ISSN 0105-8517

Through the Interface
– a Human Activity Approach to User Interface Design

Susanne Bødker

DAIMI PB – 224
April 1987

AARHUS UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
Ny Munkegade 116 – DK 8000 Aarhus C – DENMARK
Telephone: + 45 6 12 83 55 *Telex: 64767 aausci dk*

Contents

Dansk Resumé (Danish Summary)

Denne afhandling, hvis titel bedst kan oversættes til "Gennem Grænsefladen - en Virksomhedsteoretisk Forståelse af Brugergrænseflade-design", omhandler brugergrænseflader og deres design, set ud fra et perspektiv, som fokuserer på brug af edb-baserede artifakter eller redskaber. Dette gøres med udgangspunkt i virksomhedsteorien, som er en psykologisk/antropologisk forståelse af menneskeligt liv og arbejdsvirksomhed, og menneskets udvikling af, og brug af, redskaber i denne virksomhed.[1]

Hensigten med afhandlingen er at søge et bedre teoretisk fundament for forståelse af brugergrænseflader og deres design, end det som er baseret på den type af psykologisk teori, "cognitive science",[2] som i dag er fremherskende indenfor disciplinen. Derigennem er det også hensigten at kunne forklare forskellige aspekter af brugergrænseflader bedre, og at kunne give råd til designere af brugergrænseflader, så de kan designe bedre brugergrænseflader. Det er ligeledes hensigten med afhandlingen at nå frem til operationelle metoder, som kan anvendes i den omhandlede type af design. Sådanne metoder har i afhandlingen mest karakter af eksempler, idet en egentlig udvikling af et metode-sæt, baseret på teorien, ligger udenfor afhandlingens tids- og omfangsmæssige rammer. Det er altså hensigten med afhandlingen at udvikle *en ny datalogisk teoridannelse om brugergrænseflader og deres design.*

Afhandlingen henvender sig til studerende og forskere med interesse for dette felt, altså mennesker som læser fagtidsskrifter og deltager i konferencer om emnet.

Empirisk har afhandlingen sine rødder i den skandinaviske fagbevægelses-tradition,[3] og især i Utopia projektet,[4] som jeg deltog i. I dette projekt arbejdedes bl. a. med at designe brugergrænseflader sammen med brugere, og undervejs afprøvedes forskellige mere eller mindre etablerede arbejdsformer. Erfaringerne fra bl. a. dette arbejde viste, at de færreste af metoderne var egnede, når det drejede sig om at designe brugergrænseflader sammen med brugere.

Dette har ledt til, at jeg har søgt teoretiske forklaringer på, hvorfor det er nødvendigt at gå andre veje end anvendelsen af traditionelle beskrivelsesteknikker m. m., når man i design vil nå frem til en forståelse af kernen i brugernes forhold til det fremtidige redskab.

Den psykologisk/antropologiske teoridannelse, som jeg har valgt at anvende, kan tildels ses som et brud med såvel den fremherskende psykologiske skole indenfor brugergrænseflade-design som med den tradition, som har været fremherskende indenfor systemarbejde på Datalogisk Afdeling; nemlig et opgør med ideen om, at en god beskrivelse af menneskers såvel som maskiners "opførsel" er en forudsætning for godt design.

Afhandlingen placerer sig på den anden side i linie med nyere arbejder indenfor den fænomenologiske tradition, angående forståelsen af design af edb-anvendelser.[5] Disse arbejder slår fast, at det er brugernes daglige praksis og brug af edb-anvendelserne, som må være udgangspunktet for design, snarere end designernes tænkning om disse.

Det virksomhedsteoretiske udgangspunkt for at begribe redskabers rolle i brug kan kort opsummeres som følger: menneskeligt arbejde indgår på den ene side i en kollektiv virksomhed, hvor en gruppe af mennesker sammen udfører et arbejde med et vist formål eller rettet mod en vis genstand. Det enkelte menneskes individuelle virksomhed bidrager til gennemførelsen af den kollektive virksomhed. På den anden side består den individuelle virksomhed af en række handlinger, som udføres med en bestemt hensigt i relation til virksomhedens formål eller genstand. Disse handlinger realiseres gennem en række operationer. Operationer er karakteriserede ved, at de ikke udføres med en bevidst hensigt, men som et resultat af menneskets omgang med redskaber, genstande og andre mennesker. Disse operationer kommer til som handlinger, der gennem gentagen udøvelse glider fra at blive udført bevidst til at blive udført ubevidst, udløst ved mødet med bestemte materielle betingelser. Gennem konceptualisering f. eks. i sammenbrud kan mennesket igen blive bevidst om operationer. Sammenbrud optræder, når der opstår en uoverensstemmelse mellem operationerne og de materielle betingelser, som udløste dem. I arbejdsvirksomheden omgiver mennesket sig såvel med redskaber og genstande som med andre mennesker. Nogle af de handlinger og operationer, som mennesket betjener sig af, er rettet mod genstande og redskaber, de instrumentelle, mens andre er rettet mod andre mennesker, de kommunikative. En gruppe af mennesker, som sammen udfører en bestemt arbejdsvirksomhed, deler en praksis, som de samtidig er med til at opretholde og forandre. Praksis afspejles på den ene side i de redskaber, det sprog, den arbejdsorganisering og de normer som ligger til grund for arbejdet. På den anden side kommer den til udtryk i det enkelte menneskes repertoire af operationer, som anvendes i arbejdsvirksomheden.

Når vi går fra at tale om menneskelig virksomhed og brugen af redskaber i denne, til at tale om redskaberne, som de forudsætninger de er for virksomheden, og samtidig fokusere på edb-baserede redskaber, skifter vi fra et antropologisk/psykologisk til et datalogisk domæne. *Edb-baserede redskaber eller edb-anvendelser og den måde, hvorpå de fremtræder i brug, er i fokus når vi ønsker at analysere eller designe brugergrænseflader.*

Afhandlingens konklusioner om edb-anvendelsen, og hermed om brugergrænsefladens udformning, kan opsummeres som følger:

Edb-anvendelsen står imellem brugerne og den egentlige genstand for virksomheden; den er ikke selv genstand for brugerens handlinger under normal brug, d. v. s. at brugeren normalt kun retter operationer og ikke handlinger mod selve edb-anvendelsen.

Edb-anvendelsen kan mediere såvel menneskets omgang med genstande som med andre mennesker.

Menneske-maskin interaktion kan karakteriseres som menneskets operation af en edb-anvendelse. Brugergrænsefladen er de dele af software og hardware, som understøtter denne.

Brugergrænsefladen understøtter operationer mod redskabet som fysisk genstand, mod redskabet som redskab og mod den egentlige genstand eller det menneske, hvormed der kommunikeres gennem redskabet.

Konsekvenser for design/systemudvikling kan opsummeres som følger:

Brugergrænsefladen udfolder sig kun helt i brug, d. v. s. at det kun er gennem brug, at vi kan afteste alle brugergrænsefladens egenskaber, fordi det er gennem brug at disse egenskaber afsløres for brugerne.

Brugergrænseflader må primært udvikles gennem brug (og forskellige simulerede brugssituationer), ikke gennem beskrivelse.

I brug eller simuleret brug kan brugerne afprøve dels deres eksisterende repertoire af operationer, og dels mulighederne for at udvikle et nyt, fordi det er muligt at afprøve mulighederne for at rette handlinger mod den egentlige genstand eller den egentlige kommunikations-partner gennem edb-anvendelsen. I situationer, hvor dette ikke lykkes, kastes der, gennem sammenbrud, lys på brugergrænsefladens rolle; dens manglende evne til at understøtte handlinger mod den egentlige genstand gennem edb-anvendelsen. I sådanne situationer bliver edb-anvendelsen altså selv genstand for brugerens handlinger.

I design må man, udover brugerens repertoire af operationer, også tage hensyn til virksomhedens andre forudsætninger, på det individuelle såvel som det kollektive niveau. Brugernes praksis er udgangspunktet for design, ligesom praksis forandres gennem design.

Design(systemudviklings)-metoder må tage udgangspunkt i den konkrete brugsvirksomhed og praksis snarere end i abstrakte formalismer.

Indholdsmæssigt ligger hovedvægten i afhandlingen i kapitlerne 2, 4 og 5. I kapitel 2 udvikles afhandlingens grundlæggende begrebsramme med udgangspunkt i virksomhedsteorien. I kapitel 4 uddybes denne med henblik på forståelse specielt af brugergrænseflader, idet begrebsrammen anvendes til analyse af forskellige aspekter og udformninger af brugergrænseflader. I kapitel 5 sættes fokus på design af brugergrænseflader ved at begrebsrammen anvendes til analyse af konkrete designmetoder. Som et resultat heraf diskuteres mulighederne for at nå frem til bedre designmetoder, byggende på en virksomhedsteoretisk forståelse af brugergrænseflader og deres design.

Derudover består afhandlingen af en introduktion (kapitel 1), en præsentation af afhandlingens empiriske grundlag (kapitel 3), og et konklusionskapitel (kapitel 6), hvori afhandlingens resultater udmøntes i en række anbefalinger til brugergrænseflade-designeren.

Afhandlingen er indleveret til bedømmelse til den naturvidenskabelige licentiatgrad. Arbejdet er for den største dels vedkommende udført ved Datalogisk Afdeling, Aarhus Universitet under vejledning af Morten Kyng.

---

1 A. N. Leontjew: **Activity, Consciousness, and Personality**, Prentice-Hall 1978, A. N. Leontjew: **Problems of the Development of the Mind**, Progress Publishers, 1981 (Problemer i det psykiskes udvikling, København 1977), A. N. Leontjew: **The Problem of Activity in Psychology**, i J. V. Wertsch, ed.: **The Concept of Activity in Soviet Psychology**, M. E. Sharpe Inc. 1981. Danske fortolkninger af virksomhedsteorien kan findes i flere artikler i tidsskriftet **Psyke og Logos**, f. eks. B. Karpatschof: **Grænsen for automatisering**, Psyke og Logos 2, 1984.

2 Se for eksempel S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum 1983.

3 En oversigt og diskussion af den skandinaviske fagbevægelsestradition findes i P. Ehn & M. Kyng: **The Collective Resource Approach to Systems Design** i G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987.

4 En kortfattet oversigt over Utopia-projektet findes i S. Bødker et al.: **Graffiti 7. UTOPIA-projektet et alternativ i tekst og billeder**, Datalogisk Afdeling, Aarhus Universitet 1985 (findes også på engelsk som **Graffiti 7. The UTOPIA project. An Alternative in Text and Images**, Arbetslivcentrum 1985). En gennemgang og diskussion af projektet findes også i S. Bødker et al.: **A Utopian Experience**, i Bjerknes et al., op. cit. (note 3).

5 H. & S. Dreyfus: **Mind over Machine**, The Free Press 1986 og T. Winograd & C. F. Flores: **Understanding Computers and Cognition: A New Foundation for Design**, Ablex Publishing Comp. 1986.

Chapter 1

Introduction

> Design is where the action is in the user interface, not evaluation. (Allen Newell)[1]

This dissertation places itself in a field of tension between research traditions, which historically have their roots far from each other, in psychology, in computer science, and elsewhere, but which at this point of time are approaching the same issue: *Design of user interfaces.*

I have chosen the title: *Through the Interface – a Human Activity Approach to User Interface Design* to say that the dissertation deals with user interfaces and their design, from a theoretical perspective which focus on human work activity, and on use of computer applications in human work activity. 'Through the Interface' tells us that a computer application, from the user's perspective, is not something that the user operates *on* but something that the user operates *through* on other objects or subjects. In this dissertation, the user interface is seen as the parts of software and hardware which support this effect. When I use a text editor to write this chapter, the user interface supports my work on the form and content of the document, and if the user interface is a good one I am capable of forgetting that I actually work with a computer between the document and myself.

The traditions

I base this dissertation on the research tradition which has often been called the Aarhus-Oslo school. Its field has been systems development in its widest meaning: analysis and design of computer based systems and their surrounding organizations as well as the study of impacts of such systems on labor.[2] The background of this school has been a critical attitude towards traditional phase-oriented systems development methods,[3] which have in turn mainly been dealing with development of large batch-oriented computer systems[4] from a management perspective.

The character of computer applications is, however, changing these years. There is a shift from large mainframe computers to personal workstations, from data-entry and number-crunching to interactive, graphics oriented applications which are no longer only administrative. Technology provides new possibilities. The data/information processing paradigm which has been central in our conception of computer applications, breaks down in more and more situations, where we deal with new types of applications. Furthermore, users demand constructive influence in more and more cases; not just a veto against managements suggestions. Management, too, sees its reasons to involve users in design.[5]

These changes pose new challenges to our tradition: we need to deal with a new kind of technology: with new types of computer applications, with new aspects of the applications, and with new methods of design. One of these new areas of concern is design of user interfaces.

One tradition has so far been the leading, theoretically, in dealing with the user interface. This tradition is rooted in psychology: the tradition which is called human factors research or cognitive science, depending on the speaker's nationality and perspective.[6]

This tradition has been analyzing the users of different computer applications and their reactions towards different user interfaces, e.g. the difference between keyboard commands and menus for text editors. It has moved from a point where measurement of key-stroke speed and the like was the main issue, to more advanced analysis of use situations based on theory from cognitive psychology. Some of the promoters of this tradition[7] have come to the conclusion that the tradition is at present at a point, where a shift is needed from a quantitative analysis approach to a qualitative design approach: *design is where the action is in the user interface*. The cognitive science tradition will be analyzed in this dissertation in the quest for a renewed understanding of user interfaces and their design.

Perspective

My main concern is for computer support for purposeful human work, whereas I recede from the study of games, casual use of automated bank tellers, etc. The reasons for this choice can be found the history of how I came to write my dissertation, and in the theory that I have chosen to deal with. Based on empirical experiences and on the theoretical framework we must consider what an experienced user does in her daily work, very different from what a casual user of a bank teller does. These differences may give rise to very different kinds of user interfaces. As an example, let us consider one aspect of bank tellers: for the customer to use an automatic bank teller, robustness and security is important, and the designers may choose to have push-buttons for six or eight different amounts of money which the consumers can take out. This type of solution would, however, be much too unflexible for the professional banker. It is, however, no problem to train her, as a regular user, to handle decimal points and large amounts of codes.

When writing or reading a dissertation like this, we face the problem that we cannot learn what we do not already know. Writings are not representations or explanations of the world, but rather intended to trigger some awareness by the reader towards his or her own experiences. The challenge is for the writer to trigger the "right" awareness. What we practically understand or know is more important than, and precedes, a theoretical understanding, no matter what domain we talk about: research, design, or typography. This means that any kind of concepts arise from, and exist in close connection with the material world. In most cases, we cannot give complete definitions of concepts, but rather point at certain rules and prototypical cases. Through new theories and frameworks we can create new distinctions in our knowledge of, and actions in the material world; but only through the readers' own practical experiences can their value be tested.

It is a basic idea of this dissertation that participation is needed for epistemological reasons, to improve quality of the design process as well as of the computer application, especially when it comes to the user interface. I have chosen *not* to focus on the more political aspects of user participation in design, e.g. questions of resources and power. In line with my tradition and empirical background I shall, however, advocate a democratization of design. Participation in design is not democratic design, only a possible step towards it; trade union investigations, technology agreements and various negotiation systems are part of the mechanisms needed to move towards democracy.[8] In line with this we must see design as an activity which is a process of negotiation between different groups, with conflicting interests, and with different resources and power to pursue their goals.

The approach taken

In the dissertation I shall present an approach to understanding user interfaces and their design. The idea is to develop the human activity theory which deals with purposeful human work in two directions: one which focuses on computer applications and especially user interfaces, and one which deals with the design of such. What we achieve is not a new theory, but a refinement and enrichment of an existing one. But by shifting the focus from human activity to the computer application and its role, we move from a psychological domain to a computer science domain in which we make use of a psychological theory.

The approach presents a point of reference to which actual design processes can be compared. It creates possibilities of seeing some things at the same time as it creates blindness towards others.[9] In this way, the approach is, of course, like all approaches, normative.[10] The actual choice of using the term 'design' and not the more common term 'development' is an example of this: by using 'design' I want to stress the similarities of the computer expert's trade with other kinds of design, such as architecture, carrying norms such as quality just like these other professions do. At the same time, to choose not to use the term 'development' has meant less direct focus on the role that design of computer applications plays in the organizational development process.

Domain and purpose

In design, the user interface is one aspect of the future computer application; the user interface is an object for design. In use, however, the user interface determines how the computer application appears to its user. To understand the user interface we must study the relation between the human and the computer application. We call this relation **use**. To study this relation has been an issue for psychology, anthropology, and philosophy.

The cognitive science school has combined computer science with one school of cognitive psychology. This school believes that a good user interface represents/is based on a model of the user, and even that the user and the computer are structurally alike. Other, more technical definitions[11] see the user interface only from a programmer's perspective, as technical components.

In this dissertation, I shall explore the possibilities of *offering a new conception of user interfaces*. For this purpose a framework building upon the anthropological/psychological theory of A. N. Leontjew will be presented. I found that the work by Leontjew offered valuable explanations in this area, because the theory is quite operational and detailed in explaining what human beings do when they operate artifacts.[12]

Dealing with human work activity means dealing with what is specifically human, compared to objects or animals, especially the ability to design artifacts with the future use activity in mind. Animals may apply sticks and the like as artifacts with specific purposes in specific situations, but only human beings conduct design, imagining the future use of the artifact. Human beings do not conduct design as a well-planned process where we first determine the goals and then acts, but design as a process where we change our actions as we move along based on the meeting with the material world.

Design of computer applications, seen *not* as a rationalistic and well-planned process, but as a process characterized by arationality and action in situations, has been the issue for many authors recently. T. Winograd & F. Flores, among others,[13] argue that we need a different theoretical foundation for design. Their suggestions are all based on a combination of thoughts of several different philosophers. Based on my discussions of these theories with the authors and others,[14] the challenge arose to investigate how far I could get along the same lines by *trying out the Soviet psychology* as presented by A. N. Leontjew and his followers, at the same time as I direct special attention towards the user interface.

Because a practical understanding is needed to make use of our theoretical understanding, new suggestions for design methods, and for user interfaces must follow such new theoretical considerations. Furthermore, new design methods and new user interfaces must be rooted in the specific design situation, which in turn is directed towards the future use situations. For the dissertation this means that I cannot expect to give new general guidelines for user interfaces and their design, neither can I sit down at my desk and construct examples, without actually trying out the theory in practical design situations. It is not possible within the time range of my dissertation project to follow up the theory with practical experiments. Instead I shall discuss examples from my empirical background within the theoretical frames. Especially, it is a purpose of the dissertation to elaborate on the design ideas of the Utopia project, the so-called tools approach[15] – *how can we explain the tools approach and which methods can we develop as part of it?*

Through my choice of perspective I have delimited my object area: computer support for purposeful human work is my specific interest. My main goal is not to give a new method to cover user interface design in all kinds of cases but to demonstrate to my readers that user interface design under certain conditions can be conducted with success in a specific way. To offer an explanation to this, and a vocabulary. My readers can use these in their own conception to change their practice. If I am successful, they will understand something new about design, realize new needs, see new goals, and perhaps try out new design methods.

It is not my main purpose to discuss how such conditions can be achieved. For detailed discussions of the practical constraints of design, the reader is referred to the MARS project.[16] The political constraints are discussed by, e.g., P. Ehn & M. Kyng.[17] The dissertation argues that there are strong epistemological reasons for user participation in user interface design. For democratic reasons such participation must of course be followed by resources for unions to support the users.[18] I shall assume that the needed changes for the less powerful parties: the users and their organizations, can be brought about. How this can be done will only be touched upon marginally.

It is first of all my interest to reach the audience who otherwise reads journals and textbooks about human-computer interaction and design of user interfaces. I imagine that my reader is a researcher, teacher, or advanced student in this area, or perhaps a person responsible for introducing new ideas or design methods in an industrial organization. My direct aims have not been to write a textbook to be used as an introduction to the area.

My background

Looking at my own history, there are several paths leading to the subject of user interface design. As a student I favored two, at that time to me very distinct subjects: systems development, and computer graphics. My master thesis work[19] revealed to me, both practically and theoretically, that to write computer programs and to describe human work are two different things, and that formal descriptions are perhaps not the solution in the latter case.

However, I moved away to more practically oriented surroundings, Xerox PARC, where I worked with computer workstations and programming environments. Moreover, I started to investigate why programming environments had come into being, and how they were applied by the users in their design work.

After my return to Denmark, many of my former interests and experiences were united in the Utopia project in design with users of graphics oriented applications based on modern computer workstations. The last couple of years I have spent on theoretical reflections based on the Utopia project: about design together with some former project members,[20] and about human-computer interaction with others.[21] Furthermore, I have tried to transfer some of the design methods of the project to another domain: offices at Aarhus Polytechnics.

Empirical background

Empirically, the dissertation is based on the cases mentioned above, the Xerox PARC case, the Utopia project, and the Aarhus Polytechnics project. They represent design of user interfaces together with users. The design situations as such will be presented in details in Chapter 3 together with the background and setting of the projects. The Utopia project is my main empirical case. It was followed by a smaller project, the Aarhus Polytechnics project which was intended to try out some of the ideas of Utopia in a different domain, to supplement this case. Historically, my third case, the Xerox PARC case, is preceeding the two, and my work there did by no means aim directly at gaining experience with user interface design. I did, however, gain some experiences relevant for this dissertation.

The Utopia project

Utopia[22] was a Scandinavian research project on trade union based development of, and training in computer technology and work organization, especially text and image processing in the graphic industries.[23] The overall goal of the project was to contribute to the development of powerful skill-enhancing tools for graphic workers. I.e. the project stressed both the development of technology, human qualifications, and education. Quality of work and product was very important.

Graphic workers, and computer and social researchers worked together in the project, which was carried out at the Swedish Center for Working Life, Stockholm, the Royal Institute of Technology, Stockholm, Sweden, and the University of Aarhus. The project began in 1981, and went on for four years.[24]

The aims of the project were to change the trade union's range of possible actions[25] at the local level: Instead of defending the status quo, the idea was to develop an offensive strategy, providing and applying technological alternatives; technology to improve the quality of work and the products. Technology that is dynamically changeable at individual workplaces as the employees develop their competence.

The project also aimed at producing a "demonstration example" showing that trade union development of technology is a feasible strategy under certain favorable conditions. It was hoped that the project could inspire the development of strategies on technology policy in different application domains where, e.g., the economic, technical or trade union conditions are different.

In its first phase the Utopia project investigated existing technology, practice, and training in the graphic industries, as well as the prerequisites for developing alternatives. A major aspect in this first phase of the project was the mutual learning process in which the participants: graphic workers, and computer and social researchers, established a common "knowledge platform" for the future work.[26]

The Utopia project was approached by the publisher and computer supplier Liber, who wished to cooperate around the company's development project TIPS (Text and Image Processing System). This way the project came to focus on page make-up and image processing for newspapers, and the next year it concentrated on requirement specification. This called for the development of design methods for researchers and graphic workers to formulate the requirements together. The project established a "technology laboratory"[27] with development tools to simulate different kinds of page make-up, image processing, and the surrounding organization; thus making it possible for the graphic workers in the project to develop requirements and wishes on a concrete level by actually carrying out the page make-up and image processing on simulation equipment. In this laboratory, part of the work aimed at studying and developing user interfaces. This work will be in focus for my discussion of the project in the following chapters.

The next step was professional education. Of the more than twenty reports produced in the project, the majority were written for professional education of graphic workers.

The cooperation with Liber/TIPS also included an evaluation of the TIPS system and development of work organization in connection with the first pilot installation. Here the original intentions proved difficult to realize. Due to various conflicts between the involved parties, the ideas of the Utopia project for active participation in an organizational experiment where graphic workers and journalists together could seek new ways could not be realized. The project has instead followed and evaluated how the technology is used at the pilot plant.[28]

The Aarhus Polytechnics project

The Aarhus School of Polytechnics is the public school for crafts in the Aarhus area. The school is responsible for education and training in such areas as plumbing, metal, carpentry, printing, hairdressing and many others. It has a large administration distributed at a number of different locations in the area. The administration takes care of budgets and other financial issues, management of buildings and other facilities, including construction work, registration of students, salaries and other staff administration, supplies, secretarial work, etc. Many of these functions are partly located centrally and partly locally. In this administration a large office automation project is carried out, the purpose of which can be described as follows: The project should create an integrated office automation system, which allowed for a more efficient administration of the school. The office administration system should be financed not by laying off employees, but by allowing more efficient use of such resources as classrooms and heating.

The project was initiated by management of the school. According to the technology agreement, the project is managed by a technology committee with representation of management and employees. The project has been going on since 1983. It is the general idea of the project that the employees should, in project groups, take part in designing the computer applications that they are to use themselves. The school hired a number of consultants to work with the employees in the design work. The actual realization of the computer applications was to be carried out by a computer manufacturer on the basis of the specifications and prototypes created by the users and consultants in cooperation. This case deals with one of these project groups, the journal. The purpose of the group was to find out how the journal of the school could be reorganized to be more efficient, eventually by means of a computer application.[29]

The journal is a file of all incoming and out-going documents, representing the history or memory of the organization. The retrieval process was, with the chosen structure of the journal, rather cumbersome. The journal office works as a service function for the case-workers in the administration, who aquire documents on specific issues from the journal. The project group consisted of the women working in the journal office, representatives of the case-workers who were the users of the journal, and consultants with competence concerning organizational issues as well as computers. Two researchers took part with the purpose of trying out design methods, primarily for user interface design.

The project group worked with three different types of methods: scenarios to sketch different early alternatives, i.e. different main directions in the design, simple paper mock-ups of screen images, and prototypes running on the type of computer equipment which was common in the organization. These prototypes were based on a 4'th generation language.[30] I shall discuss these methods throughout the following chapters.

The Smalltalk case

"In the early 1970's the Xerox Palo Alto Research Center Learning Research Group began work on a vision of the ways different people might effectively and joyfully use computing power. In 1981 the name of the group was changed to the Software Concepts Group or SCG. The goal of SCG is to create a powerful information system, one in which the user can store, access and manipulate information so that the system can grow as the user's ideas grow. Both the number and kinds of system components should grow in proportion to the growth of the user's awareness of how to effectively use the system."[31]

SCG has in its work been concentrating on two areas of research: a programming language, and a user interface, which supports the user in her programming effort. The programming language and system is called Smalltalk-80. Smalltalk-80 is an object-oriented language, based on a small number of concepts. The whole system is, in principle, written in the Smalltalk language itself, and as such accessible to the user.

"Smalltalk is a graphical, interactive programming environment. As suggested by the personal computing vision, Smalltalk is designed so that every component in the system that is accessible to the user can be presented in a meaningful way for observation and manipulation. The user interface issues in Smalltalk revolve around the attempt to create a visual language for each object. The preferred hardware system for Smalltalk includes a high-resolution graphical display screen and a pointing device such as a graphical pen or a mouse. With these devices, the user can select information viewed on the screen and invoke messages in order to interact with that information."[32]

In 1982-83 I spent 8 months with the SCG,[33] sharing the daily life and work of the group. During that period I participated in a couple of the projects in the group, and I conducted interviews with the group members concerning their practice and conception of the Smalltalk-80 language and environment.

What I have chosen to call the Smalltalk case is an extraction of the various experiences concerning design of the user interface that I gained from this stay. These experiences illustrate what it means for design of user interfaces to be appropriate for the users: users and designers are the same group of people, i.e. we can take the user interface of the Smalltalk-80 system as an expression of what **the users** need. We can furthermore see the Smalltalk-80 environment as expressing **the designers'** needs.

The group has access to the most advanced computer technology, to very competent computer people, and the researchers have, to a large extent, the freedom and resources to pursue their own ideas, as individuals and as a group. Although management at times intervened in the design process, this could be seen as quite an ideal situation.

Theoretical background

Theoretically, this dissertation begins and ends in computer science. My intention is to facilitate better user interface design, which means to begin and end with what computers and computer applications are, and how they can be constructed to function as "efficient" as possible. To assess efficiency, however, we need a different kind of methods than those needed to assess for instance the efficiency of algorithms. In the evolution of computer science, the theory of algorithms and their efficiency has developed out of mathematics. To deal with the "efficiency" of the user interface we need not only to deal with the computer, but with the interplay between human beings and computers. Computer science is not capable of offering its own explanation of this relation. Just as computer science has earlier borrowed theory from mathematics, language theory,[34] etc., we need to look for sciences which can help us in our current aims. Other fields of computer science, especially expert systems, are at present in a similar situation.

Within the area of user interface design the cognitive science approach is one way of using another science. I shall discuss the practical applicability of the results of this approach in this dissertation. The philosophical impacts of the approach are discussed by for instance T. Winograd & F. Flores, L. Suchman, and H. & S. Dreyfus.[35] Their critique is part of a new and evolving theoretical approach. An approach which is considering human activity, including design or use of computer applications, not as primarily characterized by rationality, planning, and reflection, but by practice and our ability to act in situations, which are more or less familiar to us, where reflection is something secondary or 'post factum'. The thoughts of a number of otherwise different philosophers are used in these approaches: Winograd & Flores, and others with them, have used the ideas of Heidegger and Gadamer in their work. The thoughts of Wittgenstein have been used by Lundequist, Göranzon, and others,[36] and all of these have inspired me in my work. The courses taught by, and my discussions with, Pelle Ehn and Morten Kyng have opened my eyes for these ideas, as well as for the thoughts of Polanyi,[37] and the Soviet psychology,[38] which I came to focus on. Where Heidegger and Wittgenstein have done their work as general philosophy, the aims of Winograd & Flores, Suchman, H. & S. Dreyfus, Göranzon, and others are to create a new foundation for design of computer applications, and for understanding the role of computers in the life of human beings, in general or in specific human activities, such as work.

I shall not go into long discussions about the similarities and differences between these schools of thought. The interested reader will find this discussed by Ehn.[39] Rather I shall repeat one reason and state another for choosing to focus on the Soviet psychology:

1. The approach taken by for example Winograd and Flores is to bring together pieces of different theories from different traditions. With this follows the problem of convincing the reader as well as oneself that the theories can be applied together, that they do not build on conflicting assumptions, and so on. I try to avoid this by staying within the limits of one school of thought, and here seek the needed explanations. The Soviet psychology seemed to offer a better chance for this than the rest.

2. Although all the theories share the fundamental idea that practice is the basis for the being and doing of human beings, they differ in the way they consider language: to Winograd & Flores, all activity is primarily

communication. The lack of consideration for the physical activity underlying the activity results in a view of practice as something which exists for each individual human being. With the ideas of H. & S. Dreyfus[40] the problem is that they primarily discuss the learning and competence of the individual, not that both are bound to the material conditions, i.e. not the fact that artifacts used in some activity, as well as the materials used, are carrying a certain practice. The work of Leontjew[41] has the advantage of considering practice both in relation to the material conditions shared by a group of human beings, and to the way this is reflected in the consciousness of the individual.

It is the fundamental idea of the Soviet psychology as presented by Leontjew and others, to get to an understanding of society or culture on the one hand, personality on the other, and primarily of the connection between the two. To do this they unite aspects of sociology, historical materialism and psychology into a theory which takes its starting point in human activity as the basic component in purposeful human work. It is not my intention to go into abstract and philosophical discussions about this theory, but, inspired by Danish psychologists,[42] to present a concrete framework about computer applications, their role in human work activity, and the impacts of this for design. By this approach I start out from a theory about human beings and the role of computer applications in their work, and apply this theory to get to a framework by which we can deal with computer application and the design of such. My special concern is for the user interface, because, as we shall see, this is essential for how the computer application appears to its user in use.

Design of user interface is a topic which relates to a number of disciplines in computer science. Technological innovations, both concerning software/hardware and design methods[43] have often arisen in close connection with computer graphics. To support the software/hardware concepts, as well as to promote various design methods, programming languages, programming methods, and support from programming environments are important parts of user interface design. Furthermore, programming environments have played a special role in the development of both user interfaces and design methods: Because programming environments are developed by programmers for their own use they reflect the needs and wishes of the future users in a special way, also where it comes to the user interface. Designers of programming environments have often been among the first to utilize technological innovations.[44] Systems descriptions, formal specifications, etc., are used in user interface design and the theories of such apply for the user interface as well as for the rest of the application. All of these disciplines play a role in making it easier, or even possible to design good user interfaces.

We can see the framework that I present in this dissertation as **one** way of extending our theoretical foundation within the areas of human-computer interaction and design of user interfaces. The value of this extension will ultimately be tried out through the explanations that is offered in the dissertation, of known phenomena, and when my suggestions for design of user interfaces have been tried out in practical design. The latter is not part of the dissertation work.

Structure of the dissertation

I have presented the theoretical and the empirical backgrounds separately, and I have stressed the fact that it was the empirical background that was driving the need for a new theory. However, the dialectics between the empirical and the theoretical results are what really matters. Although the structure of this report is determined by theory, I shall throughout the report use empirical examples to explain the theory; and I shall give concrete examples of what the new theory means for the empirical level. Furthermore, it is important to look back at traditional theoretical frameworks and methods for design of user interfaces to see how they can be viewed in the light of our new theory. Hopefully this can help explain both the discrepancy between design practice and traditional methods, and the need for a changed practice.

Naturally, the subject is centered around the computer applications or **computer based artifact**. The artifacts are employed by users in use activities to create some product, or achieve some goal. The use activity and the intended product is on the one hand part of determining how the artifact can be employed. On the other hand the actual construction of the artifact is part of determining which use situations and products can be created. Similar dialectic relations exist between the design activity and the computer based artifact, between the design situation and the design methods which can be employed, etc.

I have chosen to focus on some of these relations in the structure of the report.

In Chapter 2, I shall present the overall theoretical framework of the dissertation. The framework will be illustrated by examples from the empirical studies.

In Chapter 3, a number of design activities will be presented. They present the empirical background and they will function as examples in the following chapters.

From the theoretical framework of Chapter 2 I shall in Chapter 4 elaborate on the parts concerning the user interface, to get to a more detailed understanding of the user interface and its relations to the design and the use situations.

Whereas Chapter 4 emphasizes the user interface, Chapter 5 will focus on design of user interfaces. I shall discuss various design methods and their way of handling the user interface design. Furthermore, I will discuss the applicability of different kinds of design techniques to improve the user interface design.

In Chapter 6, I will use the conclusions of the previous chapters to give a number of concrete recommendations to designers of user interfaces. I hope that these recommendations, as well as the rest of the dissertation, can be used to give designers inspiration to change their practice.

This dissertation does by no means look at design methods as something which can be followed like a computer executing a program. Rather it views design methods as something which can point at various ways of changing design practice in specific situations. There are many limits to the kinds of design activities dealt with in the dissertation, and I do not claim that the experiences are generally transferable outside the discussed types of design situations. I think, however, that the ideas can be used for designers within other and perhaps more traditional application domains: by presenting alternatives it will allow designers to reflect on their own practice, and eventually perhaps change this. The ideas can, hopefully, be used by researchers and students who want to deal with user interfaces, either on a practical or on a more theoretical level. On the practical level I think that the dissertation offers explanations to why the human-computer interaction functions the way it does in a number of situations, as well as it offers valuable recommendations for design.

Acknowledgements

It is not possible for me to list and thank each individual who have helped me with this dissertation: students and colleagues at DAIMI, colleagues from the Swedish Center for Working Life and the Royal Institute of Technology, Stockholm, who participated in the Utopia and Dialog projects, present and former members of the SCG/SCL at Xerox PARC, the people at Aarhus Polytechnics, colleagues from the Institute of Psychology, and from the Institute of Information and Media Science in Aarhus. Thank y'all.

I appreciate the help from old friends who work as computer scientists in industry or as consultants; not least the encouraging support from a group of women computer scientists, who have insisted on the importance of getting more women to do computer science teaching and research.

Karen Møller has helped me type and edit the manuscript. Morten Kyng was my supervisor, and he, Brian Mayoh, and Ole Lehrmann Madsen constituted my 'licentiat' committee. Jan Holdam did a tremendous job commenting on the final draft of the dissertation.

John Kammersgaard and I, together, worked on the Aarhus Polytechnics case and developed many of the early ideas from which this dissertation has evolved. Most of the practical ideas about user interface design in the Utopia project was developed by the project group in Stockholm. I am very grateful for the possibility of sharing their reflections. I also want to thank Horst Oberquelle, with whom I taught the course 'Aspects of Human-Machine Communication', for many encouraging discussions.

I owe special thanks to Morten Kyng and Pelle Ehn for creating an interesting environment for many exiting discussion, as well as for many hours of detailed comments and good advice.

---

1 A. Newell, talk at the Computer Human Interaction Conference, San Francisco, April 1985.

2 A good survey of this tradition can be found in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987.

3 See e.g. SIS handbok 113: **Riktslinjer för administrativ systemutveckling**, SIS 1973 (In Swedish. Guidelines for administrative systems design).

4 See e.g. the analysis of graphical systems description tools in S. Bødker & J. Hammerskov: **Grafisk Systembeskrivelse**, DAIMI IR-33, IR-34 and IR-35, University of Aarhus 1982 (in Danish. Graphical Systems Description).

5 Historically, we have seen different trends in management's attempts and reasons to involve users: from sociotechnical satisfaction and autonomous groups (see P. Ehn & M. Kyng: **The Collective Resource Approach to Systems Design** or E. Mumford: **Sociotechnical Systems Design - Evolving Theory and Practice** in Bjerknes et al., op. cit. (note 2)), via technology agreements with user participation in the control of traditional phase-oriented systems development projects, to active user involvement in both project groups and project management. Many examples of this type of management strategy can be found in the public sector in Scandinavia today. I will examine one of these cases in this dissertation.

6 Prototypical examples of this tradition can be found in the proceedings from the CHI conferences (**Human Factors in Computing Systems**, 1983 (A. Janda, ed.) or 1985 (L. Borman & B. Curtis, ed.)). We can identify at least three different approaches within the tradition: the British human factors tradition which insists on being an analytic, psychological discipline (see e.g. N. Hammond et al.: **Design Practice and Interface Usability: Evidence from**

Interviews with Designers in Janda, ibid.), the Card-Moran-Newell school (see S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum, 1983) who was the first to go directly into design considerations, and e.g. D. Norman and his group at UCSD (see D. A. Norman & S. W. Draper, ed.: **User Centered System Design**, Lawrence Erlbaum 1986) the contribution of which can be seen as a new path in psychologists' way of viewing design of user interfaces; a path which is breaking with the cognitive science tradition.

7 Card et al., op. cit. (note 6).

8 Ehn & Kyng, op. cit. (note 5).

9 T. Winograd & C. F. Flores: **Understanding Computers and Cognition: A New Foundation for Design**, Ablex Publishing Comp. 1986.

10 Things can be normative in different ways, and especially for design we can make an important distinction between the way that certain design methods are normative in stating how design should be conducted, and the way, as described in the text, that frameworks are normative in creating certain distinctions in our conception of a phenomenon.

11 W. M. Newman & R. F. Sproull: **Principles of Interactive Computer Graphics**, McGraw-Hill 1979 or J. D. Foley & A. van Dam: **Fundamentals of Interactive Computer Graphics**, Addison-Wesley 1982.

12 A detailed discussion of the relation between the theories can be found in P. Ehn: **Human Centered Design and Computer Artifacts**, Aarhus forthcoming.

13 Winograd & Flores, op. cit. (note 9), Ehn, forthcoming, op. cit. (note 12), N. E. Andersen et al.: **Professionel Systemudvikling**, Teknisk Forlag 1986 (In Danish. Professional Systems Development), and L. Suchman: **Plans and Situated Actions: The problem of human-machine communication**, Xerox ISL-6, 1985.

14 Many of these discussions have taken place during two graduate courses taught by Pelle Ehn and Morten Kyng at DAIMI: 'The Computer as a Tool', and 'Design and Cognition', also a study group with researchers from the Institutes of Psychology and of Information Science, and a talk with Terry Winograd have been important.

15 P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984.

16 Andersen et al., op. cit. (note 13).

17 Ehn & Kyng, 1987, op. cit. (note 5).

18 See e.g. Ehn & Kyng's (ibid.) discussion of union investigatory work.

19 Bødker & Hammerskov, op. cit. (note 4).

20 See note 14.

21 Many former project members participated in the Dialog project at the Royal Institute of Technology in Stockholm, a project on human-computer interaction for graphical work - I had the honor of being a 'friend' of the project, i.e., I was invited to several of the project's seminars. Together with John Kammersgaard I did both theoretical and practical follow-up of the ideas from the Utopia project (S. Bødker & J. Kammersgaard: **Interaktionsbegreber**, working note, The Computer Science Department, University of Aarhus, 1984 (In Danish. Interaction Concepts)).

22 This presentation is based on S. Bødker et al.: **A Utopian Experience**, in Bjerknes et al., op cit. (note 2).

23 In the Scandinavian languages UTOPIA is an acronym for Training, Technology, and Products from a Quality of Work Perspective.

24 The strategic background of the Utopia project can be found in the Utopia research programme from 1980 (P. Ehn et al.: **Training, Technology, and Product from the Quality of Work Perspective,** A Scandinavian research project on union based development of and training in computer technology and work organization, especially text and image processing in the graphic industry. (Research program of Utopia) UTOPIA report no. 2. 1981, p. 7.): "The experience gained by organized labor and the research conducted by trade unions during the 1970's into the ability to influence new technology and the organization of work at local level highlighted a number of problems. One fundamental experience gained is that the "degrees of freedom" available to design the content and organization of work which utilizes existing techno - logy is often considerably less than that required to meet trade unions demands. Or expressed another way: Existing production technology more and more often constitutes an insurmountable barrier preventing the realization of trade union demands for the quality of work and a meaningful job."

25 Discussions about the Utopia project and its predecessors as union strategy and as research strategy can be found in Ehn & Kyng, 1987, op. cit. (note 5).

26 The Utopia project was a big, and geographically segregated project, and, furthermore, I did not take part in the project until 1983. For that reason I consider myself more of an observer of what happened in the project, than one who did the work. On the other hand it is hard to talk about a project that you took part in, from the outside. I have chosen to talk about **we** - we the project, but the reader should know that **I** often took no part in what **we** did. I shall try to use notes when this distinction is important.

27 See S. Bødker et al., op. cit. (note 22) or S. Bødker et al.: **Graffiti 7. The UTOPIA project. An Alternative in Text and Images,** Arbetslivcentrum 1985.

28 M. Bartholdy et al.: **Studie av Datorstödd Bildbehandling på Aftonbladet,** UTOPIA report no. 21, 1986 (In Swedish. A Study of Computer-Based Image Processing at 'Aftonbladet').

[29] For a more detailed discussion of the project and the specific case see B. H. Kristensen et al.: **Retningslinier for valg af faglige strategier på kontorområdet - et case studie over Århus tekniske Skoles kontorautomatiseringsprojekt,** Department of Computer Science, University of Aarhus 1986 (In Danish. Guidelines for trade union strategies in the office area - a case study of the office automation project of the Aarhus School of Polytechnics).

[30] Although the work in the project group took place during working hours the participants still had to do their regular work too. No resources were available for hiring extra employees to take over some of the work. This made it hard for the employees to put priority on the project work. Management, on the other hand, had different priorities. Because they had more time most of the initiative came from management, both where it came to initiating new projects and to concrete design suggestions. Furthermore, the role of the consultants was quite problematic: On the one hand they were intended to work for the technology committee, meaning that they should act unbiased in the project group. On the other hand they were hired by management and management made use of them in other situations. This meant that suggestions from the consultants were often interpreted as directives from management, and the employees did not have similar possibilities for having their own consultants.

[31] A. Goldberg & D. Robson: **Smalltalk-80. The language and its implementation,** Addison-Wesley 1983, the introduction.

[32] Ibid.

[33] Now SCL (Systems Concept Laboratory)

[34] E.g. N. Chomsky and others who worked on syntactic descriptions of languages, including formal languages.

[35] Winograd & Flores, op. cit. (note 9), Suchman op. cit. (note 13), and H. & S. Dreyfus: **Mind over Machine,** The Free Press 1986.

[36] L. Wittgenstein: **Philosophical Investigations,** Oxford 1953, J. Lundequist: **Norm och Modell,** The Royal Institute of Technology, Stockholm 1982 (In Swedish. Norm and Model), and B. Göranzon, ed.: **Datautvecklingens Filosofi,** Carlsson & Jönsson, 1984 (In Swedish. The Philosophy of Computer Development).

[37] M. Polanyi: **Personal Knowledge,** Rutledge & Kegan Paul 1967.

[38] A. N. Leontjew: **Problems of the Development of the Mind,** Progress Publishers, 1981, A. N. Leontjew: **Activity, Consciousness, and Personality,** Prentice-Hall 1978, or M. Cole & J. Maltzman, ed.: **A Handbook of Contemporary Soviet Psychology,** Basic Books 1969.

[39] Ehn, forthcoming, op. cit. (note 12).

[40] H. & S. Dreyfus, op. cit. (note 35).

[41] Leontjew, 1981, op. cit. (note 38).

42 See e.g. B. Karpatschof: **Grænsen for automatisering**, Psyke og Logos 2, 1984. (In Danish. The Limit of Automation).

43 Newman & Sproull, op. cit., Foley & van Dam, op. cit. (note 11).

44 There are of course many programming environments that do not make full use of the technological possibilities.

Chapter 2

Human Activity and Human-Computer Interaction

> It is clear that (and has been widely recognized) that one cannot understand a technology without having a *functional* understanding of how it is used. Furthermore, that understanding must incorporate a *holistic* view of the network of technologies and activities into which it fits, rather than treating the technological devices in isolation. (Terry Winograd & C. Fernando Flores)[1]

Throughout recent years, the use of computers has shifted from calculation or information processing to a wider variety of applications, and the direct contact between the human and the computer becomes more and more advanced. We have moved from a situation where a few computer specialists interacted with computers through punch cards to a situation where computers are applied by office workers, professionals, and craftsmen, as well as by unskilled workers in their work.

Our theoretical conception of what goes on between the human being and the computer when the human uses a computer for a specific task has not followed this rapid development. In this dissertation the aim is to contribute to such a theoretical conception. Taking the words human-computer communication, or human-computer interaction literally, means that we consider what goes on between a human being and a computer the same way we consider inter-human communication. The human being is "communicating" with something which is, although not human, a communication partner. It becomes the research goal for design of the user interface to exploit the computer's capabilities to act as or instead of a human being in communication situations. In design this perspective may be useful, but many applications are not of this character: a drawing program like MacPaint,[2] seen as a communication partner, would lead to requirements such as to be able to direct the pen with commands in natural language. There are obvious reasons to doubt that artists or architects think of their real pen as a communication partner, and that this type of perspective is reasonable for computer based pens.

We can also interpret communication or interaction as a human being communicating with other human beings through a computer based **medium**. We focus on how to mediate the communication between the user and the human at the "other end" of the medium, being it the programmer or the actual partner in communication. It becomes important to understand how human beings act in communication situations, and how this must be reflected in the computer medium.[3] But again, can all computer applications be conceived as media? Who is communicating with whom in a drawing program?

Other frameworks focus on the use of computer applications for specific work tasks, comparing this to the use of traditional **tools**.[4] By applying a **tools** perspective we focus on the work process, and on the human use of tools to process some material into products. The material conditions for the use process become important: the users' competence, which is reflecting the practice of the group of users. The human-computer interaction is seen as part of the actual conduction of the work process. However, we can ask whether all computer applications benefit from being perceived as tools. What do we see if we view an electronic mail system as a tool? How we can write down a text perhaps, but more important for electronic mail is probably another aspect: that we can communicate with other human beings through the mail system.

No matter which of the above approaches we choose to human-computer interaction[5] we see examples indicating that the perspective is not rich enough to cover all types of well-known applications. Neither an isolated "the-user-and-her-tool" perspective, nor a pure media or communication perspective seems sufficient for understanding human-computer interaction. We need a framework to deal with both communication and tool use in work situations where computers are involved.

This is approached in this dissertation in the following way: When conducting some work process, you do more than just interact with your computer. You create a product through the tool or you communicate through the medium. You communicate with your work mates: to learn about your tools, materials, and products in a problem situation, to coordinate the common effort. Communication does not just take place isolated from the relation with materials and products – linguistic meaning is formed in humans' joint labor activity, to contribute to the overall goal of work. Human-computer interaction is part of the specific work activity in which the computer is applied.

The main purpose of this dissertation is to understand human-computer interaction, and the conditions for this, the user interface. In literature, human-computer interaction is often considered either as something which can be considered totally independent of the specific use situation,[6] or as something which is specific for the specific user in the specific use situation.[7] The idea here is that on the one hand the human-computer interaction cannot be seen independently of the use situation. Many aspects are important, e.g. whether the interaction is part of a communication or a production of some more or less tangible product. On the other hand we need a definition of user interfaces which does not just focus on the needs of the individual user in a specific situation. The reason for this is that we need to design computer applications to be used by more than one user; a theory that aims only at design for specific individuals seems elitist and has hardly any practical applicability.

In this chapter, the goal is to reach further than just an analytic framework by which human-computer interaction can be conceived in specific use situations, and further than a design strategy for user interface design. The idea is to elaborate on a theory about human-computer interaction as part of a theory of human work.

The approach is focusing on the use of computer based artifacts[8] in human work activity, and, thus, on the role of the user interface and of human-computer interaction in a specific work activity. Specific for human life, the way it is viewed here, is that human beings, as opposed to animals or things, create artifacts to be applied in a *future* use activity. By this choice of approach I have restricted this dissertation to deal with computer use as going on in purposeful work, with a specific organization and division of work, and based on a specific practice of the users.[9]

This presentation consists of the following: a general presentation of the human activity theory, a section where I present my elaboration of this theory into a theory about computer based artifacts, a similar section on design of artifacts, and finally, a summary and discussion of the framework.

Human work activity

In this section I shall give a short presentation of the relevant part of the human activity theory. The presentation is primarily based on the work of A. N. Leontjew, [10] of L.-C. Hydén,[11] and of B. Karpatschof.[12] But my thoughts and ideas are also inspired by L. Wittgenstein,[13] M. Polanyi,[14] and T. Winograd & F. Flores.[15] The purpose of this section is to provide the reader with sufficient background for understanding the following sections in which I use the theory and elaborate on it.[16] I shall start the introduction with an example.

An introductory example

We can look at make-up of newspaper pages.[17] As the basic component, page make-up is conducted by one or more make-up persons, each carrying out their own **individual** page make-up **activity**.

The **individual** human activity, such as page make-up, is part of the **collective** activity of various groups, e.g. the make-up persons at the specific newspaper, or the persons handling the front page of a specific newspaper. The individual make-up activity has a goal, or we can say that it is directed towards an object: the newspaper page. The individual page make-up activity contributes to the goals of the collective activity, for instance the activity of the group of persons: reporters, editors and typographers, handling the front page. This goal could be to promote a certain story, perhaps in turn to make the paper sell better. The make-up person places the story on the top of the front page, he uses a certain size of headline, etc. These things have a meaning to the front page group, but probably also to the readers of the paper.

To organize, coordinate and control the collective activity, **communication** plays a role. This means that the make-up persons direct some parts of their activity towards other human beings or subjects, e.g. the editors or each other. We call this the **communicative side** of human activity, when for instance the make-up person discusses with the editor how to have the articles fit into the page. Other parts are directed towards objects, the **instrumental side**: the make-up person handles paper galleys, pictures, etc., to actually create the front page.

The make-up person uses **artifacts** in the activity. Both the instrumental and the communicative side can be **mediated** by artifacts. A knife or a pair of scissors mediate the instrumental side, the forming of the newspaper page. Lay-out sketches, production plans, etc., mediate the communicative side, the coordination of the production.

We can view the individual human activity as conducted through **actions**, which take place in a unity of time and space **with specific intentions**. Making-up a specific newspaper page may consist of placing an ad in the rightmost bottom corner, fit in some text between the ad and a picture, etc. An action is conducted through one or more **operations**, which are bound to specific material conditions. To place an ad the make-up person picks up the photo typesetter paper,[18] picks the knife, cuts off some of the white area around the ad if there is too much white, without being conscious about this. The right operation to be used in a specific situation is **triggered** by the material conditions; it is not chosen consciously by the typographer. When the typographer places an ad, we can say that the actions are what he is consciously doing, e.g. placing an ad in the right most bottom corner, whereas the operations are what he does, to realize this, e.g. hold knife, cut paper, try position on page ground.

What in some situations are actions can in other situations be conducted as operations as part of other actions: e.g. placing a picture can in some situations be something which has its own specific purpose, whereas in other situations it has not, it is conducted as as part of placing an article on the page ground.

Through learning, in special learning activities or in daily work activities, the person obtains a repertoire of operations to be used in a specific activity. He gets to share the **practice** of typographers, at the same time as he is part of constituting this practice.

The make-up person can reflect upon what was formerly operations, and to try to perform former operations as actions, e.g. if the editor tells him that he is not pleased with the product of the work of the typographer. We call this **conceptualization**. Changing the level of action means changing the object (or subject) of the actions – instead of working on the article the make-up person starts to think about headlines, pictures, and the like. Unforeseen changes in the material conditions in the specific page make-up activity may cause conceptualization. We call such situations **breakdowns**.

Situations where the make-up person's knife causes a breakdown could be:

•while learning to use a new kind of knife,

•if the knife breaks,

•if the knife is badly suited for the kind of cutting its user wants to achieve, a switch of knife can be necessary, or

•a special handling of the knife to achieve the intended result.

In such a situation the knife is no longer something which is handled only through operations. Rather the knife becomes the object for the actions, removing the focus from the real object, the newspaper page.

Summary of the human activity theory

In this section I will present the main concepts and ideas of my interpretation of the work of Leontjew and others.[19]

Human work activity is the basic component of this theory. A human being conducts each concrete **action**, through which any activity is conducted.

The concrete actions that a human being conducts contribute to an individual human activity: a process through which the human being produce some kind of relation to the physical and social world around her.

The human being can aim to achieve a goal in the activity, i.e. aim at solving a task or problem, and she can direct her activity towards some physical object, some material which is to be affected through the activity. We can say that all activity is bound to a **goal** and/or an **object**.

The characteristics of the goal or object are partially determining and structuring the activity. If a human being deals with a physical object, the physical structure of the object will delimit which actions to perform, how, and in what sequence. If the task is to collect some information from various physical and human sources, the physical and social structure – where to start, who to talk with and how, etc. – will determine the structure.

The production by the human being of her relation to the material world goes on in this way: On the one hand, some need will cause the subject to perform actions, with certain **intentions**, which intervene with the physical and social world, e.g. to change a specific object. This intervention is based on a mental reflection of the world, including the specific object. On the other hand, as e.g. a physical object is structuring the actions of the human being, the reflection of the object is created and changed through the actions. U. Juul-Jensen[20] says about reflection: "...human consciousness is a reflection of the social and material world we live in. This does not mean that consciousness consists of a special kind of images which in some ways resemble the objective world. That consciousness is reflection of the world means that the physiological structures by which it is constituted are products of our activity in this world."

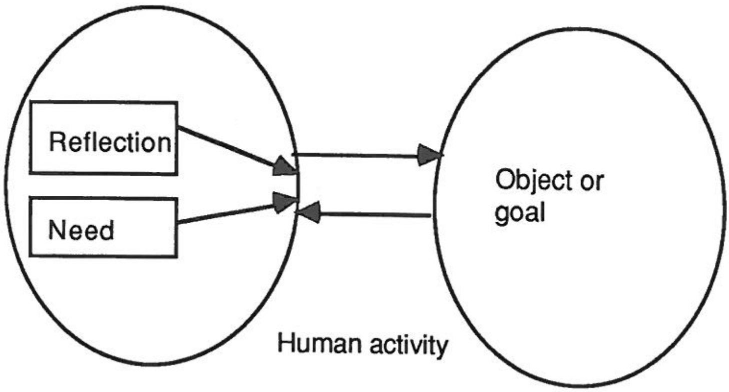


Figure 1. Reflection[21]

In the human activity, different **means** are employed: cultural techniques, such as ways of structuring the individual actions; artifacts, e.g. hammers or telephones; and languages. Through these means the human being mediates her relation with the world.

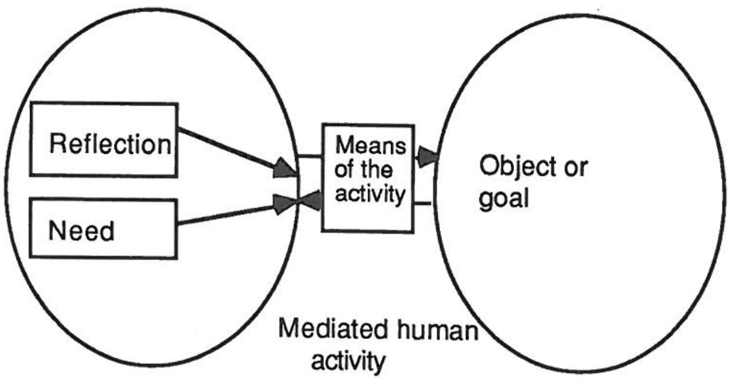


Figure 2. The mediated activity

The human beings always participate in various activities. These collective activities are structured according to a certain order of the society in which they take place. The individual will meet this order through power relations, institutions and grouping of interest in society, under which the human being lives, at the same time as she can contribute to their change. In most societies the division of labor has caused a separation between the needs of the individual and the goal of the activity in which she takes part. Furthermore, the needs of the individual as part of different collective activities might differ and even conflict.[22] We can say that the human being has not one need in the concrete activity, but a whole cluster, some of which are conflicting.

In collective activity, language is used to coordinate work. Each individual activity consists of communication with others human beings to organize, coordinate and control the activity, and of actions directed towards things which serve as objects or artifacts in the material production. We talk about the **communicative** and the **instrumental** side of human activity.

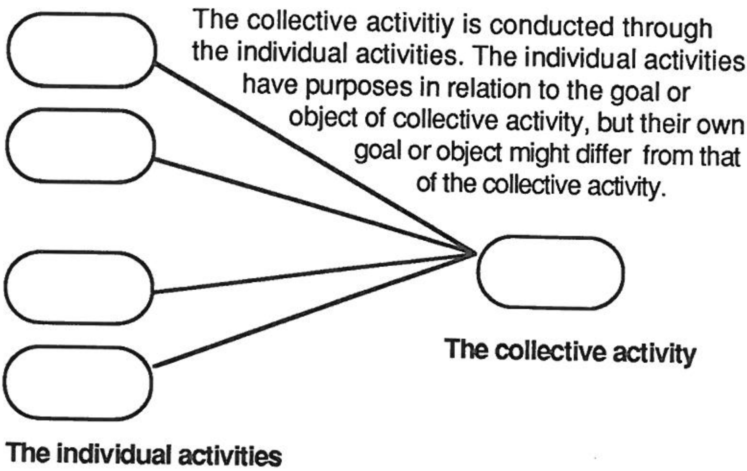


Figure 3. The outer levels of human activity

We can look at an example about programming. When I[23] use the Smalltalk-80 system to program something like some code to take care of my household finances, my activity, first of all is part of financing my household. In my activity, my actions on the budget, etc., are mediated by the Smalltalk-80 system.

My Smalltalk-80 programming is, of course, aiming at producing an overview of my household finances (the instrumental side), and perhaps I even direct my actions towards the computer application as such (instrumental) in situations where the Smalltalk-80 system becomes the object for my actions, – if I open a window which I get placed on top of what I am doing, I need to aim my actions at moving or removing the window. At the same time I might need the overview to discuss it with my family or my bank (communicative side). And in case of errors in my programs, I might contact somebody who knows more about Smalltalk programming than I do.

Actions

Each individual activity is conducted through **actions**, conducted in a unity of time and space, with specific **intentions** (*what* ought to be done). It is because of the intentional aspects of the actions and activity that we can communicate about the instrumental side of our activity – that our instrumental activity has a meaning to us. These actions are consciously directed towards an object or a subject. Each action that a human being conducts also has **operational** aspects (*how* is it done).

In the activity the intentions of the actions might not relate directly to the goal or object of the activity. To create statistics about the frequency and habits of Macintosh users, we might interview people, design questionnaires, draw tables, or ask somebody who to interview.

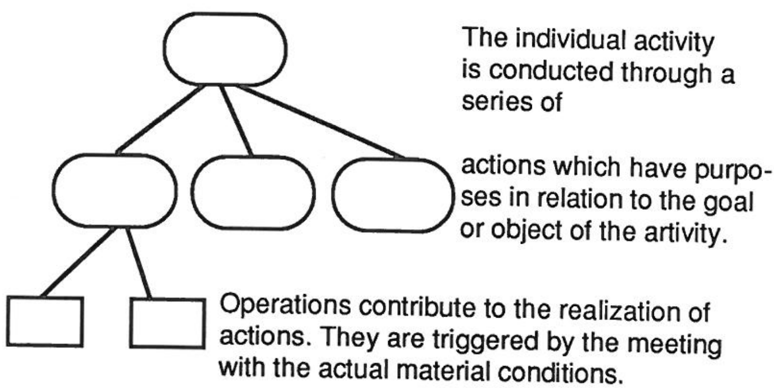


Figure 4. The individual human activity

In the Smalltalk-80 example, I use the artifacts in my activity. I use the mouse to move the cursor around the display, I press the left most button to select items on the screen, I press the middle button down to show me a menu of various things to do, I can select a command by moving the cursor to the specific entry in the menu while holding down the middle button. I can activate the program by letting go of the middle button.

However, when I program I don't think about pressing and releasing mouse buttons, etc. These are operations to me. I think about creating an object called household finances, to create a method to show my expenditures. Maybe I remember that I can copy a piece of code from some similar method. I open the browser, find the method I am interested in, and so on. These are the actions that I do, by purpose or intentionally, to fulfill the goals of my activity.

Operations

The operational aspects of actions are implemented through a series of **operations**. Each operation is corresponding to the concrete material (physical or social) conditions for conduction of the actions, and it is triggered by the meeting with the specific concrete material conditions. Operations are senso-motor units which a human being perform in a specific situation, without consciously thinking of it, to perform the actions which she is consciously aware of.

The individual human being possesses a certain **repertoire** of operations. This repertoire is part of the conditions of a specific activity because they form the basis from which operations are triggered by the meeting with concrete material conditions. For each concrete action, the human being is dependent on the triggering of a sequence of operations. If these do not exist she must conduct different actions consciously. Most likely, she must conduct more detailed actions due to the lack of operations. A simple example is writing a letter with a new word processor – we need not only be conscious about writing the letter, but also about turning on the computer, opening the editor, etc. Actions can be **operationalized**,[24] i.e. turned into operations. Operations can be **conceptualized**. Conceptualization means to articulate for one self what is otherwise self-evident. When we use a new word processor we know, or become conscious of that we cannot just operate it the way we did with the old one, and we conduct what were formerly operations as actions.

The operations applied in a specific action are not conscious to the human being. But through conceptualization they can be made conscious to us as the actions they once were; we can name a specific sequence of operations, and understand and explain reasons for their application at the level which was the level of the former actions. In specific situations, or after conduction of a specific activity, we can ask a person how and why she did what she did.

Conceptualization can take place in **breakdown situations,**[25] situations in which some unarticulated conflict occurs between the assumed conditions for the operations on the one side, and the actual conditions on the other; between the human reflection of the material conditions, and the actual conditions. Let's assume that we can use the new word processor like the old one, and we start to write a letter. We might succeed with this for a while, but sooner or later something will probably be different and we are forced to see letter writing as something which requires other operations, and thus, other actions.

For the purpose of design it is interesting to focus on the character of the operations and their material conditions: In design we are going to change operations and their conditions for a specific activity, and for that reason we will like to focus on both actual operations and conditions, and future changed ones.

However, we cannot ask the person to predict her future operations in a future action. She will not know these until they are done; they are triggered by the material conditions, by the meeting with the actual nature or culture, not by any quantifyable set of conditions. We say that the operations are usually **non-articulated.**[26] The material conditions are often **non-articulable,** which gives an action a certain character of unpredictability. Even though it is possible to get to know something about which repertoire of operation is possessed by the human being for some purpose, neither the person herself nor any observer can predict which operations come into play in the specific activity of use.

In my programming, I sometimes operate at one level of actions: When I have programmed my "show expenditures" method, I want to try it, and I issue the command "do it". Other times I operate on other levels. If I want only a part of the method executed, I might choose another strategy: select the program text that I want executed and then issue "do it".

Things can happen which bring the former operations into my consciousness – if a wrong menu shows up when I want to say "do it". Did I press the wrong mouse button? Or did I select a wrong item before opening the menu? To find out and to have the right thing done, I need to do my former operation step by step: select the right place, press the leftmost button, move the cursor (and mouse) to do the selection, release the leftmost button when the text has been selected, press the middle button, move the cursor until "do it" is highlighted, release the middle button.

I often found myself in similar situations when I started to use Smalltalk-80: I needed to think consciously about every tiny step, but after some training my focus moved to the programs that I made, I operationalized some of my former actions into operations.

In this example, I have actually articulated some of the usually non-articulated aspects: my operations on the artifact, the Smalltalk-80 system, and on the problem of financing my household. This articulation will, however, not lead to any predictions about what I will do in another similar activity, because the material conditions might at that time trigger a different combination of operations.

Practice

A group of human beings who conduct a collective activity with a specific object or goal shares a **practice**. The practice of a group arise from, and is carried by some common goal or object, as well as by the conditions of the collective activity, e.g. materials and organizational surroundings, and the **means** of the activity. Practice is reflected in the repertoire of operations of the individual member of the group, at the same time as the individual member is part of constituting and producing the practice of the group through her actions and operations.

The means of the activity are important carriers of practice. The **cultural techniques** which are specific ways of doing things, and spoken language, belong to one category: they can be made explicit through articulation of the non-articulated, but they are only present in the activity through actions and operations being carried out by persons on physical objects or subjects. Written language and artifacts, which are either **passive** or **active externalized** can be present as objects independent of the actions of a human being. A passive, externalized artifact facilitates a person's conduction of certain operations, whereas an active externalized artifact besides from facilitating certain operations also replaces certain former operations.

Some aspects of practice can be made **explicit**. They can be formulated in guidelines and theories. Cookbook recipes, text books about food and nutrition, dictionaries for chefs, and books about organizing work in big or smaller kitchens present examples of the explicit practice of cooking. They represent the articulable aspects of practice.

We can, however, only through practical experience learn the difference between a hand mixed cake or a machine mixed, the difference of using four large eggs or four smaller ones, of an oven which is warmer or an oven which is colder. Likewise can we learn the exact result of asking the kitchen assistant for three ripe tomatoes, or how to know when a certain steak cooked a certain way is 'medium' as ordered by the customer. Only through practical experience do we operationalize actions, so that later the "right" operations will be triggered by the right conditions: e.g. to choose a fork or a whisk to beat the eggs for a certain omelet. We call these aspects **personal** or **tacit**.[27]

Building up a repertoire of actions and operations is one result of learning and socialization into a group of human beings, e.g. a group of skilled workers', collective practice. But to master practice means not just to be able to conduct certain operations in which certain artifacts and languages are applied. Practice reflects both the instrumental and the communicative side of actions and activities, and it reflects both the operational and the intentional aspects. To master practice means to know the meaning of, or intentions behind the words and the way of organizing work. An apprentice needs to know not only how the work is organized and how products come into being with this or that way of organizing work, but also what it means if the master journey man asks him to do something, compared to if his fellow apprentice does. The secretary needs to know how to approach, in writing and speaking, people at different levels in the organization in which she works, and in its surroundings. This include both general rules and norms, and knowledge about the specific organization, and the individuals around her.

It is important not only to conduct some specific operations and actions to achieve a specific product from certain materials and with certain artifacts, but also to distinguish the quality of the materials, know a good product from a bad one, etc. For a carpenter who is to make a dining table, the meanings of the words 'oak' or 'teak' denote not only two different kinds of wood, but also certain styles of products, certain artifacts and operations, etc. To the secretary, a letter to the chairman of the board, and a draft of next year's budget implies different styles of writing as well as different typography and layout.

It is through the intentional aspects that e.g. the quality of materials and products become communicable among human beings who share the same practice, not because each individual is able to make explicit the choices that were made and the reasons for these, but because they share values and experiences.

We can return to the example of Smalltalk-80. What does it mean to master the practice of Smalltalk-80 programming?

First of all, to possess the repertoire of operations and actions, previously discussed. Secondly, to know a lot of legends about the history of the system, the work of its developers and their ideas behind the system, e.g. Alan Kay's original Dynabook concept.[28] To know the structure of the system and the programming style that is part of the tradition.

Reading books about Smalltalk-80 and about programming and programming languages in general, and working with the group of skilled "Smalltalkers", watching what they are doing, studying examples, and trying things out oneself, is part of this. To talk about Smalltalk-80 and to know the special vocabulary connected to this is important too. Goldberg explains some of this: 'Some people refer to selection using a click as **bugging**. Their expressions take the form of "bug that command" or "bug outside the view".[29]

To master Smalltalk-80 programming means to know the tradition, conventions and style; to recognize good Smalltalk-80 programming style when encountered; to know the programming styles that are applied by "Smalltalkers",[30] to use visual programming, to start out from how one wants things to look on the screen, to know how to start out by copying and modifying pieces of code, that does 'almost what one want', not writing new code, and to know 'debugging into existence', a way of creating pieces of code one by one when they are needed.

Some of this competence is explicit: some of the history and folklore, the programming language and the structure of the system.[31]

Other parts are personal, e.g. knowing good programming style. And one doesn't find ones way through the system by remembering its structure explicitly – but by knowing, by experience, that a method that looks almost like the needed one is hidden in some specific class in the hierarchy.

The cultural techniques used in the programming activity are the programming style, the interaction style (such as the way we activate menus by pressing the buttons of the mouse), and the language by which we communicate about Smalltalk-80 programs.

The communicative and the instrumental side

The communicative side as well as the instrumental side of human activity consists of operations and actions. What is on the instrumental side a certain combination of actions realized through operations triggered by the meeting with the material conditions, is on the communicative side a combination of actions realized through operations triggered by communicative actions[32] and operations conducted by another human being. This meeting will be mediated by language, but perhaps also by artifacts, such as telephones. The interpretation of the utterances is based both on practices of the communication partners which they to some extent share, and on the actual operational 'setting' around the communication, e.g. whether one can give orders to the other or not.

The objects and subjects that we direct our actions and operations towards are not only nature, they are created out of culture. With subjects we share some practice, and only in rare occasions do we meet another human being as nature, as an object, towards which none of our communicative practice works. Good examples are difficult to come up with, but the situation where a lady tries to explain to a hunter from a rare tripe in a remote corner of the world that her dog wants to 'go to the bathroom' could end up as one example. In our relation with things, we often meet artifacts, man-made things. Artifacts are not meant to be objects to our work, but to mediate our relation with other objects: materials, etc. Artifacts can be the object for our actions and operations if these aim at producing the artifact, but the artifacts can also become nature to us another way, if our operations stop working. In such total breakdowns we will no longer recognize a hammer, but only pieces of wood and metal. To me, most mechanical devices, such as my bike, stop being anything but pieces of metal and plastic the moment they stop working.

Learning

Learning is to socialize into the practice of a group through both reading of theory and practical experience. In this process a person is not necessarily able to reproduce all aspects of practice. We can say that her personal competence level has not yet reached the level of an expert who masters a specific practice, with respect to that specific practice.

An important part of learning is to build up a repertoire of operations. When a new artifact is brought into a practice this practice will change. Even the most competent expert will probably have to change her repertoire of operations, and for a while she is returning to a lower level of competence.

For this reason it is important to know how we learn to use a new artifact, and how this differs from the routine use activity. Furthermore, it is important to know what impacts this difference has for design of the artifact.[33]

We can summarize investigations and discussions in the human activity literature[34] about how human beings learn, more specifically how they develop their repertoires of operations, in the following:

1. Activity on material objects cannot be learned without practical experience.

2. Activity which has an abstract goal, such as solving a mathematical problem is easier learned[35] and carried out in connection with physical objects than with representations of such. This is in turn easier than in connection with language, which is easier than activity which is totally based on mental reflection. E.g., adding is at first performed by kids by counting physical objects, then they move on to master adding based on figures, then to a state where adding works best if they are allowed to talk, etc.

3. When operationalization takes place it is at first very situation specific, but as the human being meets new conditions, the variation of situations which can be handled by operations grow.

4. By the novice the activity takes place at a very detailed level of actions, where each action is consciously planned. With experience the human being moves on towards an unreflected totality. This is achieved through generalization, through operationalization of planning actions, and through abbreviation, an operationalized skipping of certain operations due to the conditions for them and a knowledge about the result. E.g. in multiplication, knowing the result of multiplication with 1 instead of conducting the operation, communicating in situations in which the context is obvious to both communication partners, in carpentry when smoothing some wood, to skip the sandpapering because you already did well with the plane.

5. The person is brought "down" from one level of competence to another either due to some pedagogical questioning of the former operations and their conditions, or because she is trying to apply old operations on the new artifact, and is encountering a breakdown. If, and how fast, she can be brought back to her old level of competence or above depends on the artifact, how much she can rely on the generality of her operations, which education is given, and whether she can make use of experiences from other types of activity.

6. The use of an artifact is, if the artifact works well, operationalized. Ideally, learning starts out with actions towards the artifact and ends without.

The challenge for design of the artifact is to build on existing practice in design to avoid that all experienced users are turned into novices, both in their more general practice of the activity and in the specific use of the artifact. Furthermore, design changes practice by introducing the new artifact, but practice can also be influenced through education: the challenge is to design the needed education to change practice to make use of the new artifact. Design is a process of learning for the participants, i.e. if competent users take part in design, both their general practice, and their specific practice in relation to the future artifact change while the activity is going on.

We shall move on in the next section with more elaboration on the theory. The next issues will be the computer based artifact, and especially the user interface – what can we say about these issues using the theoretical background.

Computer applications

In this section, we shall shift focus from the human activity as such to the computer based artifacts which are used in human activity. *I shall use the human activity theory in an attempt to develop a framework of my own which allows for focus on the computer based artifact.* The idea is to get to a conception of computer based artifacts, and especially user interfaces, which can be applied in design. Such a conception is not necessarily recognizable by the user in a specific use situation: they belong to the domain of design of computer applications. In this domain the computer application is an object for the activity. In the domain of use the specific computer based artifacts will have a meaning just like other artifacts we work with, as artifacts, not as objects that we work on.

I will first discuss artifacts in general; and following that, the characteristics specific to computer based artifacts as compared to other artifacts. Finally, I will discuss how we can define concepts such as 'user interface'. I have stressed in the introduction that these words have very different meanings in literature, and I want, in the following, to give a definition which is operational from a design perspective.

When shifting the focus from the human activity to the artifacts themselves, we need a different vocabulary. I will no longer talk about the user's activity, its object or goal, intentions, actions and operations, but about artifacts – the type of human activity which they (are intended to) support, the kind of practice they are part of, etc.

Artifacts

Artifacts are things which mediate the actions of a human being towards another subject or towards an object. When we employ artifacts we direct actions and operations both towards the artifact and towards the subject or object, towards which we direct out actions through the artifact. In situations where an action can be seen as communicative, some of the operations which realize it can be instrumental, and vice versa.

As artifacts are not themselves objects for the activity, users are normally not meant to conduct actions towards the artifacts. Both Leontjew and Polanyi[36] have examples of this, for human activity such as reading, shooting and hammering. Furthermore the Smalltalk-80 example is an illustration of this. Because artifacts are not intended to be objects, we cannot use a hammer or talk about using a hammer without the real object in mind – driving a nail into a piece of wood. To the users, artifacts are what they are meant for.

Artifacts have traditionally evolved in a very slow process taking years. The designer has, traditionally, been a competent craftsman, and there has been a close relation between the designers and the users so that feedback from use has lead to new steps in design, over and over again.[37]

Computer based artifacts

Computer based artifacts are developed much faster. They are complex and require the cooperation of different kinds of specialists. The designers are seldomly competent members of the future user community, and there is often no feedback from the users, because designers move on to new projects within other application domains. Furthermore[38], they are not even experts in design, i.e. they do not all possess the repertoire of operations to handle design, for which reason they tend to try to follow design methods as recipes instead.

Computer applications are inherently active externalized, and they can be applied to take over former human operations. In actual computer applications this can be exploited to different extents.

Computer based artifacts, as compared to traditional artifacts, allow in most cases no direct access to the subject or object of the actions conducted through the artifact. We cannot see, hear, or feel the subject or object directly, only through the representation given by the computer. Often the object does not even exist as something separate from the artifact: the messages that we create in an electronic mail system are only intended to exist as part of the computer application. It is part of human capabilities that we are able to project our experiences with one object onto another object, or to couple these two types of experiences. Just like a blind person who is using a stick to 'see' the surroundings when walking on the street, we can couple our experiences with the computer application with the real object and vice versa. Subjects do not need to be present in time either, the meeting between two subjects can be a meeting of one subject with some part of the computer application as "stand-in" for the other subject.[39]

This relation gives computer applications a certain flexibility because it is easy to make the same computer application mediate the relation between the user and several different subjects or objects. The computer application becomes the artifact of several different activities, and shifts between objects/subjects become part of what the user does while applying the computer application. At the same time, this makes computer based artifacts inflexible compared to other artifacts because it is determined in design which objects and subjects are possible. We can open a beer bottle with a ruler, but not with a drawing program.

The point here is not to say that all computer based artifacts are very different from traditional artifacts, but to say that they can be very different even though they play the same role in use as the traditional ones.

When we shift from talking about a specific use activity to talking about a specific artifact, we can no longer talk about one object, one repertoire of operations, etc. Instead we must talk about a number of goals or objects, or a certain type of goals or objects, about certain types of objects and subjects on which the users of the artifact can conduct certain actions, etc. Without a specific use situation in front of us we can only talk about intentions. Computer based artifacts are intended to support a specific type of use activity. They arise out of and are intended for a specific practice. They arise out of various other conditions for this type of use activity.

In a specific use activity, where the artifact is applied, the user can perform some action on some object through the artifact, other actions on some other object or subject, and still other on the artifact as an object. Some of these actions can belong to the communicative side of human activity, and others to the instrumental. Traditionally, some artifacts are primarily intended to support the instrumental side and other the communicative side. For many computer applications this distinction is less clear because they often mediate actions and operations towards both objects and subjects.

The user interface

By supporting certain specific actions, the computer based artifact supports both intentional and operational aspects –*what* can be done by means of the artifact, and *how* it can be done. Traditionally, we often denote the intentional aspects of a computer application the **functionality** – **what** can be done by means of the application. The use of this type of concept belongs to a different theoretical tradition than the human activity approach; where human activity is conceived as consciously planned action: everything is consciously planned and every step in use consciously taken. With the human activity approach, the functionality can be seen as something which only reveals itself in breakdowns and situations of reflection. In such situations it is possible to focus on properties of the results of the use process – on what was done – presupposing a situation of use.

The conditions for the operational aspects which are given by the computer application will be called the **user interface**. The user interface is the artifact-bound conditions for **how** actions can be done. Constituent parts of this interface can be conditions both for operations directed towards the artifact, and for operations directed towards the real objects or subjects at different levels. This way, **human-computer interaction** becomes human operation of a computer application.

We can look at an example to show how the human being change subject/object, and thus, level of action during the use of an artifact.

A person, Anne, is using a word processor (figure 5) to write a document to be read by some other person, Betty. Anne is of course writing the document for Betty, and she will try to explain things to Betty to make her understand. Betty is the subject for this type of actions. The actions can be realized both through operations directed towards the subject (Betty and Anne use certain words and phrases in their usual communication), and through operations directed towards the document or the word processor. Anne can, however also direct actions towards the document: she can be working on the form, e.g. the typography, or she can be working on the language, e.g. on the syntax, without directly thinking of Betty.

Furthermore, we can imagine that Betty is a television newscaster, and that the document Anne writes is her manuscript; or that Betty is answering questions from the readers of a newspaper, and Anne is writing her. In both cases, Anne, while writing the document, will probably also think of communicating with the viewers or readers – she will direct actions towards these subjects, too.

But we can imagine still other objects towards which she directs her actions, for instance actions directed towards the word processor. Anne will either conduct actions towards the word processor if she has not yet developed the repertoire of operations needed for what she wants to achieve, or if a breakdown occurs like a space key not working well.

In the example as well as in general, the user interface supports actions towards the different objects and subjects which the user is intended to deal with through the artifact. But the artifact as such is transparent in the sense that it should not be an object for the actions of the user in regular use.

Breakdowns can occur for many other reasons that just an unsuitable user interface. Other material conditions than the computer application can cause the breakdown, and even when we talk about the computer application such things as software and hardware errors can cause breakdowns. In the worst case a user will see the word processor as some boxes and moving parts.

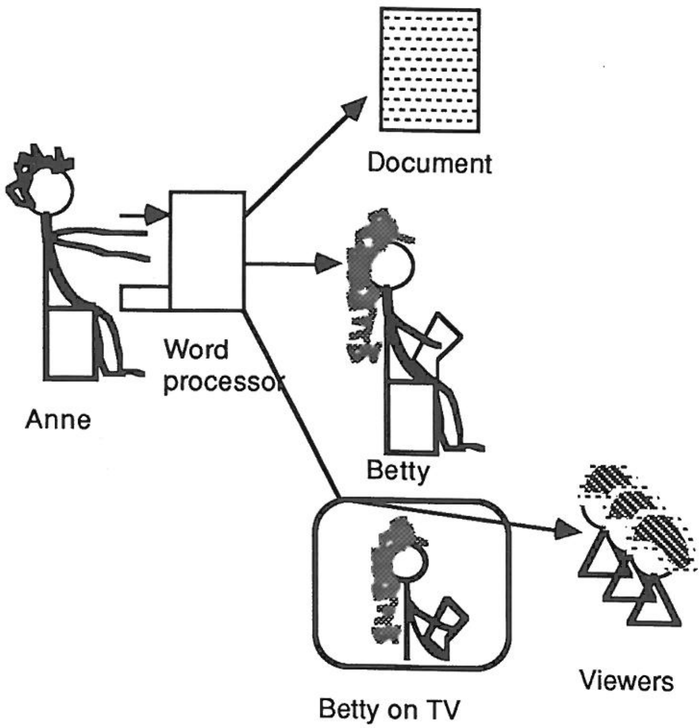


Figure 5. Anne using the artifact

How is it possible to be more specific when talking about what it means for a user to handle an object in or through the artifact, or communicating with another human being through the artifact? We can make the following distinctions between different types of situations:

1. The object is present only in the artifact (figure 6). An example of this is a spread sheet, which has no direct relation to objects outside the artifact (a print out of a spread sheet does not have the same capabilities as the spread sheet). The quality of the user interface must, for this type of artifact deal with whether the user can distinguish between handling of the artifact and handling of the object in the artifact.

2. The object exists as a physical object too, but is only present in the use activity as the representation in the computer application (figure 7). An example of this is a word processor: the object is a letter which is only

present in the use activity as what can be seen and manipulated 'on the screen'. The quality of the user interface for this type of application must relate to how the user can couple the final object and the object 'on the screen' to each other. This is the type of situation where we can view parts of the user interface as a filter between the object as it is present in the use activity, and the real object.

3. The object is present, physically, outside the artifact (figure 8). Examples of this are different kinds of control panels, where the object is handled through the artifact, but also physically accessible for inspection. For this type of user interface, too, the quality relates to the possibilities of coupling what is achieved through the artifact with what is happening with the real object.

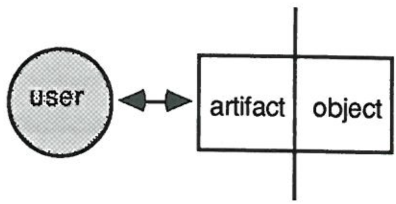


Figure 6. The object is present only in the artifact

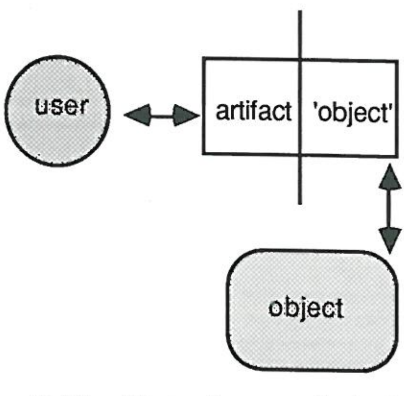


Figure 7. The object exists as a physical object, but is only present in the use activity as the representation in the computer application.

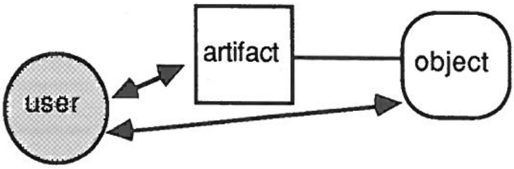


Figure 8. The object is physically present outside the artifact.

Applications similar to the above types 2 and 3 exist for the support of the communicative side:

4. The other subject is not physically present in the use activity. An example of this is a mail system. Here, too, it must be possible for the user to make the coupling between how the other subject is experienced through the artifact, and the subject.

5. The subjects are physically present but communicate (partially) through the artifact. Examples of such applications can be found in the ideas of the Xerox Co-lab project.[40]

In the following I will go into more detail about how we can understand the user interface and the role of the user interface in use. I have chosen to characterize different aspects of the user interface based on the distinction between the different objects/subjects towards which the human being directs her operations, and on the specific role and characteristics of these subjects/objects in use. I distinguish between

• the **physical aspects**, the support for our operations towards the computer application as a physical object. We will meet this object in the total breakdown or before we get to know the application. The physical aspects are the conditions for the physical handling of the artifact. The human adopts to the forms and shapes of the artifact, and a mal-adaption might prevent the forming of certain operations.[41]

• the **handling aspects**,[42] the support for operations towards the computer application. A breakdown in these operations will make the artifact appear to us as an object. The handling aspects are the conditions for the transparency of the artifact. As the artifact is a thing, the operations which are supported are inherently instrumental, no matter whether the actions are communicative or instrumental. This type of operations can, for instance in breakdown situations, be conceptualized, whereby the user can be forced to conduct actions towards the artifact as an object.

• the **subject/object directed aspects** which constitute the conditions for operations directed towards objects or subjects that we deal with "in" the artifact or through the artifact. Different parts of the subject/object directed aspects relate to different subjects or objects, but it is also part of these aspects to support the shift between subjects/objects.

The physical aspects might seem more tangible or manifest than the others, which might suggest that they have a different status than the other aspects. This is true in one way – the physical aspects constitute the nature that we can feel, see and hear in the total breakdown. Culturally, however, all three aspects are dependent on practice. *For the competent user, all aspects are equally present in a specific use activity*: They constitute a totality and a possibility of shifts between aspects.[43] Taking the example of a word processor it is only in the most total breakdown situations the word processor can be reduced to wire and plastic. The outsiders who do not share this practice will recognize a bunch of buttons, a VDT, and so on, not because they are something special, but because we all share a common culture or practice as modern human beings in our contemporary society.[44]

Let's once again take a look at Anne – how do the different aspects of the user interface of the word processor support or influence her actions and operations? A breakdown at the physical level occurs if the space key is not working well. The handling aspects support her building of operations so that she is working on a letter or writing Betty, not just pressing buttons. Breakdowns occur from different other levels of focus (focus on Betty, on the letter, on the viewers) if the word processor respond to her actions and operations in a way that Anne is not used to or not expecting. Not only software or hardware errors, but also some prompts needing to be answered before she can proceed.[45] Depending on which part of the activity we consider (Anne's actions directed towards Betty, towards the document, or towards the viewers or readers) we will see different parts of the subject/object directed aspects: The way that the document is displayed on the screen and her possibilities of giving the document the needed layout has to do with actions and operations towards the document, but also with her possibilities of "forgetting" the document and writing for Anne instead.

We can imagine, e.g., that a simple line-oriented editor will help express Anne's thoughts towards Betty just the way she likes to, but only in capital letters. But in the end, the "signals" that she sends to Betty is of course a result of her possibilities also of actions or operations towards the other objects or subjects. Breakdowns can occur when Anne is focusing on the viewers – "Betty wouldn't want to read this..." – and her focus is on Betty. If she focuses on Betty, a sudden misspelling or hopeless layout might cause her to focus on the letter, and so on.

Physical and handling aspects are conditions for possible operations towards the artifact and actions towards the subject/object. And, subject/object directed aspects added to this are conditions for possible operations towards the subject or object. In the actual use situation breakdowns can occur between any two subjects/objects, and an operation towards one subject/object might be a prerequisite for an action or operation towards another. The aspects as such are independent of each other but one piece of software or hardware can be part of constituting several of the aspects, as illustrated in figure 9 – the way a document is displayed to Anne can be a condition for both forgetting about the artifact and for thinking about the layout of the letter.

The user interface is only revealing itself fully to us in use. How we can handle this challenge in design will be discussed in the next section.

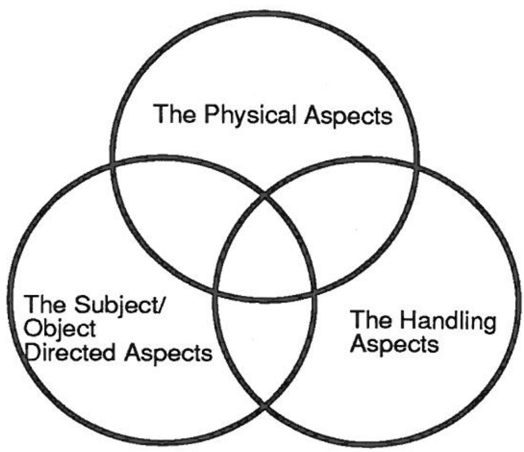


Figure 9. Software and hardware contributing to the different aspects of the user interface

Design

In this section I shall look briefly at design of computer applications seen as an anthropological phenomenon. A short discussion of design and design practice as conceived by means of the framework will be given. This presentation will be further elaborated in Chapter 5. The first look will be at the relation between use and design of computer applications. This is followed by a discussion of design practice, and finally, the special area of the user interface will be discussed.

The ideas of this section are inspired by the work of Winograd & Flores, Ehn,[46] and others, perhaps even more than it is inspired by the human activity theory. I have however found arguments for the following points in the human activity theory as well.

Design of computer based artifacts

A **need** is the origin for design as for any other activity. In our society this need is, however, not necessarily the common need of the group of users and other involved parties. Most often, computer applications are designed to fulfill the needs of managers of the use organization. Design takes place within organizational frames which tie groups with different interests together by means of power and resources.[47] Throughout the design process, many decisions and choices are made, which are not always based on rationality, but often on experience or practice of the group, and on bargaining or negotiations between the involved groups of interest.[48]

The needs of the most powerful parties will often drive design in a direction away from seeing and designing the product as an artifact. The purpose of this section is to state the conditions under which design of computer based artifacts can take place.

Design of artifacts is a process[49] in which we determine and create the conditions which turn an object into an artifact of use. The future use situation is the origin for design, and we design with this in mind.

Use, as a process of learning, is a prerequisite to design. Through use, new needs arise, either as a result of changing conditions of work or as a recognition of problems with the present artifacts through recurring breakdowns. The power relations and the division of labor are important factors for what kind of needs eventually leads to design activities and implemented artifacts.

To design with the future use activity in mind also means to start out from the present practice(s) of the future users. It is through their experiences that the need for design has arisen, and it is their practice that is to be applied and changed in the future use activity.

Design of computer based artifacts is a meeting place for many different practices, where sharing experiences is something which requires a deliberate effort. *Design is a process of learning*, both when viewed as a collective process and as an individual process for the participants. The different groups involved learn about practice of the other participating groups. For the computer experts[50] this involves learning about the work and prerequisites of the application domain. For the users who participate, learning about computers is involved together with learning about design of computer applications. For all groups the confrontation with practices of other groups contributes to learning about their own practice. This, at the same time, brings to design an innovative character: the confrontation with different practices, and thus, with one's own, is opening possibilities for new ways of doing things, and transcending the traditional practice of the users.

Design is based on and may change all aspects of the practices of the users. Conceptualization, the process of bringing into our consciousness the nature of our practice takes place in different situations **triggered** by different means. In design *we need means to trigger awareness of all aspects of practice* at the same time as some of the personal or tacit aspects of practice is dealt with better without conceptualization.

In this process two potentially conflicting goals of design come into play: that the future users must be able to assess the artifact-to-be, and that the programmers need a formal and detailed basis for their programming. This potential conflict can be dealt with in two ways: either to let the users be as detailed and explicit about their requirements as possible, or to provide the programmers with the needed competence to take part in design and help interpret breakdowns into actual programs. The human activity framework tells us that the first way is hardly feasible alone. The second way emphasizes the need for a collective learning process among the groups involved in design. Furthermore, to be a good designer means to be able to facilitate the reflection and interpretation in breakdown situations.

Although triggering of conceptualization is important in design, design rarely aims at creating chaos or the total breakdown. Each activity in design has a purpose in relation to the goal of the design activity: to achieve a design of an artifact which fulfills its purpose in use. Sharing a practice makes it possible to pursue this line, whereas we can say that conceptualization and breakdowns adjust the reflections of the future artifact of the individual participants to each other and to reality.

Any action of use of the artifact has both intentional and operational aspects. Design will often start out from the intentional aspects because these become conscious to us in breakdowns, and we are directly capable of communicating about our intentions. To fully understand the artifact, we must, however, start out from the use activity. This is the only way to focus also on the operational aspects of use, and thus on the user interface and the artifact *in use*.

To design an artifact means not only to design the object which can be used by human beings as artifacts in a specific kind of activity. As the use of artifacts is part of social activity, we *design new conditions for collective activity*, e.g. new division of labor, and other new ways of coordination, control and communication. Design of education is important too, because the artifact is to be integrated into an existing practice. This introduction is changing not only the operational aspects of the artifact, but also the other aspects of practice. A good education can facilitate this change.

Computers are capable of replacing human operations. In design of an actual computer based artifact, however, this fact can play a more or less dominant role depending on the design strategy and the application domain. The impact is that it is possible to choose in design to automate the former operations from the specific activity, or avoid such automation. A traditional accounting system is an example of a computer application, where the automation is important – earlier the human would do the calculation, now she just enters numbers and the machine does the calculation. Similarly with traditional CNC-lathes and many other applications. With modern document preparation applications, or with many modern CNC-machines it is a different matter. Of course, certain former human operations are applied, such as the use of addition for bringing up the document. However, the important aspects are that the user can enter text, design and change layout, etc., in an even more flexible way that with a typewriter, but with the same feeling of control.

Practice of designers

In this section I shall focus on the practice of designers and especially characterize different means of triggering conceptualization in design of a computer based artifact. The discussion of concrete means will be saved for Chapter 5. Furthermore, I shall discuss the role of design methods in relation to the practice of the users.

Through conceptualization, we can either direct focus from an existing or traditional[51] practice: artifacts, operations, and their material conditions, towards a new artifact, or from a new artifact towards a future use practice, especially new artifacts and the implications of such for operations and their triggering. We can distinguish two types of activities in which we make use of conceptualization: **investigation** activities where we originate from a practice to find out about a changed artifact, and **communication** activities where new or changed artifacts are evaluated, and a changed practice can be the result.

Conceptualization can be achieved through the use of artifacts and techniques by which we can construct a **materialized vision** of the future use activity, including the artifact. The mere creation, as well as the materialized visions in other words serve as triggers. Examples of such triggers are scenario techniques, systems description techniques, and prototyping. Materialized visions are the means by which we try to fix and materialize ideas about different aspects of the future artifact at the same time as they are the means or points of reference for communication between groups and individuals involved in design. There are two goals of creating such materialized visions: the communication, and the actual **construction.**

Other triggers are applied which can be very different from the above. Some of these can build upon breakdowns achieved through reading, actual conduction of the use activity, or simulated conduction. They can also be more or less structured ways of being forced to reflection about ones own practice, especially actions and operations. An important aspect of such techniques is the connection between communication and investigation activities.

Design methods are means of changing the practice of designers. They are prescriptions of the use of certain means of design, for a total design activity or for design activities with more specific goals as part of this. According to Mathiassen,[52] they prescribe the use of certain **artifacts, techniques,** and **principles of organization**[53] in design, corresponding to the cultural techniques, language, artifacts and ways of organizing work applied by the designers as part of a design practice. The prescribed means, together with a more or less explicit application domain and perspective, are the results of some explicitation of a practice: The methods are created or written down by designers who believe that they know good ways to design within a certain domain. In this process important parts of their experiences get lost, and only the explicit part of their competence is captured in the method; the design artifacts, but not all aspects of their use, and only the explicit parts of the cultural techniques.

For the use of a method this has the consequence that a method is **not** a recipe to be followed step by step, like a computer executing a program, but a set of guidelines for a certain kind of activity to be learned by the designers, and later applied through the repertoire of actions and operations.

This is only acknowledged in very few methods. Instead many authors claim a certain generality of their methods because they forget to make the application domain explicit. Furthermore, many have the attitude "Follow my method and you'll be happy". Authors of methods can of course be blamed for this attitude. On the other hand, analyzers of design methods often have the same attitude: that by following the method step by step we should be able to achieve what the author claims to have achieved through practice. Both the authors and the critics forget the importance of the meeting with the actual material conditions and of learning.

One dilemma is that methods often aim at compensating for lacking competence: by suggesting a certain way of organizing work and the use of certain means of design it is claimed that the designers need a smaller amount of design competence. But with less competence, as for all other kinds of human labor activity, the designers' repertoire of operations and actions has impacts for the quality of the product,[54] because they cannot fully act according to the "meeting" with the situation.

Design of user interfaces

In this section I shall summarize and elaborate on some important statements about the design of user interfaces.

In design we must deal with both articulated, non-articulated and non-articulable aspects of practice and with conceptualization in relation to all of these aspects. In this connection, the user interface is a special challenge: Operations as such can be conceptualized and made specific by various means, e.g. by writing down or teaching others what to do. However *the conditions which trigger a certain operation from the repertoire of operations are what we need to investigate in user interface design*. The actual triggering of **one** combination of actions and operations compared to another is part of the personal competence. This means that only breakdowns can set focus on the trigggering and the actual conditions. By using the repertoire of operations under different circumstances, such as different user interfaces and different materials or communication situations, the occurrence of a breakdown can be tested, and the conditions for the choice of operations revealed. Not only the physical aspects but also the handling and the subject/object directed aspects of the user interface are examined only through actual use situations.

The user interface creates the conditions by which the artifact does not object to the activity towards the real subjects or objects of the activity.

Design is a process that leads from a situation of numerous ways of conceptualizations, about the "old" artifacts, about practice, etc., to a situation where the artifacts don't cause breakdowns, and where it is possible to create the needed repertoire of actions and operations towards the object or subject.

The human activity approach – summary, relations and constraints

I have now presented the theory underlying my dissertation and my own approach based on this theory. I have chosen to call this a *human activity approach to user interface design*. This approach takes its starting point in human activity and allows us to deal with both communication and relations to objects as aspects of this activity. With the approach, computers can be anthropologically considered belonging to the same category as other artifacts.

Human activity is part of the social activity of various groups and it has a purpose that contributes to the goal of the collective activity. The person is part of the practice of the group.

Human activity is also a personal activity. To conduct a certain activity, the person has a repertoire of operations which are applied in conscious actions. During the conduction of the activity certain shifts of levels of action occur due to conceptualization and operationalization.

Each action performed by a human being has not only intentional aspects but also operational aspects. Likewise, the artifacts employed in the actions support these aspects.

When the person uses some computer based artifact in this activity, the most fundamental level of operation is an adaption to the physical aspects of the user interface. In addition to this, the handling aspects serve to operate the artifact. And the subject/object directed aspects support the development and use of a repertoire of operations towards subjects or objects through the application.

We have discussed how practice is important when applying, introducing and designing artifacts and especially the consequences of this for design of computer based artifacts.

In design we face a number of potential conflicts: Computers are inherently active externalized artifacts but sometimes we want to design computer applications that are passive externalized. Design is a social activity in which we need to communicate about operational aspects of the instrumental side of human activity. Design of user interfaces means conceptualization of former operations, as well as creating the conditions for new operations.

I have argued that design means dealing with practice of the involved groups. Design is fundamentally a collective activity, in which the various practices of the participants meet in a process of mutual learning. This meeting creates conflicts which create new possibilities in design.

This approach is an approach to design of artifacts for human work. No attempts have been made to claim generality of the approach outside this area. At the same time this is an indirect critique of most of the traditional "human factors" research. In order to avoid dealing with the difficult questions of competence and learning in connection with the user interface – questions which the human activity approach consider essential – they make a number of assumptions which lead them to only consider casual users and novices. For this reason hardly any existing research about "good" user interfaces for office automation are applicable in design based on the human activity approach, because all the research is the result of letting students or other novices, not professional office workers, try out the equipment.

The human activity approach has allowed for a definition of the user interface by which we can focus on different use aspects of the user interface. Furthermore, the definition stresses that competence within the application domain is important for the user interface, to design it as well as to make use of it. As a computer scientist, visiting a trade show, I am not able to "see" the user interface of a page make-up system, the way a competent typographer who has been taught to use the system in his daily work does. The definition of a user interface is not a purely technical or mechanical one. Neither is it purely individualistic, i.e. it is not a definition which claims that the user interface is different for each individual who gets in contact with the application. For design, such a definition would hardly be operational unless we also accept the assumption that we can never design artifacts except for individuals. An assumption which goes badly with our anthropological view of how artifacts have been developed through history. This, on the other hand, is not the same as saying that we cannot make user interfaces adjustable to individual needs.

One of the potential problems of the approach is its close relation between language and the organization of instrumental work. It can be argued that due to the division of labor in our society, a certain kind of work exists which has hardly anything to do with instrumental work and the production of tangible products. Rather, this work has to do with a reproduction of knowledge within the organization,[55] and with communication for its own sake. In the framework of Leontjew we can see such activity as activity which, due to the division of labor, has goals which have to do with maintenance and reproduction of the cultural techniques, ways of coordinating, and communicating about work as part of the organization's total collective activity.

The main limitation to the approach from my personal view is its lack of focus on the gender aspects of human life. I have seeked, in the human activity literature, means for pointing at specific male or female aspects of life. I have not found such which to a certain extent must account for the lack of such discussions in this dissertation. Furthermore there is a problem of translation. The English translations of the work of Leontjew, as different from the Danish translations, use the terms 'man' and 'mens' work' instead of the less gender specific words 'human beings' and 'human work'.[56] The Danish translations are in general also much more readable, for which reason I have often found it necessary to re-translate terms from Danish into English.

A narrow interpretation of skills and competence in the work of Leontjew creates problems, because it leads to a definition of skills and competence as something possessed by the traditional craftsman. A wider interpretation –

that competence or skills have to do with the extent to which a human being master a certain practice, a social practice dealing with the relations between human beings, as well as a practice where the relations to artifacts and materials are important – also covers the type of skills that women are likely to possess. Furthermore I think that the approach is one which puts many female values into focus in design. The approach stresses that everyday life and practice is the origin for design, and that human needs are driving design, not technical problems or fixes.

In other ways I have of course also made interpretations of the work of Leontjew. Interpretations that are significantly different from other interpretations in literature: Leontjew, and his successors with him, argues that operations can be automated by a machine[57] – "it is generally the fate of operations that, sooner or later, they become a function of a machine".[58] In my interpretation this means that the human being through actions and operations activates a machine, whereas it is still up the human being to 'assess' the material conditions. Some authors take a different stance and use the statement in an attempt to develop artificial intelligence;[59] achieving what I find a conflicting purpose, because they end up in discussions similar to those of cognitive science: the discussions end up with how we can articulate as much as necessary of the material conditions for the operations, and thus make machines that take over, fully, human operations. This view is closely related to the tendency in Marxist thinking of considering so-called scientific or theoretical knowledge, i.e. breakdown knowledge, as superior or more profound that everyday knowledge or practice.[60] These are, in my view, both necessary if we want to design in line with existing practice, at the same time as we want the possibility of changing this.

I have only made one restriction on the application domain of the human activity approach in this chapter; that it deals with computer support for purposeful human work. It is, however, clear that the theoretical approach carries with it the assumption that computer support for purposeful human work are artifacts. Depending on how we interpret this, we can turn the assumption into a constraint of the application domain or a perspective on quality. The statement about the application domain is, that we only deal with computer applications that can be seen as artifacts for human work, i.e. there might be other computer applications for human work activity which are not intended to be artifacts. Turned into a statement about perspective we can say that we conceive human work activity as supported by artifacts, i.e. computer applications which are to function well in human work activity, must be applicable as artifacts.

To ponder the question of generality of the human activity approach we can negate the two statements and ask which computer applications are used in purposeful human work, but not to mediate the human being's relation to other human beings or to the physical world? Random number generators and different kinds of simulators are candidates because they produce some result without any direct connection to the material world. It is, however, difficult to come up with any concrete examples of this. In the different examples I can think of, the random number generator is hidden in some other application which is an artifact to its user, or in the simulator case, the purpose is to communicate some (physical) experiences from one person to another. As a paradox, such an example would be no problem, because this computer application, as an object, would need no user interface in the sense that this concept has been developed here.

In the following chapters I shall use and elaborate on this approach along two lines: in Chapter 4 we shall go into more detail about the user interface, to use the framework to characterize user interfaces and to reach some qualitative statements about user interfaces.

In Chapter 5 we shall use the approach to study how various design methods view the computer based artifact and more specifically the user interface. We shall see what types of design activities and user interfaces come out of the use of the design artifacts and techniques that the methods prescribe. There will be three main objects in Chapter 5: a framework by which to characterize design methods and their view of user interfaces; an assessment of the cognitive science tradition; and a discussion of the possibilities of a new and better design approach.

---

1 T. Winograd & C. F. Flores: **Understanding Computers and Cognition: A New Foundation for Design**, Ablex Publishing Comp. 1986.

2 MacPaint is the Apple Macintosh standard program for free-hand drawing (See Macintosh MacPaint, Apple Computers Inc. M1502).

3 H. Oberquelle et al.: **A view of human-machine communication and cooperation**, IJMMS 19, 4, 1983.

4 See e.g. P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984 or B. Shiel: **Power Tools for Programmers**, Datamation 29, 2, 1983.

5 Both 'communication' and 'interaction' indicate some kind of equality in communication between the parties involved. I would like to stress this connotation, and ideally choose a different word to cover what I am after. The concepts are, however, both well established concepts for "what goes on between the human being and the computer". Instead of bringing about a new concept, I prefer to use the term 'human-computer interaction' throughout this dissertation. And stress, that this does not, in my use, indicate any kind of 'a priori' resemblance with inter-human communication.

[6] W. M. Newman & R. F. Sproull: **Principles of Interactive Computer Graphics**, McGraw-Hill 1979.

[7] T. P. Moran: **The Command Language grammar. A representation for the user interface of interactive computer systems**, IJMMS 15, 1, 1981.

[8] I use the concept **artifact** to stress that computer applications are made by human beings, they are culture as opposed to nature. In my use this is synonymous with such words as tool, medium, device, means, although this meaning is not connected to the word artifact in its traditional meaning. The problems with such words as tool, medium, etc., are that they are used within different traditions (see Chapter 5), that they each are more specific than the word I need, and that they do not as directly implicate that tools, media, etc., are made by human beings. I shall use the concept computer application as the generic term for some collection of software and hardware which is applied together for some purpose. I will try to avoid the use of the term computer based system, unless I explicitly deal with a computer application which is designed from a systems perspective (see Chapter 5).

[9] At the same time as it is in line with the fundamental perspective of the dissertation, this choice entails that the results about user interfaces might not hold for casual use of computers, computers for leisure, computers in education, etc.

[10] A. N. Leontjew: **Problems of the Development of the Mind**, Progress Publishers 1981 or A. N. Leontjew: **Activity, Consciousness, and Personality**, Prentice-Hall 1978.

[11] L.-C. Hydén: **Psykologi och Materialism. Introduktion till den materialistiska psykologin**, Prisma 1981. (In Swedish. Psychology and Materialism. An Introduction to materialistic psychology).

[12] B. Karpatschof: **Grænsen for automatisering**, Psyke og Logos 2, 1984. (In Danish. The Limit of Automation).

[13] L. Wittgenstein: **Philosophical Investigations**, Oxford University Press 1953 and J. Lundequist: **Norm och Modell**, The Royal Institute of Technology, Stockholm 1982 (In Swedish. Norm and Model).

[14] M. Polanyi: **Personal Knowledge**, Rutledge & Kegan Paul 1967.

[15] Winograd & Flores, op. cit. (note 1).

[16] The intention of this section is to give the reader the knowledge of the framework to conceive the rest of the dissertation. A presentation like this cannot cover all aspects of a theory, for which reason the reader who want to know more will have to go back to my sources.

[17] Page make-up is the process in which newspaper pages are put together out of articles, ads, and pictures. For a description of page make-up see e.g. S. Bødker et al.: **A Utopian Experience** in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987.

18 We face the problem here that it is difficult to talk about operations: when we name them we almost always talk about their intentions, i.e. we do not talk about the operations but about the similar actions.

19 It is difficult to point at all the sources from which I have been inspired. Notes in this and the following chapters give specific references. My main courses of reference have, however, been Leontjew, op. cit. (note 10), Hydén, op. cit. (note 11), and Karpatschof, op. cit. (note 12).

20 U. Juul-Jensen: **Videnskabsteori 2**, Berlinske Forlag 1973 (In Danish. The Philosophy of Science), my translation.

21 Figure is inspired by Hydén, op. cit. (note 11).

22 Ibid.

23 The example used here is the overall example used in A. Goldberg: **SMALLTALK-80. The Interactive Programming Environment**, Addison-Wesley 1984. The example is presented on p. 10 pp. It is an example to illustrate how one can program a class in Smalltalk-80 with which to manage ones personal finances. In the example it is possible to enter transactions, keep track of these transactions, view the transactions in different ways by means of different Smalltalk views, etc.

24 Leontjew, op. cit. (note 10) calls this automation; a term that I try to avoid because of its connotations of 'automation by a computer' (see also note 57).

25 Leontjew ibid. In the terminology introduced by T. Winograd & C. F. Flores, op. cit. (note 1) they use the terms ready-to-hand and present-at-hand which they have adopted from Heidegger. The term breakdown is used to indicate the shift in which our practice and artifacts from being ready-to-hand becomes present-at-hand, similar to what Leontjew call conceptualization. I use the term breakdown only to indicate such shifts which are caused by some unpredicted conflict between the operation and its material conditions.

26 This does not just mean that we don't talk about them, but also, that we are not aware of them and distinguish between them.

27 L. Wittgenstein, op. cit. (note 13), and M. Polanyi, op. cit. (note 14) talk about **tacit knowledge**: what we can act according to but not talk about. Furthermore they make a distinction between what we usually don't talk about, and what we cannot talk about. They see a difference between our everyday competence, and scientific or theoretical competence, where everyday competence is more profound or "richer" than theoretical, because it includes non-articulable aspects through the relation to the material world.

Theory arises from our everyday competence, but it cannot explain the full "truth" about the activity of human beings and about the conditions for this. Here we find the reasons for my initial statement about the usability of a theory: *Applicability in use is the ultimate test of the theory.*

Furthermore, general design methods and recommendations about the user interface can be useful in design of the artifact. We can however not be certain of their usefulness before we have tried them out in the design activity. We can say that methods and recommendations can help us to get in the right direction as to avoid too many breakdowns in design, but we cannot be sure that we have made the right product before the users have tested it.

For a comparative discussion of different perspectives on the relation between practice and theoretical knowledge see P. Ehn: **Human Centered Design and Computer Artifacts**, Aarhus, forthcoming.

28 See for instance A. Kay & A. Goldberg: **A personal dynamic media** in A. I. Wasserman, ed.: **Software Development Environment**, IEEE 1981.

29 A. Goldberg op cit. (note 23).

30 These results come from some interviews that I did at Xerox PARC in 1982.

31 A. Goldberg op. cit. (note 23). Also A. Goldberg & D. Robson: **SMALLTALK-80. The Language and its Implementation**, Addison-Wesley 1983, and G. Krasner (ed.): **SMALLTALK-80. Bits of History. Words of Advice**, Addison-Wesley 1983.

32 This is a little simplified, because actions and operations towards objects can play a role in the communication too.

33 This discussion will be continued in Chapter 4 and 5. In Chapter 4 the focus will be on how actual artifacts do or do not support learning. In Chapter 5 we will look at how design methods deal with learning, and what the theory, on which this dissertation builds, can tell us about the handling of practice and learning in design.

34 Leontjew op. cit. (note 10), P. Y. Gal'perin: **Stages in the Development of Mental Acts**, in M. Cole & J. Maltzman, ed.: **A Handbook of Contemporary Soviet Psychology**, Basic Books 1969. Also H. & S. Dreyfus: **Mind over Machine**, The Free Press 1986, make some observations about the nature of learning of actions and operations. Their example will be discussed in Chapter 4.

35 Gal'perin, ibid.

36 Leontjew op. cit. (note 10), and Polanyi op. cit. (note 14).

37 B. Göranzon et al.: **Datorn som Verktyg**, Studentlitteratur 1983 (In Swedish. The Computer as a Tool).

38 Ibid.

39 This is what Oberquelle et al., op. cit. (note 3) call delegation with a word borrowed from Petri. Other persons and non-computer based artifacts can "stand in" for a person as well, secretaries or answering machines answering telephone calls, are examples of this.

40 See e.g. M. Stefik et al.: **WYSIWIS Revised. Early Experiences with Multi-User Interfaces**, in **Proceedings from the Conference on Computer-Supported Cooperative Work**, Austin, Texas 1986.

41 The physical aspects are the conditions for what Leontjew calls psychological and physiological **functions** which are the basic components of operations.

42 According to Webster's New World Dictionary, **handling** means to touch, lift, operate, etc. with the hand, to manage or control, but also to respond to control [e.g. the car handles well].

43 Another example of this is a hammer: we all know what a hammer is, and nobody will probably claim that the most natural characteristics of a hammer is that it consists of a piece of metal and a piece of wood.

44 It is true, however, that the software side is developing faster than the hardware side, which makes the physical aspects more stable than the other aspects, and, hence, the computers easier recognizable for us than say text editors.

45 See further examples in Chapter 4.

46 Winograd & Flores, op. cit. (note 1) or Ehn, op. cit. (note 27).

47 For a dissertation like this, we face the problem that the needs of managers do not necessarily include quality artifacts for the employees. We do, however, see more and more cases where quality is emphasized, and I see my contribution both in supporting such efforts with a theoretical foundation and with concrete design methods, and in pointing at prototypical cases where user interface design is done differently, and perhaps better.

48 See Winograd & Flores, op. cit. (note 1), H. & S. Dreyfus, op. cit. (note 34), L. Mathiassen: **Systemudvikling og Systemudviklingsmetode**, DAIMI PB-136, University of Aarhus 1981 (in Danish. Systems Development and Systems Development Method) or P. Ehn & Å. Sandberg: **God Utredning** in Å Sandberg, ed.: **Utredning och Förändring i förvaltningen**, Liber 1979 (In Swedish. Good Investigation).

49 This is of course only true as long as we stay within the application domain of the theoretical approach. Design of a computer game, or of a piece of artwork might not be covered by this conception of design.

50 I aim at using the word designers for everybody who take part in design, a user, a computer expert or some other expert. This is difficult, however, because we often talk about professional designers such as computer experts, as designers too. I hope that it is clear to the reader when I use which meaning of the word.

51 In the Utopia project it proved valuable not only to look at the way page make-up is done with the latest generation of technology, but also how it was done with the technologies of the generations before.

52 See Mathiassen, op. cit. (note 48) or N. E. Andersen et al.: **Professionel Systemudvikling,** Teknisk Forlag 1986 (In Danish. Professionel Systems Development).

53 According to my discussion of practice, and to avoid confusion about concepts I find no need to distinguish between what is a design artifact, a suggestion for a technique to be applied, or a principle of organization, or to distinguish between methods which claims to cover all aspects of design, or fragments covering only a specific activity – for all of these I will use the term **method** from now on, even though I like the reader to remember that there is a correspondence between **artifacts, techniques,** and **principles of organization** on the one hand and the different aspects of practice on the other.

54 See e.g. Andersen, ibid.

55 See e.g. L. Suchman & E. Wynn: **Procedures and Problems in the Office,** Office: Technology and People, 2, 1984.

56 According to the quotes of Ehn, op. cit. (note 27), this seems to be a problem also with translations of the works of other Marxist writers, e.g. Israel and Kosik.

57 What I have called operationalization here is in the literature used as synonymous to automation.

58 A. N. Leontjew: **The Problem of Activity in Psychology,** in J. V. Wertsch, ed.: **The Concept of Activity in Soviet Psychology,** M. E. Sharpe Inc. 1981 p. 64.

59 O.K. Tikhomirov: **The Psychological Consequences of Computation** in J. V. Wertsch, ibid.

60 See discussions in Ehn, op. cit. (note 27).

Chapter 3

User Interface Design – the Empirical Cases

User interface design takes place in a number of different activities in design of computer applications. Decisions made early in the design process, in design activities with other goals than user interface design, will constraint the possibilities of user interface design. An example of this is decisions early in design, about the choice of a 4'th generation tool to be applied in the design activity. Lacking acknowledgment of this has resulted in design methods which prescribe user interface design to take place as a single isolated activity, late in the design process and often conducted by a different group of designers that the rest.[1] In this chapter I will discuss different examples from my empirical cases, of activities where user interface design were taking place. I deal with design activities which aim at user interface design; not necessarily with activities the aims of which are to construct the user interface, but activities where a major part of the effort is directed towards investigation or construction of the appearance of the computer application in use. These activities are collective activities, which in turn are parts of the total collective activity constituting **design** of a specific computer application and its environment.

In Chapter 2, I discussed user interface design from a theoretical point of view. It is obvious that one doesn't start to work on a theoretical approach like the human activity approach without good reasons. The best reason for me was that I felt a discrepancy between my practical experiences with user interface design and the theoretical explanations and methods in literature – why could I not find explanations to my empirical problems and solutions in literature? and why did methods prescribe techniques that didn't seem to work from the perspective of the empirical findings?

In this chapter, I will present some of these experiences. First of all, I hope that this presentation will help my readers understand more about my reasons for choosing the theoretical framework. Secondly, I will use the empirical cases as examples in the discussions in the following chapters. The reader is wondering, perhaps, why I have chosen to present the theoretical approach before its empirical background. This is done because the presentation is not only meant as a detached presentation of the empirical background, but also as a demonstration of the use of the human activity framework in the structuring of the discussion of design activities and design practice.

The chapter will start with a discussion of the common properties of the design activities of interest. Following this, six design situations, originating from the Utopia project, Aarhus Polytechnics and Xerox PARC will be presented. Finally, a short discussion of differences and similarities is given.

Design activities – the common properties

Common to the projects to be discussed is that they make use of a prototyping design strategy where the users take part in constructing and evaluating the prototypes. All the projects view the design activity as a learning activity, and they focus not only on design of the user interface, but on possible/necessary changes of practice in general as well.[2]

The projects share a, not theoretically founded, idea of making better user interfaces by involving users, and by allowing for the users to try out the user interface in prototypical use situations. In none of the projects did user interface design take place as a single, isolated activity. This means that when I present design activities in the following, I focus on the user interface design aspects, although other things were going on at the same time. For instance, demonstrating alternatives also means to touch upon the change of organization of work.

The projects, on the other hand, are different when it comes to the organizational conditions of the projects: the interests, power, and practice of the involved groups, the resources available for the projects as well as the groups; professional designers as well as users.

The domain of use for the artifacts differ, as well as the way the artifacts are intended to support the communicative and the instrumental side of human work, respectively; and the way it is to be used in individual respectively collective work.

Finally, the six examples employ different design methods, partially due to different aims of the activities: to discuss alternatives, to construct specific aspects of the user interface, etc. It is important for the discussions in the following chapters to find out about the relation between the methods employed, and the possibilities of focusing on different aspects of the user interface.

The Utopia project

It is the basic idea of design in the Utopia project that what is designed are **tools** for competent graphic work[3]:

"The design strategy of the Utopia project is based on a *tool perspective*. The computer support is designed as a collection of tools for the skilled worker to use. The tool perspective takes the *work process* as its origin rather than data or information flow. This means: *not* detailed analysis, description and formalization of qualifications *but* development of professional education based on the skills of professionals; *not* information flow analysis and systems description *but* specification of tools."

This means that the artifacts are primarily developed to mediate the actions and operations of individual human beings towards things.[4] The aims are to originate from the practice of the graphic workers, and design artifacts which are in line with the traditional ones.

An equally important part of the project was research on organization of the work process, including coordination and communication, to supplement the development of the individual tools.

The conditions of the design activities of the Utopia project can be summarized as follows: The project was sponsored by the Nordic Graphic Workers' Union, conducted by graphic workers and researchers with the purpose of providing new technological alternatives for competent graphic workers to use. There was no direct relation to a specific organization of use. Due to the researchers' background the project had access to a wide variety of design methods on different levels and focusing on different aspects of the future use activity. However, access to computer equipment was limited by what can be seen as either economic resources or commercially available technology.

Design of user interfaces was seen in the project as something done integrated with design of **use models.**[5] the main activity when developing computer applications together with competent users.

"Use models are based on the traditional concepts of the domain, but enhanced with concepts necessary to understand new possibilities and restrictions imposed by computer technology. Such models are useful as means to support design of both functionality and user interface. In education they support activities aiming at creation of conceptual competence. In use they support the user by making it possible to filter away technical distortion, i.e. to focus the awareness on the materials and products."[6]

In the following I shall present three design activities from the Utopia project.

Mock-up with paper

This design activity took place quite early in the design process, after initial investigations and some unsuccessful experiments with the use of traditional description methods. The purpose was to analyze some possible alternative solutions to how a newspaper page can be displayed, and worked on, on a screen. This analysis was to be conducted by graphic workers and researchers together.

The principle behind the mock-up is simple. Using sheets of paper, match-boxes, some chipboard and the like, one "builds" a workstation with a "high-resolution display", a "tablet", a "puck", etc. The process in which this equipment was used, is one in which page make-up was done simultaneously with the creation of the needed screen images drawn on paper. The graphic worker and the researcher worked together: The graphic worker did the make-up step by step. For each step the corresponding screen image was drawn on paper. The product was a series of snapshots simulating the make-up process done while using the workstation. This series illustrated aspects such as: What material is shown? What other information is needed? Do we need a menu, and where should we put it?[7]

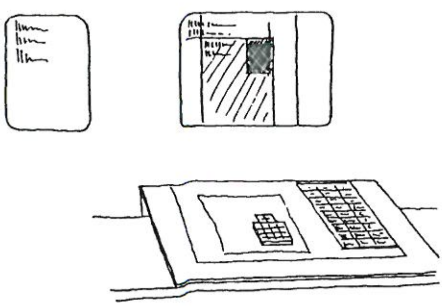


Figure 1. Paper based mock-up

The researcher took part in the process by pointing out possibilities and limitations of the corresponding "real" equipment: How much information can be shown on the screen? How can one use the puck for moving things around on the screen? Where can menus be placed and how are different kinds of menus used? etc.

Besides from experimenting with the screen images, it was possible to experiment with the interaction devices using mock-ups; the 'traditional' ones such as tablet, puck and mouse, but it was also possible to 'invent' and simulate various kinds of interaction devices, new ones as well as well-known ones, and try out their use for specific purposes. The following types of questions could then be discussed: Do we need a tablet with puck or a stylus? Will a mouse do? It was possible to try out various designs of the mouse or puck. Where to place the buttons? How many buttons?

This way, it was possible for the graphic workers to take active part in the design process. The method was very cheap as no expensive equipment or time-consuming programming was needed.

New features of the 'workstation' could be developed and added as they were needed. A major advantage of this method is that it does not limit its users to experiments with available equipment. Both equipment that just does not happen to be at hand, for economic reasons, and future computer equipment, can be simulated. One can play with super high resolution display screens large enough to show a real size newspaper page, etc. One drawback of this method is that the picture drawing on paper is time and space consuming. A totally new screen image has to be drawn for each snapshot resulting in a large and complex collection of drawings. Furthermore, aspects related to time are difficult to capture.

When the series of snapshots had been produced, the make-up process could be done over and over again "replaying" the series of snapshots. This replaying was quite cumbersome using the drawings. Also, the users were limited to redoing the same work process as initially constructed, unless additional drawings were made. For this reason the prototype helped illustrating overall principles and let the user try out the limits of the design so far. However, it could not be used to find out to what extent the new artifact would help the user build up a repertoire of operations – there was no ways of trying out what operations would be triggered by the meeting with the material conditions; only to demonstrate one or more examples of which actions/operations could be used to achieve a certain goal.

As a result of the activity, experiences were gained concerning such issues as the physical possibilities of representing graphic material, e.g. text and pictures, on a computer display. It was possible to deal with the object directed aspects and the handling aspects in two ways, which are helpful, but not sufficient in design: by letting the graphic workers go through the simulation they could find out to what extent their 'old' practice was applicable with the new artifact, thus bringing focus on this; and it was possible to deal with the static parts of the user interface: how the page could be displayed on the screen, and so on. To go further, more advanced design methods were needed.

Mock-up with color slides

The color slide mock-ups are the immediate results of this need. As the project came to focus on details, also of the user interface, mock-ups that were more realistic replica of computer equipment were needed, especially to focus on the physical aspects.

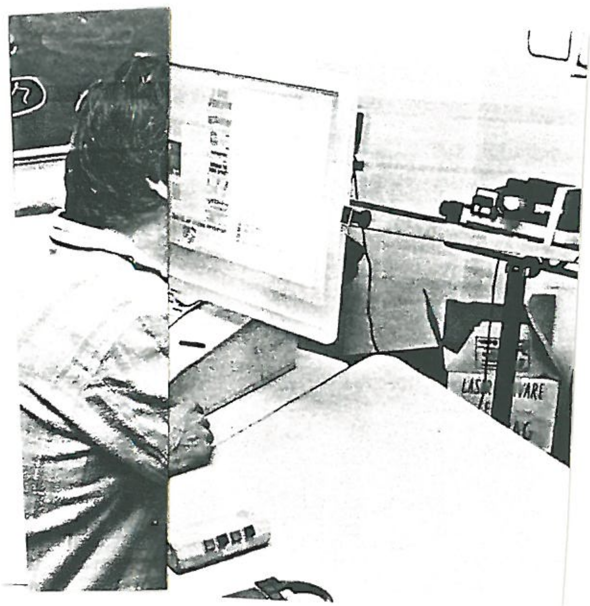


Figure 2. Mock-up with color slides

The principles of this mock-up are the same as those of the mock-up with paper but the "user friendliness" of the method has been improved using a camera and a slide projector.

The creation process was basically the same, but the drawing of screen images was replaced by snapshots on color slides. This made the production process much more efficient as parts of the screen image and other material could be re-used both when progressing the make-up process and when creating alternatives.

However, what was really gained compared to the mock-up with paper showed up when replaying the snapshots. A workstation with back-screen projection instead of display screens, equipped with "tablet", "puck", etc., was built. Here the graphic worker could sit down and redo the work process. It was still possible only to replay the sequences as they were initially constructed. But, as alternatives were easily constructed, and the sequences easily replayed, it was possible to let a number of graphic workers try out alternative work processes using the prototype. This way, it was possible to focus on their need to change practice, especially actions/operations, when applying the future application, and on their evaluation of the prospects of such a change.

Because this equipment resembled the real computer equipment, it was possible to conduct experiments on the physical aspects of the user interface, and for instance try out workstations for seated work and for standing. This meant improved possibilities for examining more details about the static parts of the handling and object directed aspects of finding out which are the different objects to be handled and especially consider 'normal' situations, because the different paths of the simulation sequence were still limited. What was still more difficult to try out was time dependant features such as dragging: how to drag a text across the screen – is it necessary/realistic to be able to read the text while dragging?

Graphical workstations

There were, as mentioned, still other important aspects of the computer application for page make-up that could not be explored using mock-ups. Some of these concerned the connection between the movements of the pointing device and movements of the cursor, such as the use of a puck for selection of text or moving (dragging) material, the use of various sorts of positioning aids such as gridding and gravitation, but also aspects of coping with the difference between the resolution of the screen and of the photo typesetter.

Experiments in relation to these issues were done on a PERQ workstation. The main problem of this was that the workstation provided a very restricted prototyping environment with no specific support for design of page make-up applications. Consequently, each experiment required a major programming effort. I will, in the following, discuss two experiments concerning the design of the user interface.

The first experiment was intended to exploit the possibilities of selecting text, moving it, and placing it on some sort of page ground as operations. It was hoped to answer questions such as: Which way is the best to select text? How is the selection indicated? Can moving be done by entering or pointing at the destination or by fetching and dragging the material? If dragging is chosen, can text be moved as a box or do we need to be able to read the text during the move? When placing the text: is it possible to place the text in the exact point or is some sort of gridding or gravitation needed? Which is preferred? This experiment related both to the handling aspects, and to object directed aspects, such as, can the graphic worker place text according to rules for typographical quality, such as aligning with pictures or columns?

The second experiment was directed towards the use of different kinds of menus and different ways of showing and changing status. The pictures of figure 3 and 4 illustrate two alternative interaction designs. The first example, figure 3, involved menus on the display screen, and changes in type size, column width, etc., were accomplished by entering numerical values from the keyboard. The second example, figure 4, involved tablet menus together with a kind of rulers which permitted analogous changes in numerical values by means of the buttons on the puck.

In the first of these examples the menus permanently occupied space on the display screen which entailed that the actual working area was smaller than is the case in the second example, where the menus did not occupy any space on the screen. However, in the first example the screen constantly displayed status information, like type size and column width as values. In the second example this information was only visible when the user changed the values.

This setting allowed for experiments with different features of the handling aspects: with the use of fixed menus or pop-up menus, display screen menus or tablet menus; alternatives, such as analogous way of changing status and a digital way (editing of fields containing the status information). It was, furthermore, possible to experiment with the selection sequence of operator and operant(s) and with various kinds of feedback from the computer.

The physical, but also the handling aspects were different in the two examples. In the first, where screen menus were applied, the graphic worker could, physically, concentrate on the display screen, whereas in the second, where tablet menus were used, it was necessary to alternately look at the screen and at the tablet.[8]

The graphic workers participated in the design of the experiments and were the ones to try out the prototypes. This way they could try out the user interface in a prototypical setting. These experiences resulted in changes of the prototypes, and also in a better understanding of what are the possibilities for alternatives, undo facilities, and the limitations for instance concerning the screen resolution of computers. The mere fact that the graphic workers had used a computer, made simulation experiments with the mock-up much more realistic to them, because they could easier imagine how it was to move something on the screen, which was not really done in the simulation situation.

The major drawback of the prototyping, as described here, is the lack of a suitable prototyping environment. Due to this, the experiments were very restricted and it was, at times, difficult for the graphical workers to relate the prototypes and the use of them to "real-life" make-up work. Better prototyping environments would definitely make it easier to make the prototypes look more realistic even though the aspects to be tried out would be the same.

Figure 3. A prototype with screen menus

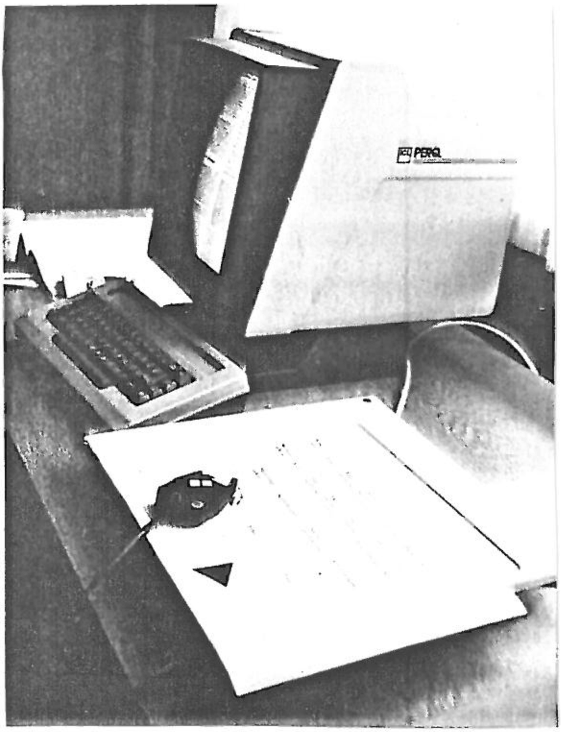


Figure 4. A prototype with tablet menus

General observations

Each of the design methods contributed to the common understanding in the project of *how* page make-up could and should be conducted when applying a computer based artifact, including requirements to and possibilities of a user interface of a page make-up application. The prototypes were supplementing each other in setting focus on different possibilities and constraints of the application, including the user interface, to the graphic workers.

In the process the graphic workers started to use the prototypes based on their previous practice and on the mutual learning processes, preceeding the design and evaluation of the prototypes. A series of steps was taken in which new breakdowns were encountered, evaluated, and eventually the prototype was changed or an alternative rejected. Through the design methods very concrete prototypes of different aspects of the use of the future tools were established. These prototypes allowed for evaluation of the user interface through the skilled workers' hands-on experiences.

The concrete results of this design process are discussed by Sundblad and others.[9] Some results point at parts of the user interface which we did differently from more traditional design of user interfaces for page make-up systems:

1. To master assessment of a page when the page cannot be shown in its full size, requires a scale of a (very) limited number of distinct reduction/ magnification sizes.

2. Bodytext must be at least 18 pixels high (Åp) to be readable. Text must be displayed in reduced sizes as gray scale patterns, not as algorithmic reductions of the text.

3. Placing an article on the page ground can be done by a new kind of tool, the ruler, a way of letting the text "float" into the empty space on the page under direct control of the user.

The Aarhus Polytechnics

In the Aarhus Polytechnics case the design group consisted of all the women working in the journal office, a number of selected case workers from different parts of the organization, a social scientist and three computer scientists. The project was supervised by a technology committee, according to the technology agreement, a local extension of the central agreement between the office workers' union (HK) and the institutions of the Danish state.[10] This agreement makes use of a consensus strategy, where the technology committee must aim at reaching an agreement, and only in case of failure will real negotiations between the parties come into play.

From the beginning, the group worked with three different alternatives:

- a restructuring of the existing paper file without use of computers,
- a restructuring of the paper file with computer support for retrieval of documents and computer based mail lists to inform the case workers about incoming mail, and
- a computer based journal where all documents are scanned in upon arrival in the journal office, and with computer based retrieval and mail lists.

These alternatives came into being partly as a result of management's wishes, and partly as examples from the range of possible solutions that the group encountered in the early discussions about needs, wishes and technical possibilities. In this process the group, among other activities, had demonstrations of various journal systems, scanners, and other relevant equipment, available on the market. The project dealt with support for communication within the group, with support for individual activity as well as collective: a case, and with this a journal, is a collective artifact because it makes it possible to collect and coordinate contributions and actions of the different case workers who hold the case throughout time. It is, however, also an individual artifact because it can be used by the individual case worker to take actions concerning the contents of the case in a specific situation.

In the following, I shall discuss the specific prerequisites, contents, and results of two different design activities of this process: A situation where scenarios were used and one where 4'th generation tools were applied.[11]

Scenarios

Scenarios of the use of the three alternative types of files were outlined by the computer and social scientists in order to intensify the discussions and make them more concrete. The scenarios were focusing on the overall actions to be conducted, their sequence, preconditions, purpose and results; the intentions was dealt with as well as some operational aspects, especially by focusing on the difference between the old practice of how documents are handled, and the need for a new practice in relation to this, organization of work, and the physical devices of the computer application. The scenarios consisted of plain text descriptions in which each step in typical daily journal work with the new technology was described.

The background of using the scenarios was a big difference between the interests, language, and focus of the involved parties, which made the discussions stand still for a long time. Much time in the group was spent discussing if a paper based journal was needed in the future, and how this should be organized. Furthermore, it was a key issue for the women in the journal office whether they needed to use a document scanner or not. Much time was also spent on discussing the communication between the case workers and the journal office. In the traditional practice, it was up to the journal office to know the organization well enough to be able to distribute the right information to the right people. This caused the problem that the case workers, as a precaution, got much information that they didn't need, at the same time as they felt that they lacked other information. The scenarios were intended to break this barrier and presented, in the same language and style, the three different technical solutions and work activities around them.

Based on these scenarios, it was possible for the participants to discuss more freely and to deal with specific details, because it became possible for them to separate technical possibilities from needs and wishes, and they got a vocabulary which helped in the discussions and comparison of the suggestions: for instance, it became possible for the group to discuss the detailed pros and cons of the suggestion for a computer based journal, whereas before, the women from the journal office had rejected to discuss this, due to their resistance against the use of a document scanner.

Example

This example illustrates the use of scenarios in the discussion of "computer based journal or not". The specific issue is whether the qualities of paper could be reproduced on the computer terminal and in which cases the group could see a need to save paper originals. The example also illustrates that the aspects of the user interface were addressed in many ways, directly or indirectly, at this early stage of the design process.

The scenario of the computer based journal reads like this: "In the computer based journal, all 'documents' are stored in the computer. (...) The 'paper mail' must be scanned into the computer in order to be filed. Afterwards it can be distributed via electronic mail."

About the journal office: "The women start their day with the 'opening' of mail. The electronic mail must be removed from a queue, journalized, and distributed. The paper based letters are sorted, and the letters which are to be journalized are scanned in and distributed."

About the case work: "When a case is retrieved via the computer, the computer presents information about the case, which compares to the information on the folder of paper based files. It is possible to search for information in the documents of the case or to select documents for reading."

This description gave rise to, among others, the following remarks in the working group and with management: "The letters often contain drawings and similar original material to be used by case workers. It would be quite irritating to have the letter sent by electronic mail and the drawings, etc., in an envelope."[12]

The legal questions of scanning drawings, and other official documents, were raised. Furthermore it was discussed whether one could preserve the differences between letters, such as paper color and logos, or whether the letters would all look alike on the screen.

Experiences showed that the scenarios can be used early in the process to reach a mutual understanding of what are the key problems concerning the user interface. It is possible to set focus on problems with the present practice this way, but it is only possible to address issues about the future practice that the participants already have some awareness about: e.g. the participants in this case had already experienced that the qualities of paper mail (color, etc.) were not present when they prepared texts with their computer equipment.

The next step in the process was to address the user interface issues in more detail for the computer supported journal, the solution which the group chose to work with.

4'th generation tools

An initial paper based sequence of screen image simulations was made by some of the computer scientists. Regrettably the future users did not have enough time to participate directly in this work.[13] The screen images were discussed with the working group: which information was needed about each letter, how should it be structured, how to get from one screen image to another, etc. On the basis of these discussions, the computer scientists built a prototype by means of a 4'th generation tool.[14] The first versions of this were used in demonstrations as part of the ongoing discussions. As the versions of the prototype got more stable they were also used in the real use setting: the women in the journal office used the prototype to create mail lists. Programming the prototype required much programming experience because much of what the group wanted, concerning the cursor movements and the like, could not be programmed directly in the 4'th generation tool, but had to be programmed in Pascal.

The 4'th generation tool allowed the group to run the prototype on the computer that was also used for other purposes in the office. On the one hand, choosing to use the existing terminals meant restrictions to the user interface; on the other hand all future users had easy access to the prototype. The existing terminals would have been severe restrictions if the group had decided to pursue the most ambitious solution: the computer based journal. The women wanted to keep the characteristics of paper, the font, logos, and the like which were helpful in distinguishing and retrieving documents of paper; an aspect which seemed hard to support, especially since it was obvious that there would be a need to be able to handle the documents as text too; e.g. to do key word search.[15]

The aspects which could be illustrated had to do with how mail lists worked: which information was needed, who was going to enter this, how could the information be presented to the user, and so on. This prototype allowed for experiments with both the physical aspects, the handling aspects, and the subject/object directed aspects of the prototype, but within very constrained frames.

The main conclusion of this process is, that it was difficult to involve the users actively, because the programming effort was too big for them to spend time on. The 4'th generation tool used was too limited in the facilities provided. It was possible to focus on all aspects of the user interface, but especially the physical aspects were difficult to change. The weaknesses of scenarios in general are discussed in Chapter 2, in this case they served the purpose of getting discussions going quite well.

Xerox PARC

In this example we shall look at activities as part of the development of the Smalltalk-80 system. The goal of these activities to create some "application part" of the Smalltalk-80 system, not, for instance, parts to handle virtual memory, scheduling or other basic primitives. The researchers who did the design had a long experience as Smalltalk programmers. Some had formal computer science training, some had other kinds of background. They developed the systems for their own use. It is characteristic that the design of the Smalltalk-80 system was embedded in an unusually long tradition of both design and of use, which can be seen as a strength, at the same time as this practice becomes very hard to transcend.

The applications were developed, primarily, to support the individual activity, the instrumental side, but did also include communication and coordination facilities. The Smalltalk-80 system as such played an important role as constraint to the situation – the researchers were capable of introducing new software and hardware facilities if such were needed. They could change all the rules of the system if they wanted, but there were obvious advantages for the possibilities of applying their repertoire of actions and operations in use, of keeping the user interface within certain frames to make it as uniform as possible.

The principle of the design practice is to start out from an idea of how the parts are to look on the display screen, and other aspects of the future use. The next step was to pick the right objects from the system; objects which had some of the needed properties, and modify these objects into what was wanted. The process is very experimental: to start out with crude fragments of the wanted user interface, try it out, improve it, try it out, and so on.

This can be illustrated by an example: Assume, that I want to develop a graphic representation of the figures of the financial history example discussed in Chapter 2.[16] I want a user interface where the user can enter the transactions of the financial history (the money spent and received, and the reasons for this), and see a graphical representation of the expenditures (a pie chart or a bar chart) together with a curve of the cash on hand. This graphical representation must be updated after each transaction.

I can, of course, start out by deciding how we want the expenditures displayed, and write some code for this. But I can also browse through the system to look for different representations which are part of the system for other purposes. Perhaps I find a bar chart somewhere. I can try to use this with the values I want displayed. Perhaps this is not really what I want, and I might want to change the layout of how the values are displayed by editing the code; or choose a different representation. In the first try, I might want transactions entered just by typing numbers in a field. For this purpose I can use an instance of the class called 'FillInTheBlank' which allows for typing and editing of values. Later on, when I have tried how this works, I might change this because I want to enter the transactions by pointing at numbers instead of typing, or by dragging or pushing the bars in the bar chart. Again, I can browse through the system to pick components that are helpful to me, but I may also need to write some code of my own.

This design practice ensures a slow evolution of the user interface, at the same time as it makes it easy to design user interface parts which are consistent with the rest of the user interface. What cannot be seen from the above description, is how the physical aspects, except the "screen look", can be dealt with in design. Due to the possibilities of technical assistance, it was however possible to experiment also with other input devices, etc., although this was of course much more complicated than the use of devices that were at hand.

Discussion

It is remarkable that the most distinct differences between the three projects relate to organizational oriented issues such as the difference in practice between the involved groups, the connection to a specific use situation, and the available resources:

In the Utopia project there was, throughout the design process, but especially in the beginning, a big difference between the practice of the involved groups. This stressed the needs for a process of mutual learning. The goals of the project related both to the project as a trade union conducted design project, and as a research project, which created some potential conflicts. In the Aarhus Polytechnics case, there were similar problems of differences between the groups of users, and between the users on the one hand and the computer and social scientists on the other, the latter also because these were, from the users' side, supposed to represent the goals of management. In the Smalltalk case, the learning is different than that of the other cases, it is a long term socialization into the practice of the group of designers/users. In the other cases, the learning is something going on due to the specific, immediate design activity.

In the Aarhus Polytechnics case, we can see the biggest differences to the other two cases in the way that **resources** were distributed among the involved groups. The journal office workers and the case workers did not get any relief from their daily work, while participating in the design activity. And they did not have the possibility of bringing in their own consultants or the like. This was important because the project was initiated by management – much **power** was, directly or indirectly, in the hands of management. At the same time the three projects have important **goals** in common concerning the development of high-quality products for and with the users. Resources were present for this, and the practice of the design groups were developed in this direction.

In the Aarhus Polytechnics case, the project dealt with a development of a computer based journal for the specific organization. The future users were directly involved in the process which was initiated by management. In the Utopia case, the project was governed by the trade unions to provide alternatives which were intended to be applied at different newspaper plants. The involved users were only representatives of the future users. The Smalltalk case, again, is different by its almost total integration of design and use.

The different conditions for the properties of the user interface are seen in the differences in the connections of the projects to a given technology, and the quality of this technology. The projects did, to different extents, focus differently of the artifacts to be developed – whether they were mediating instrumental or communicative activity, whether they were artifacts for individual use or collective use; and on different aspects of the user interface.

The six design activities described, focussed differently on the different aspects of the user interface. In both the Utopia and the Aarhus Polytechnics case it was important to preserve many of the properties of the objects (the newspaper page/the document) when this was to be worked on through the computer application. There was a need in all the cases to experiment with all three aspects of the user interface, although the applied methods allowed differently for this. It was characteristic that the more details it was necessary to get into, the more inflexible in suggesting different types of user interfaces were the design methods, and vice versa. Sketching alternatives were possible in all the cases, but different types of alternatives. Also it seems to be one thing to create breakdowns about current practice, but something much more complicated to create an idea of the repertoire of actions and operations to be applied in the future use situation. Scenarios and primitive prototypes can be used for the former, but are not sufficient for the latter. Only the Smalltalk example was any near a situation where it was possible to come somewhat close to the future use situation in trying out the prototype. In the other examples it was only possible to demonstrate some aspects.

In the following chapters I shall look into both the properties of the user interface, and the relation between these, the design activity, and the design methods applied.

---

1 See e.g. D. A. Norman & S. W. Draper, ed.: **User Centered System Design**, Lawrence Erlbaum 1986, or the discussions in Chapter 5 of the approaches of Yourdon (E. Yourdon: **Managing the Systems Life Cycle**, Yourdon Press 1982) and Card, Moran & Newell (S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum 1983).

2 For the Smalltalk-80 case this goal was not explicit, even though the considerations practically involved this kind of thinking for a specific practice: Smalltalk-80 programming. Examples of the consideration of practice can be found in connection with design of support for programmers to communicate while using of the system, and of support for coordination of the programming activity.

3 The Utopia project has been presented in the introduction of this dissertation. In Chapters 5 and 6 I shall examine the impacts of the tools perspective on the user interface and on design. This quote is taken from S. Bødker et al.: **A Utopian Experience**, in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987. The tool perspective is outlined in P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984.

4 This does not mean that the collective level was not considered in design, only that the parts of the project in focus here did not aim at producing artifacts for this level.

5 Quoted from Bødker et al., op. cit. (note 3). See also discussion in Chapter 5.

6 Ibid. or Ehn & Kyng, op. cit. (note 3).

7 In the Xerox STAR project, the painting program MARK-UP was used in a similar way to draw sketches of the screen images. According to W. Newman (EuroGraphics 84) STAR would never have come into being without MARK-UP.

8 P. Ehn et al.: **Utformning av Datorstödd Ombrytning för Dagstidningar**, UTOPIA report no. 12, 1984 (In Swedish. Computer-aided Page Make-up for Newspapers).

9 Examples of these discussions can be found in several papers of Yngve Sundblad, Staffan Romberger and other of the Utopia project members from the Royal Institute of Technology. See e.g. Y. Sundblad, ed.: **Quality and**

Interaction in Computer-aided Graphic Design, UTOPIA report no. 15, 1987 or Ehn et al., ibid.

10 An extensive analysis of this agreement and its use, as analyzed from the point of view of the local HK union, can be found in B. H. Kristensen et al.: **Retningslinier for valg at faglige strategier på kontorområdet – et case studie over Århus tekniske Skoles kontorautomatiseringsprojekt**, Department of Computer Science, University of Aarhus 1986 (In Danish. Guidelines for trade union strategies in the office area – a case study of the office automation project of the Aarhus School of Polytechnics).

11 Also this specific situation is discussed by Kristensen et al., ibid.

12 Quoted from my own notes from the work in the group.

13 Kristensen et al., op. cit. (note 10).

14 Oracle is marketed in Denmark by Oracle Corporation Europe.

15 At the same time as the users started to use the prototypes, management got more and more dissatisfied with the work of the group, primarily because the group had chosen not to work on the solution preferred by management: the computer based journal. When the prototype had been in use for a while, management declared this to be a system to be used in daily work; something that was never intended by the group and to which purpose the prototype wasn't fit.

16 The result of this process could be the pieces of program presented by A. Goldberg & D. Robson: **Smalltalk-80. The Language and its Implementation**, Addison-Wesley 1983 and A. Goldberg: **SMALLTALK-80. The Interactive Programming Environment**, Addison-Wesley 1984. The described process is, however, purely hypothetic.

Chapter 4

User Interfaces

A general framework, as presented in Chapter 2, is a first step in the direction of a wider conception of what user interfaces are, and which role they play in use of a computer based artifact. It is, however, more like a new pair of glasses, which can reveal new angles and details to us, and thus, sharpen our curiosity. This curiosity can make us move in many directions where we can seek answers to different types of questions. Many of these questions have perhaps been asked before, but without satisfactory answers.

In this chapter I shall try to use this "new pair of glasses" – the human activity approach – to discuss different kinds of questions, to which I claim that we can find more satisfactory answers. This does not mean that the framework gives new, better or complete answers to all our questions about the user interface, and it is only possible to discuss a limited number of questions in a chapter like this.

The discussions in this chapter are prototypical examples of both how we can elaborate on the human activities approach about user interfaces, and of how we can work with different aspects of the phenomena which we call user interfaces and human-computer interaction.

Furthermore, it is important to understand what are the consequences of the new framework for more traditional conceptions of these phenomena. Will we have to throw all existing general recommendations away? Or can we make use of them, perhaps differently, with the human activity framework?

Many relevant questions can be asked, and I have chosen to discuss a few of these. The choice of questions has been influenced by the needs that I have found when teaching human-computer interaction, and by people around me who have challenged me to deal with specific issues. This chapter primarily seeks explanations rather than recommendations. Recommendations concluding my discussions will be given in Chapter 6.

The basic role of the user interface is to support the user in acting on an object or with a subject through the artifact. This means that the user interface can often better be discussed negatively: when does the user interface prevent the user from carrying out the intended actions, and in what ways? How do the different aspects of the user interface support or prevent different actions and operations on a specific subject or object?

This chapter will start with such a discussion, and after this, the following will be discussed:

- What is the relation between the competence of the users and the user interface?

- How can we discuss and explain differences between two user interfaces?

- How can the human activity approach be used to discuss various more traditional approaches?

- How can the widespread notions of natural language interfaces be dealt with, and how can the relation between the human-computer interaction and language be conceived?

- What are the relations to the more technical possibilities of designing the user interface. In design, what are the technical possibilities of influencing the different aspects of the user interface?

The role of the user interface

The user interface is influencing which objects and subjects we can focus our actions on while applying the artifact: the computer as a collection of buttons, the artifact as an object, other objects e.g. text documents, and so on. A good user interface will allow the user to focus on the objects or subjects that the user intend to work with. A bad user interface, on the other hand, will perhaps force the user to focus on other objects and subjects than the intended. Reflections like this can of course be used to assess a specific user interface in a specific use activity. It is, however, more interesting to be able to talk about the user interface more generally, and even give general design recommendations. The aims of the following sections are to find out how the user interface influences shifts in level of actions: can we identify more specific kinds of shifts, and can we see the role of the user interface in such shifts?

Shifts of object/subject take place in learning situations as well as in more routine use situations, with the only difference that breakdowns and creation of new operations are more frequent in learning situations than in routine use. In this section, the idea is just to identify different situations of shifts between actions and operations, which are also shifts between different subjects and objects. These shifts are caused by various material conditions some of which are due to the artifact directly in a negative way: The physical aspects or the handling aspects can prevent the user from being conscious only of the real objects and subjects or the subject/object directed aspects can prevent the user from performing some parts of the activity towards the subjects/objects as operations. The artifact can also cause shifts in a positive way because the same aspects support such shifts.

Derived from the definition of the role of the user interface in use, the aims of a user interface are to support the intended or operationalized shifts in relation to the use activity, and to prevent the ones which are not intended.

In the following I shall distinguish three kinds of situations:

- shift of focus between different objects/subjects other than the artifact (and between aspects of these),
- shift of focus to/from the artifact as an object,
- shift of focus with the artifact as object.

Shift of focus between subjects/objects

The type of shifts that I deal with here, are constrained by the subject/object directed aspects of the user interface.

Modes

The most obvious example of what can prevent an intended shift of focus from one subject/object to another is that the parts of the application focusing on one subject or object more or less segregated from other parts, focusing on other subjects or objects. These can be separate applications, running on the same computer. Also modes, and perhaps inconsistency of style of interaction, effect of commands, different icons, etc., can contribute to this segregation. Modeless interaction and integration of the parts of the user interface which in particular support actions and operations towards specific objects, can of course not, as such, prevent unintended shifts of object/subject.

In the following, I will present examples where the subjects/objects are present only through the artifact. For the use of artifacts towards physically present subjects/objects, the discussion is different because shifts depend on the actual physical presence of these subjects/objects.

Imagine a traditional text editor where '$e' means 'exit' in the text mode but 'erase' in command mode. This is one of the problems with modes: a specific command means something and is issued one way in one mode, but means and works differently in another mode.

The Macintosh and some of the Xerox systems are made to facilitate that different applications with different subjects or objects can be applied in the same kind of activity. Modes are avoided, and the style of interaction integrated: all applications look alike where it is possible, and the same command name position in the menu and activation is used throughout the applications.[1] This way the artifacts are, to a larger extent, capable of supporting shifts between subjects/objects.

Mistakes

Human beings make mistakes, when they deal with a certain object or subject. They issue wrong commands or input, or misunderstand the feedback through the artifact. In short, they get the "wrong" operations triggered.

Undo facilities are parts of the user interface that facilitate such breakdown situations.

Undo facilities help the user avoid doing any harm because of a mistake, by making it possible to instantly reverse any operation that the user had just done. The operations on a physically present object, or towards a physically present subject, may delimit the possibilities of regretting or undoing something.

In a funny way, undo support places itself in between the subject/object directed aspects and the handling aspects. On the one hand, doing and undoing is something done to the real object or subject through the artifact, which means that undoing can be part of operations towards subjects or objects – for instance a typographer trying out different font sizes for a headline. On the other hand, the need to undo can occur because the user has perceived the feedback from (through) the artifact wrongly, or because the user understands that something could have been 'undone', only in the situation the artifact is preventing her from doing so – in both cases a breakdown can occur whereby the artifact becomes the object to the user.

Shifts of focus to/from the artifact as object

Obviously, the support of the user interface concerning shifts to focus on the artifacts, should aim at avoiding them. At the same time, if they occur, a shift from the artifact as an object should be facilitated. The aspects of the user interface dealt with here are the handling aspects.

This type of situations are breakdown situations. Both the breakdowns and the support for shifting back focus are closely related to the kind of education given to the user about the handling of the computer application. We can identify the following different situations:

Breakdowns due to errors in hardware and software

This kind of breakdowns should clearly be avoided, at the same time as they are very hard, if not impossible, to avoid. *Error messages, recovery* and *back-up* facilities help to handle such situations, by making it easier for the user to return to normal activity and to identify the problems. Learning plays an important role: knowledge about the functioning of the system helps the user understand the problem and maybe avoid it or solve it. This, however, requires that the error handling is part of the competence of the user, and that error handling is kept within the domain of use. For instance, text editors can often not open a document after the occurrence of an error, whereas the document can perhaps be repaired through a bit-oriented editor – it seems, however not reasonable to consider the application of such a bit-oriented editor part of the natural competence of use of a text editor.

Help facilities

Help facilities, as mentioned, can also support the user in returning from a breakdown: but what I have in mind here are neither active help facilities,[2] because in their attempt to adjust to the behavior of the user they might very well cause even more breakdowns by changing the material conditions, such as the properties of the artifact for the user. Nor do I have simple syntax help in mind, but rather explanations of how a command is applied and of its effect which can be triggered by the user when needed.

Recurring inappropriateness

I shall start out with two different examples of recurring inappropriateness: In an operating system we copy one file to another by means of the command **copy 'newfile'='oldfile'**. For most users it is impossible to remember on which side of the = to put the 'oldfile' and on which to put the 'newfile'. So, every time a user uses the command she must focus her conscious to this question.

Pop-up menus give rise to another example: Imagine a drawing program applying pop-up menus. There is a problem of the menu always "popping up" in the middle of the area where the user is just working, because that's where the cursor is located. The user can develop some kind of technique to move the cursor before activating the menu; on the other hand, then she must focus on where to place the cursor on the screen. In both cases her focus is drawn to the menu or the cursor, either because she can no longer inspect the things that she is working on, or because she has to find some area of the screen with no importance.

Through practical experience the user can, to some extent, learn to avoid these situations. It is not possible to give detailed general guidelines to how the user interface should look to delimit the number of this kind of breakdowns. Careful design of all parts of the user interface is important. *Flexible* handling aspects can be part of the answer. Flexibility can both mean that there are more alternative ways for the user to achieve a certain goal, e.g. choose between pop-up menus and fixed menus, and that a user can change the programs to better suit her.

The latter, however, requires some programming language which, like the rest of the user interface, is rooted in the practice of the users – good examples of this remain to be seen; with the best of this type of languages,[3] known today, the users need much education beyond what is necessary to conduct the use activity as such.

Many other mechanisms of the user interface aim at preventing the artifact from being an object for the actions of the user.

The object which a user is working on through the computer application, and even the subject with which she communicates through the application, is most often *not* physically present in the use situation. Through practical experience the user develops a understanding of what she is working on or who she is communicating with. When we discuss objects, the imagined object is just as present to her as any physical object, and the user will consider a bad correspondence between the 'real' object and the representation on the screen, etc., as a filter between herself and the real object, and similar with subjects.

For newspaper page make-up, e.g., the representation of the newspaper page is important, i.e. facilities must be provided so that the user can "see" the newspaper page without too much distortion from the artifact. Throughout this chapter we shall return to similar examples.

Shifts with parts of the artifact as an object

Shifts with focus on the artifact occur in breakdown situations where the focus is already on the artifact as an object. They occur as further breakdowns because the computer doesn't behave the way it was expected to, or when recovering from a more or less total breakdown back to the computer application. A typical situation is that the user by mistake ends up in a part of the underlying computer system, e.g. the operating system, which behaves differently from the rest, and which she didn't even know existed. Education is of course one way of solving this, but it is rarely a good strategy that the user must know all technical details to use an application (I can drive without knowing all corners of the engine of my car). Instead it can be feasible to delimit the parts of the underlying system that the user can get in touch with – e.g., by preventing the error messages from "deep down" in the computer to get through to the user.

These types of shifts are supported/caused by the handling aspects of the user interface. They furthermore involve the physical aspects, because severe errors may involve the physical aspects – a shift to some part of the operating system where the mouse doesn't work, or to a layer where the screen image looks totally different, are examples of this.

What we have seen in this section are examples of different types of potential shifts caused by different aspects of the user interface. It is, to a certain extent, possible to relate each type of shifts to a certain aspect of the user interface, even though we have also seen examples where more aspects are involved. Furthermore, we need to know the exact situation (be in a breakdown situation) to fully discuss what happened and why. I will go into more details about this issue later in the chapter.

Competence

In the following I shall move on in the quest for refinement of our understanding of the role of the user interface in use.

Competence will be in focus in the following. Competence is achieved through various types of learning of which I, in this dissertation, focus on learning through operationalization and conceptualization in practical use of an artifact.

From the discussion about learning in Chapter 2, we can see that the use of artifacts where the objects or subjects are physically present will most likely be easier to learn to use, because the subjects or objects are physically available for 'inspection', than will the use of artifacts where the subjects/objects are not physically present. Such artifacts will in turn be easier to learn than artifacts where there are no physical subjects/objects outside the artifact. For the latter two categories a 'direct' graphical display of an object (in the artifact) and a direct interaction style will be easier to learn to handle than a less direct display and interaction style, e.g. based on the typing of commands, etc. As for many other general statements made in this chapter, the above must be interpreted in the context of a specific use situation where specific conditions might have more influence, and "overrule" this effect. On the other hand, considerations like the above can be useful in design.

A user will learn to master a certain artifact through practical use, but also through education. Educational material and documentation are important, but I have chosen to delimit myself from more elaborate examinations of this area.

I shall look at the relation between the required competence of the users and the user interface: can we from an analysis of the user interface see what type of competence is required from the user? In which ways can the user interface support or prevent learning?

I shall use the conclusions about learning in Chapter 2, and the ideas about competence, presented by H. & S. Dreyfus,[4] who, in an operational way, present a framework and some explanations, which can be understood in terms of the human activity approach, although they cannot be derived from this.

H. & S. Dreyfus deal with competence achieved especially through reading and practical experience, called instruction. They talk about five levels of competence, which with the concepts of Chapter 2 can be described the following way:

The **novice** conducts the work process by conscious actions following rules, also actions towards the artifact. These rules are learned as theory and applied, based on a theoretical competence about the material conditions. Operations are scarce and very specific. It is often not clear for the novice what is the connection between the actions and the goals of the total activity. The activity is consciously planned. Many breakdowns occur.

The **advanced beginner** has developed some set of situation specific operations which are based on situation specific practical experience, and some of which are directed towards the artifact.

The **competent** has more general operations and begin to use abbreviations, i.e. to assess, operationalized, the material conditions and skip operations. Many breakdowns occur because the competent often overestimate the generality of operations, and apply these to "wrong" material conditions. The competent has some experience in understanding how different planned actions contribute to the total activity, and is able to choose between important and less important parts of the activity.

The **proficient** relies on a wide repertoire of operations, master the use of the artifact in many types of activities, and master abbreviation in a, still consciously planned, activity. The proficient does normally not conduct actions towards the artifact.

The **expert** can conduct the activity totally operationalized, so that even the "planning" of the conduction is something which is triggered by the meeting with the material conditions.

In the following, I shall use these levels of competence in the discussion of how user interfaces support the use at different levels of competence, as well as shifts of levels of competence. Again, this **how** is easier determined negatively: how do user interfaces prevent use at some level of competence, as well as shifts between levels? I do not use many of the details of the levels of H. & S. Dreyfus, rather I use the difference between the novice and the expert, and some of the important changes along the road from novice to expert.

From many practical examples it can be experienced that the user interface can prevent novices from efficient use without supplementing education. Operating systems with a complex, and often powerful, command syntax are well-known examples of this.[5] According to the above discussion, complexity in how things are done can be one explanation to this phenomenon. In the following we shall discuss whether the opposite can be the case as well, i.e., whether user interfaces can be designed to prevent users from becoming experts, and especially prevent users from treating the computer application as an artifact.

Prompts

The first example of user interface elements which specifically support some levels of competence in neglect of others deal with prompting: Often user interfaces are built to support the novice user, meaning that it will help her choose the right commands, get the command or input format right, etc. This can be done in various ways through interaction processes controlled by the computer, e.g. through *prompts* to which the user has a very restricted amount of answers, answering yes or no or a number, or pointing.

Or through very complex instructions from the computer, such as

? copy

which file from, answer with name followed by cr? oldfile

which file to, answer with name followed by cr? newfile

These kinds of prompts and similar means help the novice user to get started, guide her to follow the rules of interacting with the artifact, get the commands right, etc., but they are inefficient and restricting for more competent users: Many responses are needed to achieve even simple results, and input must be entered in the exact order requested by the computer. This kind of interaction makes it hard for the competent user to keep track of the overall goal.

Furthermore, such prompting can prevent operationalization and especially abbreviation, because there is no way of getting around answering all the questions each time, no matter whether the answers are necessary or not. In general, there are many ways out of this – in this situation, to be able to type ahead might help, and more advanced solutions could be for users to be able to change the prompts, or to choose between different defaults, and change this choice as she gets more experienced.

Operations and automation

According to H. & S. Dreyfus, the expert has operationalized the switch between different operations. However, switches occur on the lower levels of competence as well, although based on conscious analysis and choice. In general, a user works at a relatively high level of action in familiar situations, whereas she switches to a lower level in new or less frequent situations, because she has to use unfamiliar features of the artifact or because she needs more detailed control of the product of her activity. For the novice, there is a big distance from the objects that she focus her actions on, and the real goal or object of the activity. The following is an example of a shift in an unfamiliar situation: a user is creating a document applying some kind of editor. For most of the text she applies a standard format, but for a couple of paragraphs she wants a non-standard format. Depending on the document editor, the standard format is either applied via the computer (e.g. a macro in TEX[6]) or via some set of operations that she does herself (copying a ruler in MacWrite,[7] and pasting it in the right place, etc.). Now she gets to the place where she wants to do something special. The question is what, and how she can achieve this. In some applications she can only choose another standard format, the application is automated "up" to the level of formats. This makes it *simple* to use for less competent users, but an *expert* user will often find it too inflexible, because she wants to have a more *detailed control* of the formats. In other applications, e.g. MacWrite, formats can be changed more directly. On the other hand, in MacWrite the user must always do all the operations for each paragraph. The user will have to create formats by her own set of operations, they become a kind of cultural techniques, but they cannot be programmed into the application.

There are, like in the previous example, solutions to this problem. In the Camex[8] newspaper system, formats can be built like in MacWrite, and copied without copying the text.

Response times

Response times[9] is also something which can cause recurring breakdowns and thus prevent the forming of operations at different levels of competence. For the novice, response times are not very important, because novices conduct low-level actions step by step without feel for the overall goal. For the levels above it is different: Take a login procedure of some computer:[10]

xx login: Bodker ↵	
UNIX system version 3.4 of Sept. 10, 1985	*The user answers, time passes*
	Time passes
VT100_ ↵	
	The system expects an answer *the user answers*
You have mail	*Time passes*
Read mail now?_no ↵	*Time passes* *The system expects an answer* *The user answers*
There is news	*Time passes*
Read news now?_	*Time passes*

This primarily causes breakdowns because the user rapidly learns that there is no need to think out the answers – the system needs the same answers each time. Because the system needs these answers to proceed, the user is forced to be conscious about the computer application. The main problem is, however, that type-ahead is not allowed – in most cases where type-ahead is possible the user would easily learn to type the needed information and then wait until the computer is done and ready to act as the artifact that the user needs.

Conclusions about competence

The above discussion will be elaborated on in the sections to come in this chapter. The prime lesson from the above discussion is that parts of the user interface which support some levels of competence might cause breakdowns at others. This does not only mean that novices cannot make full use of user interfaces designed for experts, but also that experts cannot make full use of user interfaces designed for novices to use.

This means that in design it is necessary to be aware of the competence that is expected from the future users. Furthermore, education must be designed for the users, so that they can be brought at least to the lowest level of competence which is supported by the computer application, and so that they can make use of the application at all intended levels of competence.

In the above, I have talked very little about the domain of competence. We can talk about expertise in a very narrow way, talking only about the handling of a specific artifact, with a narrow purpose, or more widely about expertise in page make-up or programming. In the specific situation, we must be aware of which we talk about. The response time example shows that we can become experts in using a specific computer application despite what the designers aimed at – we learn to type ahead and so on to adapt to the application. This, however does not make e.g. a good programmer.

Comparison of two computer applications

In this section I will use the human activity approach to compare the user interfaces of two document preparation applications or text editors, MacWrite[11] and Microsoft WORD.[12] The reader who is not familiar with these text editors and the Macintosh standard for user interfaces is referred to a brief description of the two user interfaces given in Appendix A. After a short discussion of the conditions for my analysis, I will analyze the levels of competence inherent in the user interfaces, and finally look specifically at the physical aspects, the handling aspects, and the subject/object directed aspects of the two user interfaces.

In the analysis, I assume that the user is an competent or expert user of the specific Macintosh text editors. This is not the same as being a Macintosh hacker. Even though some of the problems of the text editors can be solved by hackers,[13] these parts of the Macintosh system are fairly inaccessible to ordinary users, and manuals and other descriptions of the text editors do not recommend the users to deal with them. A person can become an expert WORD user by using the manuals, learning from other WORD users, and using WORD.

Analysis of features

The type of objects that we work on with the two text editors, are text documents (their form and content). There is no direct support for pre-defining form, e.g. standard letters, although it is possible to make use of pre-defined rulers and paragraph formats by copying those from other documents. For this reason, the form must be worked with at the same time

as or after the content. This is supported by the applications and doesn't cause
systematic breakdowns. As there is no checking of language syntax or the
like in the programs, it is possible to focus on the subject(s) to whom one is
writing, as long as one is only writing text.
Concerning shifts between the real object and seeing the artifact as an
object the what-you-see-is-what-you-get principle allows the user to work on
the document through the text editors. In software/hardware error situations,
the computer closes down, and the user will have to start all over, i.e. there
is no danger of the user ending up in "a corner of the software" that she
doesn't understand. On the other hand it is hard for the user to do anything
but start over, nomatter how much she understands of the problem. Back-up
is left to the user.
User mistakes are in both applications supported by the possibility of
changing all formats, fonts, and texts back to what they were (if the user
remembers the former properties), and by the possibility of undoing the last
cut/paste command. Undo of format and font changes, more than one step
back, seems to be a valuable improvement of both text editors.
Except for some ways of setting up standard documents by copying, and a
glossary which can be used for spelling and format standards, WORD has no
means of customization. MacWrite has no glossary and is in general more
limited.
It is possible to mention several examples of recurring inappropriateness.
In MacWrite the fixed amount of line spacing is one example: Good
typography requires more flexibility, e.g. a 12 point text with 2 extra points
of leading instead of the 12 points with a fixed, unspecified extra leading (and
multiples of this) that one can achieve in MacWrite.
In WORD, some of the prompts in the long run prevent operationa-
lization, e.g. the prompt that you get when asking for a footnote (figure 1):
When working on a text with numbered footnotes, you just want the next
number footnote, and it is very distracting to have to answer 'yes' or pressing
the return key each time, when what you are really thinking of is the content
of the footnote.[14]
Also, it is a problem that the footnote window, usually, but not always,
acts like the regular text window. The user often risks to get swapped from
the specific footnote to the end of the footnote file: for instance when doing a
search in the footnote file, the occurrence of the search string is indicated
behind the search prompt (figure 2), but when she clicks in the footnote
window to start editing, the text scrolls to the bottom of the footnote file
(figure 3) and the selection disappears. This means that search can only be
used in the footnote window to find out "Yes, there is an occurrence of the
text string 'qwq' in the footnote file", and if the user studies the screen image
carefully before leaving the search, perhaps also approximately where this

occurrence is located in the file. It is also not possible to cut out text the way it is done in the main text.

These are just some of many examples that can be mentioned, and the discussion will continue in the discussion of competence in relation to the two text editors.

Shifts of focus with the artifact as an object are rare, due to the Macintosh treatment of software/hardware errors. However, one type of example is worth mentioning: when in the paragraph menu wanting to change a measurement and forgetting the units, one would want to go to the 'edit' menu to check the available units. This can, however, not be done without cancelling the figure and unit, that we question, i.e. without cancelling the new figures and going to a different menu to check the valid units or measurements (figure 4).

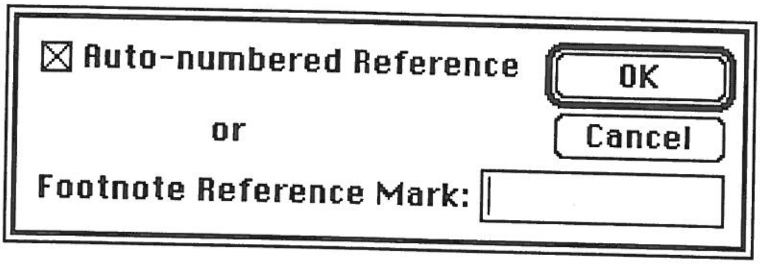


Figure 1. The footnote prompt

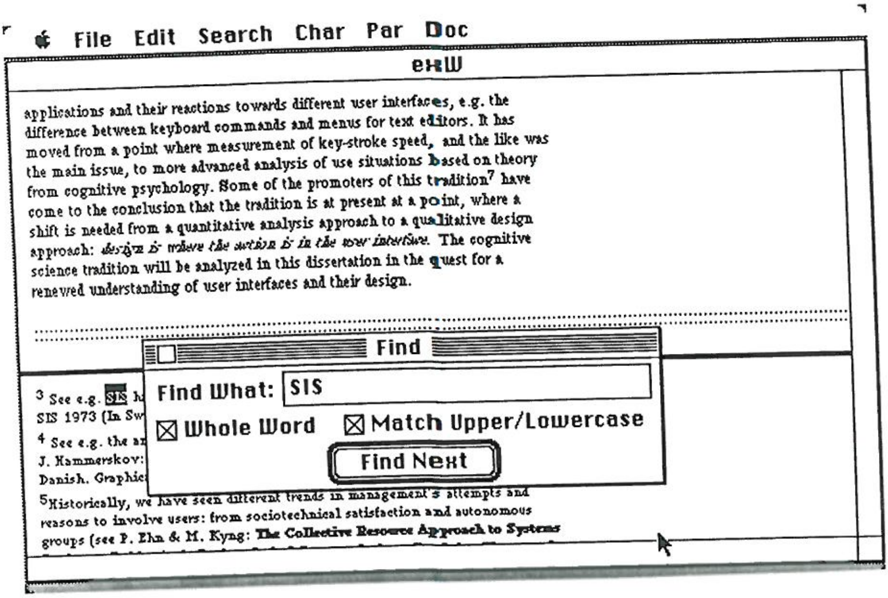


Figure 2. Search in a footnote file, with the search window on the screen. The occurrence of the search string can be seen 'behind' the search window

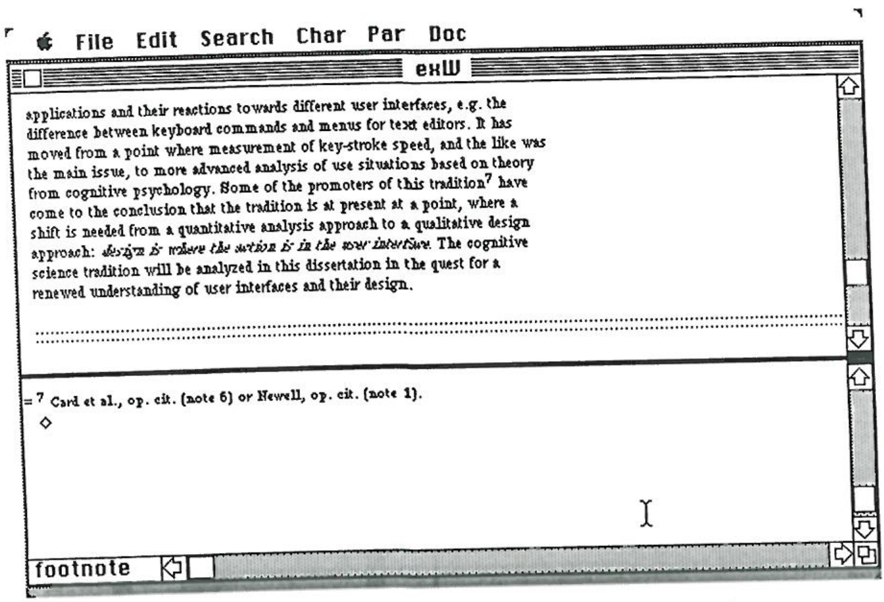


Figure 3. Search in the footnote file, after closing the search window

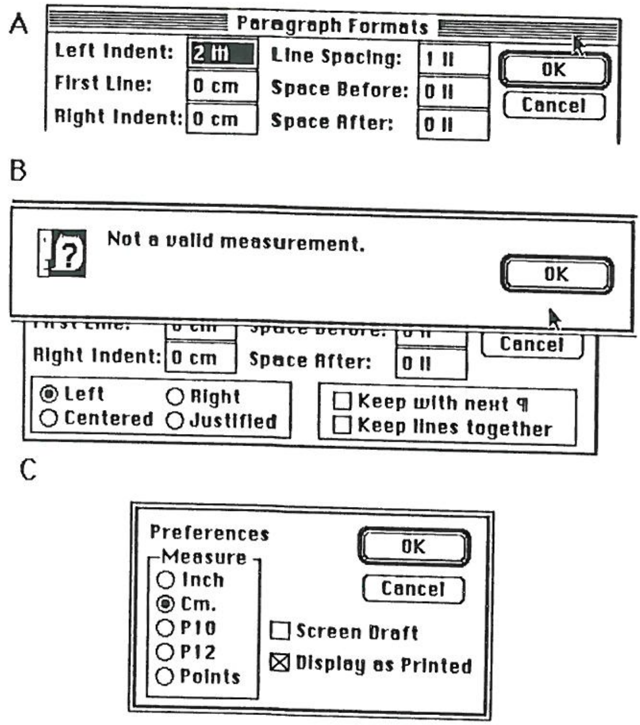


Figure 4. The user is trying to change a left indentation with 2 m, which is not a valid unit (a). An error message tells her that the unit is wrong (b). To find the valid units, the user needs to go to the preferences menu (c), which means that she has to cancel the change which she is trying to make.

Competence

If we start at the novice level, MacWrite is much easier to get started with[15] because of its direct, quite physical what-you-see-is-what-you-get principles: The icons, rulers, and menus make it easier to survey what can be done with the program and how. WORD is not as direct, and it creates a need for the user to understand the role of the invisible ¶ character and of the figures and measures in the various form sheets. If one does not understand this, mysterious things will happen.

Through its wide register of ways of issuing the same commands, WORD allows the users to evolve different patterns of operations, and in many cases use short-cuts for the routine cases. MacWrite is not as flexible, neither in this case, nor when it comes to exploiting the text editors to create good typographical quality.[16] Especially when it comes to line spacing, and to the choice of font sizes, more flexibility is needed.

For both WORD and MacWrite it is relatively easy to become experts in their use, except for the problems discussed in the previous section. WORD, however, has more to offer a skilled secretary, professional, or typographer, who is, or wants to be, an expert document designer. Furthermore, MacWrite is obviously designed to support less skilled document designers, and is an example demonstrating that when such designers want to develop their competence, what-you-see-is-what-you-get soon becomes what-you-see-is-all-you-got.

Aspects of the user interface

The physical aspects of the two user interfaces are basically the same (the use of the screen and the mouse), although the two editors make different use of the keyboard: in WORD, the arrow keys can be used instead of pointing with the mouse.[17] This doesn't apply in MacWrite. What we see on the screen, is in principle the same in the two cases, although MacWrite has some buttons to be pushed (single line spacing, etc.) which WORD doesn't have.

In the previous sections, I discussed some examples where *the handling aspects* are not sufficient. What makes the handling aspects are the "filters" that are put in between the user and the document: the direct representation of the document on the screen, and the scrolling mechanisms; the direct manipulation of the text (selection by dragging the mouse, cutting, pasting, etc.). When the user has used one of the programs for some time, there is no difference between what she sees on the screen and the printed document, except for the distortion caused by the handling aspects. MacWrite, has more direct handling aspects than WORD. WORD on the other hand has more flexible handling aspects – as an example it is possible to change the font size of a word at least three different ways: using the form sheet and the mouse, using ⌘D typing the font size, or stepping up and down a size at a time using ⌘< and ⌘>.

WORD seems to have better *subject/object directed aspects* than MacWrite, because it allows for more flexible formatting of the document – *how* we can do things to the document. In none of the editors are there any direct support for operations towards other subjects or objects than the document.

As a conclusion of this section, it is important to notice that directness in the handling aspects in this case seems to have had consequences for the flexibility in the subject/object directed aspects and vice versa. This implicates a wider conflict between the what-you-see-is-what-you-get principle and the flexibility needed by the expert document producer. How we can make better use of WYSIWYG for all levels of competence remains to be seen.

Recommendations and frameworks from literature

In this section I shall analyze two examples of recommendations for, and frameworks of the user interface. These are selected from the literature. The purpose of this analysis is to see if, and how we can make use of ideas from more traditional literature in the human activity approach. Furthermore, such an analysis serves to illustrate how the human activity framework can be used in analysis of approaches from the literature.

I have chosen two quite different articles: the classics by R. Miller: "Response time in man-computer conversational transactions"[18] and A. Thesen & D. Beringer: "Goodness-of-Fit in the User-Computer Interface: A Hierarchical Control Framework Related to 'Friendliness' ".[19] It is not possible to pick two articles from the very extensive literature and claim that they cover the results or the types of thought in the area. I think, however, that together with the other articles discussed in this chapter, they are prototypical examples of the kinds of thought which we find in the literature. I could, however, easily have chosen other examples which would have been just as representative.

Response time

In his paper, Miller[20] starts out with a discussion of the claim that no response from a computer must take longer than two seconds. Miller claims that this is too strong a demand from a psychological perspective, and he tries to identify different categories of psychological needs for response times. He bases his discussions on the idea that human activity is *clumping: that we have a subjective feeling of having completed a totality of activity. After, but not in the middle of, a totality we are willing to accept delays* longer than two seconds. The arguments for this clumping is a model of human memory as having a limited short-term memory. This results in observations like: "Novices have their short-term memory registers heavily filled with what they are trying to learn; therefore, they are not guides as to what the problem solving user (or other user) will be able to do and wants to do when he is better skilled."

Miller identifies 17 different topics of response to human actions:

1. Response to control activation – click of typewriter keys etc. – delay not more than 0.1 sec.

2. Response to "System, are you listening?"– the shorter the better but not more than 3 sec.

3. Response to "System, can you work for me?"– routine requests 2 sec., setting up jobs 15 sec.

4. Response to "System do you understand me?" – this is always an interruption of thoughts – 2-4 sec.

5. Response to identification – type in code, show identification – different acceptance depending on the type.

6. Response to "Here I am, what work should I do next?" – this is not part of an interactive use of a computer, but more of a production planning system, etc. – 5-15 sec.

7. Response to simple inquiry of listed information – 2 sec.

8. Response to simple inquiry of status – 7-10 sec.

9. Response to complex inquiry in tabular form – 4 sec. (but it depends on whether the delay is in the inquiry or after).

10. Response to request for next page – less than 1 sec.

11. Response to "Now run my problem" – response depending on the user's own problem – after 15 sec., the user is no longer "in the problem solving frame of mind".

12. Response to delay following keyboard entry vs. lightpen entry of category for inquiry – user moves faster with lightpen than with keyboard and expects faster response.

13. Graphic response from lightpen – 0.1 sec.

14. Response to complex inquiry of graphic form – 2 sec.

15. Response to graphic manipulation of dynamic models – e.g. graphic representation of a logical system – hard to estimate.

16. Response to graphic manipulation in structural design – depends on what one is doing (in the middle of a totality or after).

17. Response to "Execute this command into the operational system" – response to the fact that the system is going to work on it shortly, the actual execution can be done later.

First of all, Miller's recommendations are of course adjusted to the technology of the 1960's which make some of the categories and statements less interesting today. Also there seems to be a tendency that this type of recommendations only deals with the best available at the time when they are written down.

Miller's ideas can be discussed in two complementary ways: how can we explain them with the human activity framework? is he right? and how can we use them to make the recommendations of this dissertation more concrete?

I will not go into a detailed discussion about the underlying psychological framework of Miller: obviously, it is built on a model of the human being as an information processor,[21] which is far from the ideas in Chapter 2. Instead I will turn to the requirements and interpret them on the following basis: In general, actions are the totalities in which we conduct a use activity. We accept delays after each action because we have then completed a totality that we are conscious of. Delays in an action can cause a breakdown. According to this we can interpret Miller's recommendations in the following way:

Topics 1-5, 10, 12 and 13 have to do with operations that a human being conducts towards the artifact. As the artifact in such situations should not cause breakdowns, they should not cause any noteworthy delay. This fits well with the figures of Miller, except for 3. However, setting up jobs, as Miller calls it, can in some situations be seen as shifting of focus to a different object or goal, which can explain why more delay is no problem. However, this was perhaps the best available in 1968.

Topics 1 and 13 deal with the physical aspects of the user interface – how does a user at this level activate the artifact. According to Miller, human beings accept very short delays for this type of actions or operations. Topics 2, 3, 5, and 12 deal with rather general issues of the handling aspects, whereas 4 and 10 are more application specific.

7, 8, 9 and 14 are examples of response to low-level actions that the user can conduct on a real object through the artifact, and cover the subject/object directed aspects. The examples are rather application independent in nature. These should, for the non-novice user, be conducted as operations, for which reason it seems strange to allow as much as 10 seconds of delay (8) compared to the other figures in this category.

6, 11, 15, 16 and 17 are examples of response to more application specific actions. According to Miller's estimates these are most often conducted as actions which are not operationalized. According to Chapter 2 it seems uncertain to make that kind of assumptions unless the application has been tried out in the use situation. Furthermore, it seems likely that experts will conduct an even higher level of action, where the mentioned actions are operationalized. If we are designing for expert users we are perhaps not able to accept long response times in this category.

It is not possible, at least not without further empirical studies, to give as specific recommendations as Miller does, based on the human activity approach. What can be said is that the more fundamental operations, in terms of how widely they are applied, and how many other operations they are part of, the shorter delay is allowed, if the computer application is to help prevent breakdowns. Miller deals with operations which are complex by being conducted in many steps by the user, and operations which require much computing. For the first kind, speed becomes less crucial the more we move towards the top, even though experts conduct even the most complicated activity as operations. For the second kind, there is no argument in the human activity approach why, from the users side, there are any differences between operations conducted by means of simple and by means of complex computations.

Goodness-of-Fit in the User-Computer Interface

The idea of Thesen and Beringer[22] is that friendliness reflects both the design of software/hardware, the education, and the background of the user at the time of the dialog. They base their framework on the assumption that both the system and the user can be modelled as interacting control systems with certain expectations about each other's behavior (figure 5).

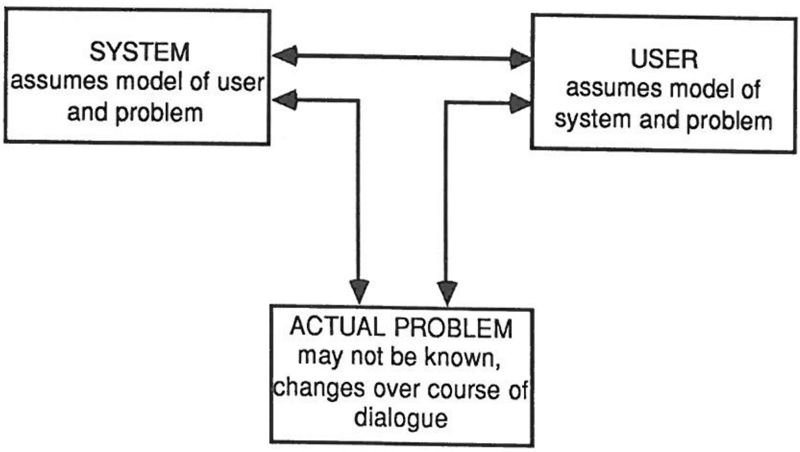


Figure 5. The basic model of human-computer interaction presented by Thesen and Beringer

The authors present many empirical cases of human beings using computer applications at different levels of action. From this, they construct a model of the interaction as taking place as a hierarchy of operations, in terms of the human activity approach. This results in a conception of the human-computer interaction as figure 6.

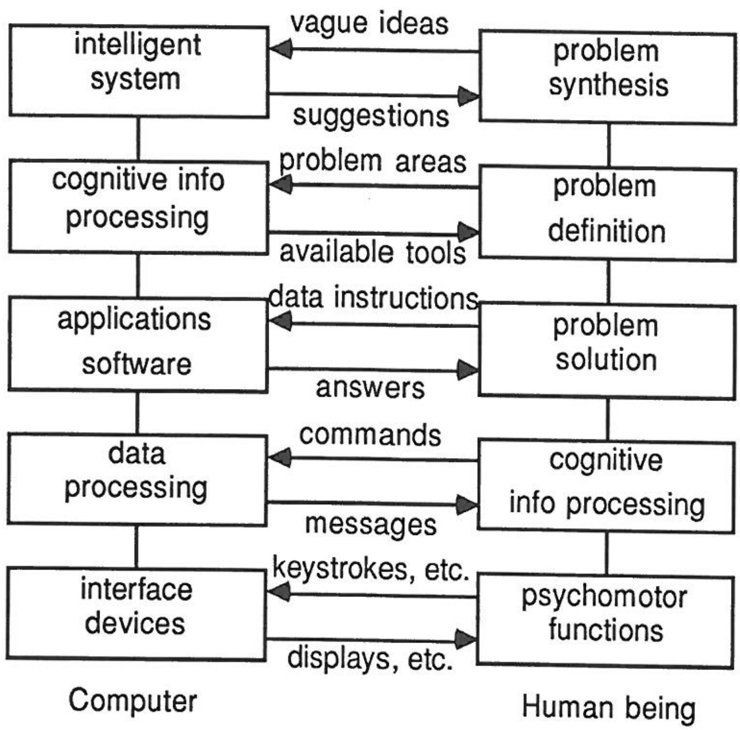


Figure 6. Human-computer interaction according to Thesen & Beringer

This way, the authors identify levels in the system which relate to different action levels of the user. Learning can change the action level, and the action level can change in the cause of the interaction (although they give no explanation to when and why). They say that we do not know how to build the topmost level today, but that it will not be distinguishable from inter-human communication. What is constructed here is, rather than one user interface supporting different levels of action, five user interfaces to be applied by the user depending on whether she needs to operate on the level where the computer application supports only pushing of buttons, on the level where she communicates a vague idea to the computer, etc. This way, the application ends up consisting of 5 "machines" which can all be manipulated by the user to reach a specific goal. As opposed to this, the idea of the human activity approach is that one "machine" must support various operations at different levels to achieve a specific goal.

The authors deal with a system as supporting actions at different, specific levels. The problem is, however, that if we look at a specific level of action, they aim at automating all operations "up" to this level. We can sketch the difference with a text editor example.

Let's assume that the user performs text editing as in figure 7. Depending on which level is convenient to the user, she might try the "write document" machine, where she tells the computer what she wants, e.g. a memo with this and that text. The computer will then come up with a suggestion for form and standard text of such document. She might also use the "write content" machine or the "change format" machine, where she can enter text or change format, but with the commands at this level. The cut and paste level is where Thesen & Beringer see commands being exchanged for messages. The Macintosh, as an example, is automated up to this level, but it can also be operated on the bottommost level.

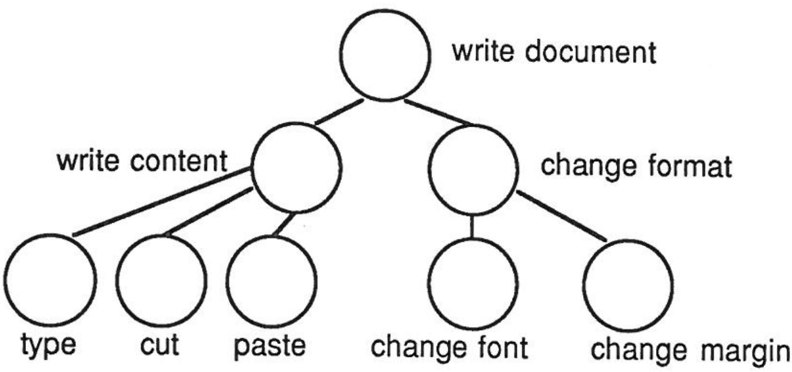


Figure 7. Text editing

With the recommendations of the human activity approach the same user interface should support 'cut' both as action and as operation, and so on. With Thesen & Beringer the user interface should have one specific mode in which type, cut, paste, etc. can be done as actions; one more automated mode where "write content" and "change format" can be done as actions, but not because the user performs 'cut', etc. as operations. The user always intends to perform one level of operations. How a switch between the levels in the model occur is not clear, e.g. whether a user can switch between the levels in the same use activity.

Furthermore, Thesen & Beringer obviously conceive human activity as something consisting of instrumental actions and operations in the bottom of the hierarchy, and communicative actions on the top. As a consequence we should prefer to interact with both subjects *and* objects through a communication of vague ideas. Based on the human activity approach we can say that although the conduction of one activity is not totally planned in advance but shaped in our meetings with the material world, we have no reason to believe that an automated execution of our vague ideas is any step forward. It is an automation of former human actions and operations, by which we cannot exploit the human's capability to trigger the right operation for the right material conditions. We will return to the discussion of the problem of the computer conducting communication after the a short conclusion.

Some conclusions about the human activity approach

From the two examples we see that the framework of the human activity can be used to reveal basic assumptions of different approaches, and that it gives ways of setting focus on both theoretical and practical problems of different approaches.

It has also been demonstrated that earlier empirical results can be used to support our theoretical explanations on the one hand, and the theory can be made more concrete by reflecting on empirical results on the other.

Communication partners and human-computer interaction

In this dissertation, I have chosen to use the words 'human-computer interaction' when I talk about what goes on between a human being and a computer application in use. Many authors prefer words like 'communication' and 'dialog'. They are all borrowed from the language that we use when we talk about inter-human activity. From a human activity approach we must claim that inter-human relations and human relations to objects are two different domains, and it makes more sense to talk about human operation of a computer application than of human computer interaction or communication. In this section we shall discuss the consequences of this separation into different domains for the prospects of natural language interfaces.

For many authors the goal is to be able to consider human-computer interaction as communication, i.e. as something which ideally has the same properties as inter-human communication. Many discussions in literature aim at analyzing and removing the present limitations to the capabilities of computers.[23] Other authors present analytic or design frameworks where a communicative level of the user interface is important.[24]

The common goal of all these authors is to imitate human communication behavior to make better user interfaces.

In the human activity approach we have identified two relations between the user, the artifact and the subject/object: the subject-artifact-object relation and the subject-artifact-subject relation (artifacts used in processing of material and artifacts used in communication). In both cases the human being conducts instrumental operations (and actions) towards the artifact (see figure 8).

If we want to have the computer application take over human communicative behavior (figure 9), the relation is reduced to a relation between the subject and an object. This makes the distinction between the handling aspects and the subject/object directed aspects of the user interface collapse because the computer can only imitate parts of the subject directed aspects. The instrumental relation is different and perhaps not as rich as the communicative relation because the behavior of computers is predictable as opposed to that of human being. I.e. we cannot make the computer substitute the human communicative behavior fully, because the computer cannot conduct the human triggering of interpretation, based on practice. This means that exploiting the properties of the instrumental relation will in the long run create better user interfaces than trying to deal with a pseudo-communicative relation.

Our artifacts, materials and language are triggers of actions and operations in our daily activity. Some of these are communicative and some are instrumental – physical objects can trigger both instrumental and communicative actions and operations and vice versa.

Making use of the fact that the human-artifact relation is inherently instrumental, means of course to trigger instrumental operations towards the artifact. Whether the triggers ought to be language signs or signs for physical objects is dependent of the specific use activity, but seems to play a minor role for the use of the artifact in many situations. For example we want a sign for electronic mail on our electronic desk top. Whether this is an icon symbolizing a mailbox, or a letter, or it reads the word 'mail', seems as such to be an almost unimportant question. What can be more important is consistency with the rest of the design, etc.

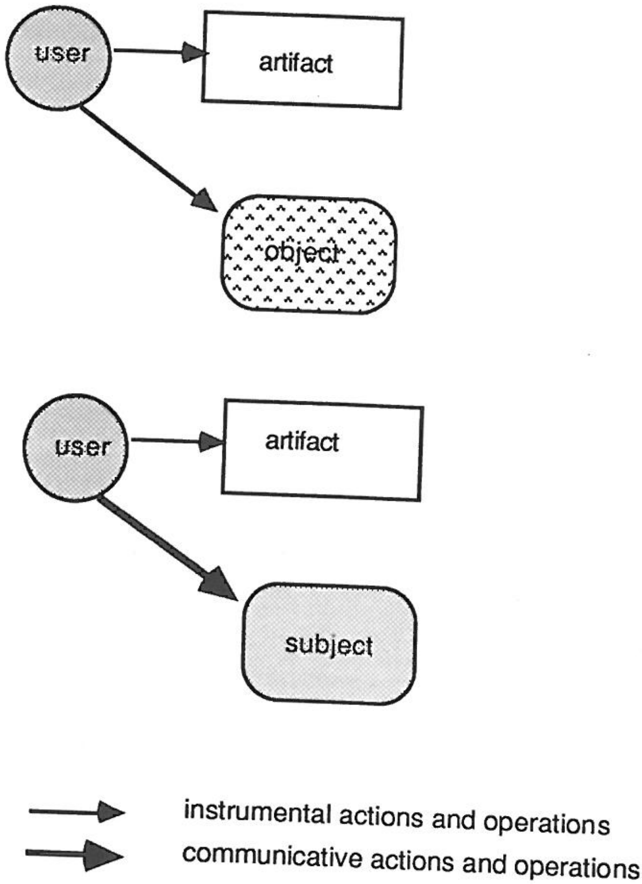


Figure 8. Two types of mediated human activity

The instrumentality also means that we must aim at precision and directness which resembles what human beings meet in their relation with objects, and not the ambiguity and incompleteness of the meeting with other subjects. From this perspective, the computer application should not try to simulate the behavior of a human being, i.e. the triggering of actions and operations based on the actions and operations of the users. For example, the possibilities discussed by Thomas[25] of having the user interface change, keep, broaden or narrow the topic of the interaction seems to be quite uninteresting. Active help facilities[26] to guide or correct the actions of the users are also examples of computer facilities which are created to try to interpret the actions and operations of the user and "trigger actions" which match this interpretation. Practically, such "user adaptable" facilities mean that just when the user has developed one repertoire of operations to adjust to the computer application, this might change.

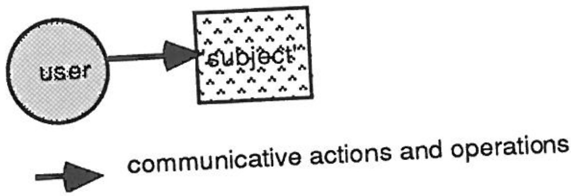


Figure 9. The communication partner reduced to an object

The handling aspects of the user interface is, and should be, supporting instrumental operations, whereas the subject/object directed aspects are, and should be, communicative or instrumental depending on whether we deal with a subject or an object. This does not prevent the use of written or spoken words for command 'languages' or menu entries. Neither does it mean that we must avoid applications which e.g. 'helps' the user of a programming language get the syntax right, as long as such help is not enforced by the application, and as long as the application does not try to 'interpret the meaning' of the commands.

Tangible as well as less tangible parts of the user interface do, of course, also play a role as triggers. The keyboard, the mouse, the icons, the documents, the menus, the error messages, etc. The more familiar these triggers are to the users, the better. Characteristic too, but not unique for computer applications, are:

- that many of the triggers used are communicative, and require communicative response,
- that the computer application is not as clear a sign for its use as more traditional artifacts – functionally very different applications might "look" very alike, as opposed to e.g. hammers, drills, and telephones,
- that the triggers are designed or selected by somebody, the designers.

This has made e.g. Peter Bøgh Andersen[27] see the computer application as an artifact for communication between the designers and the users, and study the influence of the triggers on the communicative practice of the users – the artifact as signs has influence on the language of the users.

I shall not go into more detail about this type of considerations, but point out that not only semiotics, as the work of Andersen, but also aesthetics and ergonomics, and perhaps even other disciplines have roles to play in these types of considerations. The user interface can be seen as a picture, which triggers interpretations of different kinds, and user interface design can of course result in aesthetically appealing, and less appealing user interfaces. Likewise, with ergonomics. I have stressed the importance of the physical aspects of the user interface, and ergonomics deals exactly with the human adaptation to the shapes and forms in a wider sense, of the computer based artifact.

Technical applicability of the human activity approach

Throughout this dissertation I argue that starting out from the use activity is more important than starting out from technical possibilities and constraints. Even so, it is important for an approach to support more technical considerations, as we in design have to deal with these as well. This support should not necessarily be a step by step derivation of one from the other, but perhaps a technical way of thinking which is in line with how the use activity is considered.

I have chosen one example of how user interfaces are thought of from a technical point of view: W. M. Newman & R. F. Sproull's book, "Principles of Interactive Computer Graphics".[28] This is not a random choice. In the early stages of my work I was deeply inspired by the work of Newman & Sproull. Furthermore, what I am seeking here is a framework which can perhaps be applied together with the human activity approach, not a framework to criticise.

Newman & Sproull divide the user interface into four components:

- the user's model,
- the command language,
- the feedback, and
- the information display.

The user's model is quite a fuzzy concept which can be interpreted in various ways, [29] including that Newman & Sproull aim at design of mental models, of what is going to be inside the heads of the users. I will prefer here to say that this concept is a first indication of our need to consider the relation to the practice of users in design. Newman & Sproull say that this user's model consists of a number of **objects**, to which users can apply some **actions** in use. These objects and actions are not necessarily objects and actions in Leontjew's meaning of the words. Objects are entities which relate either to the objects (subjects) that we direct our actions towards in use, these are called intrinsic, or control entities such as menus, windows, etc. The so-called actions refer to the lowest level of actions/operations with which the user can handle the artifact.

The **command language** deals with what types of commands can be issued in actions/operations and how. They identify five issues for command design: command modes, selection sequence, command abort mechanisms, error handling and command language style (keyboard dialogs, function keys, menu driven commands, etc.).

The **feedback** is the computer application's response to the commands. There are three kinds of feedback: the simple feedback, e.g. character echoing, cursor feedback, or clicks from the keyboard; the command feedback, e.g. prompts, error messages, indication of which menu items is selected, etc.; and selection feedback, e.g. highlighting of selected intrinsic entities.

Furthermore, the **information display** can be seen as a special kind of feedback. There are three aspects of information display: the overall screen layout, the display of control entities and the display of intrinsic entities. The information display constitutes our possibilities of presenting necessary properties of the artifact, as well as of the objects/subjects to the users.

We can in all four of the components recognize a distinction between the physical aspects, the handling aspects, and the subject/object directed aspects of the user interface. In other words, we can in general see the two frameworks as complementary, and apply them together. The physical aspects are the input devices, the cursor, the character echo, the sound from the keyboard, and the output devices, including the screen image as a picture. The handling aspects are the control entities of the use model, the error handling, the feedback from commands and selection of control entities, the display of menus, form sheets and other control entities, as well as the overall layout. The subject/object directed aspects deal with the intrinsic entities, as well as with modes and undo facilities according to earlier discussions of this chapter.

The dialog style can be said to belong to the handling aspects – the dialog style determines how we do things to the artifact to make it affect the object or subject, e.g. whether we click the MacWrite double line spacing button, select and fill in the WORD form sheet, or write the command 'line spacing: 2'. From the WORD/MacWrite example, we can see that the dialog style has implications for the subject/object directed aspects, too. The example dealt with how flexible we can change the line spacing – is there a fixed amount of choices to select from or can one enter any amount of extra space?

The main remaining problem of the Newman-Sproull framework is the names of the components. First of all, the user's model gives connotations to frameworks far from the human activity approach, e.g. cognitive science. I would rather prefer to talk about conceptual models, which are built upon the practice of the use domain, and are made to introduce the new artifact in this domain. We can call these use models. They too, must consist of entities or properties relating to the artifact, to the object or subject, and to the physical appearance of the artifact.[30]

'Command language' and 'information display' also give connotations which can be misleading: what is a command language? It is obviously not a language if a language is something used by human beings in communication with each other. It can be a collection of buttons to be pushed, more or less literally taken. In many situations the human being is not commanding the artifact, but doing. Information display is not only display of information, but also of physical objects, e.g. a drawing. It displays, in general, material conditions for, or triggers of, operations: icons, names, etc.

What we have achieved by this discussion is primarily to point at a number of technical aspects of the user interface, the use of which makes it possible to influence certain characteristics of the user interface, as seen from the use activity.

Summary

In this chapter we have seen several examples of concrete user interface constructs, and of their impacts for shift of action level or focus; for operations towards the artifact, as well as towards objects and subjects, i.e. of the physical aspects, the handling aspects and the subject/object directed aspects; for the required competence of the users, and for the learning possibilities. We have seen examples of how specific constructs can influence several, or only one, of the user interface aspects. E.g. a concrete dialog style is part of both the handling aspects and the subject/object directed aspects.

Furthermore, we have seen several potential trade-offs in user interface design:

- a user interface designed for experts can be inaccessible for novices and vice versa,

- flexibility of the subject/object directed aspects may be conflicting with simplicity of the handling aspects.

With the above discussion a phrase such as "a flexible user interface" has been given a richer content:

•flexible physical aspects for a specific application means for example that several different input devices can substitute each other, or that the physical devices are adjustable to the individual user or the specific purpose,

•flexible handling aspects means that the artifact can be handled in different ways to achieve the same kind of results depending on the specific situation, in a text editor e.g. to choose standard formats in most situations, but also be able to choose to adjust special paragraphs or documents individually,

•flexible subject/object directed aspects means to be able to "manipulate" the specific object or subject differently depending on the specific material conditions, but also to be able to do flexible shifts of focus among the different objects and subjects, e.g. through modeless interaction.

We can make similar precisions of other of the well-known buzz-words, e.g. simplicity:

•simple physical aspects for a specific application means for example that the input devices handle easily – that buttons are easily pressed, etc.,

•simple handling aspects means that the artifact can be handled with a few steps of actions or operations, and that the components of the user interface give clear impressions of what they are used for and how,

•simple subject/object directed aspects means to be able to "manipulate" the specific object or subject in a few steps to obtain the needed effect, but also that it is clear on which object or with which subject the user is in contact.

Yet another similar precision can be made of the concept consistency.

By means of the technical distinctions of Newman & Sproull, we have been able to make even more distinctions. By means of these categories we can consider, in design or analysis of user interfaces, how different technical aspects or solutions affect the user interface.

In the next chapter we will discuss user interface design from the perspective of design methods. In Chapter 6 we will use the results from this chapter together with those of Chapter 5 to give a series of recommendations for design of the user interface.

[1] Even though these are the aims of design, the principles sometimes fail to work in the actual cases. Examples of this will be discussed later on in this chapter.

[2] See e.g. G. Fischer et al.: **Active Help Systems**, in **Cognitive Ergonomics, Mind and Computer.** Proceedings of the Second European Conference on Cognitive Ergonomics, Mind and Computer, 1984 or G. Fischer et al.: **Knowledge-based Help Systems**, in L. Borman & B. Curtis, ed.: **Human Factors in Computing Systems**, Proceedings 1985.

[3] For some reason these languages have not followed the general development of programming languages, and they are today still macro based (reflection originally due to Kristen Nygaard).

[4] H. & S. Dreyfus: **Mind over Machine**, The Free Press, 1986.

[5] The UNIX operating system in my view is one example of this.

[6] D. E. Knuth: **TEX and METAFONT. New Directions in Typesetting**, Digital Press 1979.

[7] MacWrite is the Apple Macintosh standard text editor, for a description see Appendix A. A more detailed discussion of the editor can be found later in this chapter

[8] The Camex Breeze system (discussed in P. Ehn et al.: **Utformning av Datorstödd Ombrytning för Dagstidninger** , UTOPIA report no. 12, 1984 (In Swedish. Computer-aided Page Make-up for Newspapers)).

[9] I am grateful to Peter Møller-Nielsen for pointing out this example for me, I will get back to the issue later in this chapter.

[10] This could very well be a SUN login procedure.

[11] MacWrite is the Apple Macintosh standard text editor. The version used here has number 4.5, of April 4, 1985. I use the Danish version, also in the examples, as no English version has been available to me.

[12] Microsoft WORD version 1.05 of April 24, 1985. While I write this chapter I know that a new and very different version of WORD is on the way, where probably many of the problem discussed here have been solved. I don't think that this makes any difference for my project here, as my purpose is mainly to use the human activity framework in an analysis, not to criticise any specific product.

[13] Examples of this is to use other editors to fix files that, due to some error, are no longer accessible in WORD or MacWrite; or using the debugger of the Macintosh to write in pieces of hexadecimal code in other error situations.

[14] This is of course quite a subjective experience, and from my discussions with Morten Kyng I have experienced that the footnote problem does not

bother left-handed as much as right-handed, probably because they use the mouse with their left hand and have the right hand free for activation of the return key. A right-handed person will have to let go of the mouse to do this.

[15] For both MacWrite and WORD there is an extensive manual, but a novice user of MacWrite can furthermore make use of a "Guided Tour" demonstration program, by which she is taken through all the necessary parts of the use of the program, "on-line".

[16] See e.g. Y. Sundblad, ed.: **Quality and Interaction in Computer-aided Graphic Design**, UTOPIA report no. 15, 1986.

[17] This goes only for the Mac+ version of the Macintosh.

[18] R. Miller: **Response time in man-computer conversational transactions**, FJCC 1968.

[19] A. Thesen & D. Beringer: **Goodness-of-Fit in the User-Computer Interface: A Hierarchical Control Framework Related to "Friendliness"**, IEEE Transactions on Systems, Man, and Cybernetics, SMC-16, 7, 1986.

[20] Miller, op. cit. (note 18).

[21] See also Chapter 5.

[22] Thesen & Beringer, op. cit. (note 19).

[23] See e.g. J. C. Thomas, Jr.: **A design-interpretation analysis of natural English with applications to man-computer interaction**, IJMMS 1978 no. 10.

[24] See e.g. W. Dzida: **The IFIP Model for User Interfaces**, GMD-F2G2, GMD Bonn (no date) or Thesen & Beringer, op. cit. (note 19).

[25] Thomas, op. cit. (note 23).

[26] Fischer et al., op. cit. (note 2).

[27] P. Bøgh Andersen: **Edb-teknologi set i medieperspektiv**, The Joint Studies of Humanities and Computer Science, University of Aarhus 1984 (In Danish. Computer technology seen in a media perspective).

[28] W. M. Newman & R. F: Sproull: **Principles of Interactive Computer Graphics**, 2nd edition, McGraw-Hill 1979, chapter 28.

[29] See discussion in S. Bødker & J. Kammersgaard: **Interaktionsbegreber**, working note, The Computer Science Department, University of Aarhus, 1984 (In Danish. Interaction Concepts).

[30] See discussion and appendix of P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984 or S. Bødker et al.: **A Utopian Experience**, in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987.

Chapter 5

Methods for User Interface Design

Traditional user interface design has been bound to conventional technology. The variety of choices has been limited, although it is possible to exploit the technology differently, to apply command languages with more or less complicated structure, etc.[1] With the recent advent of cheaper and better graphical screens and the like, more advanced user interfaces have become something that many designers need to deal with. Thus, access to advanced computer technology has become a challenge to a wide range of designers, who often need to change their practice to deal with the new possibilities. Many organizations, where user interfaces are designed, apply certain standards to ensure efficiency of the design process, standardization of the products, etc. These standards, too, deal with the traditional technology, and they are not sufficient for the exploitation of the new types of technology.

Design organizations need to change their methods and standards. Designers as individuals, and as members of design teams, need to change their practices in order to deal with the new technology. In this chapter I will discuss which types of design method to look for to cope with the new possibilities. To find the 'right' methods is not as easy task: most methods do not make their perspective on the user interface or their application domain explicit, meaning that it is difficult to find out what type of user interfaces and computer applications will be the result of applying the method. Furthermore, the methods available come from very different traditions. They use different concepts: some come from psychology, and talk about modelling human behavior, some have linguistic roots, and deal with the language with which the user and the computer communicate; and still others are mainly concerned with how we can technically build the user interfaces, and how we can split up the user interface, technically, to achieve this. These fundamentally different approaches make the methods hard to compare.

The aims of this chapter are to reveal the perspective on the user interface of different methods, i.e. the chapter will present a taxonomy of methods for user interface design. The chapter will furthermore investigate the possibilities of using different types of existing methods as part of a design approach in the framework of the human activity theory.

Both the taxonomy and the discussions about a new approach to user interface design serve the same purpose: to make the theory of this dissertation practically applicable in design. It is not within practical or theoretical reach of this dissertation to suggest totally new design methods. For this reason, the reader must not expect to find the complete solution to how user interfaces should be designed in this chapter; instead I will point at possible alternatives to the traditional methods of today. The conclusions of this chapter, together with those of the previous ones, will be used in Chapter 6 to give a number of concrete recommendations to be used in user interface design.

A taxonomy of design methods with special regards to the user interface

To get to a taxonomy of design methods and their view of the user interface, I shall start out from some of the key points of the human activity theory. I have chosen to deal with methods, which are primarily prescriptive about the design process, more than with standards which are prescriptive about the product. I find, however, that the parts of the taxonomy which deal with the product, apply for standards as well.

In the taxonomy, I want to deal both with the general design method and with specific methods for user interface design. Design methods prescribe how various design activities are to take place, how the design work should be organized, how the activities should be conducted, and what artifacts should be used. When we try to assess a future user interface design based on the prescriptions of various methods we deal with goals and intentions, and the purposes of various prescribed activities in relation to the total activity. Design methods can, like artifacts in general, be applied despite their intentions instead of according to their intentions. Such use cannot be captured by a taxonomy like this, but has to be based on empirical investigations. Here the idea is to reveal the perspective of the method on the user interface and its design, with the purpose of revealing the implicit focus and blindness that the method will cause the user.

Perspective on the user interface

The aims of this taxonomy are to reveal what type of artifacts and user interfaces are inherent in the application of the method. Following the human activity theory, this means both to reveal how the computer application is to function in a future use activity, and how the practice of users is dealt with in design. In a taxonomy, it is also important to identify the application domain and goals and purposes of the activities prescribed by the method, because these are fundamental for a comparison of methods. The taxonomy will be structured in reversed sequence of the above.

Application domain, goals, objects and purposes

A method carries with it an, often implicit, *application domain* which tells something about which type of organizational change, and of computer application, it is aiming at – the goal or object of the total activity. A method for design of standard accounting applications can be very different from a method for design of innovative one-of-a-kind application for some specific purpose.

A specific activity contributing to the total design activity can be characterized according to its purpose or object, which reflects the goal of the total activity. Design methods, the way they prescribe actions to be taken in different activities, prescribes these purposes, objects and goals. Obviously, some new computer application is one object of the total design activity, although it is not necessarily the goal of each of the subactivities. Nevertheless, this object is not present in a tangible form throughout the design activity, only as different participants' more or less fuzzy visions about the artifact, or their reflection about the future use activity. To get to the, in the end tangible, product of the design activity and to communicate about the artifact-to-be and the future use activity, most methods introduce some techniques and artifacts by which to create *materialized visions*. These materialized visions serve the purpose of *constructing* aspects of the future computer application, as triggers in *communication*, and they are based on *investigations*, in which visions as well as practical limitations are supposed to be uncovered. These techniques and artifacts can be language-like, or they can be of a more physical character.

The goals, purposes, and objects prescribed in a method, as well as the types of material visions to be used have important consequences for how it is possible to handle the user interface (what types of user interfaces to build and how) when applying the method or parts of it.

Means of design, breakdowns and use practice

The means of design, or the different techniques and artifacts suggested by the method, carry with them the perspective of the method, and through their use the users take over the focus and blindness of the method. I am interested in two important parts of this perspective: the perspective on the role of *use* practice, and of *breakdown knowledge* in design.

Methods see the role of the practice of use differently. Some see the practice of use as the origin for design, and some aim at starting out from breakdown knowledge, of users or even of designers. The methods can to a different extent aim at dealing with the practice of use: the articulated, the non-articulated, and the non-articulable aspects. The different focuses on the aspects of practice, and the use of breakdowns, conceptualization and operationalization have impacts for how the method on the one hand makes use of the practice of the users to enhance communication and in investigation, and on the other hand includes formalisms, etc. with which computer applications can be constructed. The materialized vision can play very different roles: a prototype to be tried out, a description of the future actions to be read, or a description of the future artifact to be read. Furthermore, in constructing the different materialized visions, the method can make use of more or less complete formalisms, the aims of which are to structure the materialized visions into what will lead to running programs.

The methods prescribe both *communicative* purposes and *instrumental* purposes of design activities.

Properties of the product

Not only for the purposes of the design activities, but also for the product of design is the distinction between *instrumental* and *communicative* actions and operations important. Is the artifact to be designed looked upon as supporting the instrumental side of human activity, the communicative side, or both? Furthermore, one can distinguish between artifacts for *collective* activities or for *individual* activities, or both.

I have argued theoretically that a good computer application should not be something that the user operates on, but something which she operates through, on other objects or subjects. Not all methods carry this ideal of the product. A characterization of how a method consider the product of the design activity must consider how this product is intended to appear to its user in use: Does the computer application support operationalization, both when it comes to learning how to handle the artifact, and to operations towards the real subject or object through it?

Furthermore, we can distinguish between products of the design activity which are more or less *active* or *passive* externalized artifacts. Methods can aim to develop applications which aim at getting close to passive, externalized artifacts, or they can exploit the computer's capabilities of automating former human operations.

Aspects of the computer application

A return to the distinction between whether the method primarily focuses on practice in use or on breakdown knowledge also leads to the distinction between whether the method when applied will lead designers to start out from the *intentional* aspects of the future artifact (the functionality) or from the *operational* aspects (the user interface).

Methods focus differently at the different aspects of the user interface: *the physical aspects, the handling aspects*, and *the subject/object directed aspects*. Furthermore, the way that an object or subject is considered in relation to the artifact can differ according to the discussions of Chapter 2.

According to Chapter 4, aiming to deal with properties of the user interface means aiming to deal with the conditions for *avoidance of, and recovery from breakdowns*. Methods aim differently at design for *competence*: Some methods do not deal with competence at all; neither as a condition for efficient use nor as the development of education in relation with the use of the computer application. The conditions for use can be dealt with through such features as help facilities, possibilities for the user to adjust the application to her needs at a given time, etc.

Flexibility, simplicity and consistency are concepts that can be applied to characterize the aspects of the user interface. Technically, we can look at how the method aims at exploiting different kinds of technology such as display screens, pointing devices and the like, what types of dialog styles are aimed at, and so on.

Design by means of different approaches

In the following four sections I shall apply this taxonomy to point at important differences between design approaches in their view of the user interface.

I have chosen four different types of approaches which span the spectrum of artifacts for collective work, respectively individual work on the one hand, and support for instrumental, respectively communicative actions/operations on the other. Furthermore, the four examples represent important schools or trends in design of computer applications.

I shall start out with traditional systems development.[2] I will discuss this type of methods in general, with an example from literature where I have chosen the method of Yourdon,[3] and some examples of methods aiming specifically at user interface design. Systems approaches in general prescribe design of artifacts for the instrumental side of the collective level, i.e. they consider the computer application as means to coordinate and control work.

Card, Moran and Newell's approach is a psychologists' alternative to the above. It is one of the few design methods which have come out of cognitive science. By its view of the interaction as "a communicative dialog whose purpose is to accomplish some task",[4] the approach is an example of a dialog partners approach. These approaches deal with the communicative side of the individual activity, i.e., the computer application is intended to be something with which the human being can communicate to pursue some goal.

After the discussions of these two well-established methods, I shall turn to a couple of new approaches, which are far less elaborated. The first type of approach deals with computer support for the instrumental side of human activity, support for the individual activity, i.e. the computer application is intended to support the individual human being in her actions and operations towards an object. The example that I deal with is called the tools approach, a name which fist in well with our every-day word for 'artifacts for the instrumental side of individual work'. The second type of approach focuses on the collective level, the communicative side, i.e. the computer application is intended to support communication among human beings. In line with our every-day language, and with these approaches I shall denote this type of approaches, media approaches.

The systems approach

Systems approaches have their historical roots in the idea that the way of thinking used in computer programming, can be used also when dealing with systems consisting of both machine and human components. Human beings and computer components, other machines, and organizational structures can be considered/described by means of the same types of concepts, and complex components can be decomposed into a number of more simple components which can in turn be considered/described.

I have included two types of examples of systems approaches: the general method of Yourdon[5], and a group of methods which are directed towards user interface design. Because Yourdon, and many other of the systems approaches are older methods we cannot expect them to deal with the new types of user interface technology, such as graphical displays. Instead the following discussion can be seen as an attempt to uncover whether it is feasible to elaborate on methods of this type to handle the new possibilities, or whether there are some more fundamental problems related to the use of the approaches.

Application domain, goals, objects and purposes

This type of methods deals with the design of computer applications for administrative data processing, typically viewing organizations as hierarchical systems surrounded by political and organizational boundaries. In the world of Yourdon,[6] the system that is going to be changed consists of a number of processes in which data is transformed. The processes can be conducted either by human beings or by computers. No matter which of the the two, the processing can be described in the same way by a hierarchical decomposition into subprocesses. When designing new computer applications it is important to focus on the data-flow between such processes. Other of these methods are based not on data flow, but e.g. on decomposable information precedence relations.[7]

Yourdon characterizes his method in the following way:

1. A structured (top-down) and iterative way of conducting the design activity.

2. Meaningful paper models (descriptions) of the future system are built.

3. It emphasizes quality design and better code.

Design is seen as description and change of an (organizational) system with both manual and automated processes. The design of the computer applications as such becomes a kind of side-effect of this.

This type of methods aims to create a structured activity which will lead professional designers through an investigation of existing organizational structures, and of actions applied in use before change. This is intended to lead to a construction of materialized visions of the future computer application (and the changed organizational surroundings) from which the programs can be derived. The suitability of these descriptions for communication purposes is secondary to this.[8]

Means of design, breakdowns, and use practice

Yourdon prescribes the use of data flow diagrams, data dictionaries, data structure diagrams, and Structured English.

A **data flow diagram** provides graphic means of modelling the flow of data through a system, the components of which are manual or automated data processes, or a mixture of both.

A typical system requires several levels of data flow diagrams. Each of the processes, can be defined in terms of its own data flow diagram. The **data dictionary** can present a top-down definition of a complex data element. The third major element of structured analysis is the **data structure diagram.** It is applied to describe the logical structure of a data store.

We can take a closer look into where and how user interface design comes in, described in terms of the human activity framework:

From interviews with the users, the physical data processes and the data flows are identified. Each activity can be described in further detail as yet a number of activities and actions. In the next step, we go from a description of activities and actions to a description of purposes of the activities/actions only. From here the system is changed according to the needs, and when establishing the man/machine boundaries it is found out which processes are to be carried out by humans and which by the computer. Now the interface between the user and the computer application has been established in terms of which data are being entered into, and extracted from, the computer application.[9] What are designed here are possible subject/object directed *actions* of the future use. All future actions need to determined and described at this point.

The method does not make any attempts to originate from the concrete use situation, basically because users and computers are dealt with as the same kind of components. For the user, exchange of data with the computer application is not any different from exchange of data with other users.

Interviews with users form the basis for creating data flow descriptions in several levels. In this process, actions and operations are described in the same way, with emphasis on their purpose in relation to data processing.

Because Yourdon's method in the books stops with the design of what type of data processing is done, it is difficult to go into more details about user interface design.

Several authors have followed a path similar to that of Yourdon where they use Petri Nets, BNF, etc.,[10] to describe the user interface in detail. They build upon the assumption that the human-computer interaction can be described/prescribed as a set of states, between which the interaction process "moves" due to command actions from the user or the computer. This means that this type of method can be used to specify the conditions for low-level actions of use.

Within the software engineering/computer graphics tradition the so-called User Interface Management Systems[11] have evolved which have close connections to the formal specification methods for user interface design.

These formalisms lead to formal specifications of the user interface that can be interpreted by the computer. Often they are combined with a tool box of different basic components of the interface, e.g. different menu types to be used in the specification.

The common premises for User Interface Management Systems are:[12]

• that the user interface of an application can be isolated from and designed after the intentional aspects, the functionality,

• that the ideal method render all dialog styles equally accessible.

• that the method will render complex interfaces more maintainable, and facilitate portability.

• that the user interface design is inevitably intertwined with its implementation, testing, and evaluation.

User Interface Management Systems, thus, build on the assumption that user interface design prosper from a separation from the design of the rest of the application, although it is part of an iterative process where a sequence of materialized visions are constructed and evaluated. Tanner and Buxton[13] point at some critical questions by asking:

"• Is there a point where the separation of the user interface and the semantic functionality breaks down? How can semantic feedback, for example, be adequately dealt with in a methodological way?

• Do the systems really push back the complexity barrier and make user interfaces easier to implement, test, and maintain?

• The modules provided in a User Interface Management System will affect User Interface style through the bias of the path of least resistance. How can we exploit this bias to encourage a preferred and appropriate style of interaction?"

To sum up, systems approaches, as examplified with the method of Yourdon and UIMS, can be characterized by its starting point in breakdown knowledge which leads to a focus on the functionality of the system, as well as a lack of possibility of involving the users actively in the design activity.

Properties of the product

The above methods focus on the articulated aspects of the use activity, or rather, they attempt to reduce practice to only articulated aspects. It is sufficient to observe human actions and operations and to get to a number of actions which the users can perform.

Instead of emphasizing important properties of artifacts, the user is herself reduced to an object, that releases some data "when the right button is pushed", i.e. when the computer application, through prompts or trailers, tell her to do so.

Furthermore, the methods aim at active externalized artifacts, based on the assumption that human actions and operations are reducible to what computers can do.

Aspects of the computer application

The methods basically deal with what kinds of data processing can be (is) done. Except for specifying input and output of a specific data process, which can come from/go to a human being, there are no means of dealing with the user interface.

The methods try to make actions and operations explicit the same way whereafter some of them are automated. The possibilities for the user to later develop new operations are unimportant. Only the security and precision with which the human data process can receive and send data is important.

The User Interface Management Systems methods allow for focus on details of the previous assessment of user interface design from a systems approach: that the user interface design can be done separately from and after design of the functionality without specific knowledge about the use activity; the user interface is reduced to some physical and some low-level handling aspects to ensure the transmission of data between the subject and the object. The objects handled are, except for the computer application, the data being transferred. Together with the type of description formalisms used, this reduces the user to something which, as the most important capabilities, has the same type of capabilities as the computer application.

This also means that there is no concern for competence in the method; neither for education nor for the determination of competence levels or support for breakdown situations in general. Simplicity of the physical and handling aspects is important, because these aspects are used to delimit the predetermined (small number of) ways of doing actions.

The formalisms are general, aiming at creating/describing keyboard-dialog style interfaces (question/answer dialogs).[14] Furthermore, these formalisms do not include any means for discussing screen lay-out etc.

A wide range of the so-called "Human Factors Experiments" can be seen as support for systems approaches as they aim at determining the fastest way of transmitting data in specific types of systems.[15] These experiments confirm the concern for the physical aspects of the user interface: the efficiency of different pointing devices for text selection is one type of results of this type of research.

We can summarize the discussions by giving a brief example: how would a text editor, designed from a systems approach, look?

At an overall level we must emphasize the text flow between various human and computer components. The text is produced, or reproduced by these components. If we focus on two specific of these: the typist and the text editor, the same type of thinking is applied, only at this level they are perhaps passing text strings between them. The subject/object directed aspects are based on a small set of commands to be performed on the text strings. Each of these commands can be issued in one way, or a smaller set of different, but predetermined, ways (a traditional text editor can be thought of this way). The way of issuing commands provides no way of forgetting the computer application and working on the text. The physical aspects and the handling aspects (the little there is) ensure that the commands are issued the same way each time or at least with very little variety, e.g. by providing simple and easily remembered command names, or showing clear and easily conceived prompts.

The psychologists' alternative

Within psychology there is a long tradition of investigating and evaluating the performance of human tasks, especially where a computer is applied. Although methods for this kind of activity have been developed by many researchers,[16] only very few attempts have been made to turn these into design methods. One of the few attempts has been made by Card, Moran & Newell.[17] They argue that the evaluation methods, to have the best effect, must be included in design because this is where the choices are made. They also say that a post-factum evaluation of a computer application is, of course, much easier than an evaluation of an application yet-to-be.

Application domain, goals, objects and purposes

Based on information processing psychology, Card, Moran & Newell claim the following application domain of their method:

"(W)e are creating a new arena of human action: communication *with* machines rather than operation *of* machines. What the nature of this arena is like we hardly yet know. We must expect the first systems that explore the arena to be fragmentary and uneven."[18]

Basically, they are only interested in the dialog between one user and one computer application when the latter is applied to accomplish some task.

"The key notion, perhaps, is that the user and the computer engage in a communicative dialog whose purpose is the accomplishment of some task. It can be termed a dialog because both the computer and the user have access to the stream of symbols flowing back and forth to accomplish the communication; each can interrupt, query, and correct the communication at various points in the process. All the mechanisms used in this dialog constitute the interface: the physical devices, such as keyboards and displays, as well as computer's programs for controlling the interaction." [19]

The basic design strategy is based on the following: that a model of the user, based on the tasks to be performed, can lead to a specific design of the computer application.

The design method deals with what Card, Moran and Newell call the performance of the human-computer system (the user interacting with a computer to accomplish a task). Their method starts out with a situation where the structure of the system, i.e. the task, the user, and the computer application, is relatively fixed which means that the intentional aspects of the application are already determined. The method is basically an investigation method, and neither the actual construction of materialized visions nor communication in general play any role.

Means of design, breakdowns, and use practice

Performance models are designed to predict the performance of the system. Card, Moran & Newell suggest three kinds of models: experimental models, symbolic models and database models.

"Experimental models consist of actual human users with actual running programs or physical mock-ups. Such models are *run*, and performance variables are *measured.* Symbolic models are calculational, algebraic, or simulation models. They are represented on paper or in a computer and have no actual human component (although, of course, they model the user). Performance values are obtained by *computation* (by hand or computer). Database models are stores of pre-measured or pre-calculated data. Performance values are obtained simply by *look-up*. Each of these different kinds of performance models has its place in the system design process."[20]

It is a basic assumption that the design process is iterative, i.e. that it "proceeds in a complex, iterative fashion in which various parts of the design are incrementally generated, evaluated, and integrated."[21] Most of the book consists of presentations of ways of making a special kind of symbolic models, the calculation models, whereas the other kinds are only touched upon.

The three types of calculation models are: GOMS type models, the Keystroke-Level model and the Unit-Task-Level model.

In the GOMS model, the user's cognitive structure consists of four components: a set of **Goals**, a set of **Operators**, a set of **Methods** for achieving the goals, and a set of **Selection rules** for choosing among methods for goals.

Card, Moran & Newell present the following example of the basic concepts of a GOMS model: a particular model of manuscript editing with the line-oriented POET editor. When the user begins editing she has, they say, a top level goal of the activity:

GOAL: EDIT-MANUSCRIPT.

A user segments the larger activity of editing the manuscript into a sequence of small, discrete modifications, such as to delete a word or to insert a character. Although it is often possible to predict the user's actual segmentation of the activity into parts from the way the instructions are expressed on the manuscript, the user decides the parts or actions. The term unit task is used to denote these user-defined action.

GOAL: EDIT-MANUSCRIPT
• GOAL: EDIT-UNIT-TASK *repeat until no more unit tasks.*

GOAL: EDIT-UNIT-TASK is a subgoal of GOAL: EDIT-MANUSCRIPT, the subgoal is to be invoked repeatedly until no more unit tasks remain to be done.

In order to edit a unit task, the user must first acquire instructions from the manuscript and then do what is necessary to accomplish them:

GOAL: EDIT-UNIT-TASK
• GOAL: ACQUIRE-UNIT-TASK
• GOAL: EXECUTE-UNIT-TASK.

Each subgoal above will itself evoke appropriate operations.

A reasonable explanation to how they are able to handle goals and operations at the same time, seems to be that they consider routine operations as something which can be dealt with as breakdown knowledge, i.e. they do not distinguish between actions and operations.

The GOMS model construction may start with a task analysis, and it may involve such things as observation of human-computer systems and simulation of user behavior. The Keystroke-Level Model and the Unit-Task-

Level Model are models of user performance of specific levels in the GOMS model.

The Keystroke-Level Model has been developed especially through comparative studies of text editors done by Roberts. Roberts & Moran[22] point out that the method works only when performance and errors can be measured by an objective set of parameters, and that the evaluation presupposes a running implementation of the application. They suggest that designers use some of the parameters for comparison during design.

This presupposed that a number of similar applications have been measured. Furthermore, this kind of measurement can only be done for applications where possible error situations can easily be identified, and errors handled as routine operations. The type of parameters that Roberts and others[23] think about, is dealing with efficiency of data exchange between the human and the computer.

Only evaluation of actual computer applications and estimation of the performance based on specific (predetermined) characteristics of the applications are included in the method, not, e.g., how one gets from a performance model *to* a running computer application.

It is likely that the evaluation methods of the approach are useful in connection with prototypes, but this issue plays a very peripheral role in literature about the approach. Based only on calculation models, this kind of approach removes the possibility of the user getting hands-on experiences; we hardly deal with an experimental strategy any longer, but a strategy where the user interface can be obtained from a stepwise description of the user's task. Although the theoretical background of this method is very different from that of the traditional systems descriptions, the result is very much the same.

The method claims to start out from the specific activity to be performed by means of the computer application. On the other hand, it is clearly the designers who are to make the task analysis (the investigation). The method gives no hints to how this is done, except that observing users' present work seems to be in line with the method.

Existing patterns of communication and coordination are not dealt with due to the focus on one user - one computer. It is not an explicit goal of the method to treat existing repertoires of operations directly, but it is possible to include observations of traditional work activity in design.

The users are not supposed to contribute to the construction of models and the models are not used in communication in the traditional sense. Rather, users are being modelled and, if prototypes are created, observed. The performance models start out from frameworks such as GOMS. They are application independent concepts arising from the design method. The evaluation criteria have to do with performance; not with implementability, etc.

From the above we can see that Card, Moran & Newell very much take a designer's perspective, as opposed to a use perspective, concerning practice. It is, however, the perspective of an evaluator, focusing on how to view human beings, rather than the perspective of a computer specialist focusing on how to view computers.

Properties of the product

The method deals only with individual use. The user uses the computer application to achieve some task, such as creating a text. The computer application is seen as mediating the instrumental side. However, what goes on between the user and the computer is seen as communication between two components which both have access to a stream of symbols, i.e., the handling aspects of the user interface are made communicative.[24] This is the reason why I denote this type of approach the dialog partners approach. The method does not contain any specific aims of automating former human operations, and it is in general very little specific about what type of product it aims at.

Aspects of the computer application

The authors of the method recognize a need for operations and they admit that their method cannot handle such: "'automatic behavior' (-operations) could imply use of a structurally different process than cognitive skill behavior (...). This is simply another place where our simplified Model Human Processor does not yet reflect some important psychological issues, and we do not pursue it further here".[25]

Basically, the method can deal only with activities where there are no object directed *operations* and where the object directed *actions* are determined on beforehand. Operations towards the artifact (which are really not distinguishable from actions) are determined once and for all, which means that competence and learning cannot be dealt with. However they refuse to see this as an important problem: all the important cognitive aspects of a human being's use of a computer application can be explicitly described.

Again, we can use a text editor as our example: The typical text editor would approach some kind of natural language interaction style,[26] e.g. a language with a limited vocabulary and syntax. The subject/object directed aspects are rather non-existing except for the fixed set of actions available to the user. The handling aspects will include such support as spelling correction, but also ways of "guessing" the intentions of the user and act according to this. Flexibility of the user interface can come in this way, at the same time as this type of adjusting to the needs of the users easily causes unpredictability, i.e. what the user does as an operation at one point of time might not work at another time. The inherently instrumental handling aspects are approached to communication this way – the subject is supposed to feel that she communicates with another subject to get this to do something on the text, or that she communicates with the text. The object directed aspects exist only indirectly through the handling aspects, the relation between the user and the text is always hidden in the communication with the dialog partner, but the user is to feel the relation with the text through communication.

Flexibility in applying a repertoire of operations, towards the artifact or towards the object, is not dealt with by the method.

We see that although the dialog partners approach takes the step into understanding the human performance of tasks (human activity) and the psychological background of this, the approach still represents quite a "from-the-outside" view of what goes on in a specific activity. Card, Moran & Newell allow users of their method to focus on the articulated and non-articulated aspects (as seen from an observer) of human practice.

Furthermore, the approach is rather individualistic, and we are unable to treat the collective aspects of human activity (practice is dealt with only as the competence of an individual, etc.).

The establishment – some conclusions

Generally, we can say that a systems approach means a "bird's-eye" perspective on the organization of use, and a dialog partners approach means focus on the relation between the human being and the artifact. From different points of departure the two types of methods prescribe that design of user interfaces is done by means of description of sequence of (predictable) events or states of the discourse. Modelling the users as origin for design is fundamental for the two approaches.

The basic problem from the human activity framework about user interface design is, however, the lack of acknowledgement for repertoires of operations as important for the interaction, or rather the actions as something which is changing due to the meeting with the material conditions.

The concern of the systems approach for the functionality only, and eagerness for automation mean that each action as ideal should be realized as "one handle to pull". Attempts to make user interfaces where users can exploit a repertoire of operations (to choose the right handle, etc.) is conflicting with the idea that all steps can be determined and described on beforehand.

The dialog partners approach assumes that the functionality is given on beforehand, and that the subject/object directed aspects supports only a fixed set of actions to be employed. Operations towards the artifacts are determined once and for all and are considered as something fixed.

Based on the theory both kinds of approaches have three severe problems in their perspective on the user interface:

1. The methods apply an outside-in view of the use activity. They use some formalism, which is not based on the use practice, for this purpose. Both kinds of methods start out from breakdown knowledge, although the breakdowns causing the awareness are not dealt with by the methods. The description are meant to be read, and the methods can at best be used for observing the non-articulable aspects of practice.

2. The methods do not deal with changes in the level of actions in use, or with competence. Furthermore, the handling aspects are not designed to fulfill the role of making the computer application transparent.

3. The methods aim to automate former human operations.

In Card et al.[27] we have seen an attempt to combine a dialog partners approach with a systems approach. This is done by letting efficient transmission of data between the user and the computer be the key issue in the user interface design. This way it is possible to use the design methods of the dialog partners approach where traditional systems approach methods stop: where the functionality has been designed, but an efficient transmission of data between the user and the computer need to be constructed.

In revealing the perspectives of the two types of methods on the user interface, we have seen that it is not likely that it is possible to apply methods based on these types of perspective in the human activity design approach. Rejecting the systems perspective type methods means that we have no direct methods for a stepwise derivation of the computer programs from the description. In the human activity approach we must look out for different ways of doing this. Also, I would rather not totally reject the use of the evaluation methods of Card, Moran, Newell, and Roberts, but so far these have obviously not been developed in the direction of experimental design.

The new approaches

In the following, a couple of new types of approaches to design of computer applications will be dealt with. They are not general prescriptions of how to do design like the two preceeding methods. They represent, however, some of the few attempts to design which aim to originate from the practice of the future users, and from their work tasks. Due to the lack of completeness, I will focus the discussion on some principles and examples.

The tools approach

The tools approach,[28] as it is dealt with here, was developed in the Utopia project, inspired by the work of people who design workstation based applications[29] on the one hand, and the studies of design for traditional crafts on the other.[30] In the Utopia project, new technological alternatives were developed as alternatives to existing technology, and to traditional tools. The tools approach is deeply influenced by the way the design of tools has taken place within traditional crafts.[31] The idea is that a new tool is developed as an extension of practice within the application domain.

Presenting the tools approach as a method means that I have to be careful, because the approach does not claim any generality, and it is not really prescriptive. Instead the presentation of the tools approach[32] is formulated as experiences from a specific project, and with the idea that other designers can use these experiences to reflect on their own practice.

Application domain, goals, objects, and purposes

When viewing the use of computers from the tools approach one focuses on the individual use activity as part of a collective work activity. A computer application is seen as providing the user with a tool kit containing tools which, under complete and continuous control of the user, are applied to fashion materials into more refined high-quality products. The user is a person who possesses competence rooted in practice of the domain. As a consequence of this, design must be carried out by common efforts of skilled, experienced users and computer professionals. Users possess the competence necessary as basis for design, but to develop their technical imagination they have to gain insight into technical possibilities as well.[33]

The tools approach primarily deals with communication among users and professional designers. In this communication, the future tools must be tried out in the work process, or in a simulation of this.

Means of design, breakdowns, and use practice

According to the tools approach, tacit knowledge relevant when using a tool neither can nor shall be made explicit and formalized. The intention is not to automate parts of the work process, but to build computer based tools which are rooted in the craftsman's original competence.

An important prerequisite for applying computers as tools in a work process is that it is possible for the users to relate the computer based tools to their competence. To facilitate this the approach has developed what is called **use models**,[34] based on the traditional concepts of the application domain, but enhanced with concepts necessary to understand new possibilities and restrictions imposed by computer technology.

Although the tools approach does not claim to have the full answer to how to organize the design process, and which methods to apply, it does give some hints:

- For a user to recognize a good tool from a bad one the tool must be tried out in the work process. In the design process experiments are needed.

- The experienced end-users, the skilled workers, must play an important role in the process. They possess the personal competence that forms the basis for analysis and design.

- The designer has to spend a lot of time trying to gain some insight into the specific work process. Not to become a make-up person or the like, but to be able to contribute constructively in the design process.[35]

- Use models and education must be developed as part of design.

According to the approach, a group of people with the necessary profession-oriented, technical, and organizational skills must be brought together from the beginning of a design project with the purpose of *mutual learning*. Building up a mutual understanding of the specific work processes of the profession and of the technical possibilities and limitations is the purpose, and discussions, visits to workplaces with different "generations" of technology, as well as visits to research laboratories and vendors are important activities activity.

For the more design-oriented activities the use of mock-ups[36] and computer based prototypes are recommended.

The tools approach stresses the need for hands-on experiences for the users, i.e. that the users can try out the design suggestions in use. Although tools are not intended to mediate communication in work, it is necessary to deal with the coordination and communication in the collective work activity. For these aspects the approach apply methods for working out descriptions of the work organization. These are wall-graph methods where the symbols are specific symbols from the domain of use, and not abstract categories. They are applied in the process, not as much to make a description, but to make the participants reflect on their practice. Although the concrete descriptions seem to be more accessible for outsiders than more abstract ones, they are still descriptions.

Operation repertoires and their material conditions are dealt with through the mutual learning and through the experimental design.

Characteristics of the product

The tools approach carries, besides from the aims to design artifacts for individual, instrumental activity, also with it the aims of designing passive externalized artifacts which are under full control of the users.

The method aims at creating artifacts through which one can operate on objects, i.e. the method deals with artifacts for the instrumental side of human activity.

Aspects of the computer application

The method deals explicitly with the possibilities of developing a repertoire of operations. It also considers practice the basis for design. The physical aspects, the handling aspects, and the object directed aspects are dealt with throughout the process.[37]

The idea is that the tool must be under complete and continuous control of a competent user. This means that the user interface must support the skilled user in his work after some education, whereas there need not be direct support for the novice user. The method does not directly aim at one specific dialog style, but it is obvious that the ideas of direct manipulation[38] suit well for the approach.

We can again look at the text editor example: The approach stresses direct contact between the user and the document: that the document, on the screen, look as similar to the printed document as possible, that the text can be moved, ideally by selecting and dragging the parts away. And that iconic menus are applied by which tools (scissors, glue, etc.) to use, can be selected. The user has the initiative and controls the interaction.

Flexibility, as possibilities of changing level of action is stressed by the approach, whereas flexibility to adjust the user interface to individual needs has been discussed but no general recommendations given. For the specific case of page make-up and image processing recommendations have been outlined. Through the design of education, the method deals with what repertoire of operations the user must have to make use of the artifact.

The linguistic approach

Another new, and as design method even less elaborated, type of approach is a linguistic offspring. The example presented here originates from Peter Bøgh Andersen.[39] His approach is only one example of media approaches. Regrettably, most examples are quite analytical, like Oberquelle et al.[40] and Bannon,[41] and do not suggest any prescriptions of how design should be done. The common idea of the approaches is that human beings communicate, and they use computers, like many other media, in this communication. This way they focus on the relation between the work processes and communication in organizations.

Application domain, objects, goals, and purposes

The idea is that computer applications serve as media for communication within organizations. Furthermore, the designers/programmers of the application are part of the communication as well. The approach has no explicit formulation of what is the application domain, but some of the examples that are mentioned are electronic mail and operating systems. The method aims at designing an artifact for the communicative side of human activity. In the approach, there are no prescriptions of how the design process is to be structured, or of the goals and purposes of the individual activities. It seems however, that the analysis of language games and their relation to work processes is important. There are no specific activities where user interface design is to take place.

Means of design, breakdowns, and use practice

The approach gives no direct methods for how the design activity can be carried out, although it generally suggests descriptions based on speech-acts. The descriptions deal with speech-acts (action in my terms), both their intentions and operational aspects on the one hand, and triggering of interpretation on the other.

It is difficult to give detailed characteristics of the role of practice in design according to the media approach. We can however make the following observations: The approach is aware of, and tries to handle, conflicting communication practices of involved groups. That descriptions are to be made, as basis for the reflection of the users of the computer applications. It is not clear who are to make the descriptions, some third party; the designers who are also one of the groups communicating by means of the application; or the users and designers together, neither is it clear how this can be done. The method focuses only on the communicative side of practice. What products are created and communicated about in the activity seems to be less important.

Properties of the product

Communication by means of a medium is something which can take place over time, e.g. what is sent by the programmer may not be received by a specific user until years later. Computers can only process expressions – the users are the ones who put meaning into expressions. They do that based on their practice. The crucial issue of design is how do users **interpret** the expressions which are communicated through (and eventually changed by) the computer? Furthermore, the difference in practice between the designers and various groups of users must be dealt with to understand how expressions created by one group of users (e.g. the designers) will be interpreted by other groups. As computers are **active** externalized they can create new expressions, but not new meaning as such for which reason it is important for the users to see when the computer has created a new expression.

Aspects of the computer application

The user interface is dealt with as the relation between the intentions and the operations of the speech-actions. Operations towards the artifact as a thing is not dealt with. The method recognizes the need for operations on the communicative side, i.e. the subject directed aspects of the user interface, and the communicative competence is dealt with. For the instrumental side, the approach is less clear, which means that there are no explicit descriptions of how to handle the handling aspects, and the physical aspects of the user interface.

A text editor, from this approach, can be viewed in two ways, both as part of a communication between the user and the person who is to receive the document, and as part of a communication between the designer and the user. In both cases we get the result that the text editor must support the user in imagining or knowing the situation of the receiver, and in conducting the proper speech-actions in relation to what she wants to achieve. And furthermore, that the receiver can know the intentions of the sender from the document. We could imagine, if we use the text editor to write internal documents within an organization, that all documents are clearly marked with what type of message they contain – an order, a request, a first draft of a note, etc., and that the text editor enforces this marking. Furthermore, it is part of the consideration that texts, trailers, etc., which are used in the text editor, are chosen by the designer with a certain purpose.

The main point of the approach is that the user can know what kind of person is "in the other end of the line", i.e. that the language used in the communication gives possibilities of knowing the competence, etc., of the partner in communication.

Towards a new design approach? – some conclusions

In this section, I shall summarize the discussions about the tools and the media approaches, about perspective on the user interface. Furthermore, I shall start the discussion of the implications of these perspectives for the usability of such methods in a human activity approach, and relate these to the conclusions made about the more traditional approaches, the systems and the dialog partners approaches.

The media approach focuses on the collective level on the communicative side. It does not explicitly deal with operations and actions that the user conducts towards the medium as an object in its view of the user interface, only with operations and actions conducted towards other human beings through the medium, i.e. the actions and operations directed towards the receiver of the message. The approach requires passive externalized artifacts, but as an alternative, an explicitation of what the medium does to expression, as active externalized, and what the humans do through the medium.

The tools approach deals only with artifacts for the instrumental side, individual level, i.e. tools for the actual conduction of the processing of a product. The users' practice within other areas is dealt with as background for design, but **not** supported by the artifact.

What I am after in the human activity approach is to some extent a combination of the tools and the media approaches to design computer applications that serve both to create specific products and to communicate with other human beings. This way it is possible to deal with both the collective and the individual level, as well as with the operations of the user towards the artifact. The question is whether the two perspectives on the user interface, represented by the tools and the media approaches, are combinable.

The idea of the media approach of dealing with the designer as one who takes part in the communication is a problem from the tools approach: it is easy to see the data creator (who can be the designer) as part of the communication; but the designer as programmer is merely providing the artifacts which make the communication, as well as the production possible, i.e. the designer provide some of the material conditions for the actual use activity, being these physical or social.[42]

The tools approach treats language aspects of the user interface, not primarily by seeing the user interface as a sign, but by focusing on the language use around the use of the computer application in the use models, which seems to be a too narrow perspective on the role of language from the media approach.

We can recall the main problems with the traditional methods: the use of "formal" descriptions when dealing with the user interface, the the lack of acknowledgement of the importance of operations in use, and the focus on automation on former human operations.

The tools approach does differently about these points. The media approach is quite open concerning the first question, although the main emphasis is on analysis of language games if we look only at Bøgh Andersen's approach. Other authors, such as Oberquelle et al.,[43] come from a different tradition where the choice of formal descriptions is more obvious.

For the second point, too, we need to distinguish between different media approaches: Bøgh Andersen deals with the design of artifacts, but only with communicative actions and operations. For other approaches this is less clear. The approach do not aim at automation of former human operations, except in some situations where the result of the automation is made explicit.

For both types of approaches there are reasons to believe that we can move on with a discussion of their use in relation to a human activity design approach, although they do not directly apply together.

The taxonomy – some conclusions

In the next section I will continue the discussion about a possible new design approach along the lines of the human activity theory. First I will, however, make some conclusions about the taxonomy.

What I have tried to do in the examples of the use of the taxonomy is to outline examples of the differences between different methods. On the side of this, I have tried to prepare the reader for the following discussion about which means to apply in a human activity design approach. I find that the taxonomy has proved quite useful in setting focus on and differentiating between the different aspects of a method which influence, and reflect at the same time, the perspective on the user interface.

We have seen examples of the use of the taxonomy on four different design approaches, carrying different perspectives on the computer application and the user interface. We have also seen that a prospective human activity design approach will need to cover both design of individual artifacts and collective, and both support for the instrumental and the communicative side of human activity.

A human activity approach?

As outlined above it is problematic to unite the media and the tools approaches and get to the the approach that we need according to the theoretical approach. What I try to do here, is rather than to unite the tools and the media approach, to expand the tools approach, to enable it to handle the communicative side of human activity as well as the instrumental, and to deal with artifacts for both the collective activity and the individual.

As conclusions of the previous discussions, it is clear that

• a new design approach must take the specific use activity as its origin, i.e. the use practice must be the origin for design together with the specific purpose of the use activity. Design can, besides from a new artifact, result in a need for changing practice;

• it must be possible to design both the subject/object directed aspects, the handling aspects, and the physical aspects of the computer application, thus

• it must be possible to build materialized visions in design which focus on the user interface by allowing the user to try out the computer application in use.

I emphasize here the types of design methods which deal with communication, and with construction of materialized visions which are especially suited for communication about the user interface, i.e. for the examination of triggering of operations towards the artifact as well as the subject/object directed operations. The reason for this choice is the epistemological reasons for, and problems with, user participation in user interface design. There exists many methods by which programmers can specify their programs, etc., but only scarce ideas about how we communicate with users about user interfaces. How these two types of methods can work together, is yet another problem which I will not deal with. With this as with much of the discussion here I will rely on the competence of the professional designer – that she is able to use examples and principles to change her practice.

The design approach is an approach where the computer application can be seen as an artifact standing between the user and her object, or as an artifact between the user and the subject with which she communicates, in both cases ideally without being noticed. In the use of an artifact many different objects and subjects can be involved which pose different requirements to the interaction, and thus to the different aspects of the interface. The approach must cover design of artifacts for both collective and individual use, and looks at the collective activity as origin for design even though the final artifacts may end up as individual artifacts.

It is important in the approach to identify the different objects and subjects that are to be dealt with in the future use activity, and to deal explicitly with actions towards all these objects and subjects. For all such objects and subjects the user interface has physical aspects, handling aspects and subject/object directed aspects to be developed, i.e. for all objects and subjects the user needs support for operations towards the artifact, as well as operations towards the object or subject.

The ideal is a passive externalized ideal, i.e. it does not aim at automation of former human operations, neither on the communicative nor on the instrumental side, which take away from the user the control of the artifact.

The user interface

The user interface is intended to support the users' development of operations towards all the objects and subjects of the activity, as well as shifts between these. Furthermore, it is intended to support the development and use, only, of operations towards the artifact. The handling aspects and the physical aspects together are constituting the needed support for operations for all possible real objects or subjects of the use. The conditions for forming operations towards the subject or object at one level might be identical to, or overlapping with, the conditions for operations towards the artifact at another level. The functionality is depending on the levels of action which is determined in the process of designing the user interface. Thus, the design of the functionality, in this approach, is something which can be seen only in situations of after-design reflection.

In accordance with the change of practice which is a result of the new artifact, the design of education is important. Especially the education needed to achieve the level of competence where the user conducts only operations on the artifact in the regular use situations, i.e. the level where the user interface causes no breakdowns due to lack of competence.

Means of design

To elaborate on a new design approach, we need more elaborate suggestions for user interface design methods. In this section I shall discuss a number of candidates for such.

I will start with a rejection of the kind of methods which consider human activity and computer programs structurally alike. Furthermore, I suggest that formal specification methods, etc., are only applied by designers while doing the programming of the application. Descriptions that start out from any abstract concepts, such as information flow, are in general problematic because they do not start out from the practice of the users in the concrete use activity.

Instead I will turn to methods which allows the users to gain hands-on experiences with the artifact-to-be, prototyping methods. Mock-ups, simulation, 4'th generation languages, and exploratory programming can all be considered prototyping in the every-day meaning of this word, although Floyd,[44] when trying to define the common meaning of the term among software engineers, talk about prototyping in a more narrow sense: "prototyping refers to a well-defined phase in the production process, where a model is produced in advance, exhibiting all the essential features of the final product, for use as test specimen and guide for further production."

Mock-ups

Mock-ups, the way the term is used here, are different kinds of non-computer based prototypes. I even see a reason for calling some computer based prototypes mock-ups, e.g. such prototypes which are build only by means of a drawing program. The reasons for applying mock-ups are technical and economical, but also the enhancement of imagination through discussions of the "ideal user interface", partly independent of the available technology. For example, in a project like Utopia it seemed natural to start out from a display screen which in size and resolution resembles a newspaper page.[45] When realizing that such a solution is not implementable with available technology, one can start to discuss how windows, etc. can be applied to avoid the problem, or to find out whether a different kind of screen technology can be purchased. If none of the solutions are satisfactory, one may want to wait another 10 years for the solution. Mock-ups is a way of enhancing imagination.

4'th generation tools

The terms "4'th generation tools" or "application generators" cover a range of computer applications the aims of which are to support users in developing applications for their own use; typically applications which can be characterized by the following keywords: data bases, reports and statistics. The application domain is office information filing and retrieval. Most 4'th generation languages consist of a data base, an editor for creating forms on the display screen which are to be used when data is entered into the data base, a report generator, by which output lists and statistics can be structured, and a query language, by which queries for the retrieval of data can be constructed.

The 4'th generation tool often imposes constraints on the user interface, as in the Aarhus Polytechnics case where the following example comes from[46]: in the paper based journal such things as paper color and texture, logos, and writing style are important in the retrieval process. We had no possibilities of trying out whether this kind of characteristics was something to work for in the computer based journal.

4'th generation tools are meant to be used **by** the future users, i.e. it is a claim that the users themselves, without assistance from computer specialists can design their own screen images, report forms, etc. It is the experience from Aarhus Polytechnics that it is easy to get started with the design, but that to do a little more advanced things very low-level programming[47] was needed. Furthermore, it is fair to say even though the concepts applied in the 4'th generation tools are application specific, the design must start out, **not** from the use activity as such, but from a discussion of what types of documents are available, and so on. In order to take part in these discussions the users must have some amount of design/programming competence.

4'th generation tools share many characteristics with the mock-ups. Furthermore, they allow for trying out aspects of the user interface, e.g. how it feels to actually select something by moving the cursor, at the same time as they impose constraints to how the user interface can look, and forces the designers to think more explicitly in terms of what can be done with this specific 4'th generation tool. This means that the 4'th generation tools should be used together with the users, but not by them without computer technical assistance.

4'th generation tools, traditionally, provide a tool box of components to help build a specific kind of user interfaces for form-filling style interaction which makes the application domain very specific. Like all types of computer applications these get more and more advanced, but for most of them, all the prototypes designed in the process should be running computer programs, and the design process is supposed, step by step, to lead to a running application.

Exploratory programming

Another example of a prototyping environment is Smalltalk-80.[48] The Smalltalk-80 system is used in the process of designing the system itself. This literally illustrates that the purpose of design is the purpose of the future use.

Smalltalk-80 is on the one hand a very flexible environment which allows the user to experiment with major and minor changes of e.g. the user interface. On the other hand the prototyping environment strongly supports a specific practice of the future user.

As Smalltalk-80 is designed by its users (and used by designers) this is an example of what kind of design tools and techniques and user interfaces these designers want, or think they want, for themselves:

1. Flexible prototyping with the possibility of changing all parts of the product when needed. Examples and prototypes to work on instead of strictly enforced rules.

2. Programming by copying and changing instead of rewriting, which supports experimental design.

3. A graphics oriented user interface with windows, menus and a mouse.

Compared to 4'th generation tools, Smalltalk-80 is very different in the way the users can apply the rules of its use. On the one hand, it provides an extreme flexibility in how the user interfaces can look. On the other hand, it relies heavily on the practice of the designers. They must be capable of staying within the limits of the style when that is needed, and changing the style when they need that. Furthermore, it takes some effort to make use of the facilities in Smalltalk-80. Compared to the 4'th generation tools, these are intended to require much less learning and computing practice than Smalltalk-80.

Prototyping

The computer supported design methods discussed here exhibit to us a potential conflict between the accessibility, in terms of computing competence and programming effort needed, and flexibility, both in terms of how the artifacts can be applied, and in terms of which user interfaces can be designed. This indicates that to achieve better computer based prototyping support we must

- focus on a specific application domain and use practice(s),
- provide a tool box of basic components which supports different styles of interaction,[49]
- have access to a variety of interaction devices,
- use both mock-ups and more advanced computer based prototyping methods.

As a final comment, prototypes need not only be simulations in which one user try out one computer application; a prototype can be a whole play where the users are brought into a world where computer prototypes play some parts, and other parts are played by human beings. However, even the best mock-ups or prototypes are only simulations of the meeting with the real world.

Scenarios

In the above, I have discussed formal description methods and prototyping methods and rejected the former. The question is whether we have to give up descriptions all together, or if there exist types of descriptions which apply as part of the human activity approach. Naturally, descriptions can never substitute the meeting with the material world, i.e. description cannot trigger the operations that the meeting with the material would do. But we might have a chance of approaching this situation, perhaps as supplements to some kind of prototyping.

Scenarios are a way of describing computer applications and their use to users. Scenarios can have many different forms, common to which is that they are descriptions. In the Aarhus Polytechnics case scenarios in natural language were applied to sketch the different overall solutions for a computer application: what text should the user enter to register the mail, how were copies produced, which information could be retrieved by the case workers etc.? Scenarios can focus on the sequence of actions/operations taken by the user and the feedback from the application.

From the human activity point of view, what is needed is something suited for communication, which can help the users start to reflect upon their practice and the way it may change due to a new artifact. Scenarios seem not to be suitable for construction at a detailed level.

Suggestions for new methods?

The means of design have strong impacts on the user interface. This goes both for which aspects of the computer application one comes to focus on, and for the properties of the future user interface. An implication of this can be that one must either choose to pursue a certain kind of user interface, or choose means of design and be aware of what possible types of user interfaces are cut off by the choice.

One recommendation is, in the early stages of design, to choose means of design which are as general as possible when it comes to which kind of user interface to end up with, but still methods which allows to start out from the use practice of the users, not from breakdown knowledge. Later on, one can choose more specific means, because that makes construction of a materialized vision of the interface easier.

Both the mock-up, the exploratory programming, and the 4'th generation methods can be included in a tool box of design means which the professional designer has access to. Not as three different tools that can be applied directly in any given user interface design activity, but as concepts underlying the tool box. This designer's tool box should then be a set of computer based and non-computer based means, by which the specific design means for a specific design activity can be built.

What I have suggested here is not a method, in the traditional sense of a method; not a recipe to be followed step by step. The reason for this is the relation between methods and design practice discussed in Chapter 2: professional design is not done based only on a step by step procedure, but based on the repertoires of the professional designers, of operations in which different means are applied. The discussions and examples presented here aim in different ways directly at presenting such new means. To be applied in design practice, designers must try them out and include them in their repertoire of operations.

In Chapter 6 I shall go into more details about both the human activity approach and about the research to be undertaken to develop this approach. I aim at giving designers concrete recommendations for the design of user interfaces, based on the theoretical and practical discussions of this chapter and the previous chapters.

---

1 See e.g. my discussion of R. Miller: **Response time in man-computer conversational transactions**, FJCC 1968 in Chapter 4. The theoretical approach of this dissertation is not bound to this new technology, and could have been developed also without this. The new technology has, however, resulted in a larger emphasis on user interfaces, also in research.

2 For a theoretical presentation of the systems approach, see E. Holbæk-Hanssen et al.: **System description and the DELTA language**, Norwegian Computing Center, 1975. The term is also used by C. W. Churchman for a different kind of approach which is not covered by the characteristics presented here (See e.g. C. W. Churchman: **The Systems Approach**, Dell Publishing Co. 1968).

3 E. Yourdon: **Managing the Systems Life Cycle**, Yourdon Press 1982 or T. DeMarco: **Structured Analysis and System Specification**, Yourdon Press 1979. See also e.g. C. Andersen et al.: **SYSKON - en bog om konstruktion af datamatiske systemer**, part 1 and 2, G.E.C. Gads Forlag, Copenhagen 1972 (In Danish. SYSKON - a book about construction of computer systems), or M. Lundeberg et al.: **Systemering**, Studentlitteratur 1978 (In Swedish. Systemeering)

4 S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum 1983.

5 Yourdon, op. cit (note 3).

6 Yourdon, ibid.

7 Lundeberg et al., op. cit. (note 3).

8 S. Bødker & J. Hammerskov: **Grafisk Systembeskrivelse**, DAIMI IR-33, IR-34 and IR-35, University of Aarhus 1982 (in Danish. Graphical Systems Description).

9 I know that Yourdon and his people now teach a method which is a little different from this, based on the idea of events. It has not been possible for me to get written material about this, but the ideas seem to resemble those discussed below as methods for user interface design based on a systems ideal.

10 H. Oberquelle: **Semi-formal Graphic Modelling of Dialog Systems**, Universität Hamburg, FB Informatik, Bericht Nr. FBI-HH-B-113/85, 1985. D. L. Parnas: **On the Use of Transition Diagram in the Design of a User Interface**

for an Interactive Computer System, Proceedings. ACM 24'th Nat. Conference, 1969, pp. 378-385. P. Reisner: **Formal Grammar and Human Factors Design of an Interactive Graphics System**; IEEE Trans. on Software Engineering, SE-7, 1981, pp. 229-240. B. Shneiderman: **Multiparty Grammars and Related Features for Defining Interactive Systems**; IEEE Trans. on Systems, Man, and Cybernetics, SMC-12, 1982, pp. 148-154.

11 A. I. Wasserman: **USE: a Methodology for the Design and Development of Interactive Information Systems**, in H.-J. Schneider(ed.): **Formal Models and Practical Tools for Information System Design**, North-Holland 1979, pp. 31-50 or A. I. Wasserman: **Software Tools and the User Software Engineering Project**, in W. E. Riddle & R. E. Fairley: **Software Development Tools**, Springer Verlag 1980, pp. 93-113.

12 P. Tanner & W. Buxton: **Some Issues in Future User Interface Management System (UIMS) Development** in G. Pfaff (ed.): **User Interface Management Systems**, Springer Verlag 1985.

13 Tanner and Buxton, ibid.

14 In J. Caspersen et al.: **Description Techniques for Interactive Systems**, Ålborg Universitetscenter 1985, a number of this type of techniques are presented, and discussed in detail.

15 B. Shneiderman: **Human factors experiments in designing interactive systems**, Computer, December 1979, p. 9-19, mentions several experiments. S. K. Card et al.: **Evaluation of a mouse, rate controlled isometric joystick, step keys, and text keys for text selection on a CRT**, Ergonomics 21, 8, 1978 pp. 601-614 compares the speed of different devices for text selection.

16 For a discussion of different research groups and their methods, see panel about psychological research methods in the human use of computers in L. Borman & B. Curtis, ed.: **Human Factors in Computing Systems**, Proceedings ACM 1985.

17 Card et al. 1983, op. cit. (note 4).

18 Ibid.

19 Ibid.

20 Ibid.

21 Ibid., p. 406.

22 S. K. Card et al.: **The Keystroke-Level Model for user performance time with interactive systems**, CACM 23, 7, 1980, and T. L. Roberts & T. P. Moran: **The Evaluation of Text Editors: Methodology and Empirical Results**, CACM 26, 4, 1983.

23 See various contributions in Borman & Curtis, op. cit. (note 16).

24 This is not the way I view communication in this dissertation, what in my theoretical approach is a triggering of interpretative operations, by which the

communication moves ahead, is in Card, Moran & Newell's terms a planned exchange of a sequence of information.

25 Card et al. 1983, op. cit. (note 4).

26 Card et al. 1980, ibid., Roberts, op. cit., (note 22) etc. deal with other types of interaction style as well.

27 Card et al. 1978, op. cit. (note 15).

28 P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984.

29 See e.g. B. Shiel: **Power Tools for Programmers**, Datamation, 29, 2, 1983.

30 B. Göranzon, ed.: **Datautvecklingens Filosofi**, Carlsson & Jönsson 1984 (In Swedish. The philosophy of the Computer Development).

31 See e.g. Göranzon, ibid.

32 Ehn & Kyng, op. cit. (note 28).

33 See e.g. S. Bødker et al.: **A Utopian Experience**, in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987.

34 For a detailed discussion of use models see J. Kammersgaard: **On Models and their Role in the Use of Computers**, Preceedings of the Aarhus Conference on Development and Use of Computer Based Systems and Tools, Aarhus 1985.

35 Bødker et al., op. cit. (note 33).

36 For a description of mock-ups and the other methods applied by the Utopia project refer to Chapter 3.

37 For a detailed description, see Chapter 3.

38 See e.g. B. Shneiderman: **Direct Manipulation: A Step Beyond Programming Languages**, Computer, August 1983, pp. 57-69.

39 P. Bøgh Andersen: **Edb-teknologi set i medieperspektiv**, The Joint Studies of Humanities and Computer Science, University of Aarhus 1984 (In Danish. Computer technology seen in a media perspective).

40 H. Oberquelle et al.: **A view of human-machine communication and cooperation**, IJMMS 19, 4, 1983.

41 L. Bannon: **Computer-Mediated Communication** in D. A. Norman & S. W. Draper, ed.: **User Centered System Design**, Lawrence Erlbaum 1986, pp. 433-454.

42 For a discussion of the role of the user interface as signs in communication, see Chapter 4.

43 H. Oberquelle et al., 1983, op. cit. (note 40).

44 C. Floyd: **A Systematic Look of Prototyping** in R. Budde et al., ed.: **Approaches to Prototyping**, Springer Verlag 1984, p. 21.

[45] S. Bødker et al.: **Graffiti 7. The UTOPIA project. An Alternative in Text and Images**, Arbetslivcentrum 1985.

[46] See discussion in Chapter 3.

[47] See also B. H. Kristensen et al.: **Retningslinier for valg af faglige strategier på kontorområdet - et case studie over Århus tekniske Skoles kontorautomatiseringsprojekt**, Department of Computer Science, University of Aarhus 1986 (In Danish. Guidelines for trade union strategies in the office area - a case study of the office automation project of the Aarhus School of Polytechnics).

[48] A. Goldberg: **SMALLTALK-80. The Interactive Programming Environment**, Addison-Wesley 1984, p. 20. This is perhaps a rather unorthodox interpretation of the aims of applying the Smalltalk-80 system.

[49] Smalltalk-80 supports the basic idea of this by the Model-View-Controller concept, which allows the programmer to change the user interface parts (View-Controller) independently of the underlying application specific objects. Such change can of course not be made at random in reality, as we have argued that the user interface is **not** something which can be considered independently of the other aspects of the application.

Chapter 6

User Interface Design – Advice to the Designer

> Do not automate the work that you are engaged in, only the materials. If you like to draw, do not automate the drawing; rather program your personal computer to give you a new set of paints. If you like to play music, do not build a "player piano"; instead program yourself a new kind of instrument. (Alan Kay)[1]

This chapter contains the conclusions of this dissertation. An obvious question to ask is: *What does the reader know about user interfaces and their design, now, that she or he would not have known without reading this dissertation?* Was it worth the effort for the reader to work the way through this report, and was it worth the effort for me to write it? My goal with this dissertation has been to elaborate on the theory about design (of user interfaces) to make it contain ideas of hands-on experiences and prototyping which have so far only been empirically based. In short, the idea has been to look into the possibilities of bringing the theory up-to-date with the best of existing practical knowledge. At the same time the prototypical or ideal cases described here, as well as the theoretical ideas, can probably help designers of computer applications change there design practice to do better user interface design.

The chapter is structured as follows: a summary of the conclusions, at a theoretical as well as a more practical level; a presentation and discussion of some recommendations to designers; and a discussion of the possibilities and perspective provided by the human activity approach. The conclusions are meant to summarize the theoretical aspects of the dissertation: what are user interfaces? what is design? whereas the recommendations are meant to outline actions to be taken to do better design and make better user interfaces. The aims are to help designers change their practice to make better user interfaces, based on the practice of the future users. By this type of concluding chapter, I hope to inspire the readers not only to reflect on the conclusions, but also to change their practice, as researchers or designers.

Conclusions

The main goal, seen in retrospect, of this dissertation has been to redefine the user interface concept, and make this definition operational in design.

Through the theoretical approach, the focus has been on *how the computer application appears to its user in use, and especially on the operational aspects of this appearance.* The distinction between operational and intentional aspects of use, applied in this context, suggests that we ought to talk about human *operation* of a computer application rather that of human-computer interaction. The user interface can be defined as the software and hardware *supporting the human operation of the computer application* in a specific type of use activity, constituting some of the material conditions for triggering specific operations in a specific use situation.

This triggering is part of the non-articulable aspects of practice which play a special role in user interface design: we cannot deal with them without dealing with the specific use activities in which the artifact is to be applied.

The relation to practice makes it possible to deal with user interfaces not just for the individual user but for groups who share a practice.

The definition stresses the difference between the use situation where the computer based artifact is operated while focusing on some other object or subject, and the design situation where the computer based artifact is one of the objects.[2]

Theoretically, as well as practically, there is still much work to be done in including a more detailed understanding of the role of artifacts in collective work, and the consequences of this for the design of artifacts, in the human activity approach. There is definitely a trend in research that we are heading in a direction where computer support for collective work becomes increasingly important,[3] but so far the ideas presented have been without any real theoretical setting. I find it obvious that the human activity approach can play a role in this. The problem with the basis in the work of Leontjew might turn out to be that he only sees communication as coordination of physical work. To go on with the communicative aspects, we might need to look for a different theoretical basis or supplement.

Before going into concrete details of the conclusions it is also important to stress that the human activity approach means a farewell to human-computer interaction as language – what a human being does in relation to a computer application is inherently instrumental; and to the ideal computer application as simulating human behavior – the human being is capable of handling the artifact as what it is, a thing. Furthermore, as design must deal with more than the articulable aspects of practice, it is not sufficient to see design as description. The human activity approach shares this type of conclusions with the approaches of Winograd & Flores, H. & S. Dreyfus, and Ehn.[4]

The human activity theory has some far-reaching consequences for the relation between the use directed activities of design and the more technical activities, the implementation of the artifact: in traditional systems design practice the use directed activities get structured in a way which more or less ensures the implementability of the envisioned artifact. The focus of the human activity approach on the practice of use does not provide this "automatic" relation, for which reason the technical considerations are not supported directly.

With these more general conclusions, let's move on to some more specific ones. The main purpose is to give specific answers to what has been achieved by the human activity approach. For the sake of completeness, I shall add that some of these conclusions are common also to more of the above approaches.

The user interface

The relation between the user interface and the specific type of use activity is demonstrated in the following conclusion: *The user interface cannot be seen independently of the goal or object, or of the other conditions of the use activity.*

There are two types of consequences of this, one concerning the use of general recommendations or guidelines for the user interface, and one concerning the possibilities of assessing the user interface:

1. *We can give certain general recommendations for the user interface, which are based on general cultural characteristics of the human use of computers. We have, however, no warranty that such recommendations are applicable in the specific case.* This means that we can try to apply general principles in the specific case, but that the evaluation of the user interface may result in giving up on some of those general principles.

2. In the evaluation or assessment we must be aware of that: *The user interface is only revealing itself, fully, to us in use.*

The following three statements define the user interface:

1. *The user interface constitutes the conditions for operations, which are determined through the artifact. Furthermore, by the support or prevention of certain operations, the user interface constitutes conditions for possible actions.*

2. *The user interface consists of conditions for operations towards the artifact and for operations towards subjects or objects through the artifact.*

3. *The user interface consists of physical aspects, handling aspects and subject/object directed aspects.* The physical aspects support the physical adaption of the user to the artifact, the handling aspects the operations directed towards the artifact, and the subject/object directed aspects the operations towards an object or a subject through the artifact.

In Chapter 4 we have seen that although it is the purpose of the handling aspects to avoid breakdowns which bring the artifact into the awareness of its user, it is also important to anticipate such breakdowns and make it possible for the user to shift her focus back to the object or subject. This supports, and gives new reasons for, the following conclusion:

The user must be able to handle the breakdowns with regards to the artifact within the domain of use. Support in these situations is part of the handling aspects, but also a matter of education.

We have also seen a close connection between the competence of the users and the user interface; competence both in a narrow sense of operating the artifact, and competence more widely in relation to the practice of the user: *What is a good user interface at one level of competence can prevent efficient use at another level, this being a higher or a lower level of competence.* It is well-known that user interfaces designed for experts can be difficult to use for novices, but in Chapter 4 I have shown examples of the opposite as well, *that user interfaces designed for novices can prevent efficient use by the experienced or expert user.*

From the discussions about learning it seems that user interfaces based on physical objects, where the user can actually see and touch the real, physical objects, seem easier to learn than user interfaces based on "tangible" graphic objects, such as the Macintosh interface. These are, in turn, easier than more abstract or language based user interfaces, of course depending on the specific conditions.

No matter which type of user interface chosen, when applying an artifact, some of the user's actions and operations are always instrumental. The physical aspects as well as the handling aspects support instrumental actions and operations. Natural language interfaces are aiming to make instrumental aspects, i.e. handling aspects, communicative. In my concepts, this is impossible or self-contradictory. If, however, the goal of the activity is achieved through communication, the communicative side is supported through the subject directed aspects.

The subject/object directed aspects support communicative operations and actions when directed towards subjects, respectively instrumental operations and actions when directed towards objects.

In Chapter 4 I have demonstrated that it is possible to define the user interface technically in a way that is complementary to the use derived definition given here.

Such terms as flexibility, consistency and simplicity can be defined in relation to the use activity. *Flexibility* has to do with the possibilities of shifting focus among objects and subjects, but also with the possibilities of achieving the same goal by different paths (different actions and operations). *Consistency* means that the computer responds to the actions of the user in the same way in situations that are the same to the user. *Simplicity*, to how many actions it takes for the user to achieve a certain goal.

From the comparison of the text editors Microsoft WORD and MacWrite we know that simplicity in the handling aspects might contradict flexibility in the subject/object directed aspects.

The design activity

The design activity, as viewed by the human activity approach, and in line with, e.g., Ehn or Winograd & Flores,[5] is characterized by a conflicting meeting between different practices, different needs for articulation of operations and their material conditions, etc. In this meeting, design becomes rooted in existing use practice, at the same time as it becomes possible to take steps to change the practice of use by means of the practice of the designers and groups of users.[6] *In design we must handle different practices, at least the practice of the users and that of the designers.*

Only users, i.e. human beings who share the practice of the group of future users, can evaluate all aspects of the user interface in design. This is quite a serious and challenging problem for design – to deal with a not yet existing practice. One place to start is to let the users and their practice be the origin for design.

In design of the user interface we must be able to handle all aspects of practice, the articulable, as well as the non-articulable, the instrumental as well as the communicative side.

User interface design is not something which is to take place late in the design process after all the important decisions, but something which is going on all the time throughout the design process. From the point-of-view of use, user interface design is the main activity in design, together with design of other physical and social surroundings. The technical description of the artifact must be derived from this. Oppositely to what many cognitive psychologists say,[7] user interface design cannot be conducted independently of the rest of the design, by "user interface experts", because this prevents the mutual learning process. This does not, however, exclude the need for such competence in design, only must the work of the "user interface experts" be integrated with the rest of the design process, also with the cooperation with users.

Design means conceptualization of former operations and creation of new ones. Furthermore, design may mean automation of former human operations. Design deals with operations and the conditions by which they are triggered. We design new conditions for the collective as well as the individual activity.

As design originates from the practice of use, *the design activity must be structured according to the use practice and not according to the technical components of the user interface*[8] or any other abstract or formal framework.

The human activity approach supports the idea that design is both construction of the future artifact and communication about it. The materialized visions constructed in the design activity are means of triggering conceptualization about present or future practice, especially operations: *Design of user interfaces is a process in which breakdowns serve to detect problems of the future use. In the design activity we try to anticipate breakdowns all the time: the design activity makes use of breakdowns at the same time as the aims are to create a situation with no breakdowns in the final use. If successful, design leads from many to few breakdowns in anticipation and use of the future artifact.*

We can distinguish between different types of triggers which help approach or simulate the meeting with the material world in different ways:

- *Scenarios* can be used to draw attention to the present practice of the users, and to the aspects of this which will be changed due to the introduction of a new artifact. Scenarios are meant to be evaluated by reading. They cannot necessarily draw attention to how aspects of practice will be changed.

•*Prototypes* can be used to let the user try out actions and operations, in a real or simulated setting, and by this experience (aspects of) the meeting with the material world.

Design methods

Design methods can be either prescriptions of how a total design process is to take place, or prescriptions which aim at a specific part of the design activity.

Methods can be characterized according to how they aim to create artifacts to *support instrumental actions and operations, or communicative.* Furthermore, one can distinguish between methods which aim at artifacts for collective activities or for individual activities only. According to the discussions of Chapter 5, where a method places itself according to these distinctions has consequences for both the physical aspects, the handling aspects and the subject/object directed aspects of the user interface: how much it is possible to focus on them and how they are dealt with.

Methods can aim at exploiting the inherent capabilities of computers differently: *products of the design activity are more or less active or passive externalized artifacts.*

We can make the following conclusions about the methods examined in Chapter 5:

The established methods for design of computer applications in general, and for user interface design, do not intend to originate from the practice of the users. They are based on a detached observation and description of the work activity to be changed by the new artifact, and they apply some formalism of description in this description.

Description methods, such as those of Yourdon or even Card, Moran & Newell[9] *are not sufficiently capable of dealing with the user interface,* because they do not allow communication about the non-articulable aspects of practice, the meeting with the material world.

It is important to get to methods which take into account the need to distinguish between mediation of the inter-human relation and of the relation between human beings and things. Furthermore, we need to get to methods which allow for the inclusion of different aspects of practice in design: the articulable, as well as the non-articulable, the instrumental as well as the communicative side. Last, but not least, however, the approach gives support for and emphasis on the following conclusion:

Good design methods must prescribe that the means applied in a specific design activity must originate from the use activity in question. We cannot give one method to be used in design of user interfaces for all types of applications.

Recommendations to the designer

In the following I shall present and discuss some recommendations to the designer. The recommendations are primarily meant to present the specific recommendations of the human activity approach, but again, for reasons of totality, I have included a couple of more general recommendations, which in this dissertation are elaborated on as part of the approach. First I shall present and discuss the recommendations one by one; for the user interface, for the design activity, and for design methods.

The user interface

I shall first present two main statements about quality of user interfaces. These statements arise directly from the definition of the user interface and the role of the artifact in use:

A good user interface allows the user to conduct an activity as different actions and operations depending on the user's repertoire of operations and the actual material conditions.

With a good user interface, neither the physical aspects nor the handling aspects give rise to actions in ordinary use situations.

Looking more specifically at the physical aspects and the handling aspects, we have seen examples in Chapter 4 of how different design of these aspects can be used to avoid actions towards the artifact, i.e. to prevent breakdowns:

Delay in response must follow the actions of the user, and should not appear in the middle of an operation.

Prompts must be used with care: they are useful in guiding a user through unfamiliar actions, but may create breakdowns when appearing in the middle of operations.

Although the physical and handling aspects should not attract the actions of the user it is also important that they support the user if a breakdown occurs. How this is done has close relations to the competence and education of the user, but it is important that *error situations must be handled within the domain of use practice.*

It is important that *the user is able to regret or undo* her operations, if this is important in the handling of the objects or subjects. Proper facilities can prevent breakdowns towards the artifact.

Help mechanisms, error handling mechanisms, etc., which *actively* aim at predicting or explaining the actions and operations of the users, are based on a formalization of the conditions for the triggering of certain operations and may spoil existing operations by not acting according to the experiences of the user each time. Help mechanisms, error handling mechanisms, etc., must be under control of the user.

Turning to the subject/object directed aspects, the main recommendation is: *For all the objects and subjects of use, the subject/object directed aspects must support the development of operations.* The subject/object directed aspects have different characteristics depending on whether they are support for actions and operations towards subjects, or towards objects: *support the use of instrumental actions and operations towards objects, and communicative actions and operations towards subjects.*

Furthermore, *the subject/object directed aspects must support ease of shifting between subjects/objects, where this is necessary in the future use activity.* This means, e.g., that modes should be used with care, and that *the handling aspects of the artifact should be consistent for different subjects/objects.* This does not mean that it is possible to handle the artifact in the same way always, but that the same type of situations should be handled the same way.

The design activity

The recommendations for the design activity have a different character than the recommendations for the user interface. Because they are recommendations for actions they are dos and don'ts coming out of the discussions in the previous chapters.

What is important in design, the way this is dealt with by the human activity approach, is the appearance of the computer based artifact to its users in use, on the individual as well as the collective level. The operational aspects of the artifact are in focus, what we call design of the user interface.

We cannot design the user interface after determining a set of actions to be conducted, a set of intentions (what is often called the functionality), although we have some intentions in mind when initiating the design activity. Rather the physical aspects, the handling aspects, and the subject/object directed aspects must all be dealt with equally in design. Out of this we can, by after-design reflection, determine the possible intentions. The physical aspects, handling aspects and subject/object directed aspects of the user interface are equally important and often interdependent: the handling aspects cannot be fully determined without the physical aspects, and the subject/object directed aspects not without the handling aspects, but at the same time, the subject/object directed aspects put specific requirements on the physical and handling aspects, etc. One of the consequences of this is that both hardware and software have to be considered throughout the process. Many design methods prescribe software to be considered before hardware whereas this dissertation shows that software and hardware are equally important for the appearance of the artifact, the user interface.

The human activity approach gives support for the following recommendations about the handling of practice in design:

To deal with a not yet existing practice, *the use practice must be the origin for design*. This in turn means to *involve users actively in design*.

The meeting of the practices of the designers and of the users is important for design but requires a process of mutual learning. Through this meeting the possibilities of transcending existing practice is achieved. The learning about other practices and the potential conflicts between the practices open up new possibilities for artifacts, cultural techniques, language, etc. which are not brought about through one practice alone. This process, as well as to deal with different aspects of the future use practice, requires new methods. In the discussions of this dissertation, possible candidates for such new methods have been discussed, but this area need to be worked on further.

The human activity approach gives some general recommendations for the design of user interfaces:

1. *Create possibilities for the users to try out the user interface through use, not only through reflection.*

2. *Anticipate the subjects and objects of the future activity, and be aware of the shifts between them in use.*

3. *Anticipate the levels of competence of the future use(rs), as well as the domains of competence.*

4. *Focus on the difference between frequent and less frequent situations of use to deal with flexibility, consistency and simplicity of the user interface.*

In connection with this, it is important that a person can be an expert or a novice within many different domains. As pointed out in Chapter 4 it is possible to become an expert MacWrite user, but not an expert document editor by using MacWrite. To create high-quality documents, however, it is not sufficient to be an expert MacWrite user.

Designing the user interface means determining the levels of competence where the artifact can be used without special problems (breakdowns). *Education needs to be designed* to make it possible for the users to get to these levels.

From the preceeding chapters we have seen that general guidelines and recommendations about the user interface can be applied to create *realistic* visions of the future artifact. The requirements of the specific use activity are, however, still more important and may overrule the general recommendations. Creating realistic visions is important in design, but *ideal* visions can play a role which is just as important, in enhancing the *imagination* of both users and computer experts.

Design methods

Which design methods to recommend for design of user interfaces from the human activity approach is still a research issue. In the next section I shall discuss some of the research to be undertaken to achieve better design methods. Here I shall give a brief summary of the state of recommendations.

The main conclusion is that professional designers need a tool box of different kinds of methods, some of which are support for refining other methods to a specific design situation. For a later, purely technical implementation of the final artifact this type of methods must be supplemented with, e.g., programming and technical description methods, an issue which has not been touched upon in this dissertation.

Both the mock-up, the exploratory programming, and the 4'th generation tool methods can be included in a tool box of design methods which the professional designer has access to.

Mock-ups and computer based prototypes can be applied to simulate the meeting with the material world.

Better mock-up and computer based prototyping artifacts can be achieved by the focus on a specific application domain and use practice(s). For this reason the general tool box must provide basic components which support different styles of interaction, and a variety of hardware components, e.g. pointing devices, etc. Various techniques which can help the designer use a computer to simulate a user interface can be useful, for example by showing a sequence of screen images one after the other as pictures, or even as a kind of movie. The showing of such sequences could even be influenced by the steps taken by the user in the interaction. The tool box must make it possible for the designer to adjust these general components to the domain of application in an actual design situation. Also it must be possible for the designer to use exploratory programming techniques to change the components of the tool box.

To support the designer's practice in developing such methods, a set of concrete examples can be given as education material.

There is a tendency that the more advanced the computer support for the design activity is, the more specific it is concerning the types of user interfaces that can be developed. Therefore, *in the early stages of design, choose design methods which are as general as possible when it comes to which kind of user interface to end up with. Later on, more specific techniques and artifacts can be chosen.*

Scenarios can be used to bring about an awareness of the present practice of the users, and through this of possible changes. The tool box may benefit from being supplemented by different scenario methods.

The potential of the framework

The next step of exploiting the framework and the above recommendations is to get to some more operational design methods, and prove, in practice, that they are better than the methods that we know already. In Chapter 5 and above I have pointed at directions where I think we must look for such new methods. I find it beyond the possibilities of this dissertation work to create and test concrete methods that will take us in these directions. The future project, discussed below, can hopefully contribute in this area.

A future project?

The ideas of the human activity approach are included in plans for the following project, called the application simulator. The project is part of an overall programme, which deals with computer support in cooperative work.[10]

The main idea of the application simulator project is to develop a general, computer based prototyping environment to be applied in design of single-user as well as multi-user situations. The application simulator is to be used in the early stages of design to facilitate 'design by doing'.

We envision an application simulator, by means of which prototypes can be built of pluggable standard components and pieces of code written in a high-level language. The pluggable components range from simulation of a slide projector, 4'th generation type components, support for simulating pieces of underlying programs, support for simulating the capabilities and style of different computers, etc. All of these components can be changed and adjusted to the specific use situation. Furthermore the use of video signals and the like will be considered as means of simulating parts of a computer application, and likewise the possibilities of letting a human being simulate part of the computer's part of the interaction.

Surrounding this application simulator we envision also that prototypes of total use situations can be set up, e.g. plays in which the computer based prototypes are applied. Not only the application simulator, but also design methods surrounding its use is to be developed in the project.

The technical design of the application simulator is a research challenge, and so is the study of which types of products that this type of design strategy will lead to: from a human activity point-of-view is important to deal with the physical aspects, as well as the handling and subject/object directed aspects. It becomes a research challenge to develop methods to explore all three aspects of the computer application, and the relations between them. Similar considerations goes for the relation between the collective activity and the individual activity.

It is also necessary to work theoretically and practically with prototyping in a multi-user setting, and with requirements posed by different types of application domains.

According to the human activity approach, the application simulator can support certain general recommendations, but not enforce them. A consistent framework for the application of such is another research issue.

We do not yet have any general conception of the initial flexible prototypes, where imagination of the users as well as the professional designers is important, or the later, more application specific prototypes should look, only examples (mock-ups, 4'th generation style prototypes). It is not as much the idea of the application simulator to help the programmer create the inside structure of the final programs, and it is a point that the functionality need not be fully evolved to give an impression of how the computer application will appear to its user. There may still be much work for programmers to turn a prototype into a real computer application. We have no experiences in how such a prototype works as a requirement specification, and we need perhaps look for new methods for programmers to use to get from the prototype to a running computer application.

From ideal to reality?

Above we have seen that the human activity approach plays a role, both in refining concepts of user interface design, and as an ideal or a prototypical case, in suggesting how design is to take place.

The approach is normative because it states that if we want to design computer based artifacts, there are some conditions under which design needs to take place. I have discussed the ideal setting for design as consensus groups, but real life design do rarely take place in this kind of setting. It is obvious that the human activity approach can be used, and misused, depending on the political conditions surrounding design. I take the relativist position in Hirschheim's[11] meaning of the word, I am optimistic about the possibilities of making better computer applications, but at the same time sceptical to how they may be used. At least some negotiations and resources are required to set up the required type of design situations, but often the situation is much more complicated and requires the handling of different interests, etc. How this can be done, how, e.g., technology agreements can be set up to regulate evolutionary or prototypical design, how the size, timewise and resourcewise, can be estimated for this type of projects, and many similar problems need to be dealt with if we want to see better design taking place in real life situations in the future. This is in general not a matter of individual idealism, but of collective bargaining and changing societal conditions.[12]

Despite this, I hope that I have come up with reasons and ideas for changing practice to make better user interfaces; some ideas to how better design can take place, even in industry. Hopefully, designers will be able to make use of my recommendations to reconsider their practice regardless of the political conditions, because they want to do better design. Even more so, I hope that the application simulator will contribute to this, and that we will be able also to include aspects of the political conditions in the more long-term parts of the project, because this new type of design will, in the end, require new types of strategies for the involved parties, unions and management.

Design is where the action is in the user interface

I chose the quote 'Design is where the action is in the user interface' by Allen Newell as the opening line of this dissertation. This was done to stress the fact that user interface design is an important area, where much research needs to be done, and to acknowledge the opening from cognitive science towards design.

This dissertation has, however, questioned whether the path of cognitive science is the right one, and even so there seems to be more important problems with most of the traditional design approaches. If we stay with Newell, his concern is for how cognitive scientists/user interface experts get the proper influence on the computer application, and he concludes that when the functionality has been designed they must take over and act in design. With the human activity approach, this type of consideration is turned upside down: it is not a matter of designing the user interface after the functionality, but of getting to a design approach where it is almost possible to design the appearance of the artifact to the users in use, including also the social and physical surroundings of the artifact use. Design of any other aspect or issue, such as technical features, must be derived from this. The human activity approach has resulted in an understanding of computer applications where the user interface plays a much more fundamental role. A formulation of the ending line could be the following:

The user interface is where the action is in design.

---

[1] A. Kay: **Microelectronics and the Personal Computer**, Scientific American 237, 1977.

[2] There are situations in use where the computer based artifact becomes the object of use, these are exactly the situations where the user interface fails. Furthermore, there are some design situations where the artifact to be designed is applied at the same time, e.g. the use of a programming environment to design the same programming environment.

[3] **Proceedings from the Conference on Computer-Supported Cooperative Work**, Austin, Texas 1986.

[4] T. Winograd & C. F. Flores: **Understanding Computers and Cognition: A New Foundation for Design**, Ablex Publishing Comp. 1986, H. & S. Dreyfus: **Mind over Machine**, The Free Press 1986, P. Ehn: **Human Centered Design and Computer Artifacts**, Aarhus forthcoming.

[5] Ehn or Winograd and Flores, ibid.

[6] If the users are also the designers, this is at the same time good and potentially bad; good because it is easy to start out from the practice of the users, potentially bad because there are no outsiders to look through the blindness created by the common background of the designers/users.

[7] See e.g. S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum 1983 or D. A. Norman & S. W. Draper, ed.: **User Centered System Design**, Lawrence Erlbaum 1986.

[8] This goes not only for the design activity. The user interface need to be structured according to the use activity as well.

[9] E. Yourdon: **Managing the Systems Life Cycle**, Yourdon Press 1982 or Card et al. op. cit. (note 7) despite their initial talk about trying out prototypes.

[10] The programme is a long term multidisciplinary research effort planned jointly by the Computer Science Department and the Institute of Information and Media Science, Aarhus University.

[11] R. A. Hirschheim: **The Effect of A Priori Views on the Social Implications of Computing: The Case of Office Automation,** Computing Surveys, 18, 2, 1986.

[12] This is a place where I think Hirschheim, ibid. is wrong in his view of the Scandinavian trade union approach (by Ehn & Kyng called the collective resource approach, see P. Ehn & M. Kyng: **The Collective Resource Approach to Systems Design** in G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge,** Gower 1987). He is right when he calls the tradition a relativist position in the presented meaning of the word, but when he characterizes this position as dealing with freedom, affection, recognition, etc. of the individual, he forgets the fundamental distinction, which I discussed in Chapter 1, between idealism and materialism.

Bibliography

1. C. Andersen et al.: **SYSKON - en bog om konstruktion af datamatiske systemer**, part 1 and 2, G.E.C. Gads Forlag 1972 (In Danish. SYSKON - a book about construction of computer systems)

2. N. E. Andersen et al.: **Professionel Systemudvikling**, Teknisk Forlag 1986 (In Danish. Professional Systems Development)

3. L. Bannon: **Computer-Mediated Communication** in D. A. Norman & S. W. Draper (60)

4. M. Bartholdy et al.: **Studie av Datorstödd Bildbehandling på Aftonbladet**, UTOPIA report no. 21, 1986 (In Swedish. A Study of Computer Based Image Processing at 'Aftonbladet')

5. G. Bjerknes et al., ed.: **Computers and Democracy – a Scandinavian Challenge**, Gower 1987

6. L. Borman & B. Curtis, ed.: **Human Factors in Computing Systems**, Proceedings, ACM 1985

7. P. Bøgh Andersen: **Edb-teknologi set i medieperspektiv**, The Joint Studies of Humanities and Computer Science, University of Aarhus 1984 (In Danish. Computer technology seen in a media perspective)

8. S. Bødker & J. Hammerskov: **Grafisk Systembeskrivelse**, DAIMI IR-33, IR-34 and IR-35, University of Aarhus 1982 (In Danish. Graphical Systems Description)

9. S. Bødker & J. Kammersgaard: **Interaktionsbegreber**, working paper, The Computer Science Department, University of Aarhus, 1984 (In Danish. Interaction Concepts)

10. S. Bødker et al.: **Graffiti 7. The UTOPIA project. An Alternative in Text and Images**, Arbetslivcentrum 1985

11. S. Bødker et al.: **A Utopian Experience**, in Bjerknes et al. (5)

12. S. K. Card et al.: **Evaluation of a mouse, rate controlled isometric joystick, step keys, and text keys for text selection on a CRT**, Ergonomics 21, 8, 1978 pp. 601-614

13. S. K. Card et al.: **The Keystroke-Level Model for user performance time with interactive systems**, Communications of the ACM 23, 7, 1980

14. S. K. Card et al.: **The Psychology of Human Computer Interaction**, Lawrence Erlbaum 1983

15. J. Caspersen et al.: **Description Techniques for Interactive Systems**, Ålborg Universitetscenter 1985

16. C. W. Churchman: **The Systems Approach**, Dell Publishing Co. 1968

17. M. Cole & J. Maltzman, ed.: **A Handbook of Contemporary Soviet Psychology**, Basic Books 1969

18. T. DeMarco: **Structured Analysis and System Specification**, Yourdon Press 1979

19. H. & S. Dreyfus: **Mind over Machine**, The Free Press 1986

20. W. Dzida: **The IFIP Model for User Interfaces**, GMD-F2G2, GMD Bonn (no date)

21. P. Ehn: **Human Centered Design and Computer Artifacts**, Aarhus forthcoming

22. P. Ehn et al.: **Training, Technology, and Product from the Quality of Work Perspective**, A Scandinavian research project on union based development of and training in computer technology and work organization, especially text and image processing in the graphic industry. (Research program of Utopia) UTOPIA report no. 2, 1981

23. P. Ehn et al.: **Utformning av Datorstödd Ombrytning för Dagstidninger**, UTOPIA report no. 12, 1984 (In Swedish. Computer-aided Page Make-up for Newspapers)

24. P. Ehn & M. Kyng: **A tool perspective on design of interactive computer for skilled workers**, in M. Sääksjärvi, ed.: **Proceedings from the Seventh Scandinavian Research Seminar on Systemeering**, Helsinki 1984

25. P. Ehn & M. Kyng: **The Collective Resource Approach to Systems Design** in Bjerknes et al. (5)

26. P. Ehn & Å. Sandberg: **God Utredning** in Å Sandberg, ed.: **Utredning och Förändring i förvaltningen**, Liber 1979 (In Swedish. Good Investigation)

27. G. Fischer et al.: **Active Help Systems**, in Cognitive Ergonomics, Mind and Computer. Proceedings of the Second European Conference on Cognitive Ergonomics, Mind and Computer, 1984

28. G. Fischer et al.: **Knowledge-based Help Systems**, in L. Borman & B. Curtis (6)

29. C. Floyd: **A Systematic Look of Prototyping** in R. Budde et al., ed.: **Approaches to Prototyping**, Springer Verlag 1984

30. J. D. Foley & A. van Dam: **Fundamentals of Interactive Computer Graphics**, Addison-Wesley 1982

31. P. Y. Gal'perin: **Stages in the Development of Mental Acts**, in M. Cole & J. Maltzman (17)

32. A. Goldberg: **SMALLTALK-80. The Interactive Programming Environment**, Addison-Wesley 1984

33. A. Goldberg & D. Robson: **Smalltalk-80. The language and its Implementation**, Addison-Wesley 1983

34. B. Göranzon et al.: **Datorn som Verktyg**, Studentlitteratur 1983 (In Swedish. The Computer as a Tool)

35. B. Göranzon, ed.: **Datautvecklingens Filosofi**, Carlsson & Jönsson 1984 (In Swedish. The Philosophy of Computer Development)

36. N. Hammond et al.: **Design Practice and Interface Usability: Evidence from Interviews with Designers** in Janda (40)

37. R. A. Hirschheim: **The Effect of A Priori Views on the Social Implications of Computing: The Case of Office Automation**, Computing Surveys, 18, 2, 1986.

38. E. Holbæk-Hanssen et al.: **System description and the DELTA language**, Norwegian Computing Center 1975

39. L.-C. Hydén: **Psykologi och Materialism. Introduktion till den materialistiska psykologin**, Prisma 1981. (In Swedish. Psychology and Materialism. An Introduction to materialistic psychology)

40. A. Janda, ed.: **Human Factors in Computing Systems**, Proceedings ACM 1983

41. U. Juul-Jensen: **Videnskabsteori 2**, Berlinske Forlag 1973 (In Danish. The Philosophy of Science)

42. J. Kammersgaard: **On Models and their Role in the Use of Computers**, Preceedings of the Aarhus Conference on Development and Use of Computer Based Systems and Tools, Aarhus 1985

43. J. Kammersgaard: **Four Different Perspectives on Human-Computer Interaction**, DAIMI PB-203, University of Aarhus 1985 (to appear in the International Journal of Man-Machine Studies)

44. B. Karpatschof: **Grænsen for automatisering**, Psyke og Logos 2, 1984. (In Danish. The Limit of Automation)

45. A. Kay: **Microelectronics and the Personal Computer**, Scientific American 237, 1977

46. A. Kay & A. Goldberg: **A personal dynamic media** in A. I. Wasserman, ed.: **Software Development Environment**, IEEE 1981

47. D. E. Knuth: **TEX and METAFONT. New Directions in Typesetting**, Digital Press 1979

48. G. Krasner, ed.: **SMALLTALK-80. Bits of History. Words of Advice**, Addison-Wesley 1983

49. B. H. Kristensen et al.: **Retningslinier for valg af faglige strategier på kontorområdet - et case studie over Århus tekniske Skoles kontorautomatiseringsprojekt**, Department of Computer Science, University of Aarhus 1986 (In Danish. Guidelines for trade union strategies in the office area - a case study of the office automation project of the Aarhus School of Polytechnics)

50. A. N. Leontjew: **Activity, Consciousness, and Personality**, Prentice-Hall 1978 (translated from Russian)

51. A. N. Leontjew: **Problems of the Development of the Mind**, Progress Publishers 1981 (translated from Russian)

52. A. N. Leontjew: **The Problem of Activity in Psychology**, in J. V. Wertsch (86)

53. M. Lundeberg et al.: **Systemering**, Studentlitteratur 1978 (In Swedish. Systemeering)

54. J. Lundequist: **Norm och Modell**, The Royal Institute of Technology Stockholm 1982 (In Swedish. Norm and Model)

55. L. Mathiassen: **Systemudvikling og Systemudviklingsmetode**, DAIMI PB-136, University of Aarhus 1981 (In Danish. Systems Development and Systems Development Method)

56. R. Miller: **Response time in man-computer conversational transactions**, Fall Joint Computer Conference 1968

57. T. P. Moran: **The Command Language grammar. A representation for the user interface of interactive computer systems**, International Journal of Man-Machine Studies 15, 1 1981

58. E. Mumford: **Sociotechnical Systems Design - Evolving Theory and Practice** in Bjerknes et al. (5)

59. W. M. Newman & R. F. Sproull: **Principles of Interactive Computer Graphics**, 2nd edition, McGraw-Hill 1979

60. D. A. Norman & S. W. Draper, ed.: **User Centered System Design**, Lawrence Erlbaum 1986

61. H. Oberquelle: **Semi-formal Graphic Modelling of Dialog Systems**, Universität Hamburg, FB Informatik, Bericht Nr. FBI-HH-B-113/85, 1985

62. H. Oberquelle et al.: **A view of human-machine communication and cooperation**, International Journal of Man-Machine Studies 19, 4, 1983

63. D. L. Parnas: **On the Use of Transition Diagram in the Design of a User Interface for an Interactive Computer System**, Proceedings. ACM 24'th National Conference 1969, pp. 378-385

64. G. Pfaff, ed.: **User Interface Management Systems**, Springer Verlag 1985

65. M. Polanyi: **Personal Knowledge**, Rutledge & Kegan Paul 1967

66. **Proceedings from the Conference on Computer-Supported Cooperative Work**, Austin, Texas 1986

67. P. Reisner: **Formal Grammar and Human Factors Design of an Interactive Graphics System**; IEEE Trans. on Software Engineering, SE-7, 1981, pp. 229-240

68. W. E. Riddle & R. E. Fairley, ed.: **Software Development Tools**, Springer Verlag 1980

69. T. L. Roberts & T. P. Moran: **The Evaluation of Text Editors: Methodology and Empirical Results**, Communications of the ACM 26, 4, 1983

70. B. Shiel: **Power Tools for Programmers**, Datamation 29, 2, 1983

71. H.-J. Schneider, ed.: **Formal Models and Practical Tools for Information System Design**, North-Holland 1979

72. B. Shneiderman: **Human factors experiments in designing interactive systems**, Computer, December 1979

73. B. Shneiderman: **Multiparty Grammars and Related Features for Defining Interactive Systems**; IEEE Transactions on Systems, Man, and Cybernetics, SMC-12, 1982, pp. 148-154

74. B. Shneiderman: **Direct Manipulation: A Step Beyond Programming Languages**, Computer, August 1983, pp. 57-69

75. SIS handbok 113: **Riktslinjer för administrativ systemutveckling**, SIS 1973 (In Swedish. Guidelines for administrative systems design)

76. M. Stefik et al.: **WYSIWIS Revised. Early Experiences with Multi-User Interfaces**, in **Proceedings from the Conference on Computer-Supported Cooperative Work** (66)

77. L. Suchman: **Plans and Situated Actions: The problem of human-machine communication**, Xerox ISL-6, 1985

78. L. Suchman & E. Wynn: **Procedures and Problems in the Office**, Office: Technology and People, 2, 1984

79. Y. Sundblad, ed.: **Quality and Interaction in Computer-aided Graphic Design**, UTOPIA report no. 15, 1987

80. P. Tanner & W. Buxton: **Some Issues in Future User Interface Management System (UIMS) Development** in G. Pfaff (64)

81. A. Thesen & D. Beringer: **Goodness-of-Fit in the User-Computer Interface: A Hierarchical Control Framework Related to "Friendliness"**, IEEE Transactions on Systems, Man, and Cybernetics, SMC-16, 7, 1986

82. J. C. Thomas, Jr.: **A design-interpretation analysis of natural English with applications to man-computer interaction,** International Journal of Man-Machine Studies 10, 6, 1978

83. O.K. Tikhomirov: **The Psychological Consequences of Computation** in J. V. Wertsch (86)

84. A. I. Wasserman: **USE: a Methodology for the Design and Development of Interactive Information Systems,** in H.-J. Schneider (71), pp. 31-50

85. A. I. Wasserman: **Software Tools and the User Software Engineering Project,** in W. E. Riddle & R. E. Fairley (68), pp. 93-113

86. J. V. Wertsch, ed.: **The Concept of Activity in Soviet Psychology,** M. E. Sharpe Inc. 1981

87. T. Winograd & C. F. Flores: **Understanding Computers and Cognition: A New Foundation for Design,** Ablex Publishing Comp. 1986

88. L. Wittgenstein: **Philosophical Investigations,** Oxford University Press 1953

89. E. Yourdon: **Managing the Systems Life Cycle,** Yourdon Press 1982

Appendix A

A brief description of MacWrite[1] and Microsoft WORD[2]

The idea of this appendix is not to give a thorough description of the two computer applications and their use, but rather to point at what are important characteristics of the two applications for the analysis of their user interfaces. The intention is not that the reader will be able to learn to use the two text editors from this description; better material exists for this. The main idea of this description is to provide the reader with an understanding of *how* it is to use the application. This can of course only be fully experienced through the reader's own use. I hope to be able to create some understanding of the user interfaces, also for Macintosh novices, even though this will be a theoretical understanding.

MacWrite and Microsoft WORD share many characteristics, because they are both designed to live up to the standard Macintosh user interface (figure 1 and figure 2).[3] They have a menu of pull-down menus at the top, a scrollbar with which to move around in the document to the right, and a relatively big window where the document is displayed in the middle of the screen, taking up most of the space. The pull-down menus work like the following: by selecting an item in the main menu a new menu, with different entries, opens for selection.

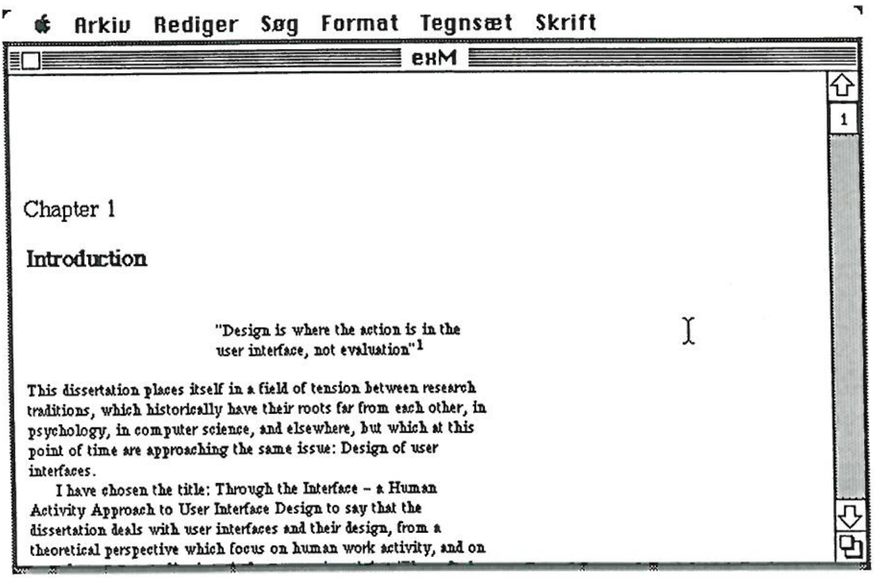


Figure 1. MacWrite with text window open, the menu of pull-down at the top, scrollbar to the right

The text editors both use the Macintosh input devices: the keyboard and the mouse with one button. Text can in both cases be entered by placing the cursor with the mouse, and typing (figure 3 and 4).

Cutting, pasting, and copying of text can be done by selecting text with the mouse and issuing a command from the edit menu (figure 5 and 6).[4]

Graphics can be pasted into the document and scaled/positioned, but the contents cannot be edited. The specific characteristics of the text editors are described in the following sections.

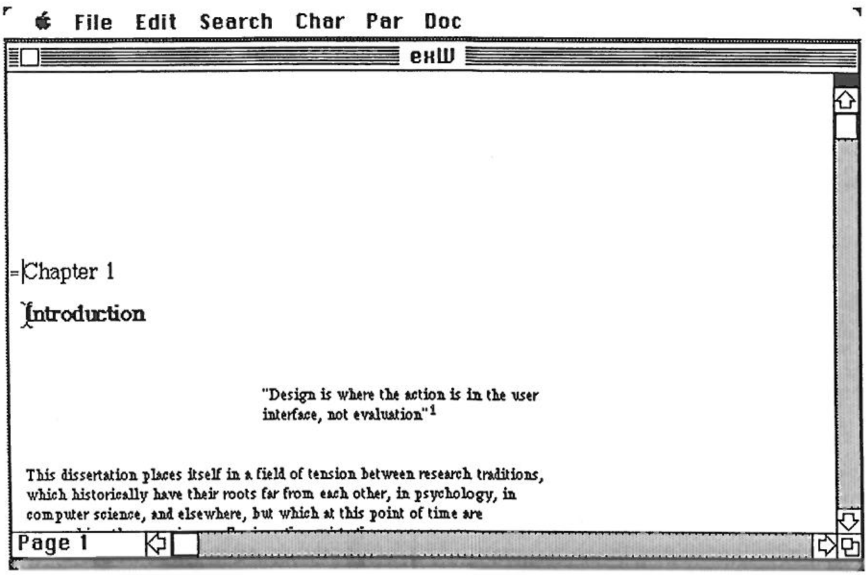


Figure 2. Microsoft WORD with text window open, the menu of pull-down at the top, scrollbar to the right and bottom

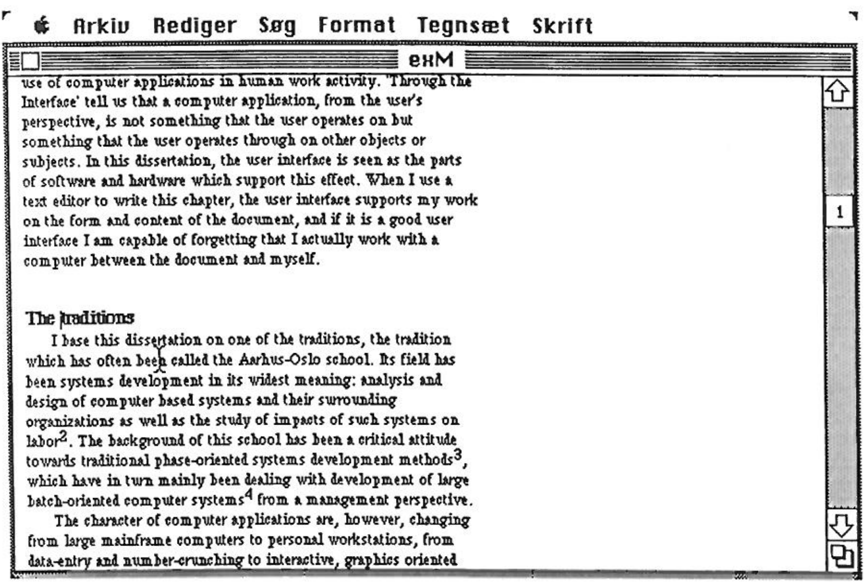


Figure 3. The text with which one works: the | indicates the point in the text where insertion of text is going to happen. This point has been selected by pointing with the cursor and pressing the button on the mouse.

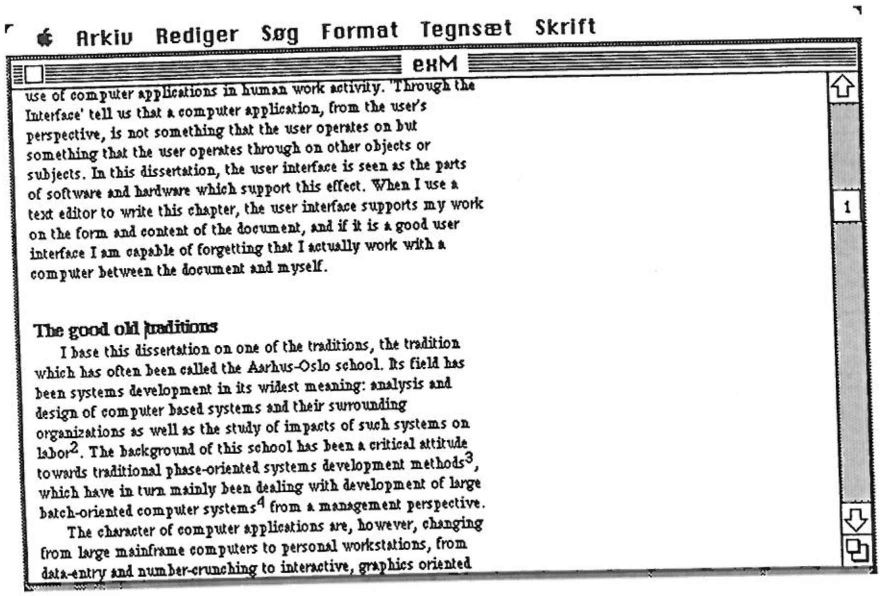


Figure 4. Text after insertion of some words ('good old') by typing

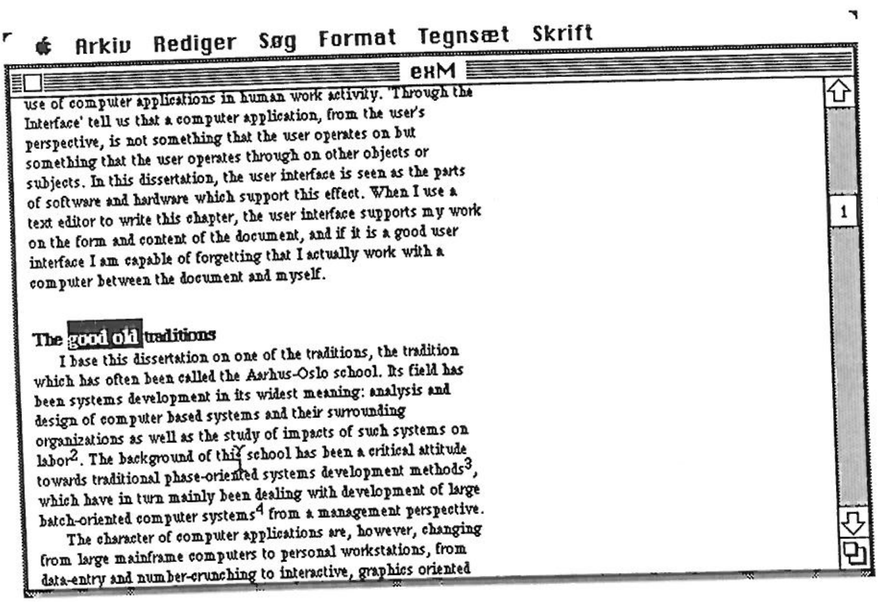
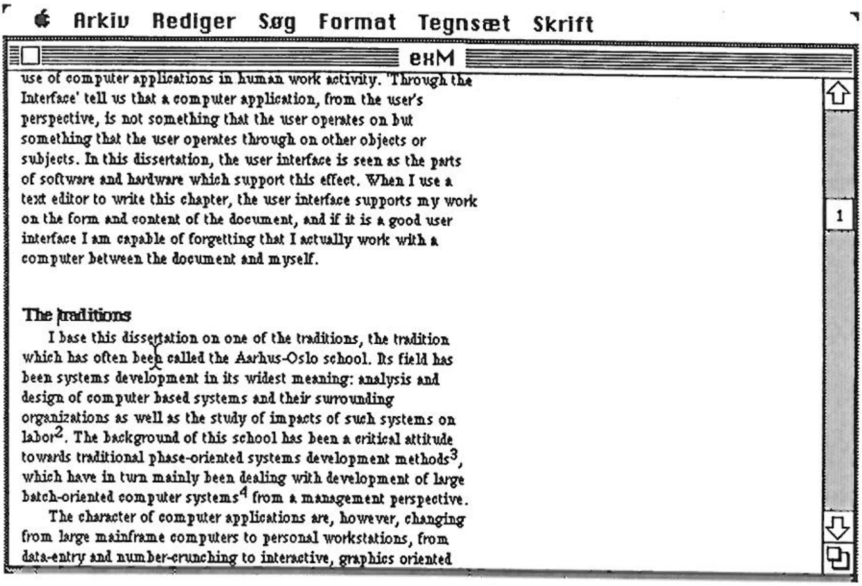


Figure 5. Selection of text by 'painting' with the mouse, the selected text is shown inversed



use of computer applications in human work activity. 'Through the Interface' tell us that a computer application, from the user's perspective, is not something that the user operates on but something that the user operates through on other objects or subjects. In this dissertation, the user interface is seen as the parts of software and hardware which support this effect. When I use a text editor to write this chapter, the user interface supports my work on the form and content of the document, and if it is a good user interface I am capable of forgetting that I actually work with a computer between the document and myself.

The traditions

I base this dissertation on one of the traditions, the tradition which has often been called the Aarhus-Oslo school. Its field has been systems development in its widest meaning: analysis and design of computer based systems and their surrounding organizations as well as the study of impacts of such systems on labor[2]. The background of this school has been a critical attitude towards traditional phase-oriented systems development methods[3], which have in turn mainly been dealing with development of large batch-oriented computer systems[4] from a management perspective.

The character of computer applications are, however, changing from large mainframe computers to personal workstations, from data-entry and number-crunching to interactive, graphics oriented

Figure 6. Text after the deletion of the words 'good old', by issuing the command 'cut' from the Rediger (Edit) menu, or by hitting the backspace key

MacWrite

The format of the document (margin width, etc.) can be changed by means of a ruler, which has some icons attached to it (figure 7). The format created this way is in effect until the next ruler appears or is placed. By pulling the margin marks of the ruler with the mouse we can change the margins like we do on a typewriter (figure 8). Tabs can be set the same way. The line width and justification can be changed by clicking the mouse button with the cursor pointing at the icon. The icons are quite self-explaining (figure 9). The latter can also be changed by entering one of the pull-down menus or by key-stroke commands.

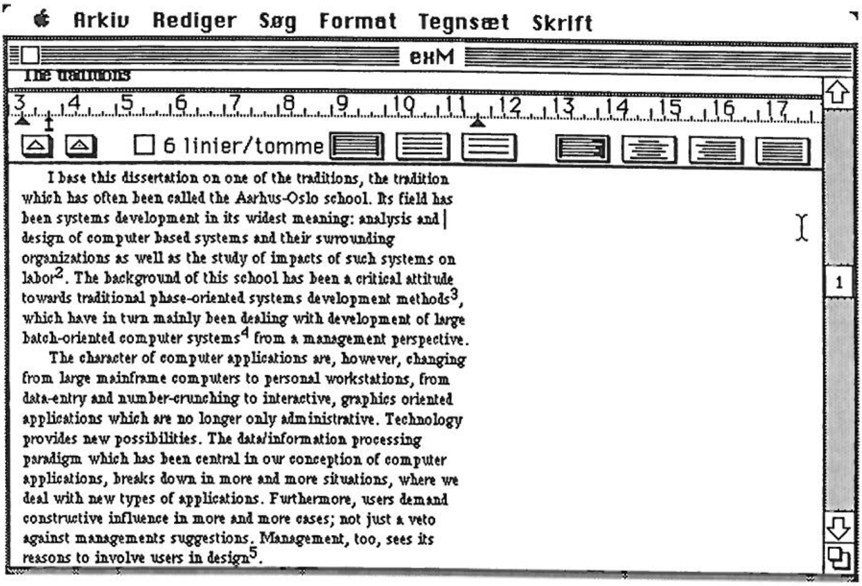


Figure 7. The ruler is shown at the top of the text window with the margin marks shown together with icons indicating linewidth and justification

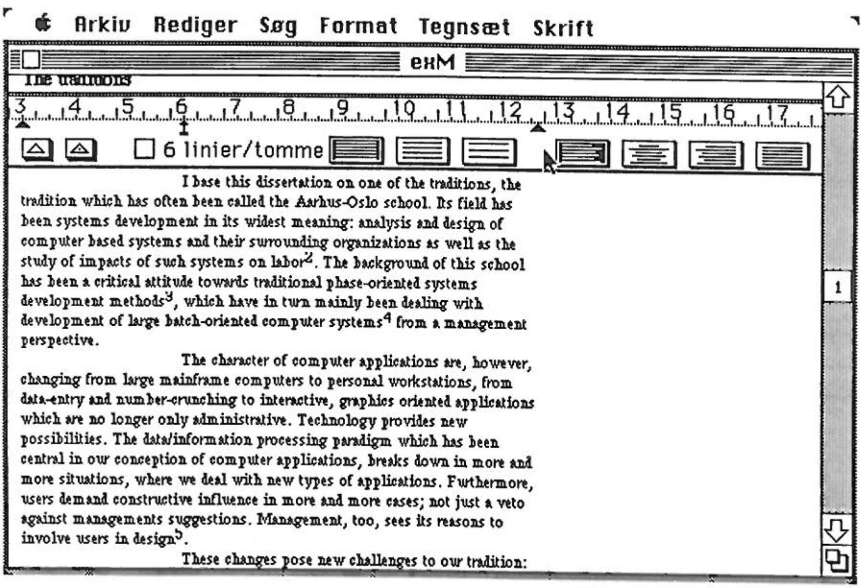


Figure 8. Format changed by pulling the margin marks, the left indention of the first line of a paragraph is now very big

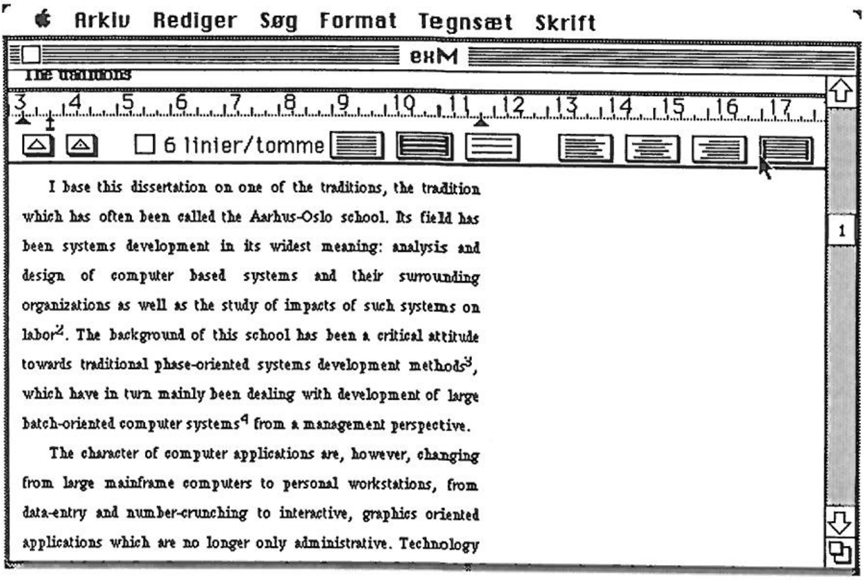


Figure 9. Change of linewidth and justification from single line spacing and left justification to one and a half line spacing and full justification

Fonts, font size and style are changed by selection in the 'tegnsæt' ('character') menu, and for the font style also as key-stroke commands. MacWrite operates with 6 fixed font sizes (9-24 points) independent of which sizes of the font are available on the computer (i.e. we can only choose "Boston" in 9-24 points even though the real font size is 36 points). Whether the font is available in the selected size or scaled is indicated in the menu by the use of different font styles. The font style menu items are icons which show the text with the specific style.

Headings and footings are created through a special window (figure 10), where the text is typed, and where page number and time stamp can be placed. After closing the window, the heading will appear as part of the text in the document (figure 11), although it can be changed only through the special window.

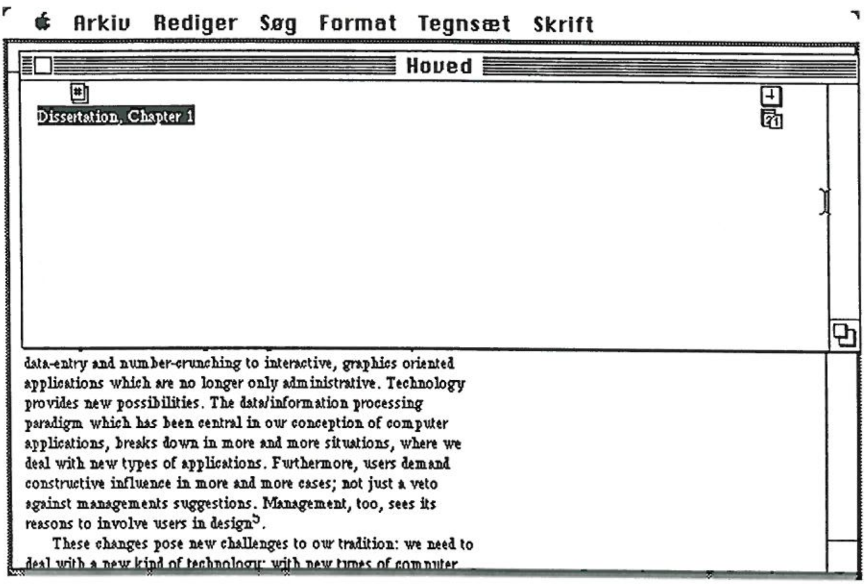


Figure 10. The special heading window is open on top of the text window

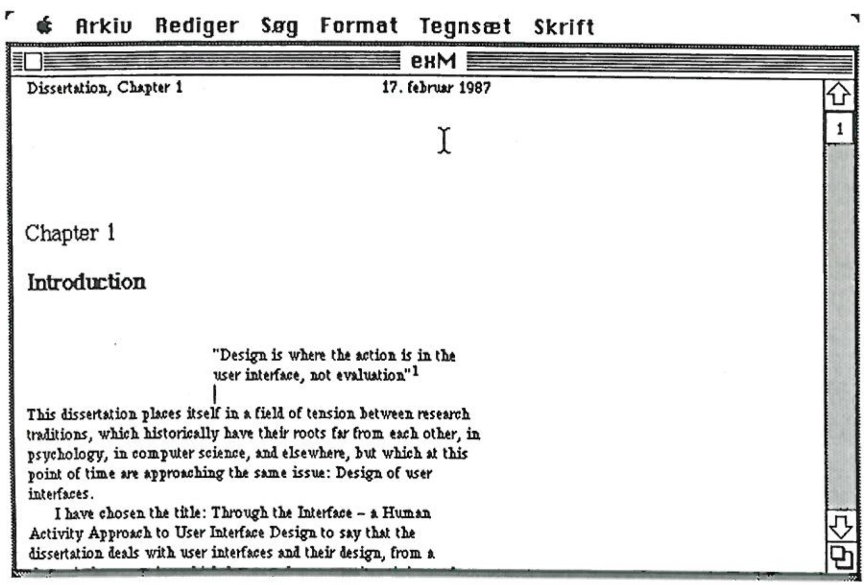


Figure 11. The main window after creation of a heading, and closing of the heading window

In the scrollbar of the document it is indicated on which page we work.

There is a limit to the size of the document of 8.5 pages, according to the manual. In the version applied here, the maximum size of a document appears to be something like 20 pages.

Microsoft WORD

The formats of the document can – through WORD – be changed in the following ways: The margin sizes can be adjusted both for the document in general (figure 12), and for individual paragraphs. For individual paragraphs a form sheet is used (figure 13). On this form sheet one can also specify the line spacing, not just in single/double spacing but in points as well as in other units. Extra space before and after paragraphs can be specified as well (figure 14).

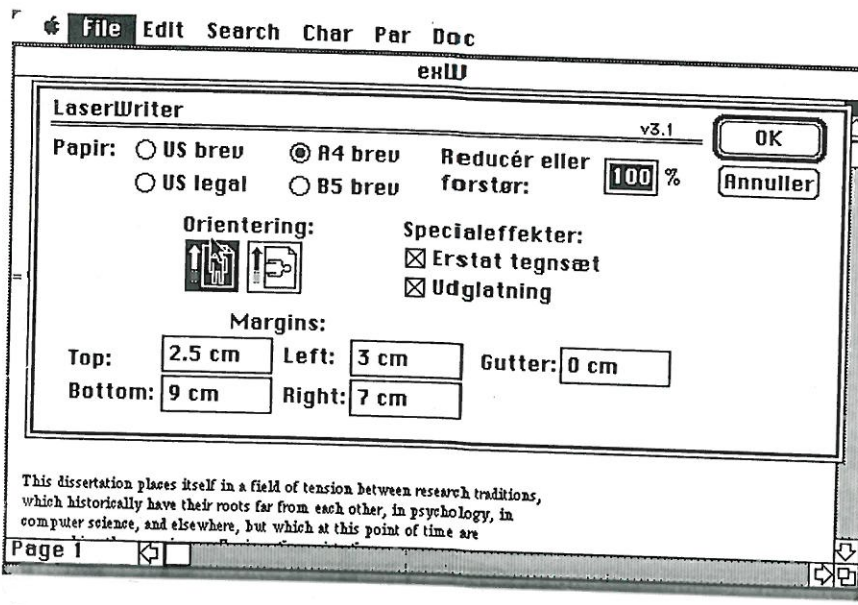


Figure 12. Setting up format for the document in general, applying 'page set-up' in the 'file' menu[5]

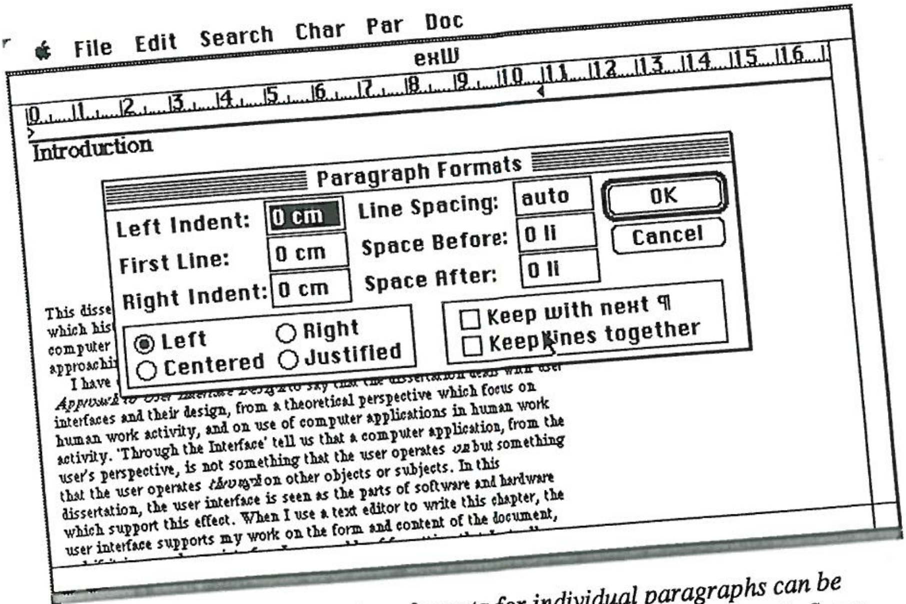


Figure 13. The form sheet where formats for individual paragraphs can be changed (within the limits of the format of the document in general). Some of the parameters can be changed by typing into the boxes (the top half), other are toggles which can be changed by clicking the mouse in the boxes (the bottom half)

Justification can be specified in the form sheet or directly in the menu. As in MacWrite there is also the possibility of using a ruler.

Paragraphs typed directly after the formatted one, get the same format, which can be changed separately. The ¶ is used to tell where a paragraph ends (¶ can be shown or hidden). The ¶, one way or another, contains the format information of the preceeding paragraph. One ¶ can be copied and pasted instead of another, whereby this paragraph gets the format of the former. A text with the ¶s visible is shown in figure 16.

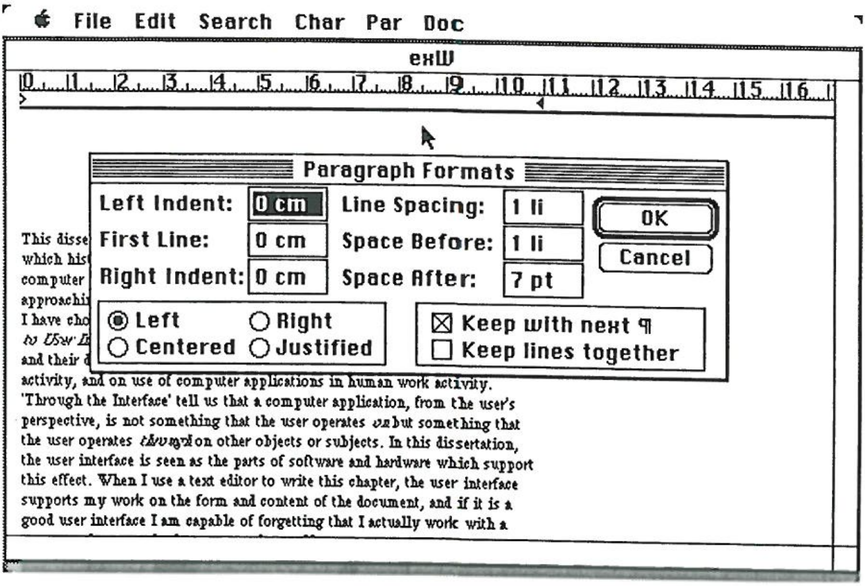


Figure 14. Some of the parameters have been changed, including those of line spacing and extra leading

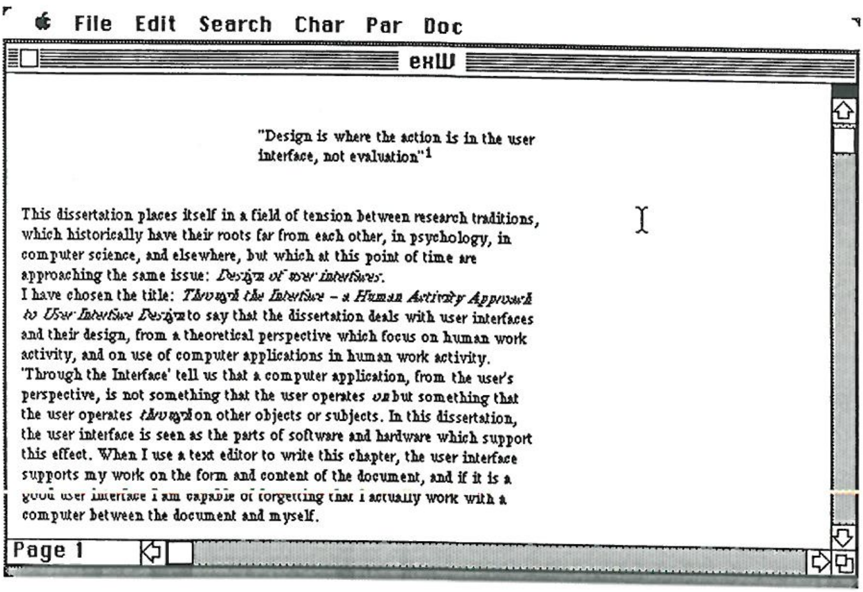


Figure 15. The text after formatting using the parameters of figure 14

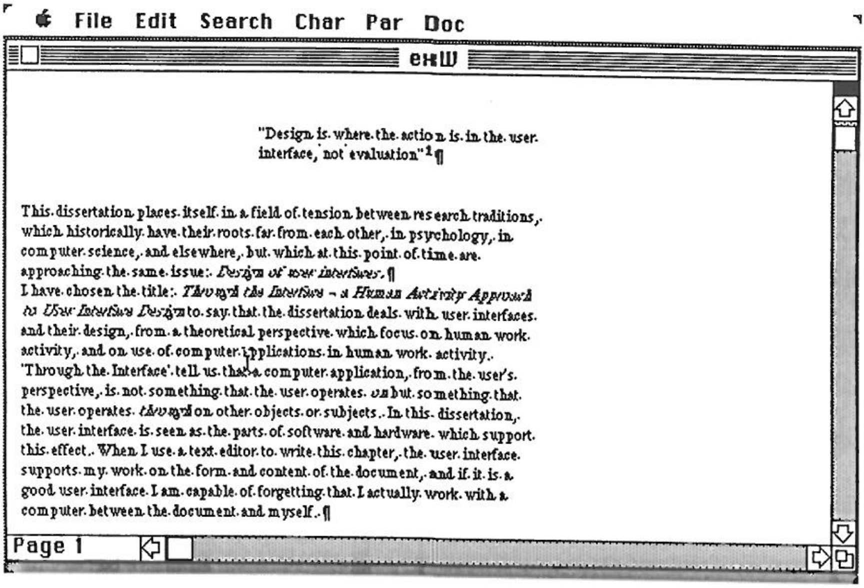


Figure 16. Text with ¶s shown, indicating the end of each paragraph in the document. The ¶s are shown by issuing the 'show ¶s' in the 'Edit" menu

Font style can be changed directly in the pull-down menu whereas fonts need to be changed via a sheet similar to the form sheet (figure 17 and 18). Font size can be changed in different ways, using this sheet or different key-stroke combinations. The available font sizes are shown to the user, but scaling to any point size can be made (between 4 points and 127 points).

Headings and footings are created typing a text, selecting it and selecting the 'running head' entry in the 'Doc' menu (figure 19 shows this and figure 20 shows the effect). The heading must be placed in absolute figures opposite to the rest of the measures in WORD, which are in relation to the margins of the pages. There is no way that one can see on the screen where the heading will be placed on paper. Automatic page numbers can be used.

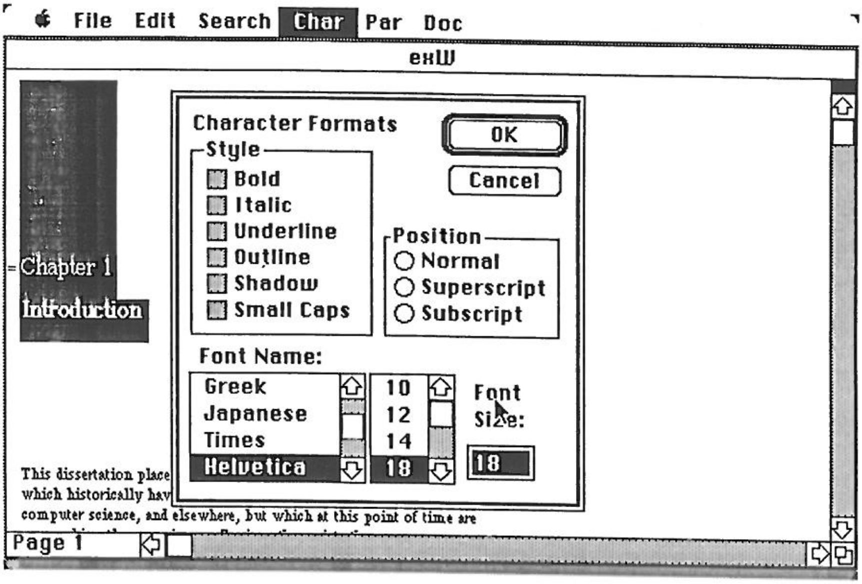


Figure 17. The character sheet which is used to change typography of text. The choice of style and position is done via toggles, whereas the font and size is chosen by pointing, or, for the size, by typing

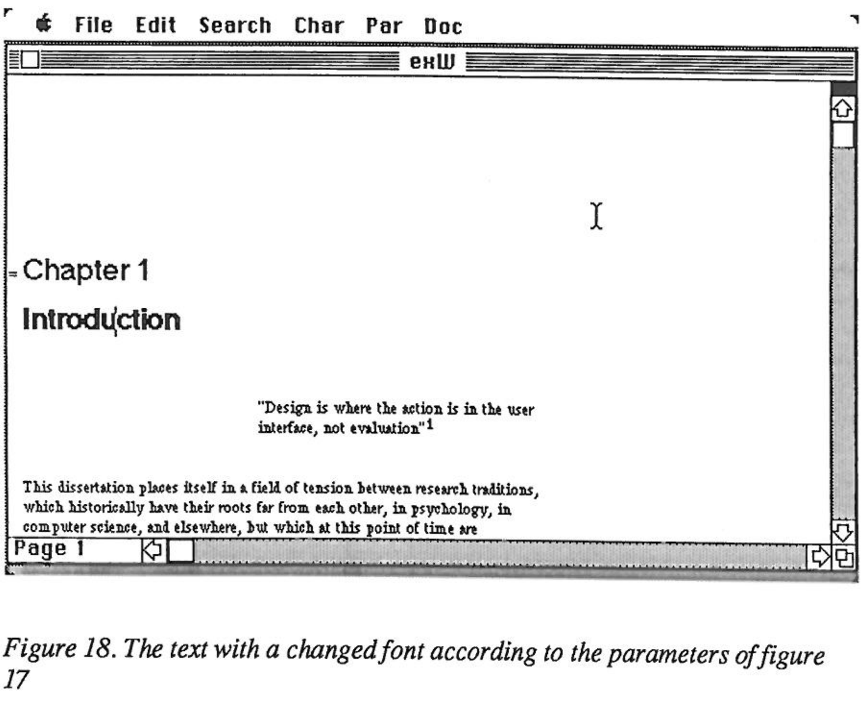


Figure 18. The text with a changed font according to the parameters of figure 17

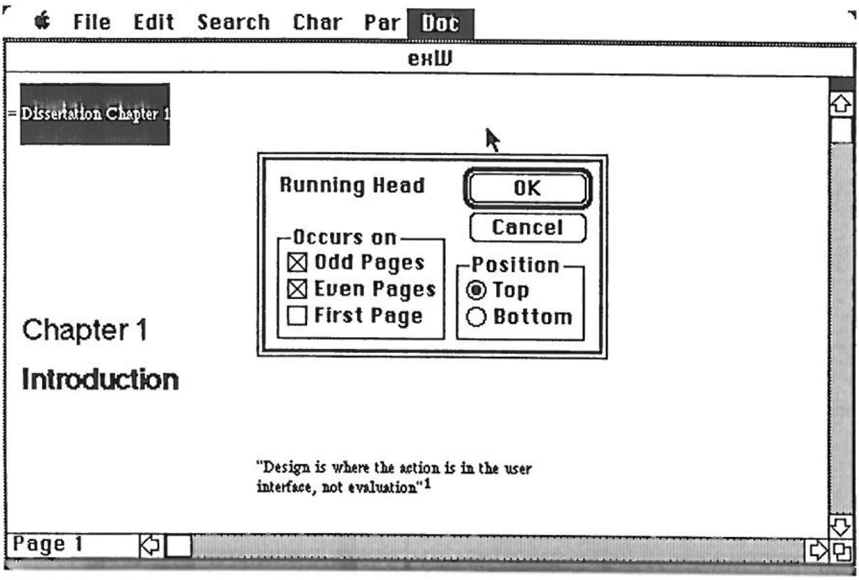


Figure 19. A running head is created by typing a text for the heading, selecting it, and choosing 'running head' in the 'Doc' menu. Hereby the above menu shows up by which the heading can be placed on the preferred pages

Footnotes can be created, and they are shown in a separate window. The footnotes can in general be treated like the text although there are some restrictions. By selecting 'footnote' in the 'doc' menu, a footnote reference is placed in the text where the cursor is located. A prompt shows up to make the user choose between auto-numbered references, or special reference marks (figure 21). After that the footnote window opens (if it is not open already), and the footnote text can be entered (figure 22).

Scrolling can be done in both directions by means of the horizontal and vertical scrollbars, respectively (figure 23). The restrictions in work are that one works with a "small document", e.g. a scientific paper, not e.g. a report like this dissertation. This means that when one approaches 100 pages or 200 footnotes, response times get extremely large.

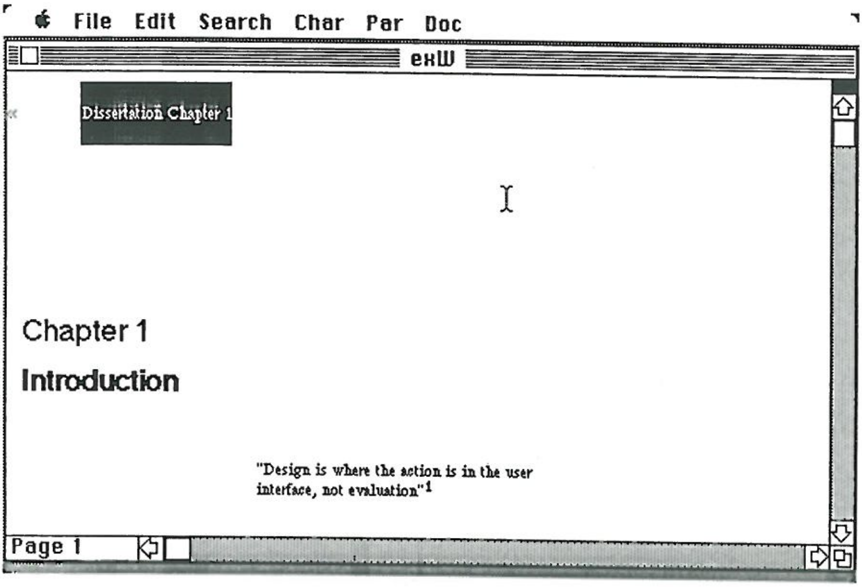


Figure 20. The running head placed in the document according to the parameters of figure 19

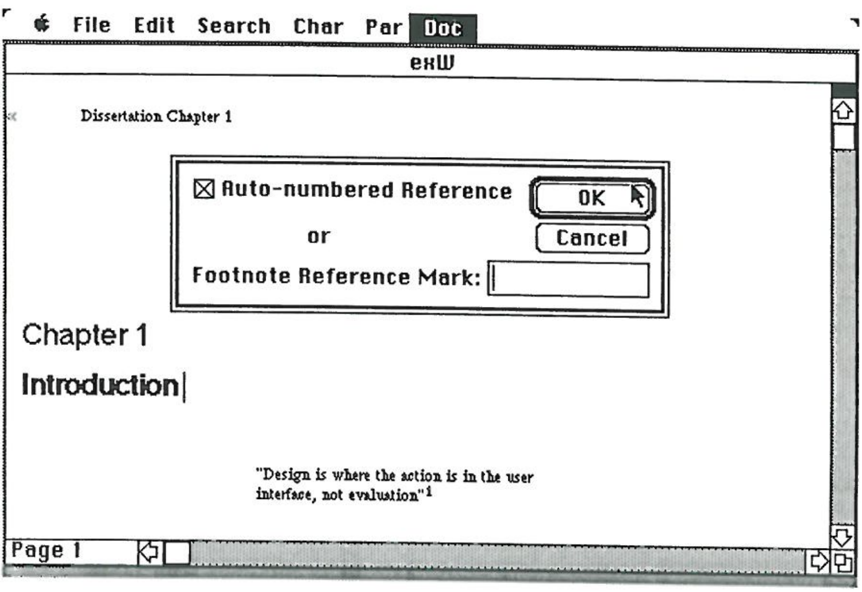


Figure 21. Creating a footnote by choosing 'footnote' in the 'Doc' menu. The footnote will be inserted where the | is placed. The user can choose between automatic numbering of footnotes, and reference marks

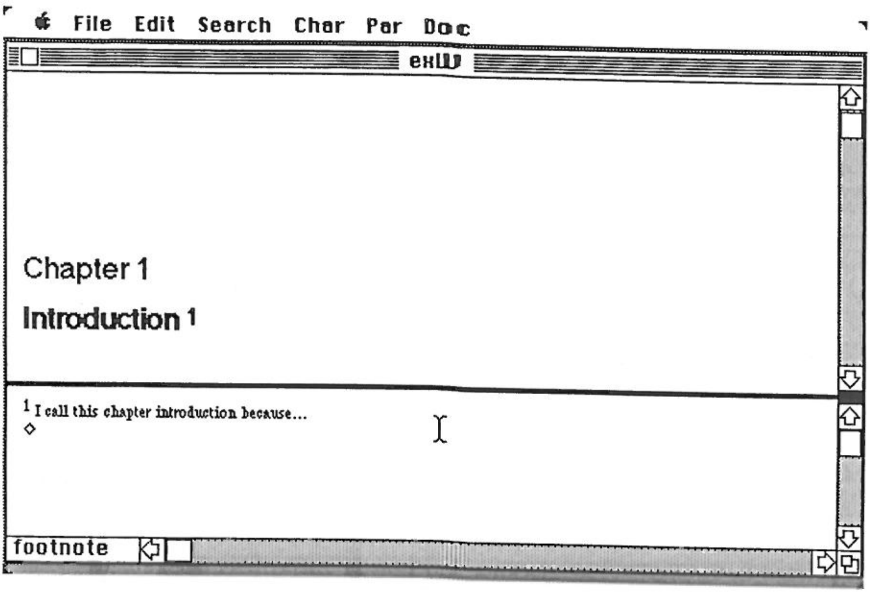


Figure 22. Typing the footnote text in the special footnote window which has appeared at the bottom of the screen

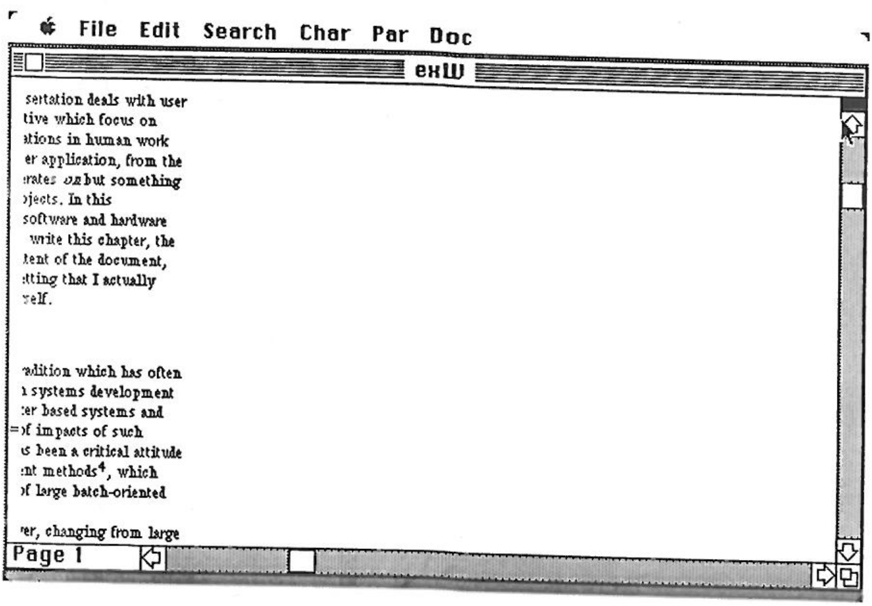


Figure 23. Scrolling in both directions, using the scrollbar at the bottom and to the right

1 MacWrite version number 4.5, of April 4, 1985, the Danish version.

2 Microsoft WORD version 1.05 of April 24, 1985.

3 Whereas I use a primarily English version of WORD in the examples, it has not been possible to get hold of an English version of MacWrite, as these cannot be purchased in Denmark.

4 The reader must excuse the lack of screen images with pull-down windows pulled down. Such snapshots are impossible to make with the software at present available at the Computer Science Department. It would of course have been possible, by physical cut and paste, to produce something which would look like a screen image, although it would take some effort to find the proper pieces in the system. I find it, however, a point here not to cheat.

5 Please note the elegant mixture of Danish and English text.