ISSN 0105-8517

A Discussion of Computer Supported Cooperative Work

Overview Paper

Pål Sørgaard

DAIMI PB – 254
May 1988

AARHUS UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
Ny Munkegade 116 – DK 8000 Aarhus C – DENMARK
Telephone: + 45 86 12 71 88 Telex: 64767 aausci dk

Sammendrag (Danish summary)

Temaet for denne afhandling, edb-støttet samarbejde, er et forskningsområde, der er vokset frem i de seneste to til tre år. Samarbejde er naturligvis ikke noget nyt. Det nye er, at man nu lægger vægt på, hvorledes samarbejde kan bevares og understøttes ved brug af edb.

Det er et efterhånden anerkendt faktum, at anvendelsen af edb har en stærk påvirkning på brugernes arbejde og på arbejdsorganisationen. Anvendelsen af edb breder sig, edb bliver taget som en selvfølge. Samtidig bliver anvendelsen mere og mere integreret i arbejdet og i organisationerne. Det er derfor nødvendigt at anskue edb-system og organisation som et samlet hele og ikke som to uafhængige størrelser. Forskningsområdet "edb-støttet samarbejde" kan opfattes som et forsøg på, indenfor et afgrænset område, at forholde sig konstruktivt til denne erkendelse.

Der er ingen generel enighed om betydningen af begreberne samarbejde og edb-støttet samarbejde indenfor dette nye forskningsområde. I denne afhandling bliver der gjort et forsøg på at give disse begreber en mere præcis og snævrere betydning. Samarbejde opfattes som karakteriseret ved, at flere personer arbejder sammen på grund af opgavens natur, at de har fælles mål, at arbejdet udføres i en uformel, flad organisation, og at arbejdet udføres relativt autonomt.[1] Samarbejde opfattes som et af flere aspekter af en organisation. Samarbejdsaspektet opfattes som forskelligt fra hierarkisk ledet arbejde. I afhandlingen fokuseres der på relationen mellem edb-systemet og det arbejde, hvor edb-systemet bliver brugt. Denne relation kaldes anvendelses-relationen. Edb-støttet samarbejde forstås som en egenskab ved denne relation, nemlig den egenskab, at edb-systemet bliver brugt på en måde, der understøtter samarbejdsaspekterne i brugernes arbejde. Denne definition af edb-støttet samarbejde har blandt andet til hensigt at udtrykke, at dette forskningsområde hverken kan gives en rent teknisk eller en rent organisatorisk definition. Det er for eksempel muligt at forestille sig, at fuldstændig individuel brug af et edb-system kan understøtte samarbejde.

Der lægges vægt på tre anvendelses-relationer af væsentlig betydning for edb-støttet samarbejde. Disse relationer svarer til situationer, hvor et edb-system bruges som et medium, som et fælles materiale, og som et værktøj.

[1] Denne definition afviger fra den gængse betydning af ordet "samarbejde". Alternative formuleringer er gruppearbejde eller kooperativt arbejde.

Anvendelsen af edb som et medium er velkendt. Elektronisk post, konferencesystemer m.v. er efterhånden ved at være udbredte edb-anvendelser. Sådanne medier understøtter samarbejde ved at gøre horisontal kommunikation nemmere og for den sags skyld billigere. Et af formålene med et hierarki er at håndtere kommunikationskompleksiteten i en organisation. Nemmere og billigere horisontal kommunikation svækker dette argument for hierarki.

Anvendelsen af edb som et fælles materiale er en idé, der udvikles i denne afhandling. Udgangspunktet er den trivielle observation, at meget arbejde koordineres ved hjælp af egenskaber ved de genstande, der arbejdes med. Et eksempel er den koordinering, der sker mellem to personer der bærer et tungt møbel. Et andet, og måske mere nærliggende eksempel, er den koordinering, der finder sted ved brugen af et manuelt arkiv. Handlinger synkroniseres, fordi dokumenterne kun kan være et sted ad gangen. En sagsbehandler, der konstanterer, at et dokument er i brug, vil nogle gange være tvunget til at finde den, der arbejder med dokumentet med tanke på at koordinere arbejdsindsatsen. Når dokumenter og andre materialer lægges over på edb, mister de ofte nogle egenskaber, der er væsentlige for den måde hvorpå arbejdet bliver udført. Der foreslås i afhandlingen, at der konstrueres edb-baserede materialer, som har nogle egenskaber, der kan understøtte og formidle koordinering mellem brugerne. Dette kan blandt andet opnås ved at bruge nogle konventionelle materialer som design metaforer.

Det er nemt at få den forestilling, at når edb bruges som et værktøj er det et individuelt fænomen, som ikke har nogen betydning for edb-støttet samarbejde. Der er flere grunde til, at det ikke forholder sig sådan. For det første bruges edb-baserede værktøjer ved manipulation af de fælles materialer og til at få adgang til de edb-baserede medier. Værktøjer og materialer står i et uadskilleligt forhold til hinanden. Ved udvikling af fælles materialer er det nødvendigt også at lægge vægt på udviklingen af de værktøjer, der skal bruges. For det andet findes der eksempler på kollektive værktøjer, for eksempel en to-personers sav. For det tredie er det muligt at individuel brug af et edb-system er af væsentlig betydning for et samarbejde. Et centralt eksempel er udviklingen af nye edb-systemer i samarbejde mellem brugere og systemudviklere. Systemudviklere kan bruge individuelle værktøjer til at konstruere prototyper, der bruges i en fælles udviklingsproces.

Afhandlingen består af en sammenfattende artikel og fem selvstændige artikler eller delafhandlinger. En af disse artikler, "Object oriented programming and computerised shared material", diskuterer mere detaljeret betydningen af fælles materiale, og hvorledes fælles materiale kan realiseres. Der argumenteres for hensigtmæssigheden af objekt-orienteret programmering, idet der i denne form for programmering lægges vægt på at modellere fænomenerne i edb-systemets genstandsområde. Der argumenteres dog for, at de fleste objekt-orienterede sprog er utilstrækkelige, idet de ikke understøtter persistente objekter eller objekter, der kan deles mellem flere brugere.

I en anden artikel, "Transaction supporting systems and organisational change", argumenteres der for, at den såkaldte "transaction cost teori" er velegnet til at karakterisere samarbejde. I denne teori laves der en distinktion mellem tre hovedformer for organisation: marked, bureaukrati eller hierarki, og gruppe eller klan. En typisk organisation vil ofte bestå af en blanding af disse organisationsformer. Der hævdes i artiklen, at alle tre former for organisation kan understøttes med edb. Der sættes et lighedstegn mellem edb-støttet samarbejde og edb-støtte til gruppe eller klan organisering. Der vil ofte være et konfliktfyldt forhold mellem de forskellige organisationsformer, for eksempel mellem et hierarki, der vil bevare kontrol, og grupper, der ønsker at styrke deres autonome arbejde. Edb-teknologi, der er egnet til at undestøtte samarbejde, kan derfor føre til, at ledelsen i en organisation udsættes for et dilemma: den nye teknologi er effektivitetsfremmende, men den kan ikke udnyttes i en hierarkisk organisation. Valget står med andre ord mellem effektivitet og kontrol. Set med et demokratisk syn på brugen af teknologi er dette en glædelig udvikling.

En tredie delafhandling er en rapport om programmeringsomgivelser og systemudviklingsomgivelser, i rapporten omtalt med det fælles begreb udviklingsomgivelser. I rapporten argumenteres der for, at resultater fra edb-støttet samarbejde bør indgå i overvejelserne, når der laves udviklingsomgivelser. Det viser sig iøvrigt, at relativt tekniske egenskaber ved en udviklingsomgivelse kan have betyding for samarbejdet. Et eksempel er den nedbrydning af skellet mellem programmører og planlæggere, der kan komme som et resultat af forbedret understøttelse af eksperimentel systemudvikling.

Edb-støttet samarbejde nødvendiggør en teoretisk forståelse af den virkning, brugen af forskellige edb-systemer har på brugernes arbejde. Siden organisationer er forskellige, er det nødvendigt at basere udviklingen af edb-systemer, der skal understøtte samarbejde, på grundige analyser af den pågældende organisation. Det er desværre ikke sådan, at god analyse garanterer et godt design. Design kræver eksperimenter, til og med leg, for at stimulere skabelsen af nye ideer. Det er derfor væsentligt at spørge om, hvorledes vi kan eksperimentere med edb-støttet samarbejde. Hvorledes kan vi ud fra et kortvarigt eksperiment vurdere effekten på samarbejdet mellem brugerne? Vi kan få en vis hjælp fra udviklingsomgivelser, der understøtter trinvis udvikling, men vi kan ikke afskaffe modsætningen mellem de analystiske og eksperimenterende tilgangsvinkler til systemudvikling.

Denne afhandling er indleveret ved Datalogisk Afdeling, Aarhus Universitet, med tanke på erhvervelse af den naturvidenskabelige licentiatgrad.

Preface

This thesis is about computer supported cooperative work. It is the result of a three year long Ph. D. study. Besides this overview paper, the thesis consists of five other papers. This preface should be read as a preface to the whole thesis.

After I finished my master's degree in May 1985 I spent the rest of 1985 working in the MARS project [22]. The MARS project was a research project on system development, doing empirical research on system development work. The MARS project had been running since 1983, and I had also participated in the project as a student. The work in the project was done in four Danish companies, in groups consisting of practitioners from the companies and researchers from the university. The empirical work, which was documented in numerous reports, was completed in the summer 1985. The second half of 1985 the seven-person project group spent writing a textbook on system development entitled "Professionel Systemudvikling" (Professional Systems Development) [1].

In the beginning of 1986 I went to Linköping University, Sweden, to work on "system development of expert systems". The idea was that more could be learned about system development when focussing on the development of specific kinds of applications. It was of special interest to me to investigate how our understanding of system development was dependent on specific technical conditions. It turned out to be very hard to find any expert system in real use, and it was equally hard to find projects aimed at the development of such a system. The time spent in Linköping was not wasted. I got acquainted with different programming environments. I also made an observation which has been of great importance to me in the work on computer supported cooperative work: when working with new kinds of computer systems it is necessary to be convinced about the *relevance* of this work. To a researcher in artificial intelligence it is relevant to work with expert systems since this is one of the most promising applications of artificial intelligence techniques. To a researcher in system development, however, it is necessary to have a *vision* about work situations where the system can be used. The classical vision of an expert system as a machine replacing an expert has effectively been eliminated by the works of Winograd and Flores [45] and Dreyfus and Dreyfus [12]. I was not able to find research in expert systems which was committed to develop more realistic visions of where the technology could be used. Instead I observed a persistent stream of new projects developing prototypes of systems for which it was hard to see any future use. I therefore wrote a paper on the evaluation of prototypes of expert systems [35].

I moved back to Århus in the autumn 1986. I was ripe for going into a new field which was based on visions of new uses of computers. At this point in time I was given the opportunity to attend the first conference on computer supported cooperative work in Austin, Texas, see [9]. This turned out to be very decisive for the topic of my thesis. At the conference many interesting systems, and some empirical investigations, were presented. There was, however, a clear lack of a shared understanding of the field. This could, for example, be seen in the great variety of systems presented. It was not always clear how the systems would support cooperative work.

Having found the topic of my thesis I started to work with the topic in various ways. I felt that there was a need to state more clearly what we mean by cooperative work and computer supported cooperative work. To get a clearer understanding I arranged a study group on computer supported cooperative work at the Computer Science Department, Aarhus University. In the study group we read papers from the first conference on computer supported cooperative work and also some literature on organisation theory. The first paper in the thesis, "A cooperative work perspective on use and development of computer artifacts", was written based on the discussions in this study group.

As mentioned above I have earlier participated in the MARS project. During my work with this thesis I have participated in two other projects at Aarhus University. The first is the research programme on computer support on cooperative design and communication [3]. The second is the Mjølner project [11], a Nordic project working on the development of an integrated, interactive, and incremental programming environment for object oriented programming. Since I focus on computer support for system development the development of a programming environment is interesting in itself. In addition to this the Mjølner project has served as a real case of cooperative work.

Based on our work in the Mjølner project Elmer Sandvad and I arranged a study group on programming environments and system development environments. The study group served as inspiration for another part of this thesis, the report "Programming environments and system development environments".

In his thesis Kim Halskov Madsen identifies three disciplines within the Århus-Oslo systems school [29]:

- technology assessment, with the DUE project as prototypical example [27],
- system development, with the MARS project as prototypical example [1, 22],
- design, with the UTOPIA project as prototypical example [5, 44].

This thesis falls within the system development discipline. There is an important difference, however, between these research projects and the work in this thesis. Most of the research in the Århus-Oslo systems school has been very empirically oriented whereas this thesis is not. The reason for this choice is explained in section 2 of this paper.

This work has been financially supported by the Computer Science Department, Aarhus University, by the Department of Computer and Information Science, Linköping University, by the Danish Natural Science Research Council, and by mag. scient., observator Julie Marie Vinter Hansens Rejselegat. I am very grateful to Aarhus University for providing a Norwegian with a scholarship for Ph. D. studies. True Nordic cooperation!

Many people have helped me with constructive comments on various parts of this thesis. I wish to use this opportunity to thank Liam Bannon, Claudio Ciborra, Joan Greenbaum, Kaj Grønbæk, Riitta Hellman, Jørgen Lindskov Knudsen, Morten Kyng, Ole Lehrman Madsen, Markku Nurminen, Peder Christian Nørgaard, Claus H. Pedersen, Lars Bak Petersen, and Elmer Sandvad. Lars Mathiassen and Sture Hägglund have been my advisers in the beginning of my Ph. D. studies. Mogens Nielsen joined Morten Kyng and Ole Lehrmann Madsen in my "licentiate committee" at Aarhus University.

This thesis has been submitted to the Computer Science Department, Aarhus University, in order to fulfill the requirements of the lic. scient. (Ph. D.) degree.

Pål Sørgaard

Århus, May, 1988

1 Introduction

Computer supported cooperative work is a new research field which has emerged during the last couple of years. Cooperative work is nothing new. What is new is that much more attention now is paid to how cooperative work can be supported by the use of computers. This shift of attention is based on several factors. One factor is the increasing awareness that the use of computers has a strong impact on human work and organisation, and that there is a misfit between classical information systems and the way work actually is performed. A second factor is the development of new kinds of multi-user computer systems, systems which directly aim at supporting work in small groups. A third factor might be that cooperative work is very much in line with new management strategies [18]. There is, however, no general agreement about the nature of cooperative work nor of computer supported cooperative work. In this thesis an attempt is made to give these concepts a more precise meaning.

In this thesis computer supported cooperative work is seen as a research field which tries to deal *constructively* with the impact of the use of computers. The starting point here is a vision of a new way of using computers, namely the use of computers in a way which supports human cooperation.

This thesis deals with computer supported cooperative work from the viewpoint of system development. Classical system development methods and practices have a too narrow perspective on work and organisation. This often leads to the ignorance of cooperative work. This thesis is far from proposing a new and better method which takes cooperative work into consideration. It is naive to believe that the solution to problems in system development is new and better methods [1]. In addition the results presented here are not yet ready to be turned into clear cut practical advice for system developers.

There is also a second kind of focus on system development in this thesis: computer support for system development is seen as a case of computer supported cooperative work. This case has been chosen for two reasons. Firstly there is much cooperative work in system development. System development is often performed in projects, a kind of organisation which is close to the definition of cooperative work used in this thesis. Secondly work in computer supported cooperative work needs to be rooted in an understanding of the work which is going to be supported. My educational background and my experiences from the MARS project [1, 22] has made system development an obvious case.

The thesis has been written as six separate papers or reports, where this paper is the overview paper. The main purpose with this paper is to provide the necessary glue needed to bind the other parts together as one coherent product. The other products are:

A: A cooperative work perspective on use and development of computer artifacts [34].

B: A framework for computer supported cooperative work [36].

C: Transaction supporting systems and organisational change [42].

D: Object oriented programming and computerised shared material [39].

E: Programming environments and system development environments [40].

This paper proceeds as follows: Section 2 provides a short description of the research process behind this thesis. In section 3 the papers of the thesis are summarised and discussed. In section 4 my work on design and development of systems aiming at the support of cooperative work will be discussed. Finally this paper, and the whole thesis, is concluded in section 5.

2 On research method

This section describes the research process which has led to this thesis. As in much research the topic was not clear at the outset. As mentioned earlier the topic came in terms of a conference. There was a fit, however, between this topic and my background in system development, with the critique of classical uses of computers, and with my personal frustration over the lacking usefulness of expert systems which I originally had set out to study [35].

The topic of computer supported cooperative work can be phrased as a vision of a new way of using computers, use of computers which in a better way support the cooperative aspects of human work. How does one set out to study such a topic? There are numerous possibilities like case studies of the impact of the use of computers on cooperative work, implementation and evaluation of experimental systems, and theoretical investigations of the nature of and possibilities for computer supported cooperative work. This thesis is the result of a mainly *theoretical* work on computer supported cooperative work. This approach was chosen for several reasons. (1) The concept of cooperative work was too vague to serve as a basis for empirical or experimental work unless an effort was made to clarify the concept. (2) The time and resources needed to make an empirical study were beyond the scope of this, largely individual, Ph. D. study. (3) Implementation and evaluation of experimental systems would also go beyond the scope of this Ph. D. study. It is my hope that the results of the theoretical work will be useful in the design of empirical and experimental research on computer supported cooperative work.

One theoretical approach taken in this thesis has been to use organisation theory, especially the transaction cost theory, to characterise cooperative work and to discuss the relationship between cooperative work and the surrounding organisation. This work is reflected in papers A and C. During the work with the thesis a framework for computer supported cooperative work has emerged. This framework is described in paper B.

A second theoretical approach has been to discuss how computers can support cooperative work. This has led to the formulation of the idea of computerised shared material as a supplement to the well established area of computer supported communication. The idea of shared material is presented in papers A and B, and in paper D the concept is treated in more detail.

A third theoretical approach is the combination of topics from the MARS project [1] and the Mjølner project [11] with my own work on computer supported cooperative work. This work has resulted in a report on programming environments and system development environments, report E.

The theoretical approach has not been the only approach pursued in this research process. There has been some work on experimental design and implementation of systems relevant for cooperative work, see section 4. A major part of my background is the empirical research in the MARS project. During the work with this thesis I have participated in the Mjølner project. I have not made organised empirical observations of cooperative work in the Mjølner project, but the project has served as a source of inspiration as a "live" example of cooperative work. It has never been the plan that my participation in the Mjølner project should be used to produce empirical reports on computer supported cooperative work. The project has, however, had a clear influence on the development of my understanding of cooperative work. One finding has been the demonstration of the enormous variety of communication situations in a project. This has served to emphasise that electronic mail never can become the central mode of communication in cooperative work. Electronic mail may, however, be a very useful supplement to numerous other kinds of communication. A second finding is that real work is a mixture of various kinds of work. We will never find "pure" cooperative work, but cooperative work can be present at various degrees. This stresses the multi-faceted nature of most organisations. A third observation is that people may very well use computers to support their cooperative work although no explicit attempt has been made to provide computer support for the cooperative work.

In a discussion of research on system development Munk-Madsen claims that increased relevance of the results will be at the expense of the correctness and vice versa [31]. He further argues that correctness can be increased by research practices where the choice of topic is well delimited, where the investigations are highly structured, where laboratory experiments are used, and where the research process itself is carefully documented. When I started the work with this thesis the state of research in computer supported cooperative work was not at a level which made it feasible to go for structured research. The discussion of the research method can therefore be rephrased by saying that this thesis emphasises the relevance of the results at the possible expense of their correctness. This choice has to some extent been made by the author, but it has also been a consequence of the nature of the research topic.

This thesis is of a multi-disciplinary nature. This is a consequence of the nature of the problems addressed in the thesis. The results reached may contribute to the formulation of new research issues, some of which may be within classical computer science, other issues may be entirely within the social sciences. It is therefore unfortunate that informatics or computer science traditionally has been regarded as a mathematical subdiscipline. It is in the way the computer is or can be used that research in informatics needs to be rooted. Sometimes this connection will be very close, sometimes it may be rather distant, but it should always be there!

The discussion of the multi-disciplinary nature of this thesis is an expression of an ongoing paradigm shift [26] in informatics. Historically the computer has been seen as a machine which performs computations, the new paradigm sees the computer as a part of human work and its organisation. Christiane Floyd describes this paradigm shift in more detail [17]. She calls the new paradigm the process-oriented perspective and the old paradigm the product-oriented perspective. She argues that the process-oriented perspective does not exclude the product-oriented perspective, instead, she claims, the product-oriented perspective is a part of the process-oriented perspective. This thesis falls within the process-oriented perspective.

3 Description of the papers

In this section I will summarise and discuss the results from the papers and reports constituting this thesis.

A cooperative work perspective on use and development of computer artifacts (A)

This paper, which was written almost a year before the other papers, is an overview of and introduction to the field of computer supported cooperative work. The paper uses organisation theory as the primary source of inspiration in a discussion of the nature of cooperative work. Cooperative work is described as a specific, prototypical, kind of work. A number of characteristics of cooperative work are stated:

- People work together due to the nature of the task,
- they share goals and do not compete,
- the work is done in an informal, normally flat organisation, and
- the work is relatively autonomous.

Here the word organisation refers to how the work is organised in the work unit where the work takes place. It does not refer to the nature of the surrounding organisation. A cooperative work unit may be embedded in a strictly bureaucratic or hierarchical, and non-cooperative, organisation. Cooperative work is therefore one aspect of an organisation. This is not stated to make the definition less stringent, but in order to take account of the multi-faceted nature of most organisations.

Two ways of coordinating human work are identified in this paper: by explicit communication and through the material manipulated in the work. This leads to the introduction of the concept of computerised shared material.[2] The concept is based on the fundamental observation that when work is computerised it is often so that the tools *and* the materials are replaced. Since much work-coordination is mediated through the shared material we should try to give computerised shared material properties that can support this coordination. In one sense shared material is a design metaphor, which can be used as proposed by Madsen [28, 29]. In another sense it is a concrete property of the computer system to be built. People share data, and this has to be reflected in the way file systems and networks are constructed.

In the field of computer supported cooperative work there is no general agreement about what kind of work one wants to support. The narrow definition provided in paper A is an attempt to create debate and to establish a clearer starting point for discussions on computer supported cooperative work. Traditionally computers have been used to support bureaucratic organisations. The definition of cooperative work stresses that computer supported cooperative work is different from the classical applications of computers. The definition can also be used to point at the insufficiencies of these classical applications. A narrow definition allows us to make more specific assumptions about the environment where the computer systems are to be used. This may lead to the identification of new possibilities.

There is another reason to provide a stringent definition of cooperative work. In order to be able to find and state results beyond very general statements about the impact of the use of computers on human work and organisation, we need to be more precise about what we are talking about. There are essentially two ways of narrowing down the scope of computer supported cooperative work. One way is to give a definition of cooperative work and then look at how this kind of work can be supported by the use of computers. This is the approach taken here. The second approach is to restrict the attention to a specific class of computer systems, i.e. to give the research field a technical definition.

[2] The concept was coined in a discussion with Morten Kyng about the UTOPIA project [5, 44]

If we fail to follow one of these approaches we have just created a new name for research on use and development of computer systems. The choice of an organisational rather than a technical restriction of the topic of computer supported cooperative work is motivated by the numerous observations of technically well working systems which fail to fit into the organisational context. Put in other words: a fancy multi-user system is not interesting in itself.

The distinction between coordination by explicit communication and through shared materials is important in the development of computer support for cooperative work. The technical properties of computerised shared material and of computer-based media will often be very different. It needs to be mentioned, however, that on an analytical level this distinction is not always clear. In a linguistic perspective [2] use of shared materials can be interpreted as communicative actions.

A framework for computer supported cooperative work (B)

This paper is an attempt to provide an overall framework for computer supported cooperative work. A distinction is made between *the context of the work, the use situation* (which is a part of the context), and *the computer system.* The context of the work refers to the totality of the work, its organisation, the kind of work, etc. Cooperative work is a characteristic of the context of the work. The use situation is more delimited, it is the situation(s) where the computer system is actually used.

The *use relation* is defined to be the relation between the context of the work, including the use situation, and the computer system. Computer supported cooperative work is defined as a quality of this relation: The quality is obtained when the computer system is used in such a way that the cooperative aspects of the work are supported. This definition denies that computer supported cooperative work should be seen as a property of the context of the work or of the computer system alone. Computer supported cooperative work is, in other words, not only a question of modern multi-user systems, it is all kinds of use of computers which support the cooperative aspect of the work. The distinction between the context of the work and the use situation makes it possible to express this. We may even talk about computer supported cooperative work in cases where the use situation is individual.

Three *use relations* of relevance to computer supported cooperative work are identified. These are the medium, the shared material, and the tool use relations. I will give examples of these use relations below.

Electronic mail systems may be used as a *medium* supporting cooperative work. Electronic mail is essentially based on controlled copying of files, sometimes across networks, from one user to another. It is normally used asynchronously by the different users, although some facilities, for example notification about new mail, may allow almost synchronous use. Electronic mail can be used in many kinds of work. Its applicability in cooperative work has several reasons. It supports horizontal communication. It supports a style of communication which has the informal character of spoken language [14] while at the same time being asynchronous. It is much faster than conventional mail and internal mail. Wynn stresses the importance of the ongoing conversation in office work [46]. The properties of electronic mail make it a good *supplement* to the the other kinds of communication being used in this ongoing conversation.

A database with datasharing can sometimes be used as a *shared material.* Take an office where customer records are kept in a database. This database is normally used concurrently and asynchronously by the group of clerks dealing with the customers. Sometimes two clerks try to access the same record. The system ought to make this situation visible, just as it would have been if the records were kept in a manual file. The collision should be detected and it should be left to the clerks to decide how future work should be coordinated. To make collision detection visible in this way may be feasible, provided the number of users is low. The definition of cooperative work in this thesis implies that there are limits to the number of persons forming a cooperative group.

A drawing program, used as a *tool,* can also be used to support cooperative work. We can think of a group working on an illustration, for example a drawing of a house. One participant knows the drawing program well enough to use it as a tool. This supports the cooperative drawing process by making it easier to produce alternatives and to make corrections. In this way a group can improve its cooperation with the help of a tool used by one person in the group. A development process where users and system developers cooperate could in a similar way be supported by a prototype generator [6]. In paper A computer supported cooperative work was restricted to communication systems and systems providing shared material. The inclusion of tool as a use relation with relevance to cooperative work reflects a development in my own understanding of computer supported cooperative work. This development has moved away from a technical understanding towards a view where any use of computers which in some sense supports the cooperative aspects of work is seen as computer supported cooperative work.

Paper B also contains lists of characteristics of the context of the work, of the use situation, and of the computer system with relevance to computer supported cooperative work.

It is a central, although perhaps simple, statement that computer supported cooperative work is a quality of the use relation. As more experience is obtained from attempts to support various cooperative characteristics of work, I believe it will be possible to be more concrete about the qualities of the use relation.

Transaction supporting systems and organisational change (C)

In this paper the transaction cost theory, with its concept of group or clan, is used to characterise cooperative work. With this characterisation "cscw"-applications can be seen as a special class of transaction supporting systems, namely as those systems which support transactions in a group.

This paper has two purposes. One is to investigate the applicability of the transaction cost theory to the field of computer supported cooperative work. The other purpose is to demonstrate that computer supported cooperative work is a field which should be taken into consideration when we discuss the impact of the use of computers on organisations.

In the literature it is discussed how the use of computers, i.e. transaction supporting systems, will impact the relative benefits of the different types of organisation. Many, for example Malone et al. [30], claim that we will observe a shift towards relatively more market, and less bureaucratic, organisation. In paper C the characterisation of "cscw"-applications as a special kind of transaction supporting systems is used as a starting point to enter this debate. This class of applications can only be fully exploited by a group, i.e. in cooperative work, and they will therefore tend to make complex organisation, with a high degree of mutual trust, more favourable. Other transaction supporting systems will, of course, tend to favour market organisation. Therefore any prediction about the impact of information technology on the relative benefits of different organisations is dependent on the class of computer systems taken into consideration. The conclusion is therefore that such predictions cannot be made.

The transaction cost theory is useful by its description of the group or clan as a kind of organisation with little or no opportunistic behaviour and with a high ability to handle uncertainty [32]. It is also useful in providing a framework for analysing the impact of computers on organisations. The framework is a kind of organisational Darwinism: The kind of organisation best suited to the task will "win". If used in a prescriptive way the theory may lead to an overly economistic perspective. In paper C, however, the transaction cost theory is used in a descriptive way. Care should always be taken when a useful descriptive theory is used in a prescriptive way.

Group organisation will typically coexist with other kinds of organisation. This relationship may be harmonious, but there may also be conflicts between management, which wants to retain control in a hierarchical organisation, and the groups which want to be autonomous. In this perspective computer supported cooperative work is a *subversive* technology undermining management's control of the organisation. Since the technology also may result in important improvements of productivity, for example by reducing communication costs [33], this may represent a contradiction for classical, hierarchical management: keep control or improve productivity? This illustrates that a stricter definition of cooperative work and the application of the transaction cost theory make the assertions about computer supported cooperative work stronger, more interesting, and also more controversial. This may cure some of the unease about the field expressed by Howard [20] and Bannon et al. [4].

Object oriented programming and computerised shared material (D)

This paper elaborates on the concept of shared material. Some arguments specific to cooperative work and also some general arguments in favour of focussing on shared material are presented. It is claimed that one should model material, not work procedures, when systems are built.

The main purpose of this paper is to demonstrate the power of the shared material concept. A second purpose is to argue for the usefulness of object oriented programming beyond issues like reuse of code.

Stretching the argument as far as possible we can claim that object oriented programming may be an important contribution in making better computer systems. The paper also contributes to a demonstration of the relationship between the different activities and lines of interest that have been pursued in the work with this thesis.

It is a fundamental idea in object oriented programming to model the phenomena in the part of the world addressed by the system. These modelling techniques can be used to implement shared material on computers. The result is a raw system providing the material and the essential primitive operations on this material. This system can be seen as a specialised programming environment which can be tailored to the needs of individual users or be modified for future needs.

The implementation of such systems requires support for *persistent objects,* objects which live across individual program executions. These objects, which implement the shared material, will also be *shared* between several users and program executions.

In some languages, for example Beta [24] and Simula [10], objects can be active processes, not only passive instances of abstract datatypes. When such objects are persistent the object base will turn into a permanently executing parallel system. Any change to this system will rely on techniques for incremental change of executing programs. In general this approach calls for a much closer integration of programming languages and databases than what we have today.

In paper D shared material is used both as a design metaphor and as a very concrete property of the actual computer systems. We must be aware that there is a difference between the shared material we talk about metaphorically and the shared material in the system. The latter is produced by a description (a program), and can *never* reflect the totality of the shared material we talked about metaphorically. This means that the use of a shared material metaphor does not solve all problems. An example given in the paper shows that the technology, in this case the graphical capabilities of the output media, results in a change in the selection of which properties of the phenomenon should be modelled. Thus, what we get out of using the shared material metaphor, will depend on the kind of technology we have in mind and on our own technological fantasy. This observation can, of course, be seen as a corollary of the general statement that good system development requires profound knowledge of the application area *and* of computer technology.

Programming environments and system development environments (E)

This report presents five dimensions for the characterisation of development environments:

1. the system development functions supported,
2. the nature of the work (e.g., cooperative work) the environment is suited to support,
3. the suitability of the environment for prototyping,
4. the application area of the systems produced by the environment,
5. the technical characteristics of the environment.

These five dimensions are used to identify a number of schools or research traditions within the field. It is interesting to note that research on programming environments often has an implicit focus on individual work, whereas the software engineering environment school has a strong focus on large, formally organised projects. It is claimed that computer supported cooperative work is an important, and overlooked, issue in the field of computer support for system development. Another result is that new results from research in programming environments, especially integration, tailorability and incrementality, may lead to changes in the role of programming in system development.

One of the aims with report E is to shed some light on the field of environments for system development by drawing on theory from the MARS project [1, 22], and by applying theory from the field of computer supported cooperative work. There is, however, some feedback the other way too. System development is a good case for cooperative work, and some of the results obtained when looking at computer support for system development may be of general interest to computer supported cooperative work. The changed role of programming in system development, or at least the prediction thereof, is such a result. The computer support may lead to a different way of performing the "same" work, and it will call for a different mix of competence. In this case programming competence will be important in the performance of almost all work functions. Even communication can be supported by programming, for example through the use of a tailorable mail system.

In the case of system development this is not a negative trend. It will contribute to the abolishment of some of the boundaries between analysts and programmers, hence it will make the work less specialised and less suited for hierarchical control by, for example, division into phases. This change of roles need not have the same "positive" effects in all kinds of work. The idea of conceiving a computer system as a (specialised) programming environment was mentioned in the discussion of paper D. This will clearly have impact on qualification requirements and on the ability of different users to remain in control of their own work. In an office where only a few, typically local management, have programming capabilities, such a system could severely strengthen the power of local management over the office workers. One conclusion to draw is that many results in computer supported cooperative work may be very specific to the professions and kinds of organisation where they originate. It also means that there are limits to what can be said about computer supported cooperative work on a theoretical level without stating in which concrete context the results apply.

4 Concrete designs

During my work with this thesis I have also spent some efforts on designing and developing systems aiming at the support of cooperative work, mainly as parts of the Mjølner environment. Unfortunately it has been necessary to give low priority to these efforts in order to get time to complete this thesis.

The main design proposal is Heimdal, a Mjølner mail handler.[3] The ideas in the design of Heimdal are described in a report written in 1986 [37]. The design of Heimdal takes its starting point in seeing system development as cooperative work. A mail system should therefore be an integrated part of an environment supporting system development. Ciborra and Lanzara have also, in a short case description, described the importance of electronic mail in system development [7].

Heimdal has two features which, I believe, are unique. Although most

[3] Heimdal is one of the Norse gods. He can see just as well at night as during daytime. He can hear grass grow, and he can also hear the wool growing on the back of the sheep. He is in possession of Gjallarhorn, a giant lur (a bronze wind instrument from the Viking age), and when he blows in Gjallarhorn it can be heard all over the world. In my opinion this god must be described as extremely communicative.

e-mail "letters" only address one topic [14], this is not always the case. The tendency to stick to one topic is perhaps caused by the simplicity of the filing capabilities of most mail handlers. Heimdal encourages the archiving of the same letter in several folders, of course without actually saving several copies. The foreseen use is that users will archive letters by the person(s) they are to or from and by the various topics they address.

In my own use of e-mail I have observed that I use electronic mail as one of several media, and that one single conversation can take place using a multitude of media. One of the purposes of a mail handler is to keep track of and document ongoing and completed conversations. This is the main purpose of Flores and Winograd's coordinator [8, 45]. A mail handler ought to be able to support these multi-media conversations. Heimdal will therefore support document types such as "note about letter" and "note about phone call" which the users can insert at appropriate points in the folders.

These two features are relatively simple to implement, and since they only affect the mail handler, they can be introduced independently of the underlying mail system and also independently of other users.

Two of the major components of the Mjølner environment are the program database [19] and the program fragment library [23, 25]. I was for a short period involved in the design of these components. Development environments need to support cooperative work. This does also apply to these components. Programmers share programs. One main purpose of a program base is to allow sharing of programs in a way which fits with the style of work. The program database should therefore support strict configuration control while at the same time allowing small groups to work on the same code. Katz et al. make a distinction between archived, semi-public, and private versions of programs [21]. The focus on material also gives some ideas about how the integration of different facilities, often called "tools", can be implemented in Mjølner. Integration of facilities should not take place by facility-facility communication or by specialised controllers, but rather by equipping the the object of the work, the material, in this case the programs or the representation thereof, with suitable mechanisms for the coordination of actions performed by various facilities. In a small working note it is described how this could be implemented in BETA [41]. The work with the program base has demonstrated that the shared material metaphor is a useful metaphor in the design of an environment for system development.

During the autumn of 1987 I also implemented a small internal information system for the research programme on cooperative design and communication [38]. The system was made as an experiment with the use of remote procedure calls, a useful technique when shared data is to be implemented in a network of workstations. The system has, however, hardly been used. This experience has served to stress that it is not enough to make a system which provides a technical solution to some problem. In this case there clearly was a mismatch between a not so strongly felt need to get such an information system and the effort needed to learn to use the proposed system. When this observation is made in a research group explicitly interested in computer supported cooperative work it emphasises that it will not be easy to get "real" users to work with prototypes of systems supporting cooperative work.

5 Concluding remarks

This final section will point at some general conclusions and some open questions. Special attention will be given to conclusions with relevance to system development. Some areas for further study will also be pointed at.

Computer supported cooperative work is, as a research field, a special instance of the interrelationship between work, its organisation, and technology. The fundamental goal is that the computer system should, in some sense, support the work, rather than automating and controlling the work. A general theory about how a computer system can support work would perhaps be useful, but the differences between professions, organisations, power relations, etc., make it hard to describe how work can be supported on a general level. Therefore we need to be more specific about what assumptions we make about the organisation and the work. This has been attempted in this thesis. In doing so a narrow selection of the kind of work to be supported has been made. The focus is on an aspect of organisations which traditionally has not been supported by computers, and where the technology now makes support possible. The cooperative aspect is only one of several aspects of organisations. We will therefore see conflicts about the use of the "cscw"-technology. Management will fear to loose control, users will want to retain or strengthen their autonomy. If this thesis can contribute to the development of a technology which makes this conflict more visible, and which makes the position of traditional management more difficult by providing efficient technology which cannot be combined with strict hierarchical control, I have reached one of my goals.

There is a need for empirical and technical research which is more specific to computer supported cooperative work. Empirical research on electronic mail can be used as an example. Much of the research in the field, for example by Feldman [16], Eveland and Bikson [15], and Sproull and Kiesler [43], has a very broad scope. In these investigations messages sent to large mail-groups dominate numerically. Feldman's primary concern is in fact messages between strangers. It is therefore hard to find central results pertaining to computer supported cooperative work in these investigations. What is needed is empirical research where use of electronic mail in a clearly cooperative work context is analysed. Such an investigation should address the pattern of cooperation, language use, etc., and must therefore be based on access to the contents of the electronic mail as well as *any other communication* from the same workplace. It is clearly hard to be allowed to make such investigations in a company. To make the problem worse, it would be desirable to do this kind of empirical research in several companies, for comparison purposes and in order to assess differences in the impact of different mail systems.

In a similar way the technically oriented research needs to be more focused. All sorts of technical facilities are now being wrapped up as relevant for cooperative work. It would be better to identify those areas where progress in computer supported cooperative work is dependent on new technical solutions. In this thesis the idea of shared material has been emphasised. The current technology is unsuited for the development of shared material. This is primarily because the expression mechanisms needed to construct shared material, especially those of object oriented programming and of some languages for parallel programming, are not yet suited to describe persistent or shared data. The development of an object oriented language which facilitates shared persistent objects is a critical issue. Much focus is now directed to the issue of persistent objects.[4] It is also in this area that I plan to do my future research.

One of the goals of my research has been to apply the general theory about system development to a more specific field. This is what was at-

[4] See, for example, the programme for the European Conference of Object Oriented Programming in Oslo, August 1988.

tempted with "system development of expert systems". The research in computer supported cooperative work can also be given this interpretation. Some conclusions of general interest to system development can be drawn on the basis of this work.

The critique of traditional system development methods can be extended by taking computer supported cooperative work into consideration. Most methods imply a focus on formal organisation and on traditional data processing. System development performed according to such methods will therefore hardly result in the invention or introduction of systems which may support cooperative work.

System development methods is one thing, system development practice is another. The goal should therefore not be to write a "cscw-cookbook". There is, however, a challenge in transferring the ideas about computer supported cooperative work from the laboratories into the repertoire of the system developers and into actual use.

In the MARS project we observed a need for an improved interplay between the different system development functions, and we stated the need for this interplay in a number of normative theses [1]. New development environments may support this interplay by making programming a natural part of most system development functions, hence making the traditional division of work between analysts and programmers less relevant.

The development of computer support for cooperative work calls for a theoretical understanding of the impact of various computer systems on cooperative work. Since the characteristics of the organisations in question are important, each development process needs to be based on a careful analysis of the user organisation. We do know, however, that careful analysis does not guarantee good design. Analysis and design need to interact. Design is creative and requires experimentation, even playing [13], in order to stimulate the generation of new ideas. We should ask ourselves, however, about how we can experiment with computer supported cooperative work. How can we evaluate the impact on cooperative work from short time experiments? A partial solution is to improve the possibilities for evolutionary system development, but we cannot escape the contradiction between the analytical and experimental approaches to system development.

References

[1] Niels Erik Andersen, Finn Kensing, Monika Lassen, Jette Lundin, Lars Mathiassen, Andreas Munk-Madsen, and Pål Sørgaard. *Professionel systemudvikling.* Teknisk Forlag, København, 1986.

[2] Peter Bøgh Andersen. Semiotics and informatics: computers as media. In Ingwersen, Rajtery, and Mark Pejtersen, editors, *Information technology and information use*, Taylor Graham, 1986.

[3] Peter Bøgh Andersen et al. *Research Programme on Computer Support in Cooperative Design and Communication.* IR 70, Computer Science Department, Aarhus University, Århus, 1987.

[4] Liam J. Bannon, Niels Bjørn-Andersen, and Benedicte Due-Thomsen. Computer support for cooperative work: an appraisal and critique. In H.-J. Bullinger et al., editor, *Information Technology for Organisational Systems*, pages 297–303, North-Holland, Amsterdam, 1988.

[5] Susanne Bødker, Pelle Ehn, John Kammersgaard, Morten Kyng, and Yngve Sundblad. A UTOPIAN experience: on design of powerful computer-based tools for skilled graphic workers. In Gro Bjerknes, Pelle Ehn, and Morten Kyng, editors, *Computers and Democracy - A Scandinavian Challenge*, pages 251–278, Avebury, Gower Publishing Company, Aldershot, England, 1987.

[6] Susanne Bødker, Pelle Ehn, Jørgen Lindskov Knudsen, Morten Kyng, and Kim Halskov Madsen. Computer support for cooperative design. In *Proceedings of the Conference on Computer-Supported Cooperative Work, September 26–29, 1988, Portland, Oregon*, pages 377–394, ACM, New York, 1988. ACM order number 612880.

[7] Claudio U. Ciborra and Giovan Francesco Lanzara. True stories and formative contexts in information systems development. In *Information Systems Development for Human Progress in Organizations*, Atlanta, May 1987.

[8] *The Coordinator Workgroup Productivity System, Workbook and Tutorial Guide.* San Fransico, 1986.

[9] *Conference on Computer-Supported Cooperative Work*, MCC Software Technology Program, Austin, Texas, December 1986. Proceedings.

[10] Ole-Johan Dahl, Bjørn Myhrhaug, and Kristen Nygaard. *SIMULA 67 Common Base Language.* Pub. S-2, Norwegian Computing Center, Oslo, 1967.

[11] Hans Petter Dahle, Mats Löfgren, Ole Lehrmann Madsen, and Boris Magnusson. The MJØLNER project. In *Software Tools: Improving Applications: Proceedings of the Conference held at Software Tools 87*, pages 81–87, Online Publications, London, 1987.

[12] Hubert L. Dreyfus and Stuart E. Dreyfus. *Mind over Machine, The Power of Human Intuition and Expertise in the Era of the Computer.* The Free Press, New York, 1986.

[13] Pelle Ehn. *Work-Oriented Design of Computer Artifacts.* Arbetslivscentrum, Stockholn, 1988. International distribution by Almqvist & Wiksell International.

[14] Kerstin Severinson Eklundh. *Dialogue Processes in Computer-Mediated Communication.* Volume 6 of *Linköping Studies in Arts and Science*, Liber, Malmö, 1986. Ph.D. thesis.

[15] John D. Eveland and Tora Bikson. Evolving electronic communication networks: an empirical assessment. *Office: Technology and People*, 3(2):103–128, August 1987.

[16] Martha S. Feldman. Electronic mail and weak ties in organizations. *Office: Technology and People*, 3(2):83–101, August 1987.

[17] Christiane Floyd. Outline of a paradigm change in software engineering. In Gro Bjerknes, Pelle Ehn, and Morten Kyng, editors, *Computers and Democracy - A Scandinavian Challenge*, pages 191–210, Avebury, Gower Publishing Company, Aldershot, England, 1987.

[18] Joan Greenbaum. In search of cooperation: an historical analysis of work organization and management strategies. In *Proceedings of the Conference on Computer-Supported Cooperative Work, September 26–29, 1988, Portland, Oregon*, pages 102–114, ACM, New York, 1988. ACM order number 612880.

[19] Anders Gustavsson and Mats Löfgren. *Yggdrasil, concepts and programming interface.* Mjølner report S-LTH-17.1, University of Lund/Lund Institute of Technology, August 1987.

[20] Robert Howard. Systems design and social responsibility: the political implications of "computer-supported cooperative work". A commentary. *Office: Technology and People*, 3(2):175–187, August 1987.

[21] R. H. Katz, M. Anwarrudin, and E. Chang. A version server for computer-aided design data. In *23rd Design Automation Conference*, pages 27–33, ACM/IEEE, 1986.

[22] Finn Kensing, Lars Mathiassen, and Andreas Munk-Madsen. *MARS A Research Project on Methods for Systems Development.* MARS-report 1, Computer Science Department, Aarhus University, Århus, July 1984.

[23] Bent Bruun Kristensen. *The Program Fragment Library: A Preliminary Specification.* Mjølner report DK-SYS-12.3, Sysware ApS, Aarhus University, Aalborg University Centre, November 1987.

[24] Bent Bruun Kristensen, Ole Lehrmann Madsen, Birger Møller-Pedersen, and Kristen Nygaard. The BETA programming language. In Bruce Shriver and Peter Wegner, editors, *Research Directions in Object-Oriented Programming*, pages 7–48, MIT Press, Cambridge, Massachusetts, 1987.

[25] Bent Bruun Kristensen, Ole Lehrmann Madsen, Birger Møller-Pedersen, and Kristen Nygaard. Syntax directed program modularization. In Pierpaolo Degano and Erik Sandewall, editors, *Integrated Interactive Computing Systems*, pages 207–219, North-Holland, Amsterdam, 1983.

[26] Thomas S. Kuhn. *The Structure of Scientific Revolutions.* The University of Chicago Press, Chicago, 1970. Second Enlarged Edition.

[27] Morten Kyng and Lars Mathiassen. System development and trade union activities. In Niels Bjørn-Andersen, editor, *Information Society, for Richer for Poorer*, North-Holland, Amsterdam, 1979.

[28] Kim Halskov Madsen. Breakthrough by breakdown. In Heinz K. Klein and Kuldeep Kumar, editors, *Proceedings of the IFIP WG8.2*

Working Conference on Information Systems Development for Human Progress in Organisation, Atlanta, 29–31 May 1987, North-Holland, Amsterdam, 1988 (forthcoming). Also available as PB 243, Computer Science Department, Aarhus University, Århus, March 1988.

[29] Kim Halskov Madsen. *Sprogbrug og Design — sammenfattende redegørelse.* PB 245, Computer Science Department, Aarhus University, Århus, November 1987.

[30] Thomas W. Malone, Joanne Yates, and Robert I. Benjamin. Eletronic markets and electronic hierarchies. *Communications of the ACM*, 30(6):484–497, June 1987.

[31] Andreas Munk-Madsen. *Viden om systemudvikling.* PB 213, Computer Science Department, Aarhus University, Århus, May 1986. Also MARS-report 13.

[32] William G. Ouchi. Markets, bureaucracies, and clans. *Administrative Science Quaterly*, 25:129–141, March 1980.

[33] Jacob Palme. *Experience with the use of the COM computerized conferencing system.* FOA Rapport C 10166E-M6(H9), Försvarets forskningsanstalt, Stockholm, December 1981.

[34] Pål Sørgaard. A cooperative work perspective on use and development of computer artifacts. In Pertti Järvinen, editor, *The Report of the 10th IRIS (Information Research seminar In Scandinavia) Seminar*, pages 719–734, University of Tampere, Tampere, 1987. Also available as PB 234, Computer Science Department, Aarhus University, Århus, November 1987.

[35] Pål Sørgaard. Evaluating expert system prototypes. In Hans-Erik Nissen and Gunhild Sandström, editors, *Quality of work versus quality of information systems. Report of the ninth Scandinavian research seminar on systemeering*, pages 187–201, University of Lund, Department of Information and Computer Sciences, Lund, December 1986.

[36] Pål Sørgaard. A framework for computer supported cooperative work. In Jens Kaasbøll, editor, *Report of the 11th IRIS (Information systems Research Seminar in Scandinavia), August 10–12, 1988,*

pages 620–640, Research Report no. 116, Department of Informatics, University of Oslo, 1988. Also available as PB 253, Computer Science Department, Aarhus University, Århus, May 1988.

[37] Pål Sørgaard. *HEIMDAL: a Mjølner mail handler.* Mjølner report DK-15.1, Computer Science Department, Aarhus University, November 1986.

[38] Pål Sørgaard. *A mutual directory for the coop programme.* Computer Science Department, Aarhus University, Århus, December 16 1987.

[39] Pål Sørgaard. Object oriented programming and computerised shared material. In Stein Gjessing and Kristen Nygaard, editors, *ECOOP '88 European Conference on Object-Oriented Programming, Oslo, Norway, August 1988, Proceedings*, pages 319–334, Lecture Notes in Computer Science 322, Springer Verlag, Heidelberg, 1988. Also available as PB 247, Computer Science Department, Aarhus University, Århus, May 1988.

[40] Pål Sørgaard. *Programming Environments and System Development Environments.* PB 252, Computer Science Department, Aarhus University, Århus, May 1988.

[41] Pål Sørgaard. The subscription model. April 1987. Informal Mjølner note.

[42] Pål Sørgaard. *Transaction supporting systems and organisational change.* PB 248, Computer Science Department, Aarhus University, Århus, May 1988.

[43] Lee Sproull and Sara Kiesler. Reducing social context cues: electronic mail in organizational communication. *Management Science*, 32(11):1492–1512, November 1986.

[44] The UTOPIA project group. *An Alternative in Text and Images.* GRAFITTI 7, Swedish Center for Working Life, Stockholm, 1985.

[45] Terry Winograd and Fernando Flores. *Understanding Computers and Cognition.* Ablex Publishing Corp., Norwood, New Jersey, 1986.

[46] Eleanor H. Wynn. *Office Coversation as an Information Medium.* PhD thesis, Department of Anthropology, University of California, Berkeley, May 1979.