MUST - a Method for Participatory Design

Finn Kensing, Jesper Simonsen, and Keld Bødker
Dept. of Computer Science, Roskilde University
DK 4000 Roskilde, Denmark
Phone: +45 4675 7711
Fax: +45 4675 4201
Email: {kensing, simonsen, keldb}@dat.ruc.dk

ABSTRACT
The paper presents a conceptual framework and a coherent method for design in an organizational context within the PD tradition. The MUST method has been developed throughout 10 projects in Danish and American organizations, and it has recently been evaluated, and adopted by IT professionals within a large Danish organization. The method is based on thorough participation with users and managers, and it combines the use of ethnographic techniques and intervention. The paper describes the application area and perspective of the MUST method, presents six general principles on which the method is based, and describes five main activities providing a stepwise decision making process in relation to the overall design process. The paper concludes with a brief comparison of the MUST method with other approaches and by summing up the main points.

Keywords
Design, conceptual framework, method, project management.

INTRODUCTION
Referring to the distinctions of Grudin(1991), the paper presents a method for participatory design in an organizational context, whether this is in-house/custom development or competitive bid/contract development.

Our main interest lies in designing for a specific organization's needs rather than designing generic products for a large market. We use the term design in the same way as architects do - focusing on the analysis of needs and opportunities, and the preliminary design of functionality and form. We do acknowledge however, that in a succeeding development process further design is needed, and that when applying a computer system, users might very well find new ways of utilising the system, as well as they might come up with additional demands. This does not negate the need for a design which is a good first approximation.

In *PDC'96 Proceedings of the Participatory Design Conference.* J. Blomberg, F. Kensing, and E.A. Dykstra-Erickson (Eds.). Cambridge, MA USA, 13-15 November 1996. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

The MUST[1] method has been developed throughout 10 projects in Danish and American organizations (Kensing, Bødker, and Simonsen, 1994), and it has recently been evaluated, and adopted by IT professionals within a large Danish organization (Kensing, Simonsen, and Bødker, 1996). We have designed IT support for e.g. nine people on an editorial board of a film company, for 50 people working in a R/D-lab, and we have designed multimedia support for 140 people working at a radio station. All the work domains can be characterized as professional work in complex settings with a very open-ended agenda for the design project: no clear statement of the problems, of the kind of IT support, or of how the project should be carried out.

According to Mathiassen (1984), a method is characterized by its application area, its perspective, and its guidelines, i.e. techniques, representation tools, and principles for organizing a project. Our suggestions according to these categories will be described where appropriate in the description below. We start out discussing why a method for design is needed and what kind of method this should be. Then we present six general principles on which the MUST method is based and five main activities that provide a stepwise decision making process. Because of the limitations of space, we can only briefly outline the guidelines of the method. We briefly relate the MUST method to other approaches, and conclude by summing up the main points.

While this paper focuses on the method with no space for examples, we refer the reader to papers describing our examples on design projects (Bødker, 1990; Kensing and Winograd, 1991; Bødker and Kensing, 1994; Simonsen and Kensing, 1994; Kensing, Bødker, and Simonsen, 1994; Simonsen, 1994; Simonsen, 1996; Simonsen and Kensing, 1994; 1996; Kensing, Simonsen, and Bødker, 1996).

WHY A METHOD AND WHAT KIND OF METHOD?
In the years of outsourcing and Business Process Reengineering, many organizations have chosen to outsource the costly and hard-to-manage software development. Bansler and Havn (1994) refer to this as "the 'industrialisation' of information systems development" and

[1] MUST - a Danish acronym for theories of and methods for initial analysis and design activities.

argue that in the future, most IT systems will be based on prefabricated generic systems.

In the same way that prefabricated walls and doors have not made architectural design irrelevant, we have found that the increased use of generic systems does not rule out a need for customized design. We argue that it is the job of design, based upon a thorough understanding of the organization in question, to investigate which generic systems are adequate, as well as how to reorganize work accordingly. Also, generic systems often need to be supplemented with the design of organizationally specific systems to comprise a coherent solution. It is these parts of systems development that we call design and which our method deals with: the analysis of needs and opportunities, and the preliminary design of functionality and form. An organization may carry out a design project in cooperation with internal IT specialists or with external consultants. These we refer to as IT professionals, and they may or may not participate in the succeeding development and implementation activities.

In Figure 1 we have combined Bansler's and Havn's (1994) project model for industrial software development with our experiences. In this model the organization relies on outside contractors for software development. The organization's IT department (or external consultants) performs the design and specification of one or more coherent visions for change and prepares for a contractual bid. The chosen contractor then gets the contract of delivering generic software and/or developing organizational specific software. In parallel, the IT department facilitates the organizational implementation by working with the user departments, external contractors, and other involved parties, in what is called "delivery management." There are major managerial decision points after "design" - e.g. which of the proposed solutions to go for - and as part of "contractual bid and selection" - e.g. which contractor to choose. For IT professionals, this points towards having a role similar to architects: besides designing a building, the architect is in charge of the overall supervision when the building is constructed. A particular instance of this model occurs when the organization's IT department chooses to also bid on the contract, in which case the software development might take place as in-house development (Grudin, 1991).

The model (as well as Bansler and Havn, 1994) indicates that IT professionals need special skills to deal with people in the user organization on their own basis, and not on a technical basis. IT professionals have to handle complex and open problem situations. We see a method as one of the resources available for IT professionals facing such situations, rather than as a cookbook to be followed step by step. IT professionals need methodological support for the activities to take place in an organization before a contractual bid and selection can take place or before the organization can decide which generic systems to select and purchase. We propose the MUST method as support for IT professionals responsible for design in this context. We have learned that for IT professionals to take up new work practices, a description of the method is certainly needed, and also that it has to be supplemented with other activities. We have been involved in establishing apprenticeship relations and supervision of on-going projects for IT professionals to take up the MUST method as part of their work practice.

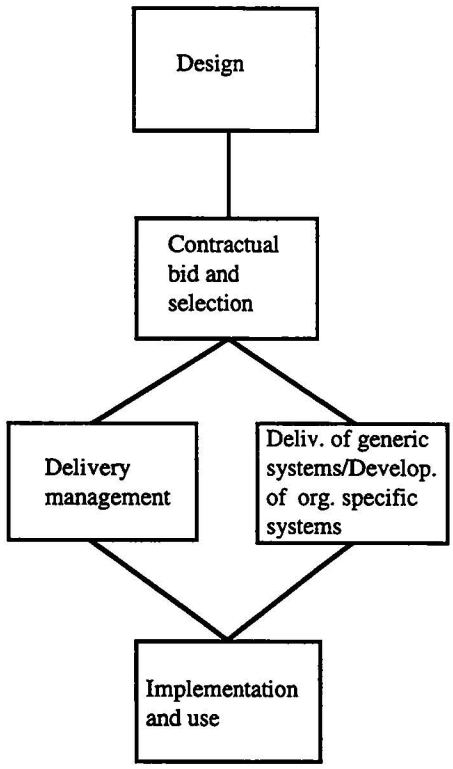

Figure 1. Project model for IT development.

Though the waterfall model has been heavily criticized by practitioners and academics, and though it has been modified - also in practice - through the introduction of iterations and prototyping, it is still the leading metaphor for the development process in the industry. The introduction of outsourcing and the use of generic software strengthen it even more. We have chosen to rely partly on this model, i.e. we do think in terms of phases. However, in line with (Andersen, Kensing et al., 1990), we suggest that "phase" means the activities which are performed between two major decision points, and that each phase includes analysis, design, programming and documentation to the degree that these activities are needed to bring about a sound basis for evaluating the distance between current status and current plans. With its separate design phase, for which we introduce five main activities and associated decision points, the MUST method provides users, managers, and IT professionals with a sustainable basis for introducing new IT systems, thus minimizing the risks of developing unrealistic, inefficient, and cumbersome systems.

The MUST method has been developed through projects following the model in Figure 1. The method is coherent in the sense that it deals with all activities within its application area: analysis of needs and possibilities, generation of

visions for change, project management, and planning for technical and organizational implementation. Most Scandinavian PD researchers, coming from a background in trade union projects, have not explicitly dealt with activities related to management (see however e.g. Bødker, 94). We want to stress that for design ideas to be implemented, establishing and maintaining relations with management is crucial when designing in an organizational context.

Design in an organizational context is an open-ended process. The objective of the design project is to investigate the situation and provide information for a decision about how to proceed. If appropriate computer support can be identified, the overall functionality and form of such systems are outlined. We see the results of a design project to include a conceptual design in terms of a written document, sketches, mock-ups, and prototypes. We consider an evaluation of consequences of implementing the design, as well as a plan for the implementation, to be a part of the result too. Based upon a design proposal, it should be possible for the organization to say "go", "no go", or "more design is needed". The project may proceed to development and implementation, but we consider these parts of systems development to be outside the application area of our method.

We see organizations as frameworks for cooperation as well as for conflicts. Groups and individuals participating in design should be expected to have common, as well as conflicting goals. The role of IT professionals is neither to cover up nor to solve political conflicts in design. Rather they should help the parties to formulate their visions, and leave it to them to solve conflicts in relevant fora (see Principle 2 below). A good design most often is a mix of tradition and transcendence (Ehn, 1988). One reason for bringing in IT professionals is to transcend the tradition. At least someone in the organization has considered some of the current ways of doing things to have lost their rationale, or found that new technological opportunities are worthwhile investigating. However, IT professionals need to understand traditions in the organization, both as a way of maintaining (or establishing!) credibility, but also in order to understand the rationale behind phenomena that otherwise can be perceived as odd by an outsider.

We want to emphasize that an important ethical issue for an application of our method - and for participation in general - is that if management wants job cuts or other drastic changes, this should be announced up front. Otherwise an important ethical principle will have been violated, and as a consequence participation will be made more difficult in the next project. This does not imply that drastic changes cannot be realized by a participatory approach. We have experienced drastic changes in work organization as part of design projects as well as job cuts just before the project started, but the users knew and accepted the objectives beforehand (Kensing, Simonsen, and Bødker, 1996).

THE SIX PRINCIPLES

Our method is grounded in six principles and it offers a set of techniques and ways of representing current work and the envisioned computer based systems. We consider the principles to be indispensable, while the techniques and representation tools may be chosen by the IT professionals based on their preferences and understanding of the situation in question. In this section we present each of the six principles and give references to the techniques and representation tools when appropriate.

Principle 1: Participation

A large proportion of the software installed in organizations is never used and the primary reason for this is that IT professionals have not "got the requirements right" (Boehm, 1981; Lyytinen and Hirschheim, 1987). Participation is a way of increasing the chances that a design corresponds to real needs and will be used as intended.

There have been both pragmatic and political arguments to participatory design. The pragmatic argument stresses that participation between IT professionals and users enables a mutual learning process and facilitates the development of an envisioned computer based system. This pragmatic view has been argued for theoretically, e.g. by reference to Wittgenstein and Heidegger (Ehn, 1988; Greenbaum, 1993).

The pragmatic view emphasizes that IT professionals need knowledge of the use context, that the users need knowledge of the technological options, and that these should be developed in a co-learning process. This view has in our projects been acknowledged by users, management and IT professionals. Further in this paper, we argue, that for a design vision to be realized, an IT solution needs not only to be technically correct, but also a design team needs to focus on anchoring the vision in the organization (see below). This requires the design team to engage multiple participants in the design endeavour. It is the responsibility of the IT professionals to organize a participatory design process and it is the responsibility of the management to provide users with the time and possibilities to participate in this process.

Political arguments to participatory design stress users' rights to influence their own working conditions and that this should be taken care of by their local union representatives (Nygaard, 1975; Kyng and Mathiassen, 1982; Ehn, 1988; Greenbaum, 1993). From the very start of the Scandinavian trade union projects, it has been a key issue to ensure that users get time off to participate, and that trade unions should build up their own competence apart from management controlled activities like systems development (Nygaard, 1975; Ehn and Sandberg, 1979; Kyng and Mathiassen, 1982). We still subscribe to this. However, we realize that IT professionals need to be pragmatic too. In the years of downsizing, the decrease of unions' power, and employees striving to build a career, we have experienced very valuable user representatives. They were not given time off for participation, but in spite of this they were most eager to participate in the projects while attending to their jobs.

Principle 2: Close links to project management

Project management deals with the division of labour in the project, how the project is designed as a process, quality control, and how conflicts are dealt with. We deliberately include establishing close interaction between project man-

agement and activities related to the design proper as a principle, because it has not been dealt with explicitly in the participatory design literature.

We advocate a division of labour between a design team and a steering committee. The design team should consist of a combination of IT professionals and future users. They are responsible for carrying out the project and for informing management and all future users. The steering committee should include managers of the involved organizational unit(s), the manager of the IT department - if any, and 1-2 user representatives[2].

The design team must decide how they will organize the process of developing an understanding of the organization's needs and possibilities, developing visions of future computer based systems, and sketching plans for the technical and organizational implementation. Designing the project as a process is of special concern in dealing with the early design activities, since they are characterised more as problem setting than problem solving (Lanzara, 1982; Schön, 1983). We do acknowledge that for a group of IT professionals to be efficient, they need to rely on a set of standard techniques, representation tools, and ways of conducting projects. However each project needs to be designed according to an understanding of the specifics of the actual context.

As described in further details below, we suggest the project to be designed around the following five main activities: 1) project establishment, 2) strategic analysis, 3) in-depth analysis of selected work domains, 4) developing one or more visions of the overall change, and 5) anchoring the visions. Each activity produces the knowledge which allows the design team to inform all future users, and which allows the steering committee to focus on the type of decisions that the design team requires in order to proceed. This enables the steering committee to take decisions on a qualified basis, thus minimizing risks in the ongoing interpretations of the project's goals, and of developing unrealistic visions.

Design is also a political process where groups and individuals have common as well as conflicting goals (Andersen, Kensing et al., 1990). The steering committee is responsible for supervising the design project, dealing with potential and manifest conflicts, and making decisions based upon information provided by the design team. We suggest to leave it to the steering committee to deal with the conflicts generated or becoming manifest in relation to the project. It is not up to the design team to solve the political controversies, but it does have a role in providing a sound basis for dealing with them, and in seeing to that they are dealt with in the relevant fora. This has been emphasized in most of our projects (see e.g. Bødker, 1990; Bødker and Kensing, 1994; Simonsen, 1994; Simonsen and Kensing, 1994; 1996; Kensing, Simonsen, and Bødker, 1996).

We suggest three techniques for the design and the continuous evaluation of both the process and the product: project establishment, planning with baselines (Andersen, Kensing et al., 1990), and reviews (Freedman and Weinberg, 1982).

Principle 3: Design as a communication process
In earlier work (Kensing and Munk-Madsen, 1993) we have created a model of the communication between users and IT professionals (see Figure 2). The model is based on two distinctions - dealing with three domains of discourse and two levels of knowledge.

"Users' present work" includes work practice, organization of work, use of IT, products/services, relations to customers, clients, and suppliers, history of recent major changes, management strategies and style, etc. "New system" includes envisioned technology in relation to new work organization for the specific work domain. "Technological options" incorporates general knowledge and experiences with IT, and its relation to work organization. The domains reflect both the users' and the IT professionals' typical prerequisites in terms of knowledge and understanding prior to entering the design process. At the outset, the users have knowledge of their present work and of organizational options. The IT professionals have knowledge of technological options with regard to hardware and software. At the outset, this is all we can expect them to know. Knowledge within all domains must be developed and related in order for the design process to be a success.

The second distinction between abstract knowledge and concrete experience expresses, that we need to deal explicitly with two levels of knowledge. We need abstract knowledge to get an overview of a domain of discourse and we need concrete experience in order to understand the abstract knowledge and in order to evaluate its relevance. As will be argued in the next section (Principle 4), it is by iterating between these two levels of understanding that the design team is able to develop the needed insight.

It is the responsibility of IT professionals to choose the techniques and the representation tools that allow them to establish a communicative process with users through which they are able to jointly develop knowledge within these six areas.

Principle 4: Combining Ethnography and Intervention
We apply a combination of ethnographic techniques and intervention in an iterative approach to design. We strive to select carefully the area and the mode of intervention based

[2] In some organizations the local union (by law or agreement) has a say in relation to development of new IT systems. If this is the case, we advocate that shop stewards become members of the steering committee. If it is not the case, users should be given the opportunity to appoint representatives.

	Users' present work	New system	Technological options
Abstract knowledge	Relevant structures on users' present work	Visions and design proposals	Overview of technological options
Concrete experience	Concrete experience with users' present work	Concrete experience with the new system	Concrete experience with technological options

Figure 2. Six areas of knowledge in user-IT professional communication.

upon what we have learned by applying ethnographic techniques - in contrast to Business Process Reengineering (Hammer and Champy, 1993, p. 207). Ethnography and intervention contrast in terms of their basic approaches and intended results: ethnographers originally strove not to change the phenomena they were studying, while interventionists deliberately set up activities to change the organization in order to learn from the reactions to the change. However, we have experienced that at a practical level, combining the two approaches and iterating between them has been an effective way to learn about the organization and also an important resource in generating realistic visions of future use of technology (see e.g. Bødker and Kensing, 1994; Simonsen, 1994; Simonsen and Kensing, 1994; 1996; Kensing, Simonsen, and Bødker, 1996).

Ethnography: First hand encounters

Blomberg et al. (1993, p.125) state that "to learn about a world you don't understand you must encounter it first hand". It is crucial for IT professionals to develop a thorough understanding of users' present work in order for the design to reflect - in a realistic way - the norms and traditions of the organization. It is realistic in the sense that the design reflects an appreciation of the rationale given by members of the organization, and in the sense that the organization is geared to meet the challenges of the envisioned design. By detailed studies of the present situation we try to "measure" the organization's needs and readiness for change (Christensen and Molin, 1983). We are trying to avoid an extreme futuristic design or a design of which the greater proportion will never be used. We have found that ethnographic techniques are helpful in accomplishing this (Bødker and Kensing, 1994; Simonsen, 1994; Simonsen and Kensing, 1994; 1996; Kensing, Simonsen, and Bødker, 1996).

Blomberg et al. (1993) advocate descriptions in terms relevant to those being studied, in contrast to applying traditional IS techniques and their formalisms. The latter, when at their best, advocate interviews with future users but are relying on the IT professionals pre-defined conceptual frameworks. In Kensing and Munk-Madsen (1993), we argue that by going back and forth between observing users' work practice and producing descriptions (or interpretations if you like) of these practises, IT professionals and users are able to develop an understanding of the current practices that is relevant in design.

Formalisms play a minor role in the MUST method. In later parts of the development process they are powerful tools. But when working with users without a technical background, we can easily do without them. We suggest using plain text, freehand drawings, sketches on large sheets of paper representing e.g. communicative structures, the relation between work organization, and the use of current/envisioned technology, etc. The closest we get to using formalism with users is by modelling information flows and data structures for the purpose of prototyping.

We advocate two types of descriptions (reflecting the abstract/concrete distinction in Figure 2): One stated in a language based on users' categories, for example collages (Bødker and Kensing, 1994) and wall graphs (Simonsen, 1994; Simonsen and Kensing, 1994; 1996). The other pointing out current domains and creating envisioned structured domains that might benefit from new IT systems (Winograd and Flores, 1986; Dahlbom and Mathiassen, 1993), for example problem lists (Kensing, Simonsen, and Bødker, 1996) and maps (Lanzara and Mathiassen, 1984). The first type we have found useful in detecting and evaluating the relevance of the other type, which in turn is needed for further design purposes.

Intervention

Interventionists deliberately set up activities designed to change the organization. As Dahlbom and Mathiassen (1993, p. 169) put it: "....only by trying to change it [the organisation] will we come to really understand it." The presumption is that through creating a change, key factors of the organization and its members' perception of it become observable.

Schön (1983; 1992) describes design as "a reflective conversation with the materials of a design situation". I.e. with Schön, the intervention happens in the mind of the IT professional or in conversations among them, rather than in the physical world, e.g. by prototyping and organizational experiments. This type of intervention is cheaper in terms of time and potential consequences, and thus preferable, but sometimes imagination is not enough, and "real" experiments need to be carried out.

Iterations

Two types of iteration interplay when we combine ethnographic techniques with intervention. First, iterations between interviews and observations allow IT professionals to be aware of the discrepancies between what people say they do or want to be able to do, and of what IT professionals as outsiders are able to observe, i.e. to handle the say/do problem (Blomberg et al., 1993; Gougen and Linde, 1993). Second, iteration between using ethnographic techniques

and intervention may be used to confront users with these discrepancies. In Bødker and Kensing (1994), we used the detection of such discrepancies as the input to a design workshop[3]. Others suggest the use of rapid prototyping for similar purposes , e.g. Mogensen (1994) who suggests the term "provotyping". We suggest the two types of iteration even prior to prototyping.

Principle 5: Co-development of IT, work organization, and users' qualifications

IT are introduced because someone - usually management - wants change. However, projects far too often focus solely on IT systems, leaving it to the users to struggle with the organizational implementation afterwards, and educational aspects are reduced to training the functionality of the systems.

Since the early 1970's, Mumford and associates have worked on a Socio-Technical approach (see e.g. Mumford, 1972; Mumford et al., 1978; Mumford, 1993) advocating development of the social and technical systems in more or less parallel. The approach was heavily critiqued by Scandinavian researchers involved in trade union projects in the mid-late 1970's. The critique was two-fold: from an ideological point of view, the approach to users' participation/control was evaluated as too narrow; and from a technical point of view, the proposed techniques were evaluated as naive and as not addressing relevant aspects (Ehn and Sandberg, 1979; Kyng and Mathiassen, 1982). However, we owe to the Socio-Technical approach the double focus on organizational and technical issues, and for including management in a participatory approach. The Socio-Technical approach even included prototyping as early as 1978 (Mumford et al., 1978).

We recommend including a third issue in this co-development process: users' qualification. The reason being that we have seen many systems which are only partly used because users have never been properly introduced to them. "It seems that the money ran out" as one user stated it in one of our projects. Educational activities help users (re-)gain control over their jobs, and it allows them to be more efficient.

Users need of course the qualifications to operate the systems that are supposed to support their work. However, this is often neglected in practice and further more the training is often organized around the functionality of the system rather than around users' daily work. The design team's report should suggest who should receive education, how much education in terms of content and time, how the education will be organized, as well as an estimate of the costs. Also, if a new division of work is part of the design, or if new products or services are part of the overall vision of change, we see it as part of the design report to suggest adequate educational/training activities and an evaluation of the costs. Finally, we suggest an initial and ongoing introduction of user representatives to the method used in the project as well as to what is expected from them in terms of involvement, relation to colleagues, and the specific tasks they will be participating in.

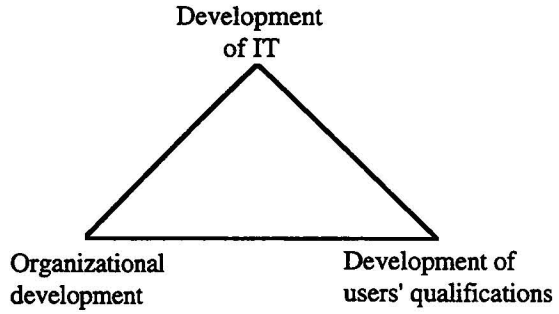

Figure 3. Co-development in related domains.

All in all, a design project needs to address, plan for, and estimate the costs of the activities taking care of technical, organizational, and educational issues, as indicated in Figure 3. This should be done to produce a sustainable basis for the organization's decision making, and for the succeeding development of the technical and organizational implementation in order to constitute a coherent whole.

Principle 6: Sustainability

The early design activities are a first step in introducing sustainable IT. We deliberately use this ecological concept as a metaphor in design. In ecology the concept refers to a balance between the utilization and the protection of the earth's resources in order not to destroy the basis of mankind. There is a growing awareness of problems, alternative products and production processes are being developed, and the market is slowly adapting. We see a start of a similar process in the development and use of IT systems:

Negative consequences have been seen. Some IT systems have been designed and/or introduced in ways that made it difficult for users to use/develop their skills and experience as part of their job (see e.g. Sachs, 1995). Often IT systems have failed economically too - expected rationalization did not materialize and projects ran far over budget. I.e. in such projects, scarce resources like money and users' qualifications were not taken properly care of.

Researchers and practitioners have developed alternatives - PD regarding processes and CSCW also regarding products, thus providing a basis for not destroying valuable resources in organizations.

Users and managers have shown an interest in alternative products and processes. Of course they might not always agree on what are positive and negative consequences of applying these. But we have seen a willingness to have such issues dealt with up front in design projects (Bødker and

[3] Beforehand we had formulated provocative statements highlighting the differences between what users had told us and what we were able to observe. These statements dealt with their current practice as well as with the relation between these and their ideas for IT support. This lead to an evaluation of consequences of the design ideas and a clarification of which work practices they preferred, which in turn resulted in a modified set of requirements.

Kensing, 1994; Simonsen and Kensing, 1994; 1996; Simonsen, 1994; 1996; Kensing, Simonsen and Bødker, 1996).

What still remains on a larger scale is for IT professionals to be introduced to a coherent method for participatory design. This is the ambitious goal of the MUST program.

What is needed is a change of attitude for most managers and IT professionals. They need to experience through practice the effects of leaving the traditional expert strategy, the result of which sometimes has been completely reversed: the way many systems work shows that rationality has lapsed into irrationality. Such cases are often reported on in the news, and have been documented by a wide range of ethnographic studies. However, in working with managers and IT professionals in most of the 10 projects informing our method development, we have experienced an increasing awareness of the pitfalls in the predominant practice as well as a willingness to experiment with alternatives.

The pragmatic argument for participation discussed above, is related to the principle of sustainability. The MUST method suggests a high degree of user participation in order that new IT systems fit with preferred work practices, as well as that it supports the organization in an up front uncovering and dealing with conflicts in relation to the introduction of IT (see Principle 2). Users, managers, and IT professionals in our projects sometimes found this cumbersome. But compared with previous experiences, they found it helpful in laying the basis for the proposed change.

FIVE MAIN ACTIVITIES CONSTITUTING THE DESIGN PROCESS

In the MUST method the overall design process is constituted by five main activities : 1) project establishment, 2) strategic analysis, 3) in-depth analysis of selected work domains, 4) developing visions of the overall change, and 5) anchoring the visions. The first four activities provide for a stepwise decision making process. Iterations are recommended, especially between activity 1 and 2 and between activity 3 and 4. The fifth activity should be seen as an ongoing concern throughout the project. Though we suggest iteration, the activities support a stepwise decision making process.

Project Establishment

Project establishment (Andersen, Kensing et. al, 1990) is a systematic technique supporting the clarification and negotiation of the aim, level of ambition, scope, and conditions of the project. The technique suggests activities for the design team to decide which tools and techniques it will use to conduct the project, as well as for establishing the team as a social unit. While many projects start out from a rather loose description, project establishment (sometimes in iteration with strategic analysis) provides the steering committee and the design team with a sound basis for the succeeding project activities.

In Lanzara's terms, project establishment is a reframing process (Lanzara, 1982). We have often experienced that management and users have rather specific ideas of which IT systems are needed, but the problematic situation leading to the solutions was not analyzed properly (see e.g. Simonsen and Kensing, 1994; 1996; Simonsen, 1994; Simonsen, 1996). We find that it is the responsibility of IT professionals to question such ideas, and project establishment is the first attempt in that direction.

Project establishment involves:

- presentation rounds to different organizational units in the organization;
- initial document analysis of the organization's own documents, like presentation materials and yearly reports;
- identification of critical success factors, i.e. what the project needs to fulfill;
- meetings where the conditions of the project are negotiated;
- a hearing[4] of all involved actors on the basis of the final (or draft on) project charter;
- and finally project planning and writing/negotiating the project charter, which is the basis for the steering committee's and the design team's decision of (and commitment to) how to approach the project.

Strategic Analysis

The purpose of the strategic analysis is to clarify and delimit which work domains should be in focus in the design project. This is often rather unclear, even if the organization has a business strategy and a related IT strategy (Simonsen and Kensing, 1994; 1996; Simonsen, 1994; 1996; Kensing, Simonsen, and Bødker, 1996). This too is a reframing process.

In some cases the strategic analysis will be a part of the project establishment. But if the organization is unable to define the focus of the design project in an adequate way, or if there are conflicts as to which areas should be given priority, we suggest that the strategic analysis be handled separately (see e.g. Simonsen, 1994; 1996).

The manifest result depends on the degree to which the organization in question already has a business strategy and a related IT strategy, and the degree to which the involved parts of the organization see the relation between these and the current project.

Strategic analysis clarifies the potentials for investments in IT support and it investigates organizational, economical, and technical limitations. It involves development of an understanding of the organization's situation on a competitive market, which parts of the organization need to be strengthened and how this relates to the current project, identification and analysis of customers and suppliers (internal and/or external), and which products and services the organization should provide. The focus is on the func-

[4] By a hearing we mean that the involved actors are informed about the given subject matter with the possibility of commenting it.

tional requirements of the environment upon the organizational units in question (Schmidt, 1988; Simonsen, 1994).

Strategic analysis is primarily a management related activity. For this activity the MUST method suggests:

- interviews of managers, the IT manager - if any, and representative users, customers, and suppliers; as well as observations of key activities;
- document analysis of (possible) strategic plans, IT strategies, and market surveys;
- functional analysis (Schmidt, 1988; Simonsen, 1994);
- that the steering committee organizes a hearing of all involved actors, hereby collecting comments for the purpose of an eventual modification of the strategic analysis and the project charter. Equally important, such a hearing ensures that all actors involved are informed about the objectives of the third activity: in-depth analysis of selected work domains.

Strategic analysis leads to a decision situation, whereby it is decided which work domains should be further analysed and succeedingly supported by IT.

In-Depth Analysis of Selected Work Domains
The work domains pointed out by the strategic analysis are in focus when in-depth analyses of current work practices are performed. The purpose is to reveal and develop an understanding of the rationale behind current work practices, ("users' present work" in Figure 2). The intention is not to map old practices into the new computer-based system. However, we have experienced that users have good reasons for what they do and that the rationale underlying current work practices is relevant for the design, even if the management aims at rather drastic changes (Kensing, Simonsen, and Bødker, 1996).

The techniques proposed for developing an understanding of the work practice, the use of current systems, and the use of information are:

- interviews and observations, where directly affected users at all levels are involved;
- document analysis of documents used in the work practice;
- thinking aloud experiments;
- mapping (Lanzara and Mathiassen, 1984);
- future workshops (Kensing, 1987; Kensing and Madsen, 1991);
- workshops where the design team, perhaps supplemented by additional users, makes rich pictures (Checkland and Scholes, 1990), collages (Bødker and Kensing, 1994), and/or wallgraphs of current work practices (Simonsen, 1994; Simonsen and Kensing, 1994; 1996).

The IT professionals might need to make preparations for these activities and subsequently carry out e.g. modelling communicative structures (Kensing and Winograd, 1991) or cultural analysis (Bødker and Pedersen, 1991), which then should be reviewed by the design team and affected users.

Even though the project establishment and strategic analysis have led to selected work domains, the analysis might lead to a conclusion that other domains need to be included in this activity too, in which case the project charter is renegotiated.

The results are descriptions of the current work organization, the use of IT, and the related problems, needs, and ideas for IT support. This is supplemented with an ordered comprehensive list of problems, needs, and related ideas for IT support and work organization. An important bi-product of this activity is that users see their own work in the light of that of others, and that IT professionals get to know users' concepts and categories, thus facilitating communication.

This leads to the third prototypical decision situation, where the steering committee decides which of the ideas for IT support should be given priority. Additionally we suggest a hearing and that the design team collects comments for the purpose of an eventual modification.

Developing Visions of the Overall Change
Developing one or more visions of the overall change is the central activity. We emphasize that the visions should not only deal with the functionality and the user interface of the suggested systems, but also include organizational change and changes in qualifications needed by the users (see Principle 5).

Ideas and visions are developed throughout the project, and they are often initiated in the very beginning of the project (Stolterman, 1991; Stolterman, 1992). They emerge in nearly all activities conducted in the project, but the purpose of this activity is especially to develop ideas and visions, and form these into one or more coherent visions.

We suggest:

- visits to "similar" work places using new IT facilities;
- future workshops (Kensing, 1987; Kensing and Madsen, 1991);
- design workshops where the design team, perhaps supplemented by affected users, sketch envisioned future work organization and its relation to new IT facilities on large sheets of paper;
- sorting out of design ideas by writing them e.g. on post-its and grouping them on a wall;
- data modelling;
- mocking IT-up and prototyping.

Again IT professionals might have to make preparations for these activities and subsequently carry out e.g. information modelling and the development of prototypes.

The result is a design report, which states the aim of the project, sums up the analyses, and describes the suggested visions. The design report is supplemented with mock-ups and prototypes of the proposed IT systems. The report holds an evaluation of positive and negative consequences of the suggested visions regarding the organization as a whole, involved organizational units and communities of users. For this purpose we suggest scenarios, outlining how the work

will be carried out when the visions are implemented (Clausen, 1993; Simonsen, 1994; Kyng, 1995; Kensing, Simonsen, and Bødker, 1996). Finally the report includes estimated costs as well as a plan for purchase/development of IT systems, for technical and organizational implementation, and for the education/training of users.

The design report forms the basis on which the steering committee decides which parts of the proposed design should be purchased as generic systems, which parts need to be developed as customized systems, and which parts should be postponed or perhaps rejected. Also, the suggested organizational changes and education/training activities are decided upon by the committee. We suggest that the steering committee organizes a hearing of all involved actors, hereby collecting comments for the purpose of an eventual modification of the proposed design.

Anchoring the Visions

We use "anchoring" as a metaphor (Simonsen, 1994) that moves beyond the design/ implementation dichotomy. In order for a vision to materialize it needs to be deeply rooted in the organization. Its rationale needs to be understood by:

- management and the steering committee, who decide if it should be implemented;
- those who will carry out the technical and organizational implementation - the latter including educational/training activities;
- the users who will have to live with its consequences.

Since the above mentioned actors are not all directly involved in developing the visions, time and resources must be directed towards making it possible for them to get to know the visions. Anchoring the visions is the job of the design team and the management of the involved parts of the organization. We see this last activity to be orthogonal to the other four. It should be given attention to in project establishment and in the strategic analysis; and both the direct participation of users, as well as the suggested hearings, contribute to the anchoring activity. The purpose is to prepare for and even start the process of organizational change, while still carrying out analysis and design activities. This guides why and by which means the design team and management interact with actors in the organization and maybe outside contractors. In this respect, anchoring the visions is contributing to seeing design as a process of change.

A participatory approach to design is the central strategy in obtaining appropriate anchoring of the visions. This includes:

- meetings and workshops developing, presenting and evaluating design ideas;
- prototyping;
- visits to other institutions using potentially relevant IT;
- demonstration of IT products;
- and scenarios describing envisioned future work practices supported by the proposed designs.

The design report and prototypes cannot convey everything that the design team learned throughout the project (Naur, 1985). Therefore appropriate anchoring requires that (part of) the design team has to cooperate, at least in an overlapping period of time, with those taking care of technical and organizational implementation. For IT professionals, this points towards having a role similar to architects: besides designing a building, the architect is in charge of the overall supervision when the building is being constructed.

RELATING THE MUST METHOD TO OTHER APPROACHES

First, it is important to note that the MUST method offers *a conceptual framework of the design process*, while e.g. Stolterman (1991), and Ehn et al. (1995) are developing a conceptual framework facilitating an ongoing evaluation of the qualities of the designed products. Yet others e.g. Schmidt (1990), Rasmussen et al. (1994), and Carstensen (1995) offer a framework for the conceptualization of users' work domain.

A second distinction is related to the application area of the proposed methods. The MUST method focuses on *the early activities in a development process* like most PD methods, BPR (Hammer and Champy, 1993), and OOA (Coad and Yourdon, 1991). It offers *guidelines for project management* (like BPR) *as well as for the design proper* (like PD and OOA). E.g. Grønbæk et al. (1995) deal with the entire development process, but focus solely on the design proper.

Third, also related to the application area, is what kind of changes the design process strives at. While downsizing is the explicit goal of BPR (Hammer and Champy, 1993, p. 212), ethical issues in relation to involving users is not dealt with. The MUST method states explicitly that if management aims at job cuts and/or other *drastic changes*, this *should be announced up front.* And if users know and accept these objectives we still recommend a participatory approach.

Fourth, the MUST method includes *management issues* in relation to design processes in an organizational context. This has not been dealt with earlier in the PD literature, where the focus has been on why and how to work with users. While e.g. Bødker (1994) does report on the role of management in relation to the future use of a system, she does not deal with the role of management in the processes of generating visions and helping them to materialize.

Fifth, the MUST method argues for *the need of a separate design phase*, including the development of visions of the overall change, in order to produce a sustainable basis for further development and implementation. Other approaches (see e.g. Grønbæk et al., (1995)), primarily strive for an accountable design through extensive user participation in prototyping.

Sixth, Plowman et al. (1995) report that the dominant approach in projects using ethnography is sociologists conducting the ethnographic studies and informing IT professionals of their findings. We and others (see e.g. Blomberg et al., 1994; Shapiro 1994) are working towards closer links between ethnography and design. The MUST method

advocate that *IT professionals start practicing ethnographic techniques themselves* in their cooperation with users. Our argument is that in some countries, e.g. Denmark, there is no tradition for involving sociologists or anthropologists in design projects. However, our experiences confirm that it is possible and valuable for IT professionals to use ethnographic techniques as part of their design activities (Kensing and Winograd, 1991, Bødker and Kensing, 1994; Simonsen and Kensing, 1994; 1996).

Seventh, while systems development methods suggest various formalisms for describing users current work and the envisioned design of IT systems (see e.g. Yourdon, 1989; Coad and Yourdon, 1991), *formalisms play a minor role* in the MUST method. Instead we suggest plain text, freehand drawings, and sketches for the production and presentation of the relation between proposed IT systems and users' current and future work practice, postponing an extended use of formalisms to later on in the development process.

SUMMARY

We have argued for the need of a separate design phase in order to produce a sustainable basis for further development and implementation of IT in an organizational context. Within the tradition of Participatory Design we have presented a conceptual framework and a coherent method for this phase. Up until now the method has proven successful even in design projects linked to job cuts and changes in work organization. However, these projects have been carried out or supervised by the authors. The degree to which the method, without our involvement, will serve as a useful resource for IT professionals in their work with users and managers is still an open question. We invite the reader to challenge the method.

REFERENCES

Andersen, Niels Erik, Finn Kensing, Jette Lundin, Lars Mathiassen, Andreas Munk-Madsen, Monika Rasbech and Pål Sørgaard: *Professional Systems Development: Experience, Ideas and Action,* Prentice-Hall, New York, 1990.

Bansler, Jørgen and Erling Havn: "Information Systems Development with Generic Systems", *Proceedings of the Second Conference on Information Systems, Nijenrode University, 30-31 May, 1994*, Nijenrode University Press, The Netherlands, 1994, pp. 707-715.

Blomberg, Jeanette, Jean Giacomi, Andrea Mosher, and Pat Swenton-Hall: "Ethnographic Field Methods and Their Relation to Design", in Douglas Schuler and Aki Namioka (Eds.): *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates, Publishers, London, UK, 1993, pp. 123-155.

Blomberg, Jeanette, Lucy Suchman, and Randall H. Trigg: Reflections on a Work-Oriented Design Project, in *proceedings of the Third Biennial Conference on Participatory Design, October 27-28, 1994, Chapel Hill, North Carolina.* Computer Professionals for Social Responsibility, Palo Alto, CA 94392-0717, USA, 1994, pp. 99-109.

Boehm, B. *Software Engineering Economics*, Prentice-Hall, 1981

Bødker, Keld. *Analyse og design i et kulturperspektiv - udfordring til systemperspektivet*, Ph.D dissertation (in Danish) [Analysis and Design in a Cultural Perspective - Challenging the Systems Perspective], Datalogiske skrifter No. 26, Roskilde University, 1989.

Bødker , Keld. A Cultural Perspective on Organizations Applied to Analysis and Design of Information Systems. In *Organizational Competence in Systems Development. A Scandinavian Contribution*, Gro Bjerknes et al. (Eds), Studentlitteratur, Lund 1990.

Bødker, Keld, and Finn Kensing. Design in an Organizational Context - an Experiment, in *Scandinavian Journal of Information Systems,* Vol 6, no. 1, (April 1994), pp.47-68

Bødker, Keld, and Jesper Strandgaard Pedersen. Workplace Cultures: Looking at Artifacts, Symbols, and Practices. In Greenbaum, Joan, and Morten Kyng (eds). *Design at Work: Cooperative Design of Computer Systems.* Lawrence Erlbaum Associates, Chichester, UK, 1991.

Bødker, Susanne. Creating Conditions for Participation: Conflicts and ressources in systems design, in *proceedings of the Third Biennial Conference on Participatory Design, October 27-28, 1994, Chapel Hill, North Carolina.* Computer Professionals for Social Responsibility, Palo Alto, CA 94392-0717, USA, 1994, pp. 13-20.

Carstensen, P., *Computer Supported Coordination,* Ph.D. Thesis, Department of Computer Science, Roskilde Universty, 1996.

Checkland, Peter and Ji Scholes: *Soft Systems Methodology in Action*, Chichester, West Sussex, UK, 1990.

Christensen, Søren and Jan Molin: *Organisationskulturer - kultur og myter, deltagerstyring, ledelse og omstilling, viden og erfaring, design,* [Organizational Cultures - Culture and Myths, Participatory Government, Management and Change-Over, Knowledge and Experience, Design], Akademisk Forlag, Copenhagen, Denmark, 1983.

Clausen, Hasse: "Narratives as Tools for the Systems Designer", *Design Studies*, Vol. 14, No 3, July 1993 pp. 283-298.

Coad, P., and E. Yourdon: *Object-Oriented Analysis.* Englewood Cliffs, New Jersey: Yourdon Press, 1991.

Dahlbom, Bo, and Lars Mathiassen. *Computers in Context. The Philosophy and Practice of Systems Design.* Blackwell, Oxford, UK, 1993.

Ehn, Pelle, and Åke Sandberg: *Företagsstyring och Löntagermakt. Planering, datorer, organisation och fackligt utredningsarbete.* Bokförlaget Prisma, Arbetslivscentrum, Sweeden, 1979.

Ehn, Pelle: *Work-Oriented Design of ComputerArtifacts.* Arbetslivcentrum, Stockholm, 1988.

Ehn, Pelle, T. Meggerle, O. Steen, and M. Svedemar: "What Kind of Car is this Sales Support System? On Styles, Artifacts and Quality-in-Use", *Proceedings of the Third Decennial Conference - Computers in Context: Joining Forces in Design, August 14-18, 1995*, Department of Computer Science, Aarhus University, Aarhus, Denmark, 1995, pp. 126-136.

Freedman, D. P. and G. M. Weinberg: *Handbook of Walkthroughs, Inspections and Technical Reviews*, Little, Brown and Company, Boston, 1982.

Gougen, Joseph A. and Charlotte Linde: "Techniques for Requirements Elicitation", *Proceedings of the IEEE International Symposium on Requirements Engineering. January 4-6, 1993, San Diego, California,* IEEE Computer Society Press, Los Alamitos, California, 1993, pp. 152-164.

Greenbaum, Joan: "PD: A Personal Statement", *Communications of the ACM*, Vol. 36, No. 4, June 1993, p. 47.

Grudin, Jonathan: "Interactive Systems: Bridging the Gaps Between Developers and Users", *IEEE Computer*, April, 1991, pp. 59-69.

Grønbæk, Kaj, Morten Kyng, and Preben Mogensen: "Cooperative Experimental Systems Development: Cooperative techniques Beyond Initial Design and Analysis", *Proceedings of the Third Decennial Conference - Computers in Context: Joining Forces in Design, August 14-18, 1995*, Department of Computer Science, Aarhus University, Aarhus, Denmark, 1995, pp. 20-29.

Hammer, Michael and James Champy: *Reengineering the Corporation. A Manifesto for Business Revolution.* Nicholas Brealey Publishing, London, 1993.

Hughes, John A., Val King, Tom Rodden, and Hans Andersen: "Moving Out from the Control Room: Ethnography in Systems Design", in *Proceedings of the Conference on Computer Supported Cooperative Work, October 22-26, 1994, Chapen Hill, North Carolina, USA,* Association for Computing Machinery, New York, N.Y., 1994, pp. 429-439.

Kensing, F.: "Generation of Visions in Systems Development - A Supplement to the Toolbox", in P. Docherty et al.: *Systems Design for Human Development and Productivity: Participation and Beyond*, Springer Verlag, 1987.

Kensing, Finn and Kim H. Madsen: "Generating Visions - Future Workshops and Metaphorical Design", in Joan Greenbaum and Morten Kyng (Eds): *Design at Work: Cooperative Design of Computer Systems,* Lawrence Erlbaum Associates, Chichester, UK, 1991.

Kensing, Finn and Terry Winograd: "Operationalizing the Language/Action Approach to Design of Computer-Support for Cooperative Work", in R. K. Stamper et al. (Eds.): *Collaborative Work, Social Communications and Information Systems*, North-Holland, The Netherlands, 1991, pp. 311-331.

Kensing, Finn and Andreas Munk-Madsen: "Participatory Design; Structure in the Toolbox", *Communications of the ACM*, No. 36, Vol. 4, 1993, pp. 78-85.

Kensing, Finn, Keld Bødker, and Jesper Simonsen: *An Emerging Approach to Systems Design - Experience from the MUST Program*, Roskilde University (Datalogiske skrifter No. 94), Roskilde, Denmark, 1994.

Kensing Finn, Jesper Simonsen, and Keld Bødker: *"Designing With Cooperating Communities"*, paper submitted for publication 1996.

Kyng, Morten, and Lars Mathiassen: Systems Development and Trade Union Activities", *Information Society for Richer, for Poorer*, North-Holland, 1982, pp. 247-260.

Kyng, Morten: "Making Representations Work", in *Communications of the ACM*, Vol. 38, No. 9, September 1995, pp. 46-55.

Lanzara, Giovan Francesco: "The Design Process: Frames, Metaphors, and Games", *System Design For, With, and By the Users*. North-Holland, 1982, pp. 29-40.

Lanzara, Giovan Francesco, and Lars Mathiassen: *Mapping Situations Within a System Development Project. An Intervention Perspective on Organizational Change.* MARS-report No. 6, DAIMI PB-179, Computer Science Department, Aarhus University, 1984.

Lyytinen, K. and R. Hirschheim (1987): Information systems failures - a survey and classification of the empirical literature. In *Oxford Surveys in Information Technology*, vol. 4, pp 257-309

Mathiassen, Lars: *Systemudvikling og Systemudviklingsmetode*, [System Development and System Development Method], DAIMI PB-136, Due-Report No. 5, Datalogisk Afdeling, Matematisk Institut, Aarhus, Denmark, 1984. [For an English version of the referred charactarization of methods, see (Andersen, 1990)].

Mumford, Enid: "Job Satisfaction: A Method of Analysis", *Personnel Review*, Vol. 1, No. 3, Summer 1972.

Mumford, Enid, Frank Land, and John Hawgood: "A Participative Approach to the Design of Computer Systems", *Impact of Science on Sociology*, Vol. 28, No. 3, 1978, pp. 235-253.

Mumford, Enid: "The ETHICS Approach, *Communications of the ACM*, No. 36, Vol. 4, 1993, p. 82.

Naur, P. Programming as Theory Building. *Microprocessing and Micorprogramming,* 15 (1985), 253-261.

Nygaard, Kristen: "Kunnskaps-strategi for fagbevegelsen", *Tidsskrift för Universitets- och Forskningspolitik. Nordisk Forum 6*, Vol. 10, No. 2, Nordiska Sommaruniversitetet förlag, Sweeden, 1975, pp. 15-27.

Plowman, Lydia, Yvonne Rogers, and Magnus Ramage: "What Are Workplace Studies For?", in H. Marmolin, Y. Sundblad, and K. Schmidt (Eds.) *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work*, September 10-14, Stockholm, Sweden, Kluwer Academic Publishers, 1995, pp. 309-324.

Rasmussen, Jens, Annelise Mark Pejtersen, and L. P. Goodstein: *Cognitive Systems Engineering*, John Wiley and Sons, Inc., 1994.

Sachs, Patricia: Transforming Work: Collaboration, Learning and Design, in *Communications of the ACM*, Vol. 38, No. 9, September 1995, pp. 36-44.

Schmidt, Kjeld: "Function Analysis Instrument", in G. Schäfer et al. (Ed.): *Functional Analysis of Office Requirements: A Multiperspective Approach*, Wiley, Chichester etc., UK, 1988, pp. 261-289.

Schmidt, Kjeld: *Analysis of Cooperative Work. A Conceptual Framework,*, Risø National Laboratory, (Risø-M-2890), Roskilde, Denmark, 1990.

Schön, Donald A.: *The Reflective Practitioner: How Professionals Think in Action*, New York, Basic Books, 1983.

Schön, Donald A.: "Designing as Reflective Conversation with the Materials of a Design Situation", *Knowledge-Based Systems*, Vol. 5, No. 1, March, 1992, pp. 3-14.

Shapiro, Dan: The Limits of Ethnography: Combining Social Sciences for CSCW, in *Proceedings of the Conference on Computer Supported Cooperative Work, October 22-26, 1994, Chapel Hill, North Carolina, USA,* Association for Computing Machinery, New York, N.Y., 1994, pp. 417-428..

Simonsen, Jesper: *Designing Systems in an Organizational Context: An Explorative Study of Theoretical, Methodological, and Organizational Issues from Action Research in Three Design Projects.* Ph.D.-thesis, Roskilde University (Datalogiske skrifter No. 52), Roskilde, Denmark, 1994.

Simonsen, Jesper: "Involving Customer Relations in Contextual Design - a Case Study", in J.D. Coelho et al. (Eds.), *Proceedings of the 4th European Conference on Information Systems, Lisbon - Portugal,* 1996, pp. 1153-1161.

Simonsen, Jesper, and Finn Kensing: "Take Users Seriously, but Take a Deeper Look: Organizational and Technical Effects from Designing with an Ethnographically inspired Approach", in *proceedings of the Third Biennial Conference on Participatory Design, October 27-28, 1994, Chapel Hill, North Carolina.* Computer Professionals for Social Responsibility, Palo Alto, CA 94392 -0717, USA, 1994, pp. 47-58.

Simonsen, Jesper, and Finn Kensing: "Using Ethnography in Contextual Design." Accepted for publication in Communication of the ACM, 1996.

Stolterman, Erik: *Designarbetes dolda Rationalitet. En studie av metodik och praktik inom systemutvekling.* [The hidden rationale of desing work - A study in the methodology and practice of system development]. Ph.D.-thesis, Research reports in information processing and computer science: Nr 14.91, Institutionene for Informationsbehandling, Administrativ databehandling, Umeå Universitet, 1991.

Stolterman, Erik: "How Systems Designers think about Design and Methods: Some Reflections Based on an Interview Study", *Scandinavian Journal of Information Systems,* Vol. 4, Institute of Elektronic Systems, University of Aalborg, Aalborg, Denmark, 1992, pp. 137-150.

Yourdon, E.: *Modern Structured Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1989.

Winograd, Terry and Fernando Flores: *Understanding Computers and Cognition, A New Foundation for Design*, Addison-Wesley, New York, 1986.