Summary

1. INTRODUCTION

The use of information systems in working life has increased rapidly during the past 20 years. In 1977, there were about 300 000 people in Sweden using computers in their work [SCB, 1991]. In 1995, the number of users had grown to approximately 2.1 million people [SCB, 1995]. This rapid development has changed the situation for many workers in our society. A considerable number of toilsome and tedious work tasks have disappeared entirely and many work tasks can be performed more effectively with the support of a computer. On the other hand, a number of new and severe problems has appeared instead. In a study by Aronsson, Åborg and Örelius (1988), more than 3000 users of computer support were examined. One of the issues investigated included how computer users were affected by their work. The study showed that numerous users had problems with their necks, shoulders and eyes. Some psychic problems were also detected. The study could not find any relation between the above mentioned problems and the physical working environment. Their were associations, however, between the detected problems and the amount of work, the types of work and the functionality of the computer support. Results similar to these have increased the interest for Human-Computer Interaction in working life.

2. HUMAN-COMPUTER INTERACTION

The main issue of Human-Computer Interaction, (HCI), is to study and understand computer-related work in order to make changes that may improve the workers' situation. In HCI, the *human* interacting with the *computer*, the computer, and the *work environment* is regarded as one system. (See Figure 1). All these parts of the system are important, because changes in one part of the system will have effects on the other parts. If, for example, the organisational structure in a company is changed, then the work performed by the workers may be influenced in a way that makes changes in the information system necessary.

Figure 1. The user, the computer and the work environment are regarded as one system.

To be able to understand and improve this system, knowledge from different disciplines is required. Knowledge in computer science is needed to understand how the computer works. To understand the human interacting with the computer, knowledge in cognitive psychology is necessary. To capture important aspects of the environment, knowledge in organisational theory, sociology, ergonomics, and other related human sciences are needed. Also, graphical and industrial design competence are important in the design of information systems.

The user's interaction with the computer has been described in Norman [1986] with a model including seven goal-driven user activities. When a user performs a task, he/she specifies an intention according to the goal and translates this into an action that is executed (e.g., pressing enter on the keyboard). This action may cause a change in the computer system state, which is perceived and interpreted by the user. The user then evaluates the systems state with respect to the goals and intentions. Depending on the computer support it may be difficult for the user to understand how to interact with the computer to reach a specific goal. It may also be difficult to understand if the goal is reached or not. The reasons for difficulties in interaction with the computer may be found in the user interface design. A user interface may be described as the appearance and the behaviour of the information system (i.e., the only part of the computer system that the user is in direct contact with). To be able to design a usable interface it is important to understand how the users think and how they perceive their environment. We can learn a great deal about these processes from cognitive psychology.

3. COGNITIVE PSYCHOLOGY

There are several discoveries within cognitive psychology that are important to consider for a designer of human-computer interfaces. One example is the behaviour of the human memory. According to Stage Theory [Gleitman, 1991], human memory can be divided into short-term memory (STM), which holds information for short intervals, and long-term memory (LTM), in which materials are stored for an indefinite period. The capacity of STM is limited. In STM, only 7±2 items or chunks can be stored simultaneously, though contents of the items can be bigger or smaller chunks. For example, a telephone number can be stored in STM as 2 2 5 6 8 6, (i.e., sex items), or 22 56 86, (i.e., three items). Storage time in STM is also limited. Items in STM may disappear because of decay or displacement. Stored material in STM decays after 10-15 seconds if the to-be-remembered material is not further processed. The function of the memory system has implications when working with a computer system. Because the capacity of the STM is limited, it is important that all information needed when making a decision or performing a work task is visible simultaneously [Lind, 1991]. If the user has to remember information from different screens, there is greater risk for unnecessary cognitive load.

Another important human characteristic is that information can be processed on different cognitive levels (See Figure 2). On a high cognitive level, the conscious level, it is only possible to deal with one process at a time. On this level complex problems can be solved. On a lower cognitive level, several processes may be handled in parallel and be performed “automatically“ [Schnider & Schiffrin, 1977]. To be able to automatise a process from a higher to a lower cognitive level, the process needs to be dealt with frequently and consistently (e.g., finding information on a screen where information always has the same spatial location).

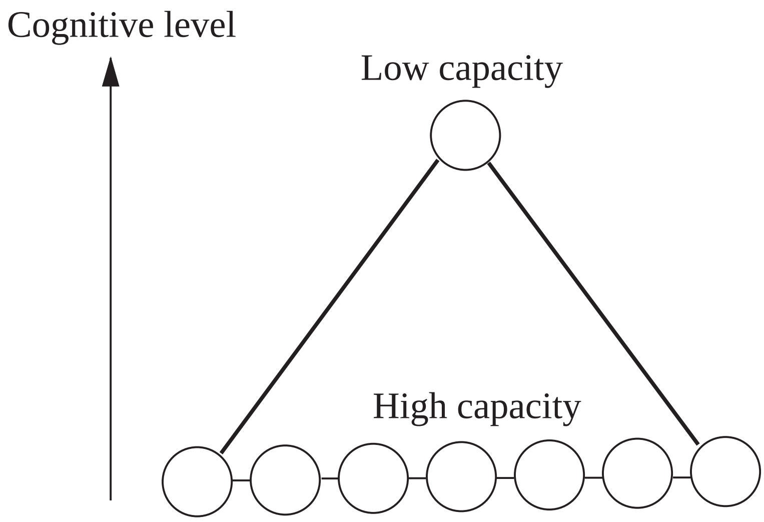

Figure 2. Processes can be dealt with on different cognitive levels. On a high cognitive level, only one process can be operated on at a time. On a lower cognitive level, processes can be handled in parallel.

Perception describes how we interpret sensations, how we apprehend objects and events in the world surrounding us (e.g., how some green colours can be formed into a bush). The Gestalt approach emphasises that objects are precieved as well-organised “wholes“ rather than as separated, isolated entities. [Matlin & Foley, 1977]. Knowledge about perception is important when deciding on how to use colours, fonts, sizes, and on how to group information on the screen to optimise the searching and reading processes.

4. DESIGN OF HUMAN-COMPUTER INTERFACES

The main goal for a designer must be to create interfaces that support the users work. The users must be able to concentrate on their work instead of dealing with of the information system. The user interface should be “obvious“ to the user [Nygren, Johnsson, Lind, & Sandblad, 1992]. Therefore, the interface has to be effective, efficient and satisfactory to the users. Effectiveness relates the goals to be reached to the accuracy and completeness with which these goals can be achieved [ISO, 1995]. Efficiency relates the level of effectiveness achieved to the expenditure of, for example, mental or physical effort, time, materials or financial cost. If the goal is to register names and addresses in a form, the effectiveness could be measured by counting the number of spelling errors. The efficiency could then be the number of registrations per time unit.

When designing a usable interface several aspects have to be regarded. The importance of domain knowledge when creating an information system has been pointed out elsewhere [Gulliksen & Sandblad, 1995]. We have defined a domain as a class of activities that bear similar aspects on the workers situation and performance regarding interaction with customers, case handling, decision making, etc. [Gulliksen & Sandblad, 1995]. Fischer [1993] argues that it is important not only to analyse the user interface but also the domain beyond the visible interface. One way to capture domain knowledge is to involve users in the development process. In user-centred design, according to, for example Norman and Draper [1986], the users express their requirements on the interface, in their domain language, and the developers interpret these requirements into computer terminology. From these demands, user interfaces can be realised as paper mock-ups or implemented using prototyping tools. The interface may then be tested by presumptive end-users, resulting in new demands and redesign in an iterative way.

It is also important to know who the end-users are. Some relevant user characteristics include knowledge, skill, and experience [ISO, 1995]. There is, for example, a great difference between skilled and novice users. For a novice user, it is important that the information system is easy to learn, whereas for a skilled user it is more important that the computer is efficient in daily use [Nielsen, 1989]. There are several examples of conflicting demands on the interface (e.g., for a skilled user labels are not that important, whereas for a novice user they are). Recommendations on how to design usable interfaces are often given as guidelines (e.g., use broad instead of deep menus). Today's guidelines often emphasise ease of learning rather than efficiency in daily use [Nygren, Allard, & Lind, 1996].

The technical environment can have effects on the interface design. Important factors are the size of the screen and it's resolution, the type of development tool, data base, etc. The size of the screen and the resolution will directly influence the amount of information that can be shown on the screen simultaneously. The type of development tool may restrict the possibilities to realise different design solutions.

A user interface should be based on standards or style guides (e.g., Motif, Windows). A style guide describes the layout and the behaviour of basic interface elements. It is of great value to make different interface solutions consistent. Consistency is important both for ease of learning and efficiency in daily use. The use of a style guide can possibly decrease development costs. A standard may have restrictions on the flexibility when designing an interface [Grudin, 1989].

5. METHODS FOR DESIGN

In a recent study on how developers work with usability during system development projects, 113 questionnaires were answered by developers representing 37 companies. Nearly all participants regarded research within HCI important, ranking “Methods for development of usable information systems“ as the most important issue [Katzeff & Svärd, 1995]. When studying methods for development of user interfaces the whole chain, including analysis, design, and evaluation, has to be included.

5.1 Analysis

When creating an information system, it is necessary to understand the user's information needs in addition to what tasks they perform at work. Designing an interface without regarding the domain will probably result in a user interface that does not support the end-users work. Domain knowledge acquisition can be the result of different analysis efforts. It can be a formalism for capturing users' mental representation of a work task to be executed with computer support [Gulliksen, Sandblad, & Lind, 1996]. Data, rules and relations needed per work task can be captured in an object model [Booch, 1990], together with 'methods' for achieving the required actions or operations. Certain factors concerning cognitive load when utilising this information, especially describing how information is physically manipulated, can be traced through analysis of information utilisation [Gulliksen, Lif, Lind, Nygren, & Sandblad, 1996]. Organisational impacts, expectations on future work situations, and task relations to work roles are described in the framework for organisation and information technology development [Gulliksen, Lind, Lif, & Sandblad, 1995].

5.2 Design

Designing a user interface is partly a creative process. It is not really possible to describe the design process as a strict method with instructions that are easy to follow and that results in an "optimal" interface. Methodologies for user interface design in human-computer interaction seldom give much support for making design decisions; rather, they support the processes surrounding these decisions, such as analysis, evaluation, etc. [Wallace & Andersson, 1993].

One approach to interface design is the *Design Rationale approach*, where the main focus is on the documentation of the reasons underlying design decisions [MacLean, Bellotti, & Young, 1990; McKerlie & MacLean, 1993]. Such methods are semi-formal notations of different design options, and an explicit representation of the criteria for choosing among these options. Design rationales are very important for the documentation of the design and for future possibilities to evaluate the rationales behind design. However, using this technique in complex development projects is time consuming and the support for making the design decisions is limited.

The usability engineering approach to interface design puts much effort into the evaluation of an interface as a basis for redesign. The purpose is to obtain a design that fulfils specific utility and usability criteria [Nielsen, 1993; Nielsen & Mack, 1994]. Here, the focus is on the definition of such criteria and on methods for evaluation and testing. This approach presupposes an already existing design, giving very little methodological support for the making of design decisions. Methods of this nature are important for a more complete design methodology.

Basing the design work on a domain specific style guide can minimise the methodological gap between the style guide and interface design for a specific application. Normally, a style guide is general to its nature, with limited design support for application development in a specific work domain. A style guide on a higher level, where domain knowledge is included, can be a more detailed and efficient support for the design process [Gulliksen, Johnson, Lind, Nygren, & Sandblad, 1993; Gulliksen & Sandblad, 1995]. Important parts of a domain specific style guide are composite interface elements corresponding to more complex information structures in the domain. This facilitates the making of design decisions but gives no structured method on how to work with design.

It is our belief that it is possible to support design more formally.

5.3 Evaluation

Numerous methods for evaluating user interfaces have been published. They all have their advantages and disadvantages, and it is not always obvious whether they are applicable in a specific system development project or not. Evaluation methods can be separated into usability testing methods (i.e., where users are actively involved), and usability inspection methods (i.e., where users are not involved).

An example of a method for usability testing is *performance measurement* [Nielsen, 1993]. The purpose of this is to measure whether a usability goal is reached or not. User performance is almost always measured by having a group of test users perform a pre-defined set of tasks while collecting data on errors and times. The tests are usually carried out in a laboratory. With such a test many usability problems will be found. One advantage of this test method is that the result is given in hard numbers that makes comparison of different design solutions easy. Unfortunately, in most system development projects there are not enough time, money or laboratory expertise to use such a method. Another problem with laboratory tests is that they are difficult to perform early in the design process because the tests demand a running prototype and require a reasonably complete data base. Further more, it is not always easy to measure more abstract goals using this method.

Other examples are *Questionnaires*, especially useful for issues concerning users' subjective satisfaction and possible anxieties [Nielsen, 1993], and *Thinking aloud* where the users verbalise their thoughts while using the system. Through such tests, users let the usability expert understand how they view the computer system.

An example of a method for inspection of the interface is *heuristic evaluation* [Nielsen et al., 1994]. The evaluator uses sets of guidelines, (i.e. heuristics), and compares those with the interface. The heuristics form a checklist that the evaluator uses during his work. It is easy to learn and inexpensive to use. With heuristic evaluation it is possible to identify many usability problems and to perform the evaluation early in the design phase. A drawback is that evaluators using this method seldom manage to identify domain specific usability problems due to lack of domain knowledge. Some kind of heuristic evaluation could, however, be useful when identifying general usability problems in an interface. The heuristics suggested by Nielsen (1993) work for a broad range of interfaces but have an emphasis on ease of learning. The heuristics are not "optimised" for identification of usability problems concerning efficiency in daily use.

Another inspection method is cognitive walkthrough [Nielsen et al., 1994]. With this method an evaluator examines each action in a solution path and tries to tell a credible story describing why the expected user would choose a certain action. The credible story is based on assumptions about the users background, knowledge and goals, and on an understanding of the problem solving process that enables a user to guess the correct action.

During our work within different projects we have found a need for an evaluation method that identifies both general and domain specific problems and places an emphasis on problems concerning efficiency in daily use.

6. DESIGN OF USER INTERFACES IN WORKING LIFE

Introduction of new computer support in working life occurs very fast today. To be able to influence this development, it is not always possible to perform experiments in a traditional scientific way. Neisser [1976] emphasises the importance of studying how people actually work in the real world. Some of the research that we have performed has been executed in the field. This may be classified as "applied research" meaning that we often work in applied development projects. That enables us to a) guide the development within these projects and b) to test our models in working life. We have recently been involved in different projects in co-operation with the Swedish National Tax Board. Within these projects we have developed and used methods for analysis, design and evaluation of information systems in in-house development projects. The users that participated were skilled professionals performing administrative case handling tasks.

In the following papers we will present some results from our research within HCI.

6.1 Overview

In Paper 1, a framework for the entire process of organisation and information system development, focusing on design issues, is discussed. We have defined the process of design in human-computer interaction as the process of creating a formal description (e.g., program code) of the appearance and functionality of an information system. The analysis phase is separated from the design phase; no design decisions should be made before the results of the analysis have been documented.

According to the definition of design as a specification in a formal language, it can never completely describe all requirements. In the development process, four different consecutive gaps of communication can be distinguished, in which important information about the actual work situation could be lost. These gaps refer to the formalisation of the process of organisational development, data modelling, work descriptions and interface design specifications. The gaps can constitute severe obstacles in the process of developing efficient and usable information systems for specific work situations. By introducing development models that cover the entire process of design, these gaps can be bridged or, at least, narrowed. Some main characteristics of such models are presented in this paper.

6.2 Analysis

The second paper describes analysis of information utilisation (AIU), a method for the specification of how information entities encountered in the information analysis are being physically manipulated in a work situation. AIU focuses particularly on factors concerning cognitive load. The method supports the human-computer interface designer with relevant and appropriate information.

The paper describes the method of analysis of information utilisation, and what information that has to be documented while performing the analysis. It suggests how the method can be incorporated into existing in-house development methodologies, and gives an example of documented analysis results from a development project within the Swedish National Tax Board.

6.3 Design

Paper 3 describes an approach to design inspired by many concepts from the “rooms design“ metaphor [Card & Hendersson, 1986]. A design goal is to specify an efficient interface for each group of individuals (or work role). By separating different workspaces (or rooms) for groups of individuals and base the design of screen documents on domain knowledge, tailored and efficient interface structures can be obtained.

The paper identifies differences between this approach of document-oriented design and the more common process-oriented approach, often used when designing based on the desktop metaphor.

Paper 4 presents a structural approach to the iterative process of design. Four different phases of the design process are identfied: Identification of the information contents, specification of the layout, dialogue structure, and the behaviour of the user interface. Design decisions made in these phases are based on a set of domain specific heuristics. Prerequisites for the user interface design process include an analysis of the domain, the users, and the technical environment. Based on this, a suitable metaphor can be chosen. A “workspace metaphor“ supporting administrative work in a broad sense performed by skilled professionals, is suggested. By creating interface elements and workspaces in an iterative design process, a first prototype can be defined. Some design examples are given.

6.4 Evaluation

During participation in some different in-house projects, we have noticed that a user-centred approach to design does not necessarily lead to a usable interface. There are often usability problems that are not captured. Paper 5 presents a method for domain specific evaluation to be used during the design of human-computer interfaces. With this method it is possible to identify both general and domain specific usability problems, and identify usability problems concerning both efficiency in daily use and ease of learning. A set of heuristics adjusted to fit evaluation of interfaces for skilled users is introduced. General problems are captured by inspecting the interface. The domain specific problems are captured by evaluating the interface together with the potential end-users. The method is to be included early in the user centred design process, as a tool to guide the development of the interface towards a design that is efficient for the end-user in his/her specific domain.

Developing information systems for skilled users is difficult because it is not always possible to foresee what problems are likely to occur when the users become skilled. It is, therefore, necessary to evaluate the computer support when it has been used during a longer period of time.

The evaluation method presented in Paper 6 is intended to be used late in the development process. The method identifies interface solutions that may lead to cognitive work environment problems. The evaluation is performed by an observer interviewing the user and studying the users work with the computer support. The result of the evaluation is documented according to a specified model. Potential problems are listed as a guide for possible improvements of the information system. The method is to be used by occupational health psychologists as a part of their investigation of the work environment in VDU-work.

7. DISCUSSION

With character based user interfaces it can be difficult to visualise and incorporate all relevant aspects of a work task. Today, interface design is not just a question of showing green alphanumeric characters on a black screen. With graphical interfaces, commonly used nowadays, it is possible to show, for instance, colours, different fonts and sizes of characters, symbols, scanned pictures, and documents on the screen. In addition to this, there are other possibilities, such as the incorporation of sound, moving images, and 3D-pictures. The introduction of graphical user interfaces does offer the opportunity to create interfaces with increased usability. On the other hand, incorrect use of the graphical tools can result in interfaces that are less efficient than the old alphanumeric ones (Niesen, 1993). The increase in number of degrees of freedom has made the designers work even more complicated.

Design is, in many respects a creative process and the role of experience is undoubtedly important. Still, in some development projects we have found that designers can create better interfaces with the support of a structured method. Our intention is not to ignore the creative and innovative part of interface design, but to enhance it through methodologies that support the creative development process.

This thesis describes a structured approach to information technology development. A method for analysis of information utilisation has been described and methods for design and evaluation have been introduced. There is, however, some work that remains, particulary concerning making the methods more distinct, and formal evaluations are needed. The need to analyse the potential end-users' competencies and the technical environment have been emphasised, but no formal methods have been described. We have also mentioned the importance of artistic capabilities in design, but have not given much further guidance on how to incorporate that. The aesthetic aspects are important for satisfaction in use, which is why there is a need to further study this area.

In future research, we will continue to identify different phases of the interface designers work. The goal is to understand and describe all phases in user interface design, and define useful methods for developers. Some of the methods are currently being incorporated in the existing development models of The Swedish National Tax Board. In this project, the entire iterative development chain of analysis, design, and evaluation is covered. This extended development model will be tested and evaluated in a number of application development projects and the results of the evaluation will be used for refinement of the model. The methods presented in this thesis have been used in several applied projects and have been shown to be useful for developers of user interfaces.

8. REFERENCES

ARONSSON, ÅBORG, & DALLNER (1994). Winners and losers from computerisation. A study of the psycho social work conditions and health of the Swedish state employees. *In International journal of human-computer interaction, Vol. 6, No. 1.*

BOOCH, G. (1990). *Object-oriented Design with Applications.* Benjamin Cummings.

CARD, S.K. & HENDERSON, A. (1987). A Multiple Virtual-Workspace Interface to Support User Task Switching. *In CHI '87 Conference on Human Factors in Computing Systems,* (Toronto, Canada, April 6-9), ACM/SIGCHI, New York.

FISCHER, G. (1993). Beyond Human-Computer Interaction: Designing Useful and Usable Computational Environments. *In People and Computers VIII* (ed. J. Alty, D. Diaper and S. Guest), Proceedings of the BCSHCI'93 conference, Cambridge University Press, Cambridge, U.K.

GLEITMAN, H. (1991). *Psychology. 3rd edition.* W.W. Norton & Company, Inc., New York.

GULLIKSEN, J., & SANDBLAD, B. (1995). Domain Specific Design of User Interfaces. *International Journal of Human-Computer Interaction, Vol. 7, No. 2*, pp. 135-151, Ablex Publishing Corporation, Norwood, New Jersey.

GULLIKSEN, J., & SANDBLAD, B. (1996). Domain Specific Design of User Interfaces – Case Handling and Data Entry Problems. *In D. Benyon & P. Palanque (eds.) Critical Issues in User Interface System Engineering,* Springer Verlag, London.

GULLIKSEN, J., JOHNSON, M., LIND, M., NYGREN, E., & SANDBLAD, B. (1993). The Need for New Application Specific Interface Elements. *In G. Salvendy & M. J. Smith (eds.) Proc. of HCI International '93, Elsevier,* pp. 15-20.

GULLIKSEN, J., LIF, M., LIND, M., NYGREN, E., & SANDBLAD, B. (1996). Analysis of Information Utilisation. *Submitted.*

GULLIKSEN, J., LIND, M., LIF, M., & SANDBLAD, B. (1995). Efficient Development of Organisations and Information Technology – A Design Approach. *In Y. Anzai and K. Ogawa (eds.) Symbiosis of Human and Artifact. Proceedings of the 6th International Conference on Human-Computer Interaction, Pacificio Yokohama, Yokohama, Japan 9 - 14 July 1995.*

GULLIKSEN, J., SANDBLAD, B., & LIND, M. (1996). The Nature of User Interface Design – The Role of Domain Knowledge. *In A.G. Sutcliffe, F. Van Assche, & D. Benyon (eds.)Domain Knowledge for Interactive System Design.* Chapman-Hall, London.

INTERNATIONAL ORGANISATION FOR STANDARDISATION (1995). *ISO/DIS 9241 (Draft). Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)* Part 10: Dialogue Principles, Part 11: Guidance on Usability, and Part 13: User Guidance.

KATZEFF, C., & SVÄRD, P.O. (1995). *Användbarhet i Praktiken: En Enkätstudie.* (In Swedish: Usability in Practice: A survey.) SISU publication. no 95:20. Swedish Institute for Systems Development.

LIND, M. (1991). Effects of Sequential and Simultaneous Presentations of Information. *CMD Report 19/91,* Center for Human-Computer Studies, Uppsala University, Sweden.

MACLEAN, A., YOUNG, R.M., BELLOTTI, V.M.E., & MORAN T.P. (1991). Questions, Options and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction, Vol. 6,* pp. 201-250.

MATLIN, M.W., & FOLEY, H.J. (1992). *Sensation and Perception. 3rd edition.* Allyn and Bacon, Massachusets.

MCKERLIE, D., & MACLEAN, A. (1993). QOC in Action: Using Design Rationale to Support Design. *INTERCHI '93 video program,* ACM: Amsterdam.

NEISSER, U. (1976*). Cognition and Reality.* San Francisco: W H Freeman

NIELSEN, J. (1989) What do users really want. *International journal of human-computer interaction. Vol 1,* pp. 137-147.

NIELSEN, J. (1993). *Usability Engineering.* Academic Press, Inc., San Diego.

NIELSEN, J., & MACK, R.L. (1994). *Usability Inspection Methods.* John Wiley & Sons Inc., New York.

NORMAN, D.A. & DRAPER, S.W. (EDS.) (1986). *User Centered System Design.* Hillsdale: Lawrence Erlbaum Associates, Inc.

NORMAN, D.A. (1986). Cognitive Engineering. *In D.A. Norman & S. Draper (eds.) User Centered System Design,* Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey.

NYGREN, E., ALLARD, A., & LIND, M. (1996). Skilled Users' Interpretation of Visual Displays. *Submitted.*

NYGREN, E., JOHNSON, M., LIND, M., & SANDBLAD, B. (1992). The Art of the Obvious. Automatically Processed Components of the Task of Reading Frequently Used Documents. Implications for Task Analysis and Interface Design. In J.P. Baursefeld, J. Bennett & G. Lynch (eds.) *Proceedings of Human Factors in Computing Systems, CHI '92,* Monterey, California, May 1992, ACM, pp. 235-239.

SCHNEIDER, W., & SHIFFRIN, R.M. (1977). Controlled and Automatic Human Information Processing I, *Psych. Rev., Vol. 84,* pp. 1-66.

STATISTISKA CENTRALBYRÅN (1991). *Datorer och arbetsmiljö. Information om arbetsmarknaden 1991:3.* (In Swedish: Computers and working environment. Labour information 1991:3.), SCB förlag, Örebro

STATISTISKA CENTRALBYRÅN, (1995). *Datorvanor 1995.* (In Swedish: Computer habits), Nordstedts Tryckeri AB, Stockholm

WALLACE, M.D., & ANDERSON, T.J. (1993). Approaches to Interface Design. *Interacting with Computers, Vol. 5, No. 3,* pp. 259-278.