Adding Usability. Methods for Modelling, User Interface Design and Evaluation.

1. Background

1.1. Evolution versus revolution

Most introductions to *Human Computer Interaction* (HCI) start by expounding how much the world has changed during the last few years, how quickly hardware and software have developed and how many more people use computers today as opposed to only a few years ago. All this, of course, is accurate; there has been a rapid change. According to a recent survey made by SCB (Statistics Sweden), there were approximately 2.1 million people, or 50% of all employees in Sweden, using computers in 1995 (SCB, 1995). In 1984, the corresponding value was 24%. Today, people use computers to a greater extent in their work than they did in the recent past. However, the important factor here is not only what has changed but also what has *not* changed. Human beings have not gone through any significant changes, at least not in the last thousand years or so. We still have the same needs and abilities and we do not perceive the world or process information differently from the way our great grandparents did. *It is clear that evolution has another pace then technical revolution*. I am not proposing that we should strive to accelerate the process of evolution in an attempt to adjust users to computers. Rather, my argument is that the technical revolution has not made sufficient progress when it comes to developing computer systems to meet the users' needs.

The introduction of computers has definitely improved the work situation for many people. Several burdensome and tedious work tasks no longer exist or have been improved. Several tasks are performed more efficiently with the use of computers. Unfortunately, new, monotonous tasks have been established partly due to the introduction of computers. A new kind of work environment problems has appeared. In a study by Aronsson, Åborg & Örelius (1988), more than 3,000 computer users were examined. One aim was to investigate how these users were affected by their work. The study showed that numerous users had problems with their neck, shoulders and eyes. Some mental problems were also detected. The study could not find any correlation between the above mentioned problems and the physical work environment. However, relations existed between the detected problems and the amount of work, the types of work and the functionality of the computer systems. As might be expected, such results have increased the interest for HCI in working life.

Computerisation should be driven by people's needs and abilities and not by the technical progress itself (Norman, 1986). There are several ways to tackle this problem. In my thesis, I focus on the development process. To be able to design an information system that supports the users in their work, it is necessary to understand the users and their needs. Issues considered in this thesis are:

- How to gather and document user requirements that are relevant when making design decisions.

- How to guide the design process towards an information system that supports the users in their work.

- How to include these methods in the system development process.

1.2. The need to study HCI

Computerisation has sometimes changed the work as well as the work environment for people beyond recognition. Several industrial processes were run earlier by hundreds of people who knew the different steps of the process in detail. These people could get an understanding of the processes involved by just watching, or even listening, to the different machines. Today, it is common that factories are run by only a few computer operators. The computer system automatically controls some parts of the process, while the users manually control other parts of it by reading and changing parameters on the computer screens. The interface towards the process has moved away from the machines to the computer screen.

Administrative work involves decision-making based on huge amounts of information. Traditionally, this information was carefully recorded in documents and in forms. Tools and methods for storing, sorting, searching, reading and writing information have developed based on knowledge gained through decades of accumulated experience. When making a decision, it is important to have the necessary information readily at hand. Previously it was possible to get an overview by simply spreading all papers, books, maps, etc. on the desk. Today, however, almost all information is stored in immense information systems. The same amount of information has to be read from a computer screen. According to Henderson & Card

(1986), a standard office desk has the area of approximately 22 standard size PC displays. Obviously, the way information is presented on the screen will have an important impact on the decisions that are made by the user. The appropriate information and functionality have to be available in an adequate way to enable the user to make proper decisions. Designing systems that effectively support the user during decision-making processes is a major goal within HCI.

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 with which the user is in direct contact). In the early days in the history of computers, there was no need, or even no possibility, to put much effort into the design of the user interface. The number of design options was rather limited. It was not possible to use graphics, sound or speech recognition in the user interface. User and programmer were usually the same person. There was no need to bother about making the system useful for other people. Today, most computer systems are intended for users with little or no skills in programming or even in using computers as such. Graphical user interfaces and various techniques for interaction have increased the possibilities for creating information systems that are easy to learn and efficient in daily use. However, the number of possible design options has increased, which can be a problem if they are not carefully examined. It is necessary that the user interface is designed to meet the requirements of users working within a certain context.

Understanding the user's interaction with the computer has been a central goal of HCI. Norman (1986) has described this interaction in a model including seven goal-driven user activities. A user performs a task by specifying an intention to achieve a goal. This intention is translated into an action that is then executed (e.g., pressing enter on the keyboard). The action may cause a change in the state of the computer, which is perceived and interpreted by the user. The user then evaluates this state with respect to goals and intentions. The potential difficulties when working with the system have been described as the "gulfs" of execution and evaluation. The *gulf of execution* is the case in which the user knows what goals to be achieved but does not know which physical variables to adjust, or in what way to adjust them. The *gulf of evaluation* is where the system has altered, usually because of the user's action, but the user cannot easily understand the change in the system's state. These difficulties may depend on the design of the user interface.

The area of HCI is extensive. It is necessary to understand how users think and perceive information, how people co-operate during work, how to create hardware and software to support users, how information systems affect people, how to create methods for building the software, how to evaluate the applications, etc. HCI is a multidisciplinary topic based on knowledge gained from several disciplines such as cognitive psychology, software engineering, computer science, organisational theory, sociology, ergonomics, systems analysis, process control and industrial design.

1.3. Cognitive psychology

Understanding how users think and perceive their environment is important when designing user interfaces. Unfortunately (or fortunately), the human brain is more complex than the computer. Visions of thinking computers have existed in the literature for a long time. So far, it has not been possible to realise this vision, partly because of the complexity of the human brain. We still do not understand how it works and we are, therefore, not able to build a copy of it. However, what we are able to do is to measure how people behave in different experimental environments. Some of these results have been valuable for the progress of HCI.

One of the most recognised models of human memory is *Stage Theory* (e.g., Gleitman, 1991). According to this theory, all humans store information in either a short-term (STM) or a long-term memory storage system (LTM). The human STM system can deal with only a limited amount of information. Based on the results of numerous studies, the evidence suggests that the limit of the human STM span was about seven units of information or *chunks* plus or minus two. Chunks refer to memory units that result from recoding units or integrating together units that are more elementary. The size of these chunks differs in the sense that, for instance, a telephone number can either be stored as 2 2 5 6 8 6 (six items) or 22 56 86 (three items). The time span that information can be held in STM is also limited. Forgetting in STM may be largely due to decay or displacement (Waugh & Norman, 1965). Stored material in the STM decays after approximately 15 seconds if it has not been further processed (e.g., through rehearsal or some other memorial strategy). In contrast to STM, the capacity of LTM is enormous. Information in the LTM system can be accessed through triggers such as a word, a smell or a sound. A trigger is often sufficient in itself to prompt the recall of a great deal of information; for example, a chord on a piano can be a trigger to the lyrics of a song. The functionality of the memory system has significant implications in relation to working with computers. Because of the limited capacity of the STM, all information needed for a decision should be visible on the screen simultaneously (Lind, 1991) as far as it can be represented visually. Otherwise, the user has to store information in the STM or make notes on paper. This will slow the users' work and leads to unnecessary cognitive load.

A simplified model of human information processing implies that cognitive processes may operate on different levels of awareness (Rasmussen, 1983; Reason, 1987). On a conscious cognitive level, it is only possible to handle one single process at a time. This level is applied to reading and understanding semantic information and for solving complex problems. On lower cognitive levels, it is possible to handle processes in parallel and largely automatically without requiering cognitive effort. Successive handling of frequently occurring tasks can automatize them to lower cognitive levels if consistently presented (Schneider & Shiffrin, 1977; Shiffrin &

Dumais, 1981). An example of such a task is learning to drive a car. In the beginning, it can be difficult to change gears, judge the surrounding traffic and keep the car on the road at the same time. Thus, for a novel driver these processes are treated on a higher cognitive level. After considerable training all these processes are usually dealt with automatically (i.e., on a lower cognitive level). The higher cognitive level can then be used to solve more demanding problems during the act of driving such as speaking in a cellular telephone with a customer. This model of human information processing can be considered when designing a computer system. The user interface should be designed so that the user can handle the interface automatically (i.e., on a lower cognitive level), leaving the higher cognitive level for solving work-related problems (Nygren, Johnson, Lind, & Sandblad, 1992).

Cognitive psychologists also study how individuals perceive the surrounding world. The term sensation refers to initial detection of energy from the physical world such as a smell or a sound. Perception involves high-order cognition in the interpretation of sensory information (e.g., characterisation of a smell). One area of keen concern is our ability to detect stimuli. According to *Signal Detection Theory* (Tanner & Swets, 1954), several factors influence this ability, including signal magnitude of the stimuli, the nature of the task, observer expectancy, and consequences following reward or punishment (Solso, 1991).

Pattern recognition is another topic that has aroused the interest and attention of cognitive psychologists. Here, a pattern refers to "a complex composition of sensory stimuli that the human observer may recognise as being a member of a class of objects" (Solso, 1991, p. 86). Several theories on how visual patterns are classified have been generated over the years. "The *Gestalt approach* emphasizes that we perceive objects as well-organized 'wholes' rather than as separated, isolated parts" (Matlin & Foley, 1977, p. 131). A collection of dot patterns can be perceived in different ways depending on how they are grouped by the observer. The Gestalt approach presents a number of laws of perceptual organisation (e.g., Gleitman, 1991). Three of these are discussed below (Figure 1):

- Proximity. The closer two figures are to each other, the more likely will they tend to be grouped together.

- Similarity. Things that look equal are usually grouped together.

- Closure. Figures with a gap are likely to be completed.

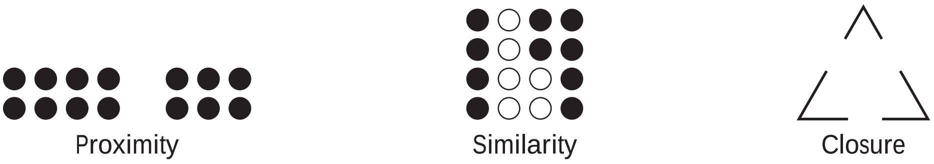

Figure 1. Examples of Gestalt laws.

Knowledge about perception is essential when deciding on how to use colours, fonts, sizes, and how to group information on the screen to optimise the searching and reading processes. Experienced users decode frequently occurring, meaningful patterns rapidly (Nygren, Allard, & Lind, In press). If a collection of objects always has the same spatial location on the screen, global patterns may emerge over time that can be applied to guide the user's reading process. It is also possible for the user to decode a pattern if the variable always has the same unique colour or shape.

1.4. Systems analysis

Another background discipline is *systems analysis* which is concerned with understanding systems by building models. A system is a limited part of the environment, and is separated from its environment by the systems boundary. An example is a bank office with customers and bank tellers. According to this approach, the computer, the user and the environment can be regarded as one system, a system that can be represented by a model. Such a model can be modified and tested, e.g. through simulations, without affecting the real world, where the result of this process can be employed to improve the actual system. A model is never a true or complete representation of a system, but describes aspects of the system relevant to a specified purpose. A model is extremely useful when several people need to have a common view of a system.

Systems analysis gives a theoretical framework for how models are formulated and validated, how they can be analysed and how conclusions concerning the behaviour of the studied system can be drawn from this analysis.

Models are frequently occurring within HCI. The users' interaction with the computer can be described with a model. A user has a conceptual model of the information system, but also of the "real" system that is the focus of interest in a work situation. An information system is usually tested with a model (e.g., a prototype). Systems analysis, and the use of formal models, has an important role in information system development. Here, systems analysis involves analysing and modelling present and future work situations. The result of this process are models suitable for developing the computer support, such as data models, data flow diagrams, use cases etc. Each model represents one view of the analysed system, is never equal to the real system and is only valid within the area defined by the purpose of the study. When formulated in a correct way and with a known and controlled quality, formal models are however extremely useful for specification of requirements, documentation of data flows and information utilisation. etc.

1.5. Usability

Nielsen (1993) claims that the term "user friendly" was introduced when developers of information systems first realised that their systems were to be operated by users with demands on the products in terms of access, etc. Nielsen means that this term is not appropriate for several reasons. "Users don't need machines to be friendly to them, they just need machines that will not stand in there way when they try to get their work done" (p. 23). Another reason is that different users have different needs in the sense that a machine can be "friendly" to one user but tedious to another. Instead, Nielsen proposes the term *usefulness* that relates to whether the system can be used to achieve some desired goal. Usefulness can be divided into two categories: Utility and usability (Grudin, 1992). *Utility* corresponds to whether the needed functionality is included in the system (i.e., if all tools needed to perform a given task exist). *Usability* is more closely related to how well a user could use this functionality.

In Draft for International Standard ISO/DIS 9241-11 (1995), *usability* is defined as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. " Here, the effectiveness of a system relates to the work objectives (goals). The efficiency relates to effectiveness in relation to the resources needed to perform the tasks. Satisfaction, according to ISO 9241, concerns acceptability and comfort.

Another important criteria is the extent to which the user can interact efficiently with the system without unnecessary mental efforts that are caused by *cognitive work environment problems*. Limitations in the work environment that hinder the users to use their skills efficiently may cause cognitive work-environment problems (Lind, Nygren, & Sandblad, 1991). Such impediments are often caused by the human-computer interface and can lead, in addition to, somatic and mental health problems, inefficient work procedures, bad performance and low user acceptance.

2. Developing usable systems

Thus far, we have limited our discussion about HCI to theoretical considerations with regard to how users interact with computers. Another question is of course how to apply these considerations in practise in order to develop information systems with high usability. According to a recent study on how developers work with usability during systems development projects, developers representing 37 Swedish companies answered 113 questionnaires. 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 systems development, the whole chain of analysis, design and evaluation has to be considered.

Systems development methods have been established both within the area of HCI and Information Systems Development (ISD). Ehn and Löwgren (1997) give a historic overview on how the trends in systems development have changed over the years. They claim that within both ISD and HCI the focus has changed from objective methods to social methods and now to subjective methods.

The *objective* view can be found in usability engineering (in HCI) and Software engineering (in ISD). *Usability engineering* as a process usually consists of three main steps: user and task analysis, where the users and their tasks are studied; usability specification, where a number of measurable goals are identified; and the iterative process of design, usability test and redesign that continues until the goals are met (e.g., Good, Spine, Whiteside, & George, 1986). Usability engineering is primarily concerned with the user's performance in terms of errors and times (i.e., objectively measurable results rather than the appropriateness of the system). The *software engineering* approach is similar to systematic methods for design found in mathematical and logical theories. The information requirements in an organisation are analysed as objective facts. "A common assumption behind most of these objective ISD approaches in the IT design field seems to be that the users must be able to give complete and explicit descriptions of their demands." (Ehn & Löwgren, 1997, p. 304).

As a reaction to the objective approach, a view on the development process in ISD grew that focused on users and their influence on the actual systems (i.e., the *social* approach). According to this approach, the users and their work are too complex to be understood by a system developer. Therefore, the users have to participate in the development process. Muller, Haslwanter, & Dayton (1997) identifies the following three motives for *participatory design*:

- Democracy. The users have the right to influence decisions that concern their work place.

- Efficiency, expertise, and quality. Efficiency and quality of the software are improved by involving real users since they are experts on their work.

- Commitment and Buy-in. It is more likely that the end-users will accept the system if they have been involved in the development of it.

Greenbaum and Kyng (1991) define demands on the participatory design process:

- The computer systems need to be designed with full participation from the users.
- When a new system is built it should enhance the workplace skills rather then degrade or rationalise them.
- Computers are tools and should be designed to be controlled by the people using them.
- Computers should not only increase the productivity but also the quality of the work.
- The design process is regarded as a political one, where conflicts will occur during the way. These conflicts should not be ignored, but should be emphasised since they are essential for the final result.
- The design process should highlight the issue how computers are used in the context of the work organisation.

Methods corresponding to participatory design have been developed within HCI. *Contextual design* (Wixon, Holtzblatt, & Knox, 1990) is described as a customer-oriented method. Analysis of the customers work are performed in their actual context through *contextual inquiry* where the potential end-users are observed and questioned (Holtzblatt & Jones, 1993). Three main principles guide the contextual inquiry process, namely context, partnership and focus. It provides an understanding of the nature of the user's work through inquiries with users during ongoing work. The principle of partnership means that it is only through dialogue with the users that designers can become aware of their experience of work and use of tools. The choice of focus guides the contextual inquiry process so that the proper information is gathered from the users. The value of the data comes from interpreting what is observed and by engaging the users in a dialogue. Interfaces are developed in co-design with users and are evaluated with further contextual inquiry. The contextual approach have been criticised for having "no systematic methodology, no conceptual framework, no explicit way to abstract from particular experiences" (Carroll & Kellogg, 1989, p. 7).

Factors such as time, organisation, costs and the kind of project to be initiated determine to what extent the users can be involved in a project. Grudin (1991) has identified three different kinds of development project to illustrate when users and developers have the opportunity to co-operate. In *contract development,* the users are known from the beginning, but the development organisation is identified after the contract is awarded. An illustration would be an organisation that identifies a need for a new system and prepares a specification. A contract is awarded to the most suitable developers. *Product development* is the commercial development of products where the application is normally marketed before the users are known. In *in-house development,* both the developers and the users are known from the start. A typical example is a bank with its own computer department developing new systems for the bank tellers.

Ehn and Löwgren (1997) further argue that today the trend has changed from a social to a *subjective* approach. The focus has moved from the users to the actual designer and his competence. Design is regarded as an art that cannot be learned from a book. The knowledge needed to create a useful design comes from experience and skill. Moreover, the concept of usability is more related to novelty, aesthetics and emotions. An information system has to be likeable, beautiful, satisfying and exciting to become a popular product. The market is of utmost importance.

My view on systems development is a mixture of these approaches. An objective view is needed to analyse and evaluate parts of the system. However, the users must be included in the process to specify how their work should be performed in the future. The question is rather how and when to involve the users in order to obtain the best results in terms of usability. The users are experts on their work, not on design or usability. Consequently, a designer with usability knowledge should have an essential role when developing information systems.

Most traditional methods for systems development are based on a *waterfall model*. Analysis, design, coding, testing and maintenance are all performed in a sequence. A disadvantage with such a model is that the users are not able to test the system until it is complete. Usability problems are discovered late in the process, whereas the possibilities for improvements are limited. Boehm (1988) claims that a waterfall model is not suitable when developing interactive end-user applications. He suggests a *spiral model* where analysis, design and evaluation are performed in an iterative way. Problems with the system can then be discovered early in the process, making it easier to remedy mistakes and make changes. However, to make this iterative process effective, the time for each iteration has to be short. In most development projects time and money set hard limits.

3. Analysis

When developing computer based information system, it is necessary to understand how the end-users perform their work and interact with the environment. In larger projects where several people are involved, it is necessary that the users' requirements are described in a way that all participants can comprehend. The documentation becomes even more important when the people involved in the modelling phase are not the same as those involved in the design of the system. Modelling the users' work normally consists of the two major activities of analysis and documentation.

3.1. Task analysis - systems analysis

Task analysis (TA) is generally concerned with what people do to get things done (Preece et al., 1994). The goal of TA is to analyse users' work and describe it in terms of their tasks. *Hierarchical Task Analysis* (HTA) is a method for decomposing tasks into subtasks and operations (Shepherd, 1989). These are represented graphically with a structure chart notation. The aim is to describe the tasks in terms of a hierarchy of operations and plans. HTA is used for identifying the steps needed to perform a task. Cognitive task analysis, on the other hand, is more concerned with representing the knowledge that people have, or need to have, to complete a task. This can be performed through *GOMS modelling* where goals, operators, methods and selection rules are specified (Card, Moran, & Newell, 1983). Goals are what the user has to accomplish; an operator is an action performed in service of a goal; methods are sequences of operators that accomplish a goal; and selection rules represent the user's knowledge of which method to use. *Task-Action Grammar* (TAG) uses a rule notation to specify a user's knowledge and interaction (Payne & Green, 1989).

There exist various techniques on how to perform TA (Jeffries, 1997). A commonly practised method for gathering task data is to *interview* people that work in a domain. Interviews can be performed with single users or with groups of users and are especially useful when an in-depth understanding is needed. If the purpose is to gather information from a wide range of users, questionnaires are likely to be more suitable though it is generally difficult to gain a deeper understanding using this method. Combining the two methods can be a fruitful solution. Questionnaires may be distributed to a great many users to gain more opinions on results gathered during interviews. When concerned about, for instance, the users' memory for task details, *retrospectives* and *diaries* can be used. Users are asked about specific instances of a task shortly after the task has been performed. Another method is to *observe* the users while they perform their work. With observations data can be identified that users otherwise overlook, e.g. because information is processed more or less automatically (Schneider & Shiffrin, 1977).

One problem with several methods of TA is that the definition of the tasks is too fine-grained (e.g., entering a single command, pressing the right mouse-button). We have observed that a minimal amount of "key-pressings" and "mouse-clicks" are performed amongst a much longer period of professional interaction with the work task, containing information search, judgement and decision making, etc. When designing the interface, a specification of bigger concatenated tasks is more useful.

Benyon (1992) claims that there are several problems with methods for TA. Some of his arguments are listed below:

- Device-dependent
  TA is not capable of being independent of the device that is used when carrying out the tasks. If TA is employed in the early stages of development, there is a great risk that current practices will be embodied in the new system.

- Objects
  Most TA techniques refer to objects. These objects are dependent on the physical device. In contrast to the concept of object, the concept of an entity is solid.

- Hierarchy
  TA uses hierarchical representation. A hierarchy is effective only if the system can be strictly decomposed; otherwise, all the power is lost. He also argues that users of the system might want to have different views, which raises a problem since by definition hierarchies only represent one view.

- Grammar
  Most TA techniques use grammar instead of graphical representations. He argues that grammar is not user centred to a great extent since it is more difficult for users to understand.

Benyon implies that, within the area of systems analysis, there already exist well-established methods for structured analysis. HCI researchers should instead contribute with modelling tools to complement existing methods for systems analysis. In systems analysis, several models can be derived to describe future work. Each model represents one view of the system (for further reading see Benyon, 1990).

- The relational data model
  This model describes how information within a system is interrelated. The relational model consists of two basic concepts: domains and relations. Domains are sets of values from which the actual value of a data element can be drawn. A data element "Income" can have the value 100, 000. The set of possible values from which this value can be drawn is called a domain. Relations can be thought of as a table. Each relation can have a number of attributes; for example, the relation *Person* can have the attributes Personal ID and Name.

- The Entity-Relationship model (E-R)
  The E-R model is a top-down approach to data modelling. It begins with general concepts like the entities and relationships and fills in the details until a suitable level is reached. An entity is defined as a group of data elements. These entities are "things" that can be identified in the analysed system (e.g., a department or an employee). The entities have certain relationships (e.g., a department has many employees and an employee works at one department).

  The E-R model is a graphical representation of the system that can be practised as a basis for discussions with the end-users. The relational data model is especially useful when developing the database. These two models can be used in conjunction.

- The Data flow diagram (DFD)
  This diagram describes the flow of data within an organisation, for example, from person to person (DeMarco, 1978). The DFD is essentially a logical design of the new system, specifying the functions that are required. The basic requirements of the process model are inputs, outputs, processes and stores of data. Processes, which are triggered by events, take data as input and produce data as output.

3.2. Function-oriented versus object-oriented methods

Methods for systems development can be divided into function-oriented and object-oriented methods. According to the *function-oriented* approach, functions and data are not intertwined. The data are treated as a passive holder of information and the functions are the active parts. A system modelled according to this approach is divided into functions and the data are sent between those functions. One problem with function-oriented methods is that all functions have to be aware of the data structure such as how the data is stored (Jacobson, Christerson, Jonsson, & Övergaard, 1992). If a data structure needs to be changed, all functions relating to these data have to be changed as well.

When using an *object-oriented* approach, functions and data are highly integrated. From the system that is modelled, different objects are identified, most of which can be identified from the real world (e.g., car, driver, wheel). Each object has its own attributes holding the data and operations describing the behaviour. Every object can be described as a "black box" with input and output. When two objects interact with each other, they do not have to be aware of the internal structure of the included data and, therefore, changes in the system tend to be local.

Several methods for system development that automatically transforms the models to executable code exist. Lately, object-oriented modelling has become more common, which has several advantages in terms of reusing objects and mapping to the real world. There are several techniques for modelling available. One is the *Unified Modelling Language* (UML) that has become very popular for object-oriented systems analysis (Booch, Jacobson, & Rumbaugh, 1997). UML is a unified version of three different object-oriented modelling techniques. The model is documented in terms of four types of diagram: use case, class, behaviour and implementation.

In UML, the requirements on the system's functionality are described in terms of *use cases* and *actors* (Jacobson et al., 1992). There are several actors (human and non-human) exchanging information with the system and represent what interacts with the system. A non-human actor is, for instance, another computer system. The actors are not actually treated as a feature of the system and, consequently, are not described in detail. In UML, an actor is regarded as a sort of class where each instance of such a class is a user. A user can play the role of several actors (e.g., manager and administrator).

A user performs work by carrying out sequentially related operations on the system. This is called a use case. Each use case is a specific way of interacting with a system. A use case is also regarded as a class, so every scenario performed by a user can be described as an instance of a use case. The instance exists as long as the use case is operating.

Describing the requirements in terms of actors and use cases provides good support when defining how the involved objects communicate in the system. The users become involved at an early stage and they are able to describe their work in a terminology that can be adopted by both the users and the developers of the system.

In general, systems analysis methods are suitable for developing several components of the information system, but they are not sufficient to meet the needs of the user interface designer. Instead, these methods invite the designer to create a design where each function or use case is represented with one window on the screen. Usually the user has to interact with several such windows to complete a task, resulting in a fragmental interface, with a large amount of windows. To be able to design a usable interface, we have found that some additional modelling is needed.

4. Design

There are several definitions of the verb *design*. Usually it means the creation of something new such as a system or a database. When the word design is used in this thesis it usually, unless otherwise stated, refers to the design of the user interface.

4.1. Approaches to design

Wallace & Anderson (1993) have identified four major approaches to interface design: craft, cognitive engineering, technologist and enhanced software engineering. According to the *crafts* approach, each design project is unique and it is therefore impossible to use general methodologies when designing an interface. Instead, skills in human factors are needed. This approach assert that good design comes from good designers. *Cognitive engineering* is an attempt to apply theories of information processing and problem solving to interface design. An example is the *Keystroke-Level Model* (Card et al., 1983), which tries to quantify properties of the users' actions (e.g., the time it takes to move the mouse or enter a letter on the key board) so they can be taken into account during the design process. The *technologists* want to free the programmers from the time-consuming and complex task of interface design by providing them with automated development tools. Finally, the *enhanced software engineering* approach claims that methods for task analysis should be introduced to extend software engineering methods to support the design process.

My approach to design is a mixture of the enhanced software engineering and the craft approach. There are methods for systems analysis that are useful when developing several parts of the computer system. However, there is a need for a methodology that can complement systems analysis methods in order to guide the development towards a design of a usable user interface. The actual design is partially a creative process that cannot be described as a top-down or bottom-up method. Therefore, the competence of the designer(s) will have a critical impact on the interface. However, to be able to make the right decisions, the designer should be supplied by a substantial model describing the users' requirements on the interface.

4.2. Making design decisions

Design is mainly a question about optimising the user interface based on different requirements on the information system. The users are experts on the *domain* and have requirements on how they want to work with the new computer system (Figure 2). These requirements are generally gathered during systems and task analysis. Different *groups of users* have different needs that have to be met in the design. Some relevant user characteristics to be considered are knowledge, skill, experience, education and physical attributes (ISO, 1995). *Rules and recommendations* describing how the interface should look and behave in terms of style guides (e.g., Windows™), and guidelines are also essential. In addition, several companies have their own corporate standard that restricts the layout and behaviour of the interface. The *technical environment* will also limit the design space. The size of the screen and the resolution directly controls how much information is possible to show on the screen simultaneously. The construction tool (e.g., for prototyping and implementation of the interface) may or may not impose restrictions on the use of graphical capabilities. The designer’s task is to optimise the interface based on all these requirements in order to create the best possible solution for supporting the users in their work.

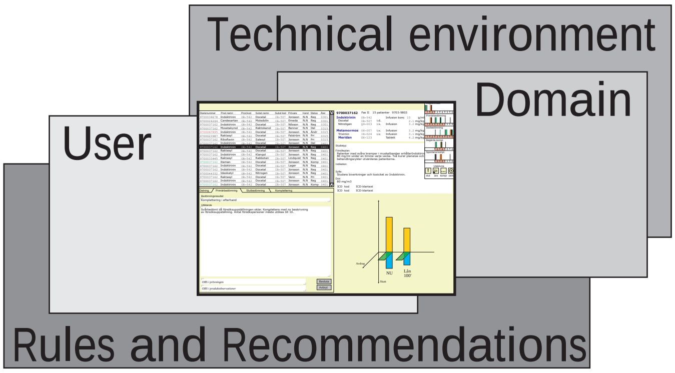

Figure 2. Design is mainly concerned with optimising the user interface based on different requirements.

In the last few years, graphical user interfaces have become common and it is now possible to use many colours, fonts, 3-D representations, etc. The number of degrees of freedom has increased rapidly. By using graphics in the interface, it has become possible to create rich, distinct and effective interfaces that are ease to use and learn. On the other hand, it has increased the risk of making the wrong design decisions. Incorrect use of graphical tools can result in user interfaces that are less effective than the old alphanumeric ones (Nielsen, 1993). The increase of the design space has put higher demands on the designer.

4.3. Supporting the design process

Optimising the user interface is not an easy task and describing exactly how it is done is not really possible. One reason for this is that there are no general solutions. What is optimal in one context can be devastating in another. However, there are some methods can facilitate this process.

One such method is *Design Rationale* (MacLean, Young, Bellotti, & Moran, 1991). The intention is to support the design decision process and the documentation of these decisions. Design questions, options and criteria are specified for each decision. A typical design question could be “how to select different operations in the interface,” where for each question, different design options are identified (e.g., “select from menu” or “select via buttons.”) Each option is documented together with a list of criteria so that the most beneficial design option can be chosen. Design rational can be a good support for structuring decision relevant information during design, though this approach to design is time consuming.

By continuously creating prototypes that are tested on end-users in an iterative way, the design can be significantly improved. This *prototyping* approach enables the designer to test design solutions at an early stage. If potential usability problems are detected at an early stage, they are easier to avoid in the final application. Prototyping is essentially a trail and error approach to design (Johnson, 1992) and hence a thorough analysis must proceed the prototyping phase to insure the efficacy of this approach.

Shneiderman (1992) has suggested a number of measurable central goals to be achieved when designing the system: *Time to learn*, *speed of performance*, *rate of errors by users*, *retention over time* and *subjective satisfaction*. Shneiderman intends that it is difficult to achieve all goals and that designers therefore are forced to make trade-offs. If the rate of error is to be kept very low, then the speed of performance may have to be sacrificed. The designer has to be aware of the trade-offs and must give priority to certain goals.

Other support for the design process is in the form of guidelines or heuristics. Foley (1990) has suggested the following design principles to be considered during the design process: *be consistent*, *provide feedback*, *minimise error possibilities*, *provide error recovery*, *accommodate multiple skill levels* and *minimise memorisation*. These principles can function as a checklist to guide the designer when optimising between design options. However, conflicts between different principles will occur and must be considered.

A user interface can be based on style guides or standards. A style guide describes the functionality and the layout of different interface elements. Normally, a style guide is characteristically general with limited design support for applica tion 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, 1996). Important features of a *domain specific style guide* are composite interface elements corresponding to more complex information structures in the domain. Style guides and standards, although necessary, might restrict the designers' flexibility when creating the interface (Olsson, G ö ransson, Bor älv, & Sandblad, 1993; Grudin, 1989).

4.4. The designer

The design of the interface does not only depend on the methods that are used, but on the person(s) making the actual design decisions. In many in-house systems development projects, software engineers are responsible for the interface design. Even if they are interested in the area of design, they seldom have adequate competence and/or time to create usable interfaces. In participatory design, the users have a very strong position in the design process (Greenbaum & Kyng, 1991). The users should have an essential role in this process since they are the experts on the work to be supported by the system. However, the users are not experts in design or usability.

To be able to create a usable user interface, knowledge is needed in software engineering, cognitive psychology and usability in addition to some artistic capability. Because the person responsible for the design needs to be well oriented in a myriad of areas, it would be more appropriate if interface design is carried out by a group of special *design experts*. One way of improving the individual's capabilities to make good design is to frequently create new interfaces and then assess their usability. The role of experience is important because systems development projects take a considerable amount of time, and software engineers often have to give software development a higher priority, leaving no time for the task of acquiring design experience. A person that could concentrate on design and usability testing only would have better possibilities to obtain such experience.

The user interface designer can be compared with an architect. The architect is partly an engineer and partly an artist. He has to be able to create a structure that is both functional and appealing, fulfil requirements such as the use of standard elements, security and keeping within a specified budget. It is necessary to establish the designer as a role within systems development. Sometimes designers with different skills should be involved such as a user interface designer (or interaction designer) and a graphic designer.

4.5. Skilled versus novice users

Nielsen (1993) identifies three dimensions on which users' experience differs: knowledge about computers in general, expertise in using a specific system and knowledge about the domain. He claims that this will have implications on the design of the user interface. A user with computer experience is able to work with systems in a different way than people without experience since they know how the computer usually reacts when using it. An expert user of the system can make use of short cuts from the keyboard while a novice user usually interacts via the mouse. A domain expert is able to understand domain language and can have a high density of work related information on the screen.

My research mainly concerns user interfaces for *skilled professionals*. A skilled professional is a domain expert using the information system for professional interaction, often several hours per day. When designing an interface for such a user it is more important that the system is efficient in daily use than that it is easy to learn. There are several examples of conflicting demands when designing for skilled users as opposed to novice users; e.g., a skilled user does not need labels to the same extent as a novice user. Today's guidelines often emphasise ease of learning rather than efficiency in daily use (Nygren et al., In press).

4.6. Aesthetics in user interface design

In HCI, much effort has been put on the efficiency and effectiveness of a system. However, the system should not only be functional. It also has to be *pleasant* and *aesthetic*. When a customer chooses between two products, he does not only compare the usefulness of the products. The aesthetic part is also important. We can be happy with a toaster just because we like the look of it even if it burns the bread. A product has to "tell" the user what kind of a product it is and what can be expected from it. Most people do not want fast cars to look like tractors. This can also be applied when designing user interfaces. It should be possible to tell if the system is intended to be practised in a bank or if it is developed for physicians in a hospital. The "look and feel" of a product can be essential for making the user feel comfortable with the system. Also, it becomes a way to compete with rival companies on the market.

Many of the guidelines concerning the aesthetics of an interface have been developed within the area of graphic design. Some guidelines of this nature cannot be regarded as general since they are dependent on trends and fashion. However, several "rules" are applicable, e.g., how to use the limited space to create a design that is balanced, pleasant and attracts the users' attention (for further reading see e.g., Marcus, 1992).

5. Evaluation

Methods for evaluation of user interfaces can be separated into, usability testing methods, where users are involved, and usability inspection methods, where users are *not* involved.

A traditional method for user testing is *performance measurement* where the purpose is to measure whether a usability goal is reached or not. User performance is usually 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 can be found. One advantage of this test method is that the result is given in hard numbers which makes comparison of different design solutions easy. Unfortunately, in most systems development projects there are not enough time, money or laboratory expertise to use this kind of method (Nielsen, 1993). Another problem with laboratory tests is that they are difficult to perform early in the design process since a running prototype and a reasonably complete database is required. Difficulties in sampling, methodological problems in planning, validity and reliability of obtained measures are other pitfalls in usability testing (Holleran, 1991). In addition, evaluation of efficiency in daily use requires skilled users.

In *Thinking aloud* the users verbalise their thoughts while using the system (Lewis, 1982). Through this test, users let the evaluator understand how they view the computer system. This is an inexpensive test that identifies users' misconceptions of the system. It is especially useful when applied by the designer of the interface since direct feedback from the users on the design can be obtained (Jørgensen, 1990). Drawbacks with this method include that it is not very natural for users to think aloud. It is also hard for skilled users to verbalise their decision process since they execute part of their work automatically (Schneider & Shiffrin, 1977; Shiffrin and Dumais, 1981).

Questionnaires are especially useful for issues concerning users' subjective satisfaction and possible anxieties (Nielsen, 1993). Questionnaires can be easily distributed to numerous users and, in addition, is an inexpensive survey method. However, it is difficult to get objective results when using questionnaires since the users' answers are based on what they think they do, not on what they actually do.

One method that includes users, developers and usability experts, and may be carried out early in the design process is *pluralistic walkthrough* (Bias, 1991). Representatives from the three categories meet and discuss usability problems that are associated with the dialogue elements in different scenario steps. In pluralistic walkthrough, the focus is on how users react in different situations. An example could be that a user might claim that, in a certain situation, he or she would "Hold down the shift key while pressing Enter." Pluralistic walkthrough is an effective method in evaluating the learnability of a user interface. It is not as effective when evaluating the efficiency in daily use since the users are not able to predict how they will interact with the system when they have become skilled.

There are also several different inspection methods available. One such method is *cognitive walkthrough* (Polson, Lewis, Rieman, & Wharton, 1992). 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 story is based on assumptions about the user's background, knowledge and goals, and on understanding the problem solving process that enables a user to guess the correct action. Cognitive walkthrough is an inspection method that focuses on evaluating a design for ease of learning, particularly by exploration. It is more difficult to evaluate efficiency in daily use. Problems concerning the content of the interface are rarely identified, due to the evaluator's limited domain knowledge.

Another inspection method is *heuristic evaluation* (Nielsen & Molich, 1990). 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/her work. The method is easy to learn and inexpensive to use. With heuristic evaluation, it is possible to identify many usability problems and it is possible to evaluate early on in the design phase. Unfortunately, it is difficult for end-users without knowledge in HCI to use this method. However, heuristic evaluation can be useful when evaluating the style (i.e., look and feel) of the interface. The heuristics that Nielsen (1993) suggests 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.

Recently, a series of methods for measuring usability has been developed in the ESPRIT MUSIC project (Corbett, Macleod, & Kelly, 1993). The usability of a product is defined through analytic, performance, cognitive workload, and user attitude measures. Analytic measures are performed early and are based on a dynamic model of the user interface and on the user tasks. It estimates performance parameters for human interaction dependent on the use of specific interface objects. Using the DRUM tool for analysis of video recording can enhance performance measurement. Cognitive workload is measured through heart rate variability and respiration and subjectively by the use of questionnaires. Additionally, questionnaires are used to measure user attitudes. This is an extensive method that can be employed in a number of evaluations. However, measuring efficiency in daily use requires that the users have learned how to use the system which can be rather time consuming.

6. The thesis.

6.1. Introduction

The decision of creating a new or improving an information system can be the result of a more thorough analysis of the company or of the users' work situation, an adjustment to a changing world or a hope for better efficiency. When changing the information system, however, it is important to consider work organisation, information handling and competence concurrently (Gulliksen, Lind, Lif, & Sandblad, 1995). If the organisation is changed there might be a need to change the information system as well. New computer support requires new competence, and so on. It is therefore necessary to perform some kind of expectation analysis, where different staff categories can specify expectations regarding work organisation, work procedures and support systems. Analysing the users' current work situation is simply not enough. It is also essential to model the users' future work, otherwise there is a great risk that current work practices will be embodied in the new information system. In this thesis the focus is on how to design the user interface of an information system in a way that it will support the users in their work.

The design of a computer system is seldom the result of one person's work. In most complex development projects, people with different background and expertise are responsible for different parts of the system. Users, software engineers, designers and several other related professionals have to co-operate to create a usable information system. To be able to communicate unambiguously and effectively, these people need a common language. It is not possible to transfer all knowledge from one group to another or from one phase to another without loosing some information. The "gaps" in which information is lost can be a severe obstacle in the development process. "Methodologies for design can bridge or narrow these gaps, improve knowledge communication, make development more efficient and economic, prevent unnecessary work and produce usable interfaces" (Gulliksen et al., 1995, p. 954).

The user interface is the only part of the system with which the user is in direct contact. "The needs of the user should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system." (Norman, 1986, p. 61). User requirements can be gathered through systems analysis which aims at describing the users' future work by specifying different models, such as a data model and a data flow diagram. Methods for systems analysis can be very supportive in the development of several parts of the information system. However, the aid for designing the interface based on these methods are limited (e.g., Floyd, 1986). A new approach to the entire process of interface design is suggested in order to narrow the gap between analysis and user interface design. By introducing useful methods for analysis, user interface design and evaluation enhance the possibilities for creating usable interfaces. The methods should be used in conjunction with existing methods for systems analysis. However, the methods as such are not the solution to all problems. The role of the designer is also important. To make optimal use of the users' domain knowledge, it is also necessary to include these methods in a user-centred design process. Three criteria are suggested to enhance the user interface design process (Figure 3):

1. Use *methods* to support user interface design

2. Iterative *user-centred* systems development

3. Include a *user interface designer* with usability knowledge

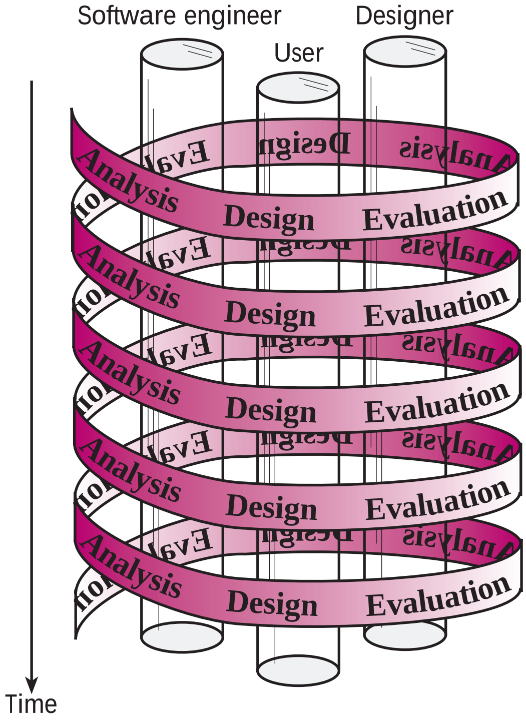

Figure 3. A user interface designer is introduced in the iterative system development process.

The kind of work we primarily study can be characterised as administrative case handling work performed by skilled professionals. These professionals only use and appreciate an information system as long as it effectively supports the main purpose of the task such as to perform case handling (Gulliksen, 1996). Computer systems supporting this work can be found in, for example, banks, health care and economic administration. More general-purpose systems, including word processing, spread-sheets or pure data-entry, are not considered here.

This thesis describes different constituents of the development process introduced to bridge the gap between analysis and user interface design (Figure 4). Two methods for analysing the users' work are presented. These methods are used as a complement to traditional methods for systems analysis in order to gather user requirements that are relevant for the user interface design process. Analysis of Information Utilisation, AIU, (Paper I) is performed through observation interviews with users while they perform their work. The method aims at specifying how information is being used in a work situation. User Interface Modelling, UIM, (Paper II) is performed in modelling sessions with users, software engineers and designers. It is primarily intended to be used as a complement to use cases. Three models are specified, suitable for the design of the user interface: A goal model; an actor's model; a work situation model. AIU and UIM can be applied separately or in conjunction with each other, depending on the particular development project. The third paper describes a workspace metaphor suitable for administrative information systems. AIU and UIM support the use of this metaphor. The actual user interface design is an iterative process, where in each iteration the user interface should be evaluated and redesigned. Paper IV presents a method, ESC, for evaluating the style (i.e., look and feel) and the content (i.e., substance) of the interface separately, to make best use of the users' and the designers' knowledge. The objective of this method is to guide the design of the user interface towards a usable system. Each such iteration has to be short to be cost efficient. Paper V describes a case study where this whole approach to design is used in an applied project. The last paper introduces a method for identifying work environment problems caused by an information system. The method is intended to be employed by Occupational Health Care experts and focuses especially on factors concerning cognitive load.

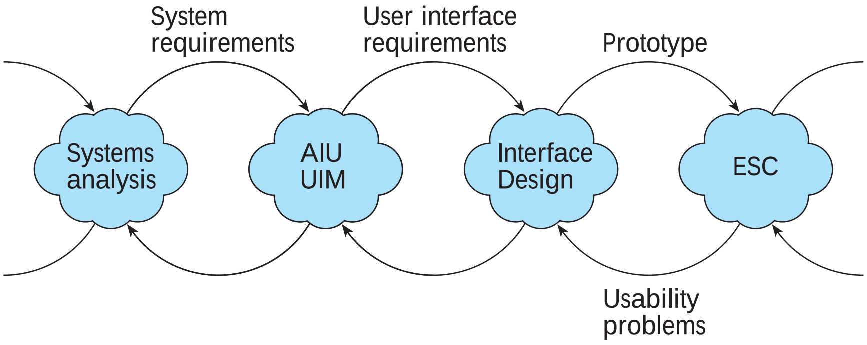

Figure 4. The gap between analysis and design can be bridged, or at least narrowed, with the introduction of methodologies for user interface design.

My approach to design can be viewed upon as a process of optimising the interface based on the different requirements on the system (Figure 5). The introduction of methods for analysing user requirements, such as AIU and UIM, can enable the design of the initial prototype to come fairly close to the "optimal" solution. By evaluating the style and content and then redesign the interface in an iterative process the prototype will come closer and closer to what the users need.

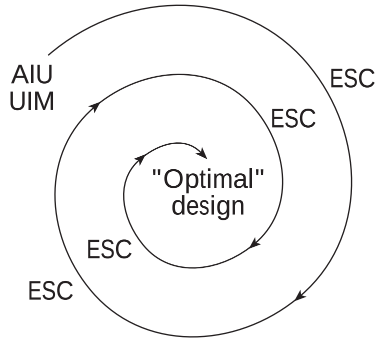

Figure 5. Analysis of Information Utilisation (AIU) or User Interface Modelling (UIM) enables the initial prototype the come fairly close to what the users require. The design can be "optimised" by, in each iteration, Evaluate the Style and Content separately (ESC) and then redesign the interface.

6.2. Method

"The purpose of research in science is to bring a higher level of confidence and certainty to our understanding than is possible by belief, faith, or reason alone. Science therefore requires a highly critical attitude." (Neale & Liebert, 1986, p. 9)

HCI research has a short history compared with many other disciplines. To help establish HCI and its contribution to the world of science, the research emanating from its corridors has to follow the established scientific "rules." HCI is a broad area of research and scientific methods are borrowed from both natural, behavioural and social sciences. The methods that are best suited depend on the purpose of the studies.

Reliability and validity

Reliability and validity are highly visible concepts in science. *Reliability* is concerned with the consistency or stability of a test or measure from one occasion to the next. When repeated measurements of the same phenomenon yield identical or very similar results, the measurement instrument is said to have high reliability. *Validity*, which should not be confused with reliability, refers to a measurement instrument or a test that measures what it is presumed to measure; the degree to which a measure or test is free of systematic error. A classic example is the IQ-test. There are several reliable IQ-tests, but how do we know that the IQ tests really measure intelligence?

Quantitative versus Qualitative studies

Within *quantitative* research, reliability and validity can be presented as numbers, making it possible to compare different tests. In *qualitative* research, the concepts are not as clear. There are different notions on the how to define whether a study is valid or not. Kvale (1989) stresses that validation in qualitative research comes from investigating, continuously checking, questioning and theorising on the nature of the investigated phenomena. In a broad sense a study can be regarded as valid if the reliability has been checked, if there is an assurance that empirical evidence exists and whether the data has been analysed in a proper way (Svensson, 1996). Gummesson (1991) emphasises that to obtain validity it is sometimes necessary to collect data from different sources. By comparing these results, better support for the validity of the results can be achieved.

It is not always obvious when a quantitative study is more appropriate than a qualitative study and vice verse. Although in quantitative studies results are given in hard numbers that are easy to compare, they do not always capture aspects of the studied situation that are not "visible." It can be difficult to "read between the lines" when using quantitative methods. Qualitative methods, on the other hand, can be the solution when a deeper understanding is sought after. Questions like why and how are sometimes easier to answer when using a qualitative method. A combination of the two is often the best solution. Questionnaires can be used, for instance, to identify different problems within a given area, and qualitative interviews can be performed to unravel a deeper understanding as to why the problems occur.

Laboratory versus Field studies

Studies can be performed in either a laboratory setting or in a field setting. A *laboratory* setting has an advantage in that it allows the researcher to control the variables under study systematically. Extraneous (confounding) variables are easier to eliminate or neutralise. A disadvantage with laboratory studies is that the subjects know they are being investigated and thus are less likely to act naturally and spontaneously. A further disadvantage is that the environment is novel (Neale & Liebert, 1986) and runs the risk of being artificial. A result that points in one direction when performed in the laboratory is not necessarily valid in the field. A *field* study has the advantage that it is performed in an environment well-known to the subjects. On the other hand, it is more difficult for the observer to foresee potential disturbance. Both methods have their advantages and disadvantages and the choice of method is highly dependent on the purpose of the study. Sometimes it can be a good idea to perform experiments, both in a laboratory setting and out in the field.

In my thesis most of the papers describe methods for analysis, design and evaluation of user interfaces. When developing methods, we are more interested in a holistic view then when studying a smaller part of the design process (e.g., how an information entity is perceived). Gummesson (1991) asserts that the whole is not always the sum of its parts. It is seldom possible to compare different methods within a project because of time limits and costs. Still, it is essential to understand how methods can be applied in a real context. Goldkuhl (1994) argues that methods can be regarded as "good" if they are "well-grounded." Goldkuhl has proposed the following criteria for such a process.

- *Internally grounded*: The method should be internally congruent; the different parts of the method should be consistent and fit well together.

- *Empirically grounded*: The method should be empirically tested and evaluated. Empirical data should be collected and the method should be evaluated in relation to these results. This can be done in different ways such as through action research, experiments in laboratories or by post-studies of usage of the method.

- *Theoretically grounded*: The method should be evaluated against and related to existing theory.

Action Research

When performing *action research*, two groups of people are involved, namely the researchers and the clients. Usually the client has a practical problem to be solved. The researchers are experts in the particular problem domain and are therefore involved in the client's work. In action research, the researcher is not an independent observer; instead, he takes an active part in the process of identifying new solutions to a certain problem. If the action research is performed properly, both parties will receive benefits from the co-operation. The client will be provided a solution to the problem and hopefully gain new knowledge and results will be readily available to the researcher that can be used in terms of scientific contribution.

Authors have defined the term Action Research differently, though the most widely accepted definition is the one given by Rapoport (1970, p. 499). According to this investigator, "Action research aims to contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration within a mutually acceptable ethical framework."

Rapoport discusses three dilemmas in action research related to the issues of *ethics*, *goals* and *initiatives*. The ethical dilemma concerns aspects related to confidentiality and protection of the subjects. Another potential conflict is that of being able to spread the results within the academia and at the same time avoid making the clients' internal secrets available to competitors. The goal dilemmas are related to the different goals of the client and the researcher. Usually the client's primary goal is to obtain a solution to a specific problem. The researchers are more concerned with results that lead to a fuller understanding within their field of inquiry. It is necessary that the researcher is given time to perform high-quality research, including in-dept understanding and evaluation. Rapoport also means that the initiative in carrying out different studies not only should come from the client but also from the researcher.

Warmington (1980) also points out some problems concerning the *goals*, *values* and *roles* in action research. He suggests that the goals are highly integrated with personal goals (e.g., that the researcher needs to gain some rewards from the research) and that the managerial has goals concerning recognition in the organisational hierarchy. Warmington further claims that the researcher and the clients often have different values. The researcher is treated to be critical to new ideas, while support and encouragement are more common in a company in order to generate new solutions. The role of the researcher is also discussed since it can be difficult to be both an academic outsider and work as a colleague to the members of the organisation.

Action research places high demands on the researcher. It is expected of the researcher to adjust and choose methods according to new requirements in a changing environment (e.g., new deadlines may appear in the project, people may switch positions in the organisation, etc.).

Choice of method

During the past several years, I have been involved in several applied development projects, most of which have been connected with the Swedish National Tax Board. The research method that has been applied to some of these projects reflects an action research approach. The co-operation with the client started about three years ago. Initially, we were invited as observers in order to collect data and to gain knowledge about the processes involved. Field studies, observations and interviews were all employed as a means to collect data. We collected documentation and attended meetings that concerned the various projects. In some projects we participated as experts in design and evaluation and supported the software engineers in the design process. This enabled us to identify weaknesses in the systems development process and define areas where new methods and tools were needed. New methods were then developed and evaluated within different projects. The results of these evaluations were used for further refinements of the methods.

Both quantitative and qualitative methods have been applied depending on the kind of studies that were performed. Most of my work has been performed under field conditions since I believe that this technique allows the researcher to study behaviour in real-life situations.

6.3. Summary of the papers

Paper I - Analysis of Information Utilization (AIU)

Analysis of Information Utilisation (AIU) is a method for specifying how information entities encountered in the information analysis are being used in a work situation. AIU aims at capturing aspects of the user's work that are not modelled in traditional information analysis. AIU focuses particularly on factors of computer-supported work that are related to cognitive load. For the skilled user, it is important to minimise the cognitive load caused by the information system to enable the user to work effectively. The results of the method support the human-computer interface designer with relevant and appropriate information.

An important concept within AIU is the work task. A work task is defined as a continuous moment of work specifically performed to reach a goal. Such a task includes some kind of judgement that is terminated by a decision, which is defined as made when it has been documented. This definition is because all administrative work includes judgements and decisions. Creating user interfaces that effectively support the user while performing these tasks will decrease the amount of cognitive workload.

AIU is performed through "observation-interviews" with representative users while they perform their work. Studying the users in their actual work situation increases the possibility to identify processes that the users are not aware of (e.g., because they have been automised; Schneider & Shiffrin, 1977). This implicit knowledge is difficult for users to mediate but, by interviewing them while they perform their work, the performance can be identified and analysed. Another reason for performing observation interviews is that professional users often apply ingenuity in solving tasks that can be of interest when creating the information systems.

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. AIU is not a new method for system development, but complements today's methods for systems analysis with more design-relevant information. To illustrate, an example is presented describing interface design based on AIU in a system development project within the Swedish National Tax Board.

Paper II - User Interface Modelling - adding usability to use cases

This paper describes a methodology for gathering users' requirements relevant for the design of the user interface. The method is to be employed in conjunction with use case modelling. Use cases are modelled for specifying the functionality of the system according to an object-oriented approach.

User Interface Modelling (UIM) has been developed with the designer in focus. The main task of the designer is to optimise the user interface based on different requirements. To facilitate this task, three different models are defined, namely a goal model, an actor's model and a work situation model.

The goal model is a specification of high level goals that the users want to achieve when working with the information system. The specified goals are important when making design decisions based on the different requirements.

The actor's model is a description of each actor using the system. An actor represents a group of users in which a single user can play the role of several actors. For the designer it is essential to know who these actors are. Characteristics that are described for each actor are position at office, product experience, task experience and frequency of use.

In the work situation model, specification of the different work situations is represented. In each work situation, which can be described as a core work task, the user communicates with different use cases. When designing the interface, it is important to support the users in these work situations. The objects needed and the actions performed by the actors are documented for each use case in a work situation. Finally, there are characteristics of the attributes and operations that have not been documented in the data model that should be considered when designing the interface. These characteristics are documented in a diagram.

UIM is performed in modelling sessions with end-users, software engineers and designers. A modelling leader is responsible for the documentation of the models and the guiding of the sessions. UIM is documented with a graphic notation based on the use case approach.

A design example is included to further illustrate the use of the method.

The method has been used and evaluated at the Swedish National Tax Board. The results of the evaluation indicate that UIM is both easy to use and a good support in the design process.

Paper III - Workspaces enhance efficiency - Theories, concepts and a case study

This paper presents a workspace metaphor intended to be applied when designing user interfaces for administrative systems. The metaphor is suggested as an alternative to the commonly used multiple-windows metaphor.

Two different approaches to design are discussed: The process-oriented and the workspace-oriented approach. The authors claim that the process-oriented approach invites the designer to create fragmental user interfaces that can cause unnecessary cognitive load. With a workspace-oriented approach the users' work is analysed in terms of work situations (i.e., core work tasks), that can be effectively supported with the use of the workspace metaphor.

Each workspace is here defined as a "full screen" that corresponds to a work situation. Such a workspace must include all tools and information needed by the user to complete a work task. Several work tasks can be performed in each workspace and the user can switch between different workspaces via a panel that can be visualised upon demand. This ability to change workspaces facilitates overview and navigation.

Some recommendations on how to design an effective interface when using this metaphor are presented.

The paper also describes a case study performed at the Swedish National Tax Board where this metaphor was implemented in an applied project.

Paper IV - Evaluating Style and Content separately during the design of the user interface

During our work within different development projects, we have seen that a user centred approach to design does not necessarily lead to a user interface that is optimal for the end-users. One reason for this is that the users' and the designers' knowledge are not utilised in a proper way during the design process. To overcome this problem an evaluation method is introduced where the style (i.e., look and feel) and the content (i.e., substance) of the interface are evaluated separately.

The first phase of evaluation (style) is an inspection of the interface performed by persons with HCI-knowledge. The second phase (content) is an evaluation performed together with a group of potential end-users based on a number of scenarios. Both phases include a checklist with "heuristics" to guide the evaluation. We have made a clear distinction between heuristics concerning the contents and heuristics concerning the style of the interface. By including this distinction, each group of evaluators can put more effort into the area where they are experts. Users are experts on the work to be supported by the system and HCI-people are experts on usability issues. When HCI experts inspect an interface, they are probably able to determine if, for example, default values are needed. However, the users themselves are probably better at determining *which* values that should be shown by default. With this distinction, the evaluation can be made more cost efficient.

The evaluation method presented in this paper is meant to be included early in the iterative design process as a tool to guide the development of the interface towards a design with high usability. It is easier to improve a prototype if the potential usability problems are discovered early. The method can be applied even if the first prototype is only a paper mock-up.

The method has been evaluated in two studies. The first study showed that it was possible to split the evaluation into contents and style. The second study showed that it was possible to identify more potential usability problems when using this method, then without support from a method.

Paper V - Adding usability - a case study

This paper presents a case study performed at the Medical Products Agency (MPA) in Sweden. The purpose of this study was to use some of the methods described in this thesis in an applied project. A combination of Analysis of Information Utilisation and User Interface Modelling has been used to accrue requirements relevant for the design. These methods served as a complement to existing models. The results of AIU/UIM were applied as an input for the design process.

The design was performed in an iterative process. During each iteration the prototypes were evaluated and redesigned. The style and the content of the interface were evaluated separately using ESC. The end-users evaluated the contents of the interface and the usability experts evaluated style.

Two studies were performed. Study one showed that the results of AIU/UIM were a good support for the design of the first prototype. Study two showed that more usability problems were identified when using ESC than when using heuristic evaluation.

Paper VI - A Practical Method for Evaluation of Human-Computer Interfaces.

This paper introduces a method (ADA) for evaluating information systems in use. This method was originally developed for the Swedish Occupational Health Care as a complement to their more traditional methods for evaluating the physical and psychosocial work environment.

The ADA method aims to identify usability and cognitive work environment problems in computer supported work. The results from this evaluation are to be utilised as a basis for further analysis and for discussions on how to improve the system.

An ADA evaluation is performed as "observation interviews" with different categories of users under actual work conditions. The method is meant to be applied in the end-users ordinary work environment. During each ADA evaluation, an evaluator identifies potential problems. An interview guide with major usability aspects is always available to support the evaluator. The findings are interpreted using an interpretation guide.

An evaluation of the ADA method is presented based on results from usage at three different systems. The results indicate that the reliability and the validity of the method are high.

7. Discussion

7.1. A new approach to design

A new approach to design of user interfaces is introduced in this thesis. I have presented methods for the entire chain of modelling, user interface design and evaluation. These methods are meant to be used in conjunction with existing methodologies for systems analyses to increase the possibilities of creating usable user interfaces. The methods should be included in a user-centred iterative systems development process. Thus far, this approach has only been employed for information systems in the administrative domain. However, there are reasons to believe that it is applicable in other domains as well.

The methods presented in this thesis have shown to be a good support for user interface design. However, to utilise the strength of the methods fully, a user interface designer with skills in usability should be involved in the development process. Unfortunately HCI experts are rarely included in system development projects, at least in Sweden. Software engineers, who seldom have the time, experience or interest needed to design a usable system are often those who are responsable for the user interface design. They are usually more concerned about creating a system that fulfills the functional requirements. This is of course important, but it is not enough, the usability of the system is also essential. There is a need to establish a role of a designer in systems development projects. A designer can function as a link between users and software engineers, as a kind of ”defender” for the users. Designing user interfaces requires experience and skill. Many system development projects extend over a long period, some as long as three to four years. The time given to design a prototype in such a project may be less then four months. Since the software engineers often participate throughout the entire project, they do not have the opportunity to acquire the experience necessary to design interfaces. It would be more appropriate to be assigned a specific role that allows one to concentrate on interface design only, and then advance to the next project to gain more skill and experience in interface design.

If a designer only creates the user interface, the design must be documented so that it can be implemented by the software engineers. Usually the prototype is the only documentation of the design. We have seen examples in which the prototype and the final product are completely different, indicating that several changes have been brought about during the long process from prototype to final product. Design decisions are continuously being made during construction, due to constraints in the development tool, change in plans, etc. However, the individuals coding the application have to realise that there is a reason for every design decision and that it is not appropriate to make changes without considering the costs in usability. One way to overcome this problem could be to continue with evaluations throughout the coding process. Both designers and users should be able to review the system several times before it is complete. We are currently planning to study the role of the designer more carefully. Interesting questions are: Which skills are required? How can the designer play an integrated role in system development? How can users, software engineers and designers communicate during this process?

7.2. HCI in practice

Bellotti (1988) found that HCI methods were not widely used in real development projects 10 years ago. Similar results have been found in Swedish settings in a more recent study by Katzeff & Svärd (1995). On the other hand, this study also shows that most companies are cognisant that they have a need for new methodologies. Every year new methods emerge from the academia, though, they have had very limited impact on the practitioners. I think there are several reasons for this paradox. First, learning to use a method takes time. Even if most project leaders know that a good method could be helpful, they are not prepared to run the risk of altering their time schedule. Another problem is that several methods developed in academia never succeed in leaving the desk. To be able to "sell" a method, it has to be integrated in tools and marketed outside the ordinary channels for reporting research results (e.g., conferences and journals). It has to become a well-defined product. Some people would argue that academia is only responsible for producing research results and that practitioners are free to use these findings in product development. Obviously, this does not work. I think that it is crucial to develop and maintain a good relation to practitioners that are interested in creating and advertising a product. In my particular field, I have had the opportunity to work in co-operation with different companies and use the methods in real development projects. This has been extremely fruitful, both because I have been able to evaluate my work in a real context and also because some of my work has become an integrated part of the practitioners' development tools.

I believe that many companies are simply not aware of the need for usability. One way to "market" usability is to show that it can be profitable. There exists a body of literature with success stories showing that a usable interface can increase the productivity in a company, but this in itself is not enough. The demands have to come from the potential end-users, both concerning commercial products and in in-house development projects. If the users can make the developers aware of the importance of usability, then more resources has to be allocated to the design process. Internet might be of some help here. With the introduction of Internet, more people operate computers on a daily basis. Media has begun to write about usability and "user-friendly" interfaces and ordinary people have acquired an interest in the subject. When users judge the usability of a product instead of just the number of functions, the market will be forced to react, which should lead to discernible changes.

7.3. HCI in the future

When discusing computers today, we usually refer to a PC with a display, keyboard and mouse. This kind of device will not disappear within the immediate future. However, computers will definitely play a vital role in our future everyday life, but not in the way, we are used to. Several of our ordinary tools today will be computer-ised tomorrow (e.g., books, pencils and clothes). The difference is that we will not refer to them as computers but simply an integrated part of our ordinary tools. Virtual reality will also have a considerable impact on many individuals' everyday life. In the future, it will not be necessary to travel to the United States to meet your relatives when you can meet them virtually in your living room. Computers will be operated by more people at work and in their homes, for both amusement as well as for solving practical problems. We have already witnessed an incredible explosion in the use of the Internet for all kinds of services. No matter what kind of technical devices that will be used in the future, it will still be important to understand the users and their needs. An interesting challenge for the future is to adjust the methods presented in this paper to support development of usable systems for 3D or virtual reality applications. I have confidence in that the usability issues will become more important as the technical revolution continues. I hope that we will see new and useful products that are not just propelled by the technical revolution but also by the actual needs of the people.

References

ARONSSON, G., ÅBORG, C., & ÖRELIUS, M. (1988). *Datoriseringens vinnare och förlorare* [Winners and losers from computerisation. Summary in English]. Arbete och hälsa. Arbetsmiljöinstitutet, 27.

BELLOTTI, V. (1988). Implications of current design practice for the use of HCI techniques. In D. Jones, R. Winder (Eds.), *People and computers IV, Proceedings of Fourth Conference on Human-Computer Interaction* (pp. 13-34). Cambridge, UK: Cambridge University Press.

BENYON, D. (1990). *Information and Data Modelling.* Oxford, UK: Blackwell scientific publications.

BENYON, D. (1992). The role of task analysis in systems design. *Interacting with computers, 4* (1), 102-123.

BIAS, R.C. (1991). Walkthroughs: Efficient Collaborative Testing. *IEEE Software, 8* (5), 94-95.

BOEHM, B.W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer, 21* (5), 61-72.

BOOCH, G., JACOBSON, I., & RUMBAUGH, J. (1997). *Version 1.0 of the Unified Modelling Language,* (On-line). Available: http://www.rational.com/uml/references/docset.html.

CARD, S.K., MORAN, T.P., & NEWELL, A. (1983). *The Psychology of Human-Computer Interaction.* Hove, England: Lawrence Erlbaum Associates, Inc.

CARROLL, J.M., & KELLOGG, W. A. (1989). Artifacts as theory-nexus: Hermeneutics meets theory-based design. In K. Bice, & C. Lewis (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 89* (pp. 7-14). New York, NJ: ACM.

CORBETT, M., MACLEOD, M., & KELLY, M. (1993). Quantitative Usability Evaluation – The ESPRIT MUSiC Project, In G. Salvendy, & M.J. Smith (Eds.), *Proceedings of the Fifth International Conference on Human-Computer Interaction, HCI International' 93* (pp. 313-318). Amsterdam: Elsevier Science Publishers.

DEMARCO, T. (1978). *Structured Analysis and System Specification.* New York: Yourdon Press.

EHN, P., & LÖWGREN, J. (1997). Design for Quality-in-use: Human-Computer Interaction Meets Information Systems Development. In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 299-313). Amsterdam: Elsevier Science B.V.

FLOYD, C. (1986). A comparative evaluation of system development methods. In T.W. Olle, H.G. Sol, & A.A. Verrijn-Stuart (Eds.), *Information systems design methodologies: Improving the Practice* (pp.19-55). Elsevier Science Publishers B.V.

FOLEY, J.D. (1990). Dialogue Design. In J.D. Foley, A. van Dam, S.K., Feiner, & J.F. Hughes (Eds.), *Computer Graphics: Principles and Practise* (pp. 391-431). Reading, MA: Addison-Wesley.

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

GOLDKUHL, G. (1994). *Välgrundad metodutveckling* [Well-grounded methods development] (Technical Report LiTH-IDA-R-94-04). Linköping, Sweden: Dept. of Computer and Information Science, Linköping University.

GOOD, M., SPINE, T.M., WHITESIDE, J., & GEORGE, P. (1986). User-derived impact analysis as a tool for usability engineering. In *Proceedings of CHI' 89 Conference on Human Factors in Computing Systems* (pp. 241-246). New York: ACM.

GREENBAUM, J., & KYNG, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

GRUDIN, J. (1989). The Case Against User Interface Consistency, *Communications of the ACM, 32* (10), 1164-1173.

GRUDIN, J. (1991). Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer, 24* (4), 59-69.

GRUDIN, J. (1992). Utility and usability: research issues and development contexts. *Interacting with computers, 4* (2), 209-217.

GULLIKSEN, J. (1996). Case Handling Models as a Basis for Information System Design. In C.A. Ntuen, & E.H. Park (Eds.) *Human Interaction with Complex Systems-II* (pp. 225-256). Norwell, MA: Kluwer Academic Publishers.

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.), *Proceedings of HCI International' 93* (pp. 15-20). New York: Elsevier Science B.V.

GULLIKSEN, J., LIND, M., LIF, M., & SANDBLAD, B. (1995). Efficient Development of Organizations and Information Technology – A Design Approach. In Y. Anzai, & K. Ogawa (Eds.), *Symbiosis of Human and Artifact. Proceedings of the 6th International Conference on Human-Computer Interaction, HCI International' 95* (pp. 951-956). Amsterdam: Elsevier Science B.V.

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* (pp. 21-36). New York: Springer Verlag.

GUMMESSON, E. (1991). *Qualitative Methods in Management Research.* Newbury Park: Saga publications.

HENDERSON, A., & CARD, S.K. (1986). Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. *ACM Transactions on Graphics, 5* (3), 211-243.

HOLLERAN, P.A. (1991). A methodological note on pitfalls in usability testing. *Behaviour & information technology, 10* (5), 345-357.

HOLTZBLATT, K., & JONES, S. (1993). Contextual Inquiry: A Participatory Technique for System Design. In D. Schuler, & A. Namioka (Eds.), *Participatory Design: Principles and Practices* (pp. 177-210). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

ISO/DIS 9241 (Draft). Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 10: Dialogue Principles, Part 11: Guidance on Usability, Part 13: User Guidance (1995). Geneva, Switzerland: International Organization for Standardization.

JACOBSON, I., CHRISTERSON, M., JONSSON, P., & ÖVERGAARD, G. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach.* Wokingham, England: Addison-Wesley Publishing Company.

JEFFRIES, R. (1997). The Role of Task Analysis in the Design of Software. In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 347-359). Amsterdam: Elsevier Science B.V.

JOHNSON, P. (1992). *Human-Computer Interaction, Psychology, Task Analysis and Software Engineering.* London: McGraw-Hill Book Company.

JØRGENSEN, A.H. (1990). Thinking-aloud in user interface design: a method promoting cognitive ergonomics. *Ergonomics, 33*, (4), 501-507.

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

KVALE, S. (1989). *Issues of Validity in Qualitative Research.* Lund, Sweden: Studentlitteratur.

LEWIS, C. (1982). *Using the 'thinking-aloud' method in cognitive interface design.* (IBM Research Report RC 9265, 2/17/82). Yorktown Heights, NY: IBM T.J. Watson Research Center.

LIND, M. (1991). *Effects of Sequential and Simultaneous Presentations of Information* (Rep. No. 19, CMD). Uppsala, Sweden: Uppsala University.

LIND, M., NYGREN, E., & SANDBLAD, B. (1991). Kognitiva arbetsmiljöproblem och gränssnittsdesign [Cognitive work environment problems and design of user interfaces] (Rep. No. 20, CMD). Uppsala, Sweden: Uppsala University.

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, 6*, 201-250.

MARCUS, A. (1992). *Graphic Design for Electronic Documents and User Interfaces.* New York: ACM Press.

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

MULLER, M.J., HASLWANTER, J.H, & DAYTON, T. (1997). Participatory Practices in the Software Lifecycle. In M. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 255-297). Amsterdam: Elsevier Science B.V.

NEALE, J.M., & LIEBERT, R.M. (1986). *Science and Behaviour. An Introduction to Methods of Research.* Englewood Cliffs, New Jersey: Prentice-Hall.

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

NIELSEN, J., & MOLICH, R. (1990). Heuristic evaluation of user interfaces In J.C. Chew, & J. Whiteside (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 90* (pp. 249-256). New York, NJ: ACM.

NORMAN, D.A. (1986). Cognitive Engineering. In D.A. Norman, & S.W. Draper (Eds.), *User Centered System Design* (pp. 31-61). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

NYGREN, E., ALLARD, A., & LIND, M. (In press). Skilled Users' Interpretation of Visual Displays. *Human Computer Interaction.*

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* (pp. 235-239). New York: ACM.

OLSSON, E., GÖRANSSON, B., BORÄLV, E., & SANDBLAD, B. (1993). Domain Specific Style Guide – Design and Implementation. In *Proceedings of the MOTIF' 93 & COSE International User Conference* (pp. 133-139). Washington, DC: Open Software Foundation.

PAYNE, S.J., & GREEN, T.R.G. (1989). Task-Action Grammar: the model and its developments. In D. Diaper (Ed.), *Task Analysis for Human Computer Interaction.* Chichester: Ellis Horwood.

POLSON, P.G., LEWIS, C., RIEMAN, J., & WHARTON, C. (1992). Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies, 36* (5), 741-773.

PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., & CAREY, T. (1994). *Human-Computer Interaction.* Wokingham, England: Addison-Wesley Publishing Company.

RAPOPORT, R. (1970). Three dilemmas in action research. *Human Relations, 23* (6), 499-513.

RASMUSSEN, J. (1983). Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics, 13* (3), 257-266.

REASON, J.T. (1987). Generic Error-Modeling System (GEMS): A Cognitive Framework for Locating Common Human Error Forms. In J. Rasmussen, K. Duncan, & J.

Leplat (Eds.), *New Technology and Human Error* (pp. 63-83). Chichester, England: Wiley.

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

SHNEIDERMAN, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (2nd ed.).* U.S.A: Addison-Wesley Publishing Company, Inc.

SHEPHERD, A. (1989). Analysis and training in information technology tasks. In D. Diaper (Ed.), *Task Analysis for Human Computer Interaction* (pp. 15-55). Chichester, England: Ellis Horwood.

SHIFFRIN, R.M., & DUMAIS, S.T. (1981). The Development of Automatism. In J.R. Anderson (Ed.), *Cognitive Skills and their Acquisition.* Hillsdale, NJ: Erlbaum.

SOLSO, R.L. (1991). *Cognitive Psychology.* Boston: Allyn and Bacon.

SCB (1995). *Datorvanor 1995* [Computer habits 1995]. Stockholm: Nordstedts Tryckeri AB.

SVENSSON, P.-G. (1996). Förståelse, trovärdighet eller validitet. [Understanding, reliability or validity]. In P.-G. Svensson, & B. Starrin (Eds.), *Kvalitativa studier i teori och praktik* [Qualitative studies in theory and practice]. Lund, Sweden: Studentlitteratur.

TANNER, W.P., & SWETS, J.A. (1954). A decision-making theory of visual detection. *Psychological Review, 61* (6), 401-409.

WALLACE, M.D., & ANDERSON, T.J. (1993). Approaches to Interface Design. *Interacting with Computers, 5* (3), 259-278.

WARMINGTON, A. (1980). Action research: Its methods and its implications. *Journal of Applied Systems Analyses, 7,* 23-39.

WAUGH, N.C., & NORMAN, D.A. (1965). Primary memory. *Psychological Review, 72* (2), 89-104.

WIXON, D., HOLTZBLATT, K., & KNOX, S. (1990). Contextual Design: An emergent View of System Design. In J.C. Chew, & J. Whiteside (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 90.* ACM/SIGCHI. Reading, MA: Addison-Wesley Publishing Company, Inc.