Activity theory as a challenge to systems design

Susanne Bødker

Department of Computer Science, Aarhus University, Ny Munkegade 116, DK-8000 Aarhus C, Denmark

Abstract

This paper suggests an improvement of the theoretical foundation of information systems research of the 1990s. This foundation is found in human activity theory. The paper deals with how human activity theory can help systems design change, theoretically and practically. Applying activity theory to systems design, certain cornerstones become apparent, on which future systems design practice and research need to build. The paper will discuss these cornerstones, and the implications of the new understanding for design practice and research.

INTRODUCTION

This paper takes a look at the present state of systems design practice and research. It argues that present systems design is really in a conflict situation for a number of reasons including that

- the material that is worked with (computers) become more and more flexible, and that the tools and techniques used in systems design today are insufficient for that kind of material and for the challenges posed by ready-made software,
- and, that real user influence is necessary in more and more cases.

In recent years, activity theory has been brought into the arena by a number of researchers in systems design. This paper will look at how *human activity theory* can inform systems design, i.e. at how human activity theory may help systems design change theoretically, and not least, practically.

This paper focuses on a change of understanding of systems design, rather than on a new approach to it. The need for a new theoretical foundation for systems design has been suggested recently a number of researchers (e.g. Winograd and Flores (1986), Ehn (1988), Nurminen (1989)). They agree that the traditional rationalistic thinking is insufficient as a theoretical basis for systems design. Their reaction is primarily to the narrow technical focus of traditional systems design. The use of such philosophers as Heidegger and Wittgenstein has become very popular in this search for a new foundation. However, some of these writings have become rather idealistic, looking very little at the actual world around people, the conditions for how people act towards each other and their artifacts, and the technical aspects of the computer application seems to become a matter of little concern. On the other hand, often materialism is seen as equalling realism. From a realism stance (Juul-Jensen, 1989) language and descriptions can be seen as representations of objectively existing structures in reality. We need to find a theory, which explains why we cannot just capture this reality in a description or a program. If we want to maintain a theoretical basis, arguing that we cannot fully predict the future use, we need to reject the realist approach. Activity theory may be an approach which makes this possible.

Human Activity Theory

Human activity theory is based upon the anthropological/ psychological theory of A. N. Leontjew (1978, 1981) and L. S. Vygotsky (1978). Its use when dealing with issues concerning the design and use of computer technology is described for this conference by Kuutti (1990). This theory is a very general philosophical framework for understanding the development of human culture and individual personality based on dialectical materialism. A historical understanding of human and societal development is important in analyses based on the framework. The theory can be seen as a framework for understanding the totality of human work and praxis, and the deliberate processes changing this, i.e. a totality encompassing organizational development, design and use of computer artifacts.

Some of the key-points of human activity theory relevant for the further discussion are:

- Human activity is always mediated by tools, language, etc. These are made by humans and they *mediate* our relation with what we produce and with other human beings. The artifacts are not as such objects of our activity.
- The ways of doing work, grounded in tradition and shared by a group of workers, we call praxis. When getting trained at some piece of work, one gets to share this praxis. At the same time each individual who holds a praxis continues the praxis, and changes it as well.

The essence for design may be that practice should not be changed from the outside, but in a collective process by the designers and the workers themselves. Engeström (1987) is looking at change processes in organizational settings and sees contradictions of various kinds as the cornerstones of such change. His starting point is in a *problem situation*, where there is a reason for somebody to want a change (in our case for somebody to want a new computer application). The artifacts that we work with are under a constant reconstruction, due to conflicts in the way they are applied. These conflicts occur at a number of levels, due to which the reconstruction cycles are different in size. For example, a banks accounting system is no longer efficient for the services that the bank wants to give to its customers. A large project is set up and a new system designed. Another example is that I all the time get new and more advanced applications available for my Macintosh. I may sometimes want to use some of these, because I get better support for what I want to do, because I need to work with a colleague who has different software from mine, etc. Systems design includes a multitude of such cycles, which goes on more or less integrated with use, using more or less well-known materials, design tools and techniques, with a more or less clear understanding of the product.

In systems design it is a main purpose to try to predict the future use activity. At the same time, we will never be able to fully make such a prediction. The future will always shape itself differently from what we predicted. This gives rise to new conflicts, which will in turn lead to new design. The strength of an activity theory framework is that this *understanding can be included explicitly*, whereas for a number of other theories, this is a side-effect that needs to be dealt with in practice, because we were not quite successful in our prediction of the future.

By this focus we can define systems design to include a variety of projects that deals with change of work using computer equipment. Some design processes are conducted in a fairly standard way, taking on, for the designers, well-known problems, employing well-known methods, etc. Other projects exist in a world where the products, conditions, etc., are continuously changing, and where no real stable platform has been found. This definition is somewhat broader than the definitions of systems design, systems development or information systems development that are normally applied within our area. A widening of the scope though seems not only relevant but even necessary, because many real-life problems of today lie outside of traditional ISD dealing with processing of large quantities of data or information. Examples of this are design of many types of workstation-based applications for office work, where the problems relate to anything from user participation to design of the user interface.

The Problems of Design

Here is a further investigation of some of the problems that systems design is facing today. The examples that I use in the following are toy examples in the sense that they are not based on systematic empirical investigations. At the same time they are based, though, on my numerous encounters with real-life design projects.

A project group works on a computer application to be used in organizational change. Depending on the kind of project and level of detail on which we are looking, such a project group uses a number of instruments in this process: description tools, 4th generation tools, programming languages, interview guides, data flow diagrams, and so on. The community surrounding the project group consists of other systems designers and technical staff in the software house or computer department, of workers and managers in the use organization, etc. In and between the design and use organizations there is a division of work, as well as there is in between groups. This division of work sets up criteria for who is involved in this project to create the specific product that we are focussing on in our analysis. Furthermore, this whole setting has established certain rules for how projects are set up, how they deal with users and managers, etc.

Looking at possible contradictions within this activity system today, we may see that a project group often contains a number of conflicts:

- The participants have different qualifications, and if users are involved in the project group, they may view their role very differently from that of the professional systems designers.
- Users and designers may often have different expectations to the role of other groups: the professional designers may, for instance, think that the users can make descriptions of screen images themselves, whereas the users think that they are in the process to give advice to the designers (Holck Lauridsen & Nielsen, 1989).

In many cases there is a major potential for conflicts in the way design instruments are used in design process since the instruments are to serve many different purposes in design:

- Data flow diagrams are used to specify the product. They are also used to communicate in the group about how work is done at present. They may not serve both purposes equally well.
- Available tools are not suited for the kinds of products that are build, or they do not support the division of work in the project organization. These are examples of conflicts between the instrument used and the product, in one case, and the division of work in another.
- Many project groups believe that they would do better design if they somehow managed to find the design method appropriate for them. This is a conflict between the design activity and an idealized one. Some projects find design methods which improve their work, but somehow the ideal method is always around the corner (Lyytinen, 1987).
- More and more advanced so-called user friendly applications are available on the market as standard products. In particular, such advanced standard products as spreadsheets and WYSIWYG technology provide different possibilities from the materials, products and instruments of "ordinary" systems design projects. This creates a certain interest and need in the use organizations of getting products similar to these advanced ones, and thus a demand to design similar applications. We see a conflict between the design activity and the use activity that is designed for.
- The project organization often have neither the competence nor the technology to build such applications. They live in a world where COBOL or Pascal based applications have been dominating. Many project organizations see that they need to look for different tools. The activities producing technical tools for design have started to build such different programming environments a long time ago but these have not yet been taken into use in most organizations.

All in all, there is a growing demand on the design organization. From the use organizations, of dealing with the users in a better way, and of developing a different kind of products than before. This in turn makes it necessary for the design organization to try to use different design methods, as well as different tools such as 4th generation tools and object-oriented programming languages. Providing a better use situation for the users seem to include both setting up a better design process, creating better use settings **and** exploiting the technical possibilities better. This is not without problems because a major change of design practice and of the role of the designers are required: As Lyytinen (1987) points out, "advancing technology offers great *opportunities* for improving all IS processes.." The development organization though, often have neither the skills nor the techniques of handling these opportunities.

In the following section, we will look at the systems design activity, and how our understanding and practice ought to be changed from an activity theory point of view.

HOW CAN ACTIVITY THEORY INFORM SYSTEMS DESIGN?

In this section activity theory will be used to discuss the specific human activity that is called systems design. Systems design will be discussed in relation to the activity that it is aiming to change, the use activity. In particular its specific object, the computer application, will be discussed, as will the many roles of computer technology, as material and as instrument of the design activity. Finally, the roles of the subject of the design activity, the designers will be discussed, as they are, ultimately, the mediators between the design and the use activities, and between the technology and the human users.

Organizational Change and Systems Design - the process

The idea of conflicts as a basis for organizational change cycles is developed further with respect to systems design in Bisgaard et al. (1989a and b). They claim that the following criteria are fundamental for this way of understanding systems design:

A. the development of a new artifact moves on simultaneously with the development of the organization in general. If the organization is not considered in together with the artifact, one may introduce new contradictions instead of helping out with the original one.

B. conflicts are considered fundamental, and used as a resource in the change process.

C. systems design starts out from the users, and it is the development of their praxis which is in focus.

D. design is seen as a collective (between users and designers) learning process where two principles are fundamental: the application of a methodological cycle, and the notion of the zone of proximal development.

The methodological cycle is moving from an analysis of the activity and the surrounding activities, to the creation of instruments by which the users can transcend their own practice, thus creating a vision of the change (as examples of methods that can be used are mentioned future workshops, reframing techniques, prototypes), to an implementation of the final new instrument into the organization. The idea here is, that inasmuch as we try to predict how praxis will change, artifacts are used differently from the original intentions, and new contradictions are always introduced, causing the need for new artifacts to arise.

Vygotsky's idea of a zone of proximal development is a concept for talking about the difference between the present qualifications/praxis of the workers, and the potential qualifications that they can readily acquire under proper guidance.

What I suggest is that in systems design we may use the idea of the zone of proximal development as a guidance for what change of work practice we can hope to achieve. For instance, when designing a point-of-sales system, can we hope to make all cashiers, some of whom may only work a few hours a day, knowledgeable in all the products in the supermarket? These two learning principles, together termed "learning by expansion" (Engeström, 1987), are discussed by Kuutti (1989) in relation to creativity. What he suggests is that it is expansive development that we ought to look for in design, rather than a metaphysical "creativity". *This turns the notion of creativity away from being a question of somebody getting a good idea, towards something that we can deal with, collectively, in design in a systematic way.*

Systems design includes many activities conducted when a computer application is designed and introduced in the workplace: investigation of goals and impacts of the change, design and implementation of the computer application, introduction of the computer application in the work activity, education and development of education material, etc. Applying activity theory to systems design means to take serious all aspects of activity: from activity motive formation to the triggering of operations under certain material conditions. From the previous discussions it is apparent that a view of systems design as something totally technical dealing only with the programs and hardware devices does not work. Neither does a view focussing on the social and organizational aspects of work. What activity theory suggests to us is that systems design must include design for operations, actions as well as activities, i.e. technical as well as social and organizational aspects. *Systems design, though, lacks tools for dealing with many of these aspects, as well as the dynamics of the understanding that activity theory provides*.

What we do in systems design is to try to predict and describe the future, partly because various specifications of the programs are needed, and partly because we need to get some understanding of the change that is caused when introducing the new computer application. In much traditional systems design this is done in description languages, such as data flow diagrams, ISAC graphs, or DELTA descriptions. More recently, less formal approaches such as prototyping and wall-graphs have become common (Bødker & Grønbæk, 1989, Kuvaja, 1989, Bjerknes & Bratteteig, 1987). Some of these techniques aim to help the designers hold on to already articulated knowledge, e. g. actions, whereas other techniques aim to help forming/experiencing future operations (Bødker, 1987), or to reshape the activity more generally (e.g. Nurminen, 1989 or Ehn, et al. 1989). Activity theory can explain to us why we are never really successful in creating a world that is like what we predicted, because it is the actual situation, the physical and social conditions of the activity that determines how the operations are carried out, not any identifiably set of conditions (see also Bødker, 1987). This could mean that our prediction tools need to be rethought - descriptions that trigger notions of change for the participants may be more important than such which aims to describe the "world" as well as possible (see e.g. Holbæk-Hanssen et al., 1975). Furthermore, the mere thinking of the new computer application as something which, when introduced, causes new contradictions, and thus new change cycles of different kinds, suggest that we *ought to think of design as re-design* (Floyd, 1987), not as "one-shot" processes. This is well in line with how artifacts, within this framework, are seen as historical devices which reflect the state of praxis up until the time that they are developed. Design issues such as designing for flexible use, or tailorability (Trigg et al., 1987, Henderson & Kyng, 1989) becomes important in this light.

The distinction between what we normally call design and what we normally call use seem to vanish or be somewhat blurred in this perspective: Some change cycles are small, including only a small change of the product produced, the devices used, or the set-up of the user interface. Engeström(1989) actually argues that any use of a piece of machinery or technology is a technological change process. He uses the picture of a large machine which is, seen over time, taken apart, and brought back together many times, while worn out parts are replaced with new ones, new work tasks arranged around the machine, new workers are brought in with different experiences, etc. Other changes are large, all-encompassing processes. Systems design seems to need a redefinition to contain all of these change processes, or in a certain way the term gets obsolete when it comprises a multitude of processes from use and adaption to routine design and design of qualitatively new types of technology. Design, in a historical light, was introduced to *separate creation and use* (Cooley, 1984). *What is suggested here is to bring the two back together.*

Human activity theory entails a perspective on design where there is a tight coupling between design and use. Where changes, including changes of the computer applications go on continuously, driven by the conflicts occuring at different levels in organizations and society. Activity theory gives certain instruments that make it possible to deal deliberately with these change processes, and it emphasizes that all levels and aspects of the activity needs to be dealt with in design. In the following we shall look more closely at the specific roles of the computer applications as instruments and materials of human activity.

Computer Applications as Instruments and Materials

Computer applications play the role of instruments in two situations, in the work activity where the outcome of design is applied, and in design. In this sense, the design process is of course also a use process, even though the computer applications used in the two situations are most likely not the same.

As pointed out in Bødker (1987, 1989), the view on computer applications as mediators is an important one, if we want to know how to design better computer applications. This view allows us to better look at the computer application not only as a product of the design activity, but also as a potential mediator of the use activity. As artifacts are not themselves meant to be objects for the activity, users are normally not meant to focus on the artifacts. *To the users, artifacts are what they are used for.* This future role of the artifact is important in design, and thus it is necessary to design both the social and physical conditions for the use activity: What is created in systems design is software and hardware supporting the human operation of the computer application in a specific type of use activity. What is created is also, perhaps, a change of the purpose of the activity or of what purpose individual actions fulfill as part of the activity. All of these levels needs to be dealt with in design, but only some of them can be communicated about in the traditional sense. The operational aspects need to be dealt with by acting, because it is only then that *the actual material conditions can be tried out* (see Bødker, 1987).

Looking back at Engeström's way of describing activities, we will see, that in systems design, computer technology can play many roles, often even at the same time. Besides from the role of instrument, computer software and hardware is included in the *materials* we work with, and an important object and outcome of systems design is the computer application. In some situations, the computer-based design instruments may be very different from the material that is actually shaped in the design process: case tools which are used to make data flow diagrams, are different from e.g. the COBOL compiler, operating system etc. on the computer where the specified system is eventually implemented. In other situations the distinction is hardly there at all, e.g. when a LISP environment is used first to build a prototype, and later on to build the final application. A data dictionary may be used to support certain rules of how variables are named in a program, even in cases where work is divided among several programmers. The use of computer-based instruments in design has changed dramatically recently. Until then, computer-based instruments, and support for cooperation were hardly found at all. Only very low-level instruments for structuring programs and for debugging were available. The rest was paper work, and techniques for structured decomposition were important. With advanced CASE tools, user interface management systems, 4th generation tools, and advanced programming environments, the situation is tending to become different. At least in research settings, we have seen that more flexible, exploratory, prototyping ways of working gains ground (Bødker 1987). With such technology, a formal specification of software functionality (the what can be done), detached from the specification of hardware, and from how the application is in use, is no longer an advantage. This effects in turn also the early stages of design where, according to activity theory, *it is important to create specific visions about possible future work activities*. Once these instruments are available, they can of course also be used here, and not just in the later realization of a vision.

To understand use, and in particular future use, it is necessary to apply an experimental design strategy where specific visions, such as prototypes of different kinds are presented to the users very early, for them to try out the conditions of the future activity. This requires for the designers that they have flexible instruments, and in particular that they know these instruments as well as the materials that they work with well. The new materials, such as advanced programming environments, require different ways of working from what has this far been known in systems design.

Who is Acting in Design ? - The Roles of the Systems Designer

No doubt, the role of the systems designer will be widened if following the methodological cycle of Engeström (see Bisgaard et al., 1989a and b). Not only does the cycle include stages which are not normally conceived as part of the design process (this includes both initial investigations of the work activity and its historical development, and following up and consolidation of the changed work activity), it has been pointed out that a collective learning process among users and designers is important, that a change of the users' praxis needs to be dealt with at all levels, including both technical, educational and work organizational conditions of the future activity. At the same time, the systems designer is the outsider who may help the users transcend their own practice. Furthermore, since systems design is the main activity for a systems designer, but not for a future user, the designer must take on the role of a facilitator, who makes sure that the process gets going in different ways. The systems designers initially get to choose the instruments to be applied, even though these may later be changed in the collective process with the users. The systems designers get to set the stage whereas later moves must be done in cooperation with the users, and instruments must, in this sense be chosen depending on the material conditions of the actual design situation, to follow the track from the characterization of the use situation. Some of these tools may be traditional description techniques, or the like, and some may be some of the "new ones", e.g. as suggested by Bisgaard et al. (1989a and b): mapping from Lanzara & Mathiassen (1985), future workshops, role playing, mock-ups, prototyping (see also Greenbaum & Kyng, in preparation), setting up of limited test situations. Furthermore some may be exploratory programming, use of 4th generation tools etc.

A well-trained systems designer, who has experience with many of these methods, may not even need to deliberately choose one of these methods, it may be routine for the systems designer to pick the right one. The technique is triggered by the actual material conditions (Bødker, 1987). The familiarity of the design activity, and thus, the learning taking place is a consequence of the material conditions: to what extend can the present repertoire of operations function in the situation? and to what extent can the activity and actions be given meaning within the previous practice? An understanding of the problem under consideration includes the past, the present and the future: the history of the artifacts used in the activity, the present use praxis, and the product's role in future use. There is an understanding of the material with which the designers work, and there is an understanding of the instruments applied. How familiar are all of these? This question needs to be asked not only on behalf of the professional designers, but also on behalf of the users, if we fully want to understand design because we are dealing with a collective process.

It belongs to the tasks of the designers to set up this process and get it going. Designers be able to handle numerous types of design instruments to emphasize different aspects of the outcome of design, from advanced programming languages and prototyping environments to organizational games (Ehn & Sjögren, 1990). The wide variety of qualifications is hardly contained in any education as of today which implies a need for a changed education of designers in the 1990s. Such an education must be practical as well as theoretical. It must deal with advanced technology as well as with human and organizational issues, as described earlier in this paper.

Activity Theory as a Basis for Understanding and Action in Design

As an analytic framework, I will suggest that activity theory is useful in analyzing work in several respects. To understand the work of the future users makes an analysis of both intentional and operational aspects necessary. This in turn, as we have seen discussed in e.g. Christiansen (1989) and Bødker (1987), leads to a better understanding of how learning takes place in work, and what role the computer application plays. Furthermore a dynamic, historical dimension is introduced in the analysis - how and why do the praxis of the workers (including tools, in particular computer applications) change the way it does? An analysis of systems design as an activity system allows for a focus on many different levels of the total activity. We can look at the materials and tools used by a couple of programmers, or we can look at the total project group in relation to the surrounding organizational units. What is the purpose of their activity, what do they do to fulfill this purpose, and how do they do it? What are the actual social and physical conditions for their work? In particular this analysis points at the many different levels where systems design tools and techniques comes in. And it points at the many parts played by computer technology in the design process.

As a practical reorganization of systems design, activity theory offers the possibility of seeing use and systems design as a multitude of change cycles, where computer applications as well as other parts of the work activity are constantly reconstructed, using more or less well-known materials, design tools and techniques, with a more or less clear understanding of the product. An explicit awareness of these cycles may change our way of doing design. This is following along the lines suggested by e.g. Floyd (1987), but taken further because conflicts can be acknowledged and taken seriously in design. The implications of taking serious conflicts in design needs to be developed much further to avoid helping only the groups in the organization which are, on beforehand, the strongest.

It is suggested to widen the scope of the design process to something which also includes initial studies of praxis, the use activity, and the devices employed in this. And to be more concerned for how the computer application is ultimately introduced in the use activity. Furthermore, since an explicit focus on both the social and physical conditions of future use is necessary, a more exploratory way of working, than what is often seen in present systems design, seems necessary. This includes working with technical and organizational issues throughout the process, not starting with one, and ending with the other.

The practical ways of acting differently when doing design or analysis of design/use situations are in the end what counts when it come to determining the possibilities for human activity theory to inform systems design research and praxis. Many of the constraints and conditions for such ways of acting remains to be explored.

SYSTEMS DESIGN IN THE 90S

Bringing up-to-date the practical design activities in itself seems to be a challenge of the 90s. What we see is a conflict between what Engeström (1987) calls the rationalized design activity where the acting subject is management, the object and outcome is predetermined, the instruments complex machinery and explicit knowledge, division of labour a hierarchy, and the rules codes of competition and control, and the humanized design activity. In the humanized design activity the subjects are overall management as well as semi-autonomous groups, the object, outcome and instruments introduced from more research-oriented activities, division of labour semi-autonomous groups and the rules, cooperation and communication within group, competition and control between groups.

The differences between these types of projects seem to indicate that there are major theoretical differences between different kinds of systems design projects, depending on the above discussed characteristics. What I would like to suggest is that these differences may be too big for us to cover under the same theme of research. Maybe the differences between different kinds of what we today denote (information) systems design projects are bigger than the similarities. And maybe the research challenges attached to these different kinds of "(information) systems design" are too diverse to belong in the same "research arena", in the 90s. And certainly, some types of "(information) systems design" projects are too standardized and familiar for the conductors to even deserve research attention.

Where then, are the research challenges of the 90s? With an eye to the above discussions, I suggest that research in the 90s should focus on how computer applications can be designed to mediate human work, starting out from human practice. Norman's (1989) design of everyday artifacts, and Carroll's (1990) task-artifact cycle are examples of early attempts of such an understanding. In Bødker (1987, 1989), and Bannon & Bødker (1989) frames are set up for how we can *understand how computer-based artifacts mediate use*, but definitely this analysis can, and ought to be taken further. The idea of computer applications as mediators emphasize the connection of human beings with each other and with the objects that they work on through a transparent computer application. To some extent this implies a totally new perspective on the computer application than than of a system or an information system.

Conceiving design tools and understanding design materials seem to be necessary ingrediences in an improved understanding. Ways for users to participate actively in design and experience their future work activity are important as well. The cooperative prototyping idea of Bødker & Grønbæk (1989, 1990) is one such approach. Here prototyping is done in a cooperative process emphasizing mutual learning between designers and future users and evaluation of prototypes in use-like settings. What we have seen today is primarily how prototyping and the like can be used to investigate the operation of the future artifact, but human activity theory suggests that we need to "prototype" the intentional aspects as well (see Bisgaard et al., 1989a). In other words, we should not fall into the trap, where we do not allow/help the users formulate or articulate their wishes for the future. *How we can help users change their total activity* in this fashion seems, from this perspective, to be a major issue. Furthermore, the framework could help us *understand what role computer-based objects as well as other objects, such as prototypes and descriptions play in actual design situations*. Examples of what can be looked for is e.g. whether the object is the same for participating users and designers, in situations where they are together using and developing a prototype, or discussing a description. The material anchoring of the theory makes the analysis of these artifacts and objects in use very important for our detailed understanding of design situations. An initial analysis of this kind has been carried out in (Bødker & Grønbæk, 1990).

Not much systems design research or practice of today includes historical analyses of the use practice in question. Often we start out from the given situation without trying to understand how the practice have evolved until then. The historical analyses of the Utopia project (Bødker et al. 1987, Ehn & Kyng, 1984, Ekdahl, 1984) of composition technology may belong to the exceptions. In this case it proved useful for the design to understand how different generations of technology had posed constraints on how a newspaper page could be set up, and thus made impacts on the concept of quality typography over time. We have little *practical or theoretical understanding of what it means to make analysis of generations of computer technology, and how and why they have succeeded each other*, even though it seems clear that many design activities would benefit from such an analysis, not least of past problems and failures.

When looking at many of the design/research projects that we know of today where conflicts are taken seriously in design, we see that they are one-party projects (see e.g. Ehn & Kyng, 1987). In such projects the conflicts of organizations are often dealt with as outside the project as such. Not many research projects have been carried out with project groups, consisting of groups with recognized conflicting interests. We do not know much about how cooperative prototyping, for instance, function in such situations. Taking human activity theory seriously, *we must investigate ways in which groups of users and designers can make use of internal as well as external conflicts as resources of design*. Furthermore, we need to know more about the specific historical and cultural setting in which these projects have taken place and investigate ways of setting up similar projects in other cultural settings. This has become important, not least, in light of the development in Eastern Europe.

With these programmatic statements, the paper is approaching its end, since this workshop is one which intends to ask questions of perceptions and challenges of the 1990s.

ACKNOWLEDGEMENTS

This paper had not been possible without the valuable comments of Heinz Klein, Preben Mogensen, Liam Bannon, Pelle Ehn and Kaj Grønbæk.

REFERENCES

Bannon, L. & Bødker, S. (in press). Beyond the Interface - encountering artifacts in use. In J. Carroll. *Designing Interaction: Psychological Theory at the Human-Computer Interface* (publication expected 1990).

Bisgaard, O., Mogensen, P., Nørby, M., & Thomsen, M. (1989a). *Systemudvikling som lærevirksomhed, konflikter som basis for organisationel udvikling* [Systems development as a learning activity, conflicts as the origin of organizational development] (DAIMI IR-88). Århus: University of Aarhus.

Bisgaard, O., Mogensen, P., Nørby, M., & Thomsen, M. (1989b), Expansive systems development. In S. Bødker (Ed.), *Proceedings of the 12th IRIS seminar*, (pp. 37-56) Skagen, Denmark, August 1989.

Bjerknes, G. & Bratteteig, T. (1987). Florence in Wonderland. Systems Development with Nurses. In Bjerknes, G., Ehn, P., & Kyng, M. (Eds.) (1987) *Computers and Democracy – a Scandinavian challenge*, (pp. 279-296). Aldershot, UK: Avebury.

Bødker, S. & Grønbæk, K. (1989). Cooperative prototyping experiments- users and designers envision a dentist case record system. *Proceedings of the First European Conference on Computer-Supported Cooperative Work, EC-CSCW,* (pp. 343-357).

Bødker, S. & Grønbæk, K. (1990). *Cooperative prototyping- Users and designers in mutual activity*, submitted for International Journal of Man-Machine Studies.

Bødker, S. (1987). *Through the Interface – a Human Activity Approach to User Interface Design* (DAIMI PB-224) Aarhus: University of Aarhus (Published as book by Lawrence Erlbaum Associates, 1991).

Bødker, S. (1989). A human activity approach to user interfaces. *Human Computer Interaction*, *4*(3), 171-195.

Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M., & Sundblad, Y. (1987). A Utopian experience. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and democracy – a Scandinavian challenge*, (pp. 251-278). Aldershot, UK: Avebury.

Carroll, J. (1990). Infinite detail and emulation in an ontologically minimized HCI. In J. C. Chew & J. Whiteside, *Empowering people*, Proceedings of CHI '90. (pp. 321-328).

Christiansen, E. (1989). *Den Realistiske Vision* [The Realistic Vision] Unpublished doctoral dissertation, University of Aalborg, Denmark.

Cooley, M. (1980). *Architect or bee?–The Human/technology Relationship*. Slough: Langley Technical Service.

Ehn, P & Kyng, M. (1987). The collective resource approach to systems design. In Bjerknes, G., Ehn, P., & Kyng, M. (Eds.) (1987) *Computers and Democracy – a Scandinavian challenge*, (pp. 17–58). Aldershot, UK: Avebury.

Ehn, P. (1988). *Work-oriented Design of Computer Artifact*. Falköping, Sweden: Arbetslivscentrum/Almqvist & Wiksell International.

Ehn, P., & Kyng, M. (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*, (pp. 211-242). Helsinki: Helsinki Business School.

Ehn, P., & Sjögren, D. (1990). From systems descriptions to scripts for actions. In J. Greenbaum & M. Kyng, (Eds.), *Design at Work*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ehn, P., Möllerud, B., & Sjögren, D. (1989). Playing in reality. In S. Bødker (Ed.), *Proceedings of the 12th IRIS seminar*, (pp. 129-148) Skagen, Denmark, August 1989.

Ekdahl, L. (1984) At bli maskinens herrar [To become masters of the machine]. UTOPIA report no. 19. Stockholm: Arbetslivscentrum.

Engeström, Y. (1987). *Learning by Expanding*. Helsinki: Orienta-Konsultit.

Engeström, Y. (1989). Lecture at the Computer Science Department, Aarhus University.

Floyd, C. (1987). Outline of a paradigm change in software engineering. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and Democracy – a Scandinavian challenge*, (pp. 191-212). Aldershot, UK: Avebury.

Greenbaum, J. & Kyng, M. (Eds.) (in production). *Design at Work*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Henderson, A. & Kyng, M. (1990). There is no place like home: continuing design in use. In J. Greenbaum & M. Kyng, (Eds.), *Design at Work*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Holbæk-Hanssen E., Håndlykken, P. & Nygaard, K. (1975). *System Description and the DELTA Language*, Oslo: Norsk Regnesentral.

Holck Lauridsen, M. & Nielsen, H.B. (1989). Designprocessen som et sprogspil - belyst gennem et empirisk studie af 3 designprojekter [The design process as a language game - seen through an empirical investigation of 3 design projects]. Master Thesis, Århus: Aarhus University.

Juul-Jensen (1989) Den kulturhistoriske psykologi: Ideologisk metafysik eller objektiv teori? [The cultural-historical psychology: Ideological metaphysics or objective theory]. In M. Hedegaard, V. R. Hansen & S. Thyssen (Eds.), *Et Virksomt Liv* [An Active Life]. (pp. 142-187) Aarhus: Aarhus Universitets Forlag.

Kuutti, K. (1989). Flaws in rationalistic thinking, excact creartivity and information systems. In S. Bødker (Ed.), *Proceedings of the 12th IRIS seminar*, (pp. 373-392) Skagen, Denmark, August 1989.

Kuutti, K. (1990). *The theory of activity and its application on IS research*, ISRA Copenhagen, 1990.

Kuvaja, P. (1989). Workshop presented at the 12th IRIS seminar, Skagen, Denmark, August 1989.

Lanzara, G.F. and Mathiassen, L. (1985). Mapping situations within a system development project. *Information & Management* 8.

Leontjew, A. N. (1978). *Activity, consciousness, and personality*. Englewood Cliffs, NJ: Prentice-Hall.

Leontjew, A. N. (1981). *Problems of the development of the mind*. Moscow: Progress Publishers.

Lyytinen, K. (1987). Different perspectives on information systems: Problems and solutions. *ACM Computing Surveys, 19*(1).

Norman, D. (1989). *The Psychology of Everyday Things*, New York, NY: Basic Books.

Nurminen, M.(1989). *Three Perspectives to Information Systems*, Lund: Studentlitteratur.

Trigg, R., Moran, T.P., & Halasz, F.G. (1987). Adaptability and Tailorability in NoteCards. In H.C. Bullinger & B. Shackel, *Human-Computer Interaction - INTERACT '87*. (pp. 723-728). Amsterdam: Elsevier/North Holland.

Vygotsky, L. (1978), *Mind in Society*, Cambridge, MA: Harvard University Press.

Winograd, T., & Flores C. F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex.