J. Greenbaum & M. Kyng (Eds.). Design at Work: Cooperative Design of Computer Systems, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 219-240.

[11] M. Kyng in cooperation with J. Greenbaum (1991). Cooperative Design: Bringing together the practices of users and designers. In H.-E. Nissen, H. K. Klein, & R. Hirschheim (Eds.): The Information Systems Research Arena of the 90's, North Holland, Amsterdam, pp. 405-415.

[12] S. Bødker, K. Grønbæk, & M. Kyng (1993). Cooperative Design: Techniques and experiences from the Scandinavian Scene. In A. Namioka & D. Schuler (Eds.), Participatory Design. Hillsdale, NJ.: Lawrence Erlbaum Associates, pp. 157-175.

[13] M. Kyng (1991). Designing for cooperation—cooperating in design, Communications of the ACM, 34, 12, pp. 64-73.

[14] K. Grønbæk, M. Kyng & P. Mogensen (1993). CSCW challenges: Cooperative Design in Engineering Projects, Communications of the ACM, 36, 6, pp. 67-77.

[15] M. Kyng (1994). Scandinavian Design: Users in Product Development. In proceedings of CHI '94, Boston, ACM, pp. 3-9.

[16] M. Kyng (1995). Creating Contexts for Design (Computer Science Department, Aarhus University, pp. 1-24). In Carroll (Ed.), Scenario-Based Design. John Wiley (scheduled for April/May 1995).

[17] M. Kyng (1994). Making Representations Work. In L. Suchman (Ed.), Representations of Work, Monograph HICSS-27, Hawaii, pp. 19-35. To appear in CACM (scheduled for September 1995).

References

Andersen, C., Krogh-Jespersen, F., & Petersen, A. (1972). *SYSKON - en bog om konstruktion af datamatiske systemer.* København: G.E.C. Gad.

Anderson, R. E., Johnson, D. G., Gotterbarn, D., & Perrolle, J. (1993). Using the ACM Code of Ethics in decision Making. *Communications of the ACM, 36*(2), 98-107.

Andriole, S. J. (1989). *Storyboard Prototyping: A New Approach to User Requirements Analysis* (First ed.). Wellsley, Mass.: QED Information Sciences.

Bannon, L. (1991). From Human Factors to Human Actors: The Role of Psychology and Human-Computer Interaction Studies in System Design. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems* (pp. 25-44). Hillsdale, New Jersey: Lawrence Erlbaum.

Bansler, J. (1989). Systems Development Research in Scandinavia: Three Theoretical Schools. *Office: Technology & People, 4*(2).

Bjerknes, G., & Bratteteig, T. (1988). The Memoirs of Two Survivors or Evaluation of a Computer System for Cooperative Work. In D. Tatar (Ed.), *Proceedings of ACM CSCW'88 Conference on Computer-Supported Cooperative Work*, (pp. 167-177). Portland, Oregon: ACM.

Braverman, H. (1974). *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century.* New York: Monthly Review Press.

Bødker, S. (1987). Prototyping Revisited—design with users in a cooperative setting. In *Report of the 10th IRIS Seminar*, (pp. 71-92). University of Tampere.

Bødker, S. (1991). *Through the interface - A human activity approach to user interface design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Bødker, S., & Grønbæk, K. (1989). Cooperative prototyping experiments—users and designers envision a dentist case record system. In J. Bowers & S. Benford (Eds.), *Proceedings of the First European Conference on Computer-Supported Cooperative Work, EC-CSCW,* (pp. 343-357). Hilton, Gatwick, London, UK.

Bødker, S., & Grønbæk, K. (1991a). Cooperative Prototyping: Users and Designers in Mutual Activity. *International Journal of Man-Machine Studies, 34* (Special Issue on CSCW).

Bødker, S., & Grønbæk, K. (1991b). Design in Action: From Prototyping by Demonstration to Cooperative Prototyping. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems* (pp. 197-218). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Campbell, R. L. (1992). Will the Real Scenario Please Stand Up? *SIGCHI Bulletin, 24*(2), 6-8.

Carroll, J. (Ed.). (1995). *Scenario-Based Design*. New York: John Wiley & Sons, Inc.

Clement, A., & Besselar, P. V. d. (1993). A Retrospective Look at PD Projects. *Communications of the ACM, 36*(6), 29-37.

Dahl, O.-J., Myhrhaug, B., & Nygaard, K. (1971). *SIMULA 67 Common Base Language* (S- No. 22). Norsk Regnesentral, Oslo.

Dahlbom, B. (1993). Introduction. In B. Langefors (Editor Bo Dahlbom), *Essays on Infology* (pp. 11-26). Göteborg, Sweden: Department of Information Systems, University of Göteborg.

Dahlbom, B., & Mathiassen, L. (1993). *Computers in context: the philosophy and practice of systems design*. Cambridge, MA: NCC Blackwell.

Dahlbom, B., & Mathiassen, L. (1994). The Future of Computing. In P. Kerola, A. Juustila, & J. Järvinen (Eds.), *Information Systems Research Seminar in Scandinavia - IRIS 17*, (pp. 2-16). Syöte, Finland: University of Oulu, Department of Information Processing Science, Finland.

Denning, P. J. (1992). Educating a New Engineer. *Communications of the ACM, 35*(12), 82-97.

Dreyfus, H., & Dreyfus, S. (1986). *Mind over Machine: the power of human intuition and expertise in the era of the computer.* Glasgow: Basil Blackwell Ltd.

Ehn, P. (1989). *Work-Oriented Design of Computer Artifacts*. Stockholm, Sweden, and Hillsdale, NJ: Arbetslivscentrum and Lawrence Erlbaum Associates.

Ehn, P. (1993). Scandinavian Design: On Participation and Skill. In D. Schuler & A. Namioka (Eds.), *Participatory Design: Principles and practices* (pp. 41-77). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ehn, P., & Sandberg, Å. (1983). Local Union Influence on Technology and Work Organization—some results from the Demos Project. In U. Briefs, C. Ciborra, & L. Schneider (Eds.), *System design for, with and by users* (pp. 427-437). Amsterdam: North-Holland.

Ehn, P., & Sjögren, D. (1991). From System Descriptions to Scripts for Action. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems* (pp. 241-268). London and New Jersey: Lawrence Erlbaum.

Floyd, C. (1987). Outline of a Paradigm Change in Software Engineering. In *Computers and Democracy* (pp. 191-210). Aldershot: Avebury.

Floyd, C. (1992). Software Development as Reality Construction. In C. Floyd, H. Züllighoven, R. Budde, & R. Keil-Slavik (Eds.), *Software Development and Reality Construction* (pp. 86-100). Berlin: Springer-Verlag.

Floyd, C., Mehl, W.-M., Schmidt, F.-M. R., & Wolf, G. (1989). Out of Scandinavia: Alternative Approaches to Software Design and System Development. *Human-Computer Interaction, 4*(4), 253-350.

Floyd, C., Züllighoven, H., Budde, R., & Keil-Slavik, R. (Eds.). (1992). *Software*

Development and Reality Construction. Berlin: Springer-Verlag.

Freire, P. (1970). *Pedagogy of the Opressed.* New York: Herder and Herder.

Gärtner, J., & Wagner, I. (1994). Systems as Intermediaries: Political Frameworks of Design & Participation. In R. Trigg, S. I. Anderson, & E. Dykstra-Erickson (Eds.), *PDC '94,* (pp. 37-46). Chapel Hill, North Carolina, USA: CPSR.

Goguen, J. A. (1992). *The Dry and the Wet* (Technical Monograph No. PRG-100). Oxford University Computing Laboratory, Programming Research Group.

Greenbaum, J. (1979). *In the name of efficiency - Management theory and shop-floor practice in data processing work.* Philadelphia: Temple University Press.

Greenbaum, J. (1990). The Head and the Hart. *Computers and Society, 20*(2), 9-16.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Grudin, J. (1991a). CSCW Introduction (special issue). *CACM, 34*(12), 30-35.

Grudin, J. (1991b). Interactive Systems: Bridging the Gaps between Developers and Users. *IEEE Computer* (April), 59-69.

Grudin, J. (1991c). Systematic Sources of Suboptimal Interface Design in Large Product Development Organizations. *Human-Computer Interaction, 6*(2), 147-196.

Grønbæk, K. (1989). Extending the Boundaries of Prototyping - Towards Cooperative Prototyping. In *Proceedings of the 12th IRIS conference* University of Aarhus (DAIMI PB - 296).

Grønbæk, K. (1991). *Prototyping and Active User Involvement in System Development: Towards a Cooperative Prototyping Approach.* Ph.D. Thesis, Computer Science Department, Aarhus University.

Grønbæk, K., Hviid, A., & Trigg, R. (1991). *ApplBuilder - an Object-Oriented Application Generator Supporting Rapid Prototyping* (PB-366). Computer Science Department, Aarhus University.

Grønbæk, K., & Knudsen, J. L. (1992). Tools and techniques for experimental system development. In K. Systä, P. Kellomäki, & R. Mäkinen (Eds.), *Nordic workshop on Programming Environment Research,* (unpaginated). Tampere, Finland: Tampere University of Technology, Software Systems Laboratory.

Grønbæk, K., & Trigg, R. H. (1994). Introduction, Hypermedia Special Section. *CACM, 37*(2), 26-29.

Hartmanis, J. (Committee Chair). (1992). Computing the Future. *Communications of the ACM, 35*(11), 30-40.

Heath, C., & Luff, P. (1992). Collaboration and Control. Crisis Management and Mutimedia Technology in London Underground Line Control Rooms. *Computer Supported Cooperative Work, 1*(1), 69-94.

Heidegger, M. (1962). *Being and Time.* New York: Harper & Row.

Hirshheim, R., & Klein, H. K. (1989). Four Paradigms of Information Systems Development. *Communications of the ACM, 32*(10), 1199-1216.

Hughes, J., Randall, D., & Shapiro, D. (1993). From Ethnografic Record to System Design. Some experiencies from the field. *Computer Supported Cooperative Work (CSCW), 1*(3), 123-141.

Jungk, R., & Müllert, N. (1987). *Future Workshops: How to create desirabel futures.* London: Institute for Social Inventions.

Järvinen, P. (1991). Cooperative Design: Discussant's Note. In H.-E. Nissen, H. K. Klein, & R. Hirschheim (Eds.), *Information Systems Research: Contemporary Approaches & Emergent Traditions* (pp. 417-420). Amsterdam, New York, Oxford, Tokyo: North-Holland.

Kensing, F., & Madsen, K. H. (1991). Generating Visions: Future Workshops and Metaphorical Design. In J. Greenbaum & M. Kyng (Eds.), *Design at Work—Cooperative Design of Computer Systems* (pp.

155-168). Hillsdale, New Jersey: Lawrence Erlbaum.

Kraft, P., & Bansler, J. (1994a). The Collective Resource Approach: The Scandinavian Experience. *Scandinavian Journal of Information Systems, 6*(1), 71-84.

Kraft, P., & Bansler, J. (1994b). Priviledge and Invisibility in the New Working Order: A reply to Kyng. *Scandinavian Journal of Information Systems, 6*(1), 97-106.

Kuhn, S., Muller, M., & Meskill, J. (Eds.). (1992). *Proceedings of the Participatory Design Conference*. Boston, MA: CPSR.

Kyng, M. (1991). The System Work Group, Computer Science Department, Aarhus University. In S. P. Robertson, G. M. Olson, & J. S. Olson (Eds.), *Human Factors in Computing Systems, Reaching Through Technology*, (pp. 477-478). New Orleans, Louisiana: ACM Press.

Kyng, M. (1994a). *Centre for Experimental System Development—Devise (in Danish: Center for Eksperimentel Systemudvikling—Devise)* (Research application No. Department of Computer Science, University of Aarhus.

Kyng, M. (1994b). Collective Resources Meets Puritanism. *Scandinavian Journal of Information Systems, 6*(1), 85-96.

Kyng, M., & Nielsen, P. A. (1993). Domain Modelling and Application Frameworks. In J. P. Bansler, K. Bødker, F. Kensing, J. Nørbjerg, & J. Pries-Heje (Eds.), *16th IRIS - Information systems Research seminar In Scandinavia*, (pp. 221-235). Copenhagen: University of Copenhagen, Department of Computer Science.

Madsen, K. H., & Aiken, P. H. (1993). Experiences Using Cooperative Interactive Storyboard Prototyping. *Communications of the ACM, 36*(6), 57-64.

Mills, C. W. (1980). On Intellectual Craftsmanship (1952). *Transaction Social Science and Modern Society, 17*(2), 63-70.

Mogensen, P. (1994). *Challenging Practice—an approach to Cooperative analysis*. Ph.D., University of Aarhus.

Muller, M. J. (1991). PICTIVE - An Exploration in Participatory Design. In *CHI'91*, (pp. 225-231). New Orleans, LA: ACM.

Muller, M. J., & Kuhn, S. (1993). Participatory Design (Special Issue). *Communications of the ACM, 36*(6).

Muller, M. J., Tudor, L. G., Wildman, D. M., White, E. A., Root, R. W., Dayton, T., Carr, B., Diekmann, B., & Erickson, E. D. (1995). Bifocals for Design and Representation in Participatory Design: PICTIVE and CARD. In J. Carroll (Ed.), *Scenario-Based Design* New York: John Wiley & Sons, Inc.

Mumford, E., & Ward, T. B. (1968). *Computers: Planning for People*. London: Batsford.

Mumford, E., & Weir, M. (1979). *Computer systems in Work Design - the ETHICS Method; Effective Technical and Human Implementation of Computer systems*. Associated Business Press.

Naur, P. (1985). Programming as theory building. *Microprocessing and Microprogramming, 15*, 253-261.

Naur, P. (1992). Computing and the So-called Foundations of the So-called Sciences (1990). In P. Naur, *Computing: A Human Activity* (pp. 49-63). New York: ACM Press.

Negt, O. (1972). *Sociological Fantasy and Exemplified Learning*. Framkfurt/M, Germany: EVA.

Noble, D. (1994). PD: Politics and Prospects. In R. Trigg, S. I. Anderson, & E. Dykstra-Erickson (Eds.), *PDC '94*, (pp. 175). Chapel Hill, North Carolina, USA: CPSR.

Nurminen, M., Kalmi, R., Karhu, P., & Niemelä, J. (1985). Knowledge and Work: Towards a Social Interpretation of Information Systems. In M. Lassen & L. Mathiassen (Eds.), *Report of the Eighth Scandinavian Research Seminar on Systemeering* (pp. 172-191). Århus: Department of Computer Science, University of Aarhus.

Nurminen, M. I. (1988). *People or Computers: Three Ways of Looking at Information Systems.* Lund, Sweden: Studentlitteratur.

Nygaard, K. (1979). The "Iron and Metal Project": Trade Union Participation. In Å. Sandberg (Ed.), *Computers Dividing Man and Work* (pp. 94-107). Stockholm, Sweden: Arbetslivscentrum.

Nygaard, K. (1984 (1988)). Profession oriented languages. In P. B. Andersen & T. Bratteteig (Eds.), *Computers and Language at Work* (pp. 21-30). Oslo, Norway: Institute of Informatics, University of Oslo. (Originally presented as Keynote speech at Medical Informatics Conference Europe, Bruxelles, September 10-13 1984).

Pehrson, B., Simon, I., Brunnstein, K., Raubold, E., Duncan, K., & Krueger, K. (Eds.). (1994). *13th IFIP World Congress 94.* Amsterdam, The Netherlands: North-Holland, Elsevier Science B.V.

Polanyi, M. (1967). *The Tacit Dimension.* Garden City, NY: Anchor Books, Doubleday & Company.

Rosson, M. B., & Alpert, S. R. (1990). The Cognitive Consequences of Object-Oriented Design. *Human-Computer Interaction, 5*(4), 345-379.

Sandberg, Å. (Ed.). (1979). *Computers Dividing Man and Work.* Stockholm: Swedish Center for Working Life.

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory Design: Principles and Practices.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Shapiro, D., et al. (1993). *Informing CSCW System Requirements* (COMIC deliverable No. COMIC Project, ESPRIT Basic Research 6225. Deliverable D2.1). Department of Computing, Lancaster University, Lancaster LA1 4YR, UK.

Snelling, L., & Jolly, C. (1994). The Work Mapping Technique. In R. Trigg, S. I. Anderson, & E. Dykstra-Erickson (Eds.), *PDC '94,* (pp. 135-137). Chapel Hill, North Carolina, USA: CPSR.

Suchman, L. (1987). *Plans and situated actions. The problem of human-machine communication.* New York: Cambridge University Press.

Suchman, L., & Trigg, R. (1991). Understanding Practice: Video as a Medium for Reflection and Design. In J. Greenbaum & M. Kyng (Eds.), *Design at Work - Cooperative Design of Computer Systems* (pp. 65-89). Hillsdale, New Jersey: Lawrence Erlbaum.

Sørensen, B. A. (1994). When 'Old' and 'New' Technologies Have to Interact and Co-exist. In R. Trigg, S. I. Anderson, & E. Dykstra-Erickson (Eds.), *PDC '94,* (p. 113). Chapel Hill, North Carolina, USA: CPSR.

Trigg, R., Anderson, S. I., & Dykstra-Erickson, E. (Eds.). (1994). *Proceedings PDC '94.* Palo Alto, CA: CPSR.

Turner, A. J. (Ed.). (1991). Computing Curricula 1991. *Communications of the ACM, 34*(6), 68-84.

Utopia (1984). *The Utopia project—An Alternative in Text and Images.* Stockholm, Sweden: Swedish Center for Working Life.

Wagner, I. (1993). A Web of Fuzzy Problems: Confronting the Ethical Issues. *Communications of the ACM, 36*(6), 94-101.

Winograd, T., & Flores, F. (1986). *Understanding Computers and Cognition—A New Foundation for Design.* Norwood, New Jersey: Ablex Publishing Corporation.

Wittgenstein, L. (1953,1963). *Philosophical Investigations.* Oxford, UK: Oxford University Press.

Wynn, E. (1991). Taking Practice Seriously. In J. Greenbaum & M. Kyng (Eds.), *Design at Work* (pp. 65-89). London and New Jersey: Lawrence Erlbaum.

Wynn, E. H. (1979). *Office Conversation as an Information Medium.* USA: University of California, Berkeley.