V

# Adding usability - a case study

MAGNUS LIF

*Center for Human-Computer Studies (CMD), Uppsala University, Sweden.*

## Abstract

This paper describes a case study that has been performed at the Medical Products Agency (MPA) which is the authority responsible for the whole process of drug approval and control in Sweden. The purpose with this case study was to apply a new approach to user interface design in the development of an information system for employees at the MPA. Analysis of Information Utilisation was used in conjunction with User Interface Modelling to gather and specify requirements relevant for the design of the user interface. The design was performed in an iterative process where the style and the content of the interface was evaluated during each iteration.

The work was divided into two phases. Phase one included analysis of user requirements and design of a first version of a prototype. In the second phase the iterative process of design was studied. The methods used in the two phases were evaluated and the results show that this approach to design increases the possibilities to create interfaces with higher usability.

## 1. Introduction

Many system development projects fail to provide usable systems. Several projects end up with fragmentary interfaces that do not give sufficient support for the users in their work. There can be several reasons for this, for example, organisational problems, lack of competence, methodological problems, and so on.

In the kind of projects where we have been involved, the users' work has been analysed with methods for systems analysis. These methods usually result in a data model, a data flow diagram (e.g., DeMarco, 1978) and some kind of functional description. Such methods are suitable for developing several parts of the system but, they do not give sufficient support for the design of the user interface (e.g., Floyd, 1986).

Today, object-oriented methods and techniques are common in systems development. The functionality of a system is often specified in terms of actors and use cases. A use case is a description of the systems behaviour during interaction with an actor (Jacobson, Christerson, Jonsson, & Övergaard, 1992). Use cases are useful in the development process, however, the support for designing the user interface is limited. Instead, it seems like this approach invites the designer to create user interfaces where each use case corresponds to a window (with secondary windows) on the screen. Usually the user has to communicate with several use cases while performing a task. Therefore, there is a risk that the screen soon is filled with windows, which forces the user to spend time on moving and rearranging windows instead of performing the actual work. Such an interface is designed to support the use cases instead of the users.

Wallace and 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. They 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 key stroke-level model (Card, Moran, & Newell, 1983), which tries to quantify properties of the users' actions, for example 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 wants 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.

Our approach to design is a mixture between the enhanced software engineering and the crafts approach. We have stated elsewhere that there is a gap between analysis and design (Gulliksen, Lind, Lif, & Sandblad, 1995). There is a need for a methodology that can complement methods for systems analysis by specifying requirements that are relevant for the design of the user interface. Attempts have been made to extend systems analysis methods with methods for task analysis (e.g., Sutcliff & Wang, 1990). However, we mean that existing methods for task analysis do not meet the needs of the designer since the definition of tasks are too fine-grained to be useful in the design process (Gulliksen, Lif, Lind, Nygren, & Sandblad, 1997). A specification of larger, concatenated, tasks would be more sufficient for the design. Furthermore, the interface should be created in co-operation with the users. Greenbaum & Kyng (1991) mean that the users should not only be involved but have a very strong impact on the design of the system. We mean that the users are of utmost importance because they are experts in the work that is to be supported by the system. However, this does not mean that the users should make all design decisions. User are *not* competent designers. Therefore it is necessary to involve a usability expert in the process who can be responsible for the design of the user interface. To assure that users and interface designers can co-operate effectively, the interface should be evaluated and redesigned in an iterative process (Gould & Lewis, 1985). The content of the interface (i.e. the substance) should be evaluated by the users, and the style (i.e. the look and feel) by usability experts. The results of these evaluations can guide the design towards a system with high usability.

We have previously defined methods for Analysis of Information Utilisation, AIU (Gulliksen et al., 1997), User Interface Modelling, UIM (Lif, 1998a) and for Evaluating Style and Content separately, ESC (Lif, 1998b), to facilitate the design of the user interface. This paper describes a case study where these three methods have been used in a system development project. The purpose with this case study is not only to evaluate the different methods but to introduce a new approach to design. We mean that the following criteria are important for a successful outcome:

- Use of methods to support the user interface design process
- Iterative user centred systems development
- Include a user interface designer with usability knowledge in the process

In this case study the three methods have been applied to enhance the design process. The methods complements existing methods for systems analysis. Following, a short introduction to use cases, AIU, UIM and ESC will be given.

# 2. Short description of the methods in use

The different methods will not be described in detail. For further reading, consult the referred methods.

## 2.1. Use cases

When analysing the users' work the functionality of the system can be described in terms of actors and use cases (Jacobson et al., 1992). Actors can be both human and non-human (e.g., another system). A human actor represents a category of users in an organisation, such as a secretary. Each actor communicates with different use cases in order to achieve a goal. A use case describes the interaction between an actor and the system. The sum of all use cases represents the entire functionality of the system.

The relations between actors and use cases are documented in a use case diagram with a graphical notation. Each use case is further described in free text, specifying how the actors interact with the system.

### 2.2. User Interface Modelling (UIM)

UIM is used as a complement to use cases in order to gather aspects of the users work that are relevant for the design (Lif, 1998a). The main task in design is to optimise the user interface based on different requirements. To facilitate this process three different models are specified: a goal model, an actor's model and a work situation model.

The goal model is a specification of high-level goals to be achieved by users when working with the information system. These goals are essential 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 category of users. One user can play the role of several actors. These actors are identified, but not described in detail, in the use case modelling. In UIM a deeper analysis is performed to specify important user characteristics relevant for the design, such as product experience, task experience, and frequency of use.

In the work situation model, the users' requirements on the work is specified. A work situation can be described as a core work task. In each work situation the user communicates with different use cases. When designing the interface it is important to support the users in these work situations. The information objects needed and the actions performed by the actors are documented for each use case in a work situation. Finally, design relevant characteristics of attributes and operations are specified.

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

### 2.3. Analysis of Information Utilisation (AIU)

AIU aims at capturing aspects of the users work that are not modelled in traditional systems analysis and focuses particularly on factors of computer supported work, related to cognitive load (Gulliksen et al., 1997). It is important to minimise the cognitive load caused by the information system in order to enable the user to work efficiently. The result of this analysis supports the user interface designer with relevant and appropriate information.

An important concept in AIU is the work task. A work task is defined as a continuous moment of work performed in order to achieve a goal. Such a task is terminated by a decision, which is defined as made when it is documented. This definition of the work task is because all administrative work includes judgements and decisions. Creating user interfaces that efficiently support the user while performing such work tasks will minimise the amount of unnecessary 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 possibilities to identify processes that are difficult for users to mediate, such as tasks that are performed more or less automatically (Schneider & Shiffrin, 1977). Another reason for observing the users while they work is that professional users often invent their own ways of solving tasks that they might not be fully aware of themselves, and that can be of interest when creating the information systems.

### 2.4. Evaluation of Style and Content (ESC)

ESC is intended to guide the design process (Lif, 1998b). The first part is an inspection of the interface, performed by HCI experts. The second part is a scenario based evaluation performed together with a group of potential end-users. Both parts include a checklist with heuristics that guides the evaluation. A clear distinction has been made between heuristics concerning the content (i.e. the substance) and heuristics concerning the style (i.e. the look and feel) of the interface. By doing this separation, each group of evaluators can put more effort into the field in which they are experts. Users are experts on the work to be supported by the system and HCI experts on usability issues. When an HCI expert without particular domain knowledge is inspecting an interface, he or she is probably able to find out if, for example, default values are needed. However, the users themselves are usually the ones who can tell *which* information to show as default values. By doing this separation the evaluation process can be made more cost efficient.

ESC is meant to be included early in the iterative design process as a tool to guide the development of the interface towards a design that is usable for the users. It is easier to improve the 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.

# 3. The case study

The different methods for supporting the design of the user interface were applied in a case study at the Medical Products Agency (MPA) in Sweden. This national authority is responsible for the whole process of drug approval and control which includes judgement of applications for clinical trials, approval of new drugs and recording data on secondary effects caused by the drugs. An information system was developed to support the employees at MPA during their work. Only a limited part of the system were studied, namely the part supporting judgement of applications for clinical trials. The study focuses on the development of the user interface.

## 3.1. Description of the users work

Pharmaceutical companies or clinics needs to submit an application describing methods to be used, procedures, design of the experiments in detail, etc., before performing a clinical trial. The application is delivered in three copies, where each copy is stored in a physical binder. Important data from the clinical trial applications are recorded in the computer system by a secretary. A clinician then decides which specialists that should judge each application. Specialists may have competence in physics, kinetics, biochemistry, pharmacology, statistics, etc. Each application is judged by several such experts to assure that the clinical trials fulfil MPA:s requirements concerning, for example, ethics and safety. Aspects that are considered during this judgement process are, e.g., number of patients, age, secondary effects etc. Based on the specialist statements, the clinician makes the final judgement that leads to a decision to accept or reject the application or to request additional information about the trial. The processing of an application should be limited to 6 weeks upon day of arrival.

## 3.2. Description of our role in the project

The software company, responsible for the development of the new system, engaged us to assist during the user interface design. The objectives of this case study was to perform an evaluation of the utility of our methods for analysis and design and to share our knowledge in usability with the staff members at the software company. In the case study we participated in parts of the project, not only as independent observers but also as staff members, performing action research (e.g., Rapoport, 1970). This enabled us to get a close insight in the user interface design process in the project.

The case study was divided into two phases. The first phase included analysis of user requirements and design of a first version of the prototype as a paper mock-up. A combination of AIU and UIM was used to gather and specify the user interface requirements (Figure 1). The first phase included the following steps:

1. Study results from the systems analysis

2. Gather user requirements using AIU

3. Specify user requirements using UIM

4. Design a low-fidelity prototype as a paper mock-up

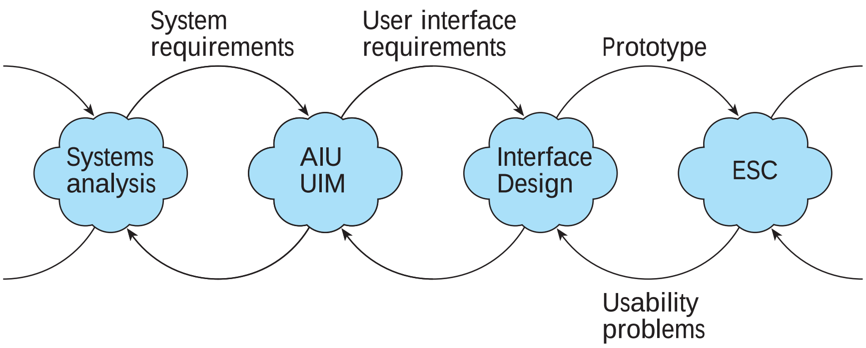


*Figure 1. A combination of AIU and UIM were used for gathering and specifying user requirements relevant for the design of the user interface. In each iteration the style and the content of the prototype were evaluated using ESC, and the interface was redesigned.*

In the second phase, the iterative process of design was studied. During each iteration the style and the content of the interface were evaluated separately in order to guide the design of the prototype (Figure 1). The second phase included the following steps:

1. Redesign the prototype

2. Evaluate the style and the content of the interface separately

3. Redesign the prototype based on the results of the evaluation

4. Evaluate the style and the content of the interface separately

# 4. Analysis and design of a first paper mock-up

The purpose with this project was to develop a computer system for supporting judgements of applications for clinical trials. Today, a main frame system with an alpha-numeric user interface is used. This was to be replaced by a system with a graphical user interface for standard PCs with a 17-inch screen (resolution 1024*768).

## 4.1. Study results from the systems analysis

When we were engaged in the project, the users' work had already been modelled by an external consultant during approximately two years. This work was performed together with the end-users, resulting in an Entity-Relationship model with a description of the attributes, a Data flow diagram, and a description of the functionality of the intended system in free text.

As a complement to these models, the software company specified the functionality of the system with use cases (Jacobson et al., 1992). The actors in the organisation and their relations to the use cases were also specified (Figure 2). These models were also specified in co-operation with the users.

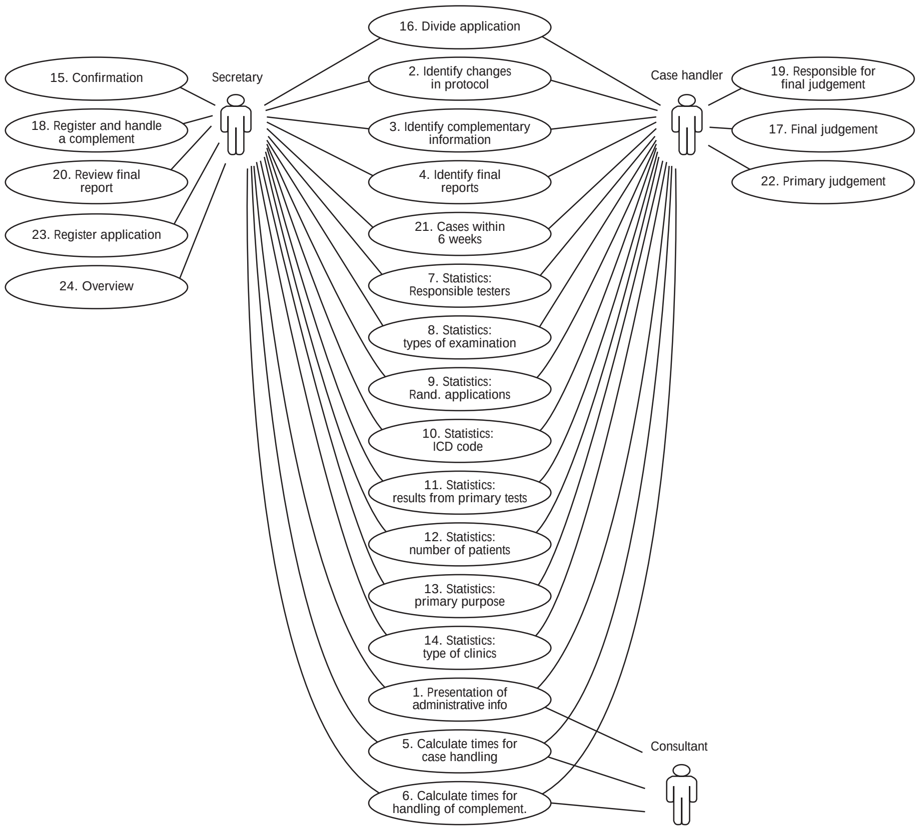


*Figure 2. A use case diagram describing the relation between the actors and the use cases.*

We studied the different models for approximately one week to get a basic understanding of the work to be supported by the computer system. However, it was difficult to get a concrete view of the users' work why we made some complementary "observation-interviews" as described in AIU.

## 4.2. Gather user requirements using AIU

All together we performed six observation-interviews (three each) with subjects representing the actors Secretary and Case handler. During the interviews we observed the subjects as they performed their ordinary work, asked questions, and discussed the process. The interviews were recorded on tape, written down and analysed. The result was used as a complement to existing models to gain deeper understanding, necessary for specifying the relevant requirements on the user interface. During the AIU we also collected documents and identified "tools" that were used by the users in their daily work. Examples of such documents and tools were: private records, used for storing information about older applications; a bookshelf where the physical binders were placed and where each shelf had a certain status; archives for searching and sorting information about the applications.

### 4.3. Specify user requirements using UIM

UIM is intended to be performed in modelling sessions with the potential end-users. This was not possible because of the users' limited amount of time. Instead we specified the requirements based on existing models, on the results from the observation interviews, and on more informal discussions with the users. As defined in UIM, three models were created: A goal model, an actor's model and a work situation model.

#### *The goal model:*

The goal model is a specification of high level goals to be achieved by the users. Each goal is given a priority. The following goals were specified:

1. Fast registration of the applications.

2. All information needed when judging an application has to be simultaneously visible on the screen.

3. The system has to be easy to use and learn in order to be used at all. (The current system is not used very much because of its complexity.)

4. Each application has to be handled within the time limit (4 or 6 weeks).

#### *The actor s model:*

The actor's model describes characteristics of each category of users that are important when making the design decisions. Following, a description of the characteristics of some of the actors:

*Secretary:*

- Usually work with registration and administration of applications.

- Experienced users of the current alpha numeric system and some experience of using graphical user interfaces.

- Will use the system several hours every day, primary for registration of applications.

*Case handler:*

- There are different kinds of case handlers. The two main groups are Specialists and Clinicians. Each application is judged by several Specialists with different competencies (e.g., biochemistry, pharmacology, and others). A clinician decides which specialists that should judge each application. A clinician also makes the final judgement by comparing the statements made by the different specialists.

- Today, they hardly use the present alphanumeric system. Their work mainly consists of reading and judging the applications which are documents of approximately 500 pages stored in physical binders. The case handlers are used to word processing.

- The case handlers will use the system every day (if it fulfils their requirements), but to a less extent than the secretaries.

### *The Work situation model*

The work situation model describes the contents of the user's work relevant for the design process. A work situation can be described as a core work task in which an actor may communicate with several use cases. Usually an actor works with a few work situations. The following questions were considered while identifying the work situations.

- Which are the major, high level, tasks for each actor?

- Which use cases are logically related?

- Which use cases are related in time?

- Which use cases requires access to the same information?

Three work situations were specified: *Registration*, *Case handling* and *Statistics*. Each work situation is represented with a rectangle with rounded corners in the model (Figure 3). The arrows indicate which work situations each actor may handle. The solid paths indicate which use cases the actors may communicate with in each work situation.

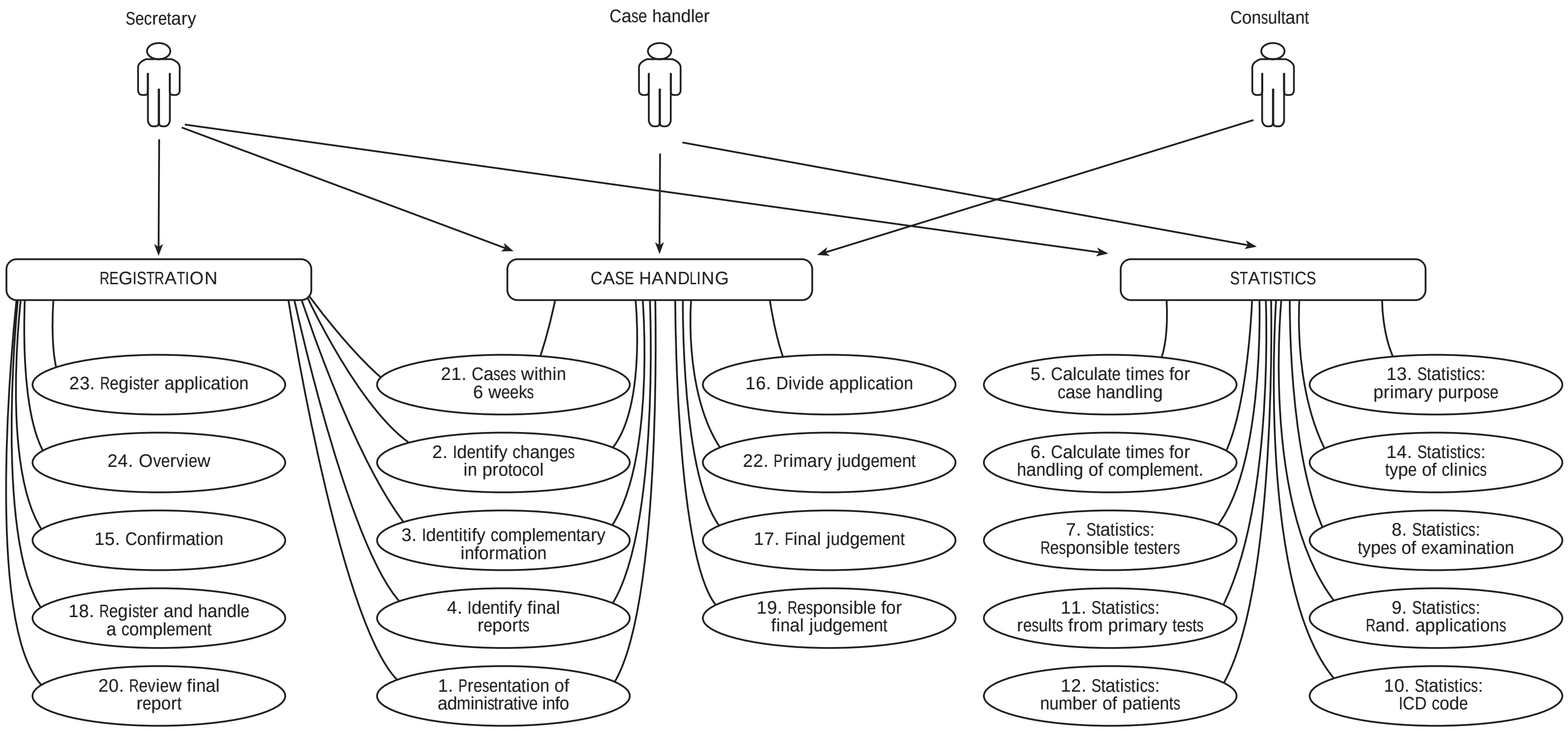


*Figure 3. Each actor communicates with several use cases in a work situation.*

### *Information objects and actions*

A model was drawn for each work situation specifying which information objects and actions that are needed when communicating with a use case. The data model and the description of the use cases were the primary input to this process. In the example, only one work situation is illustrated, namely Case handling (Figure 4).

Each information object is represented with a rectangle. Two rectangles on top of each other indicates that more than one information object of that kind is needed, e.g. several applications. Actions are documented for each use case in free text.

This model supplies the designer with an overview of each workspace. From here it is possible to identify which information objects and actions that are common for several use cases.

### *Number of information objects, attributes and operations*

Characteristics of the attributes and operations were specified in a diagram. An example of such a diagram, with only a small set of attributes and operations, is shown in Table 1. In the diagram the attributes and operations have different priorities: X for extra important; 1 for important; 2 for less important. This information is useful when deciding which information to show simultaneously on the screen.

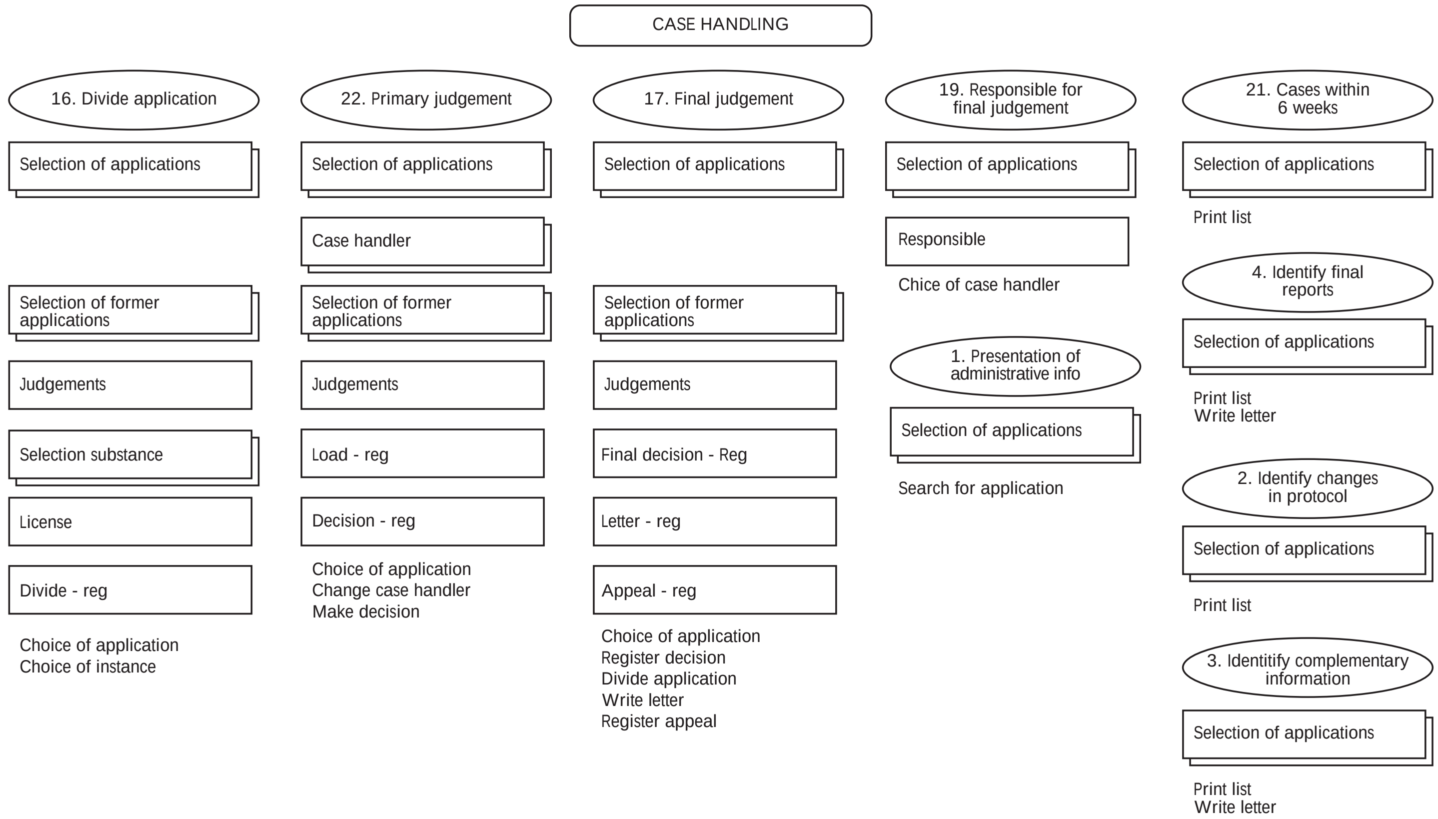


*Figure 4. Information objects and actions for each use case in the work situation Case Handling.*

| Inform. Object | No | Attribute | Priority | Other | Operation | Priority | Other |
|---|---|---|---|---|---|---|---|
| Application | | | | | | | |
| | | Registration number | X | | print | 1 | |
| | | Phase | 1 | | copy | 1 | |
| | | Number of patients | 1 | | | | |
| | | Type of study | 1 | | | | |
| | | Experimental plan | 2 | | | | |
| | | Primary purpose | 1 | | | | |
| | | Dose | 1 | | | | |
| | | ICD code | 2 | | | | |
| | | Obs. application | 2 | | | | |
| | | Pharmacy | 1 | Y/N | | | |
| | | Preclinic | 1 | Y/N | | | |
| | | Clinic | 1 | Y/N | | | |
| | | Doc_Pharmacy | 2 | Y/N | | | |
| | | Doc_Preclinic | 2 | Y/N | | | |
| | | Doc_Clinic | 2 | Y/N | | | |

Table 1. Characteristics of attributes and operations for the information object Application.

### 4.4. Design of a low-fidelity prototype as a paper mock-up

The intention here was to design a first version of the prototype using pen and paper to illustrate how the system could look and behave for a user performing *Registration* or *Case handling*.

The design of this first, paper mock-up was based on the models specified in UIM. According to the goal model the system should enable fast registration of applications. The system also had to be easy to learn and use. To enable each actor to easily overview their work, we decided to let each work situation correspond to a screen-sized workspace in the user interface, one for registration and one for case handling. A workspace must include every tool and all information needed to support one work situation (Lif, Olsson, Gulliksen, & Sandblad, 1998). When a user performs work he communicates with a number of use cases in such a workspace.

The work situation model provided a bird's-eye view of the whole system which was valuable when the overall structure of the user interface was outlined. From this model it was also possible to identify which information objects that were common for several use cases. An example of such an information object was "Selection of applications".

During our observations we discovered that the physical binders were placed on bookshelves that had different status. This bookshelf was important for the users and was therefore introduced in the prototype as an interface element where each shelf had different status and where the applications (binders) were represented as lines on the shelves (Figure 5). Another important goal, specified in the goal model, was to handle each case within the time limit (four or six weeks). Therefore a time line was presented on each shelf showing the time that had past from the arrival of an application.


*Figure 5. This "bookshelf" is used for selecting applications with a certain status and for showing how long time each application has been processed.*

Another essential goal was to have all information needed when judging an application simultaneously present on the screen. According to the work situation model a lot of information have to be visible at the same time. Parallel presentation of information is more efficient than sequential presentation (Lind, 1991). As a consequence, the screen displays have a rather high density of information. To be able to show as much as possible of the information, and still make it efficient for the users', different means for coding the information were used, such as colour, form, font and spatial location.

### 4.5. Evaluation of the utility of the methods in use

This first phase included analysis of the user requirements using a combination of AIU and UIM (AIU/UIM) and design of a first version of the prototype as a paper mock-up. In parallel to our work with analysis and design, the staff at the software company designed a paper mock-up based on existing models, i.e. without support from the models specified in AIU/UIM.

The utilisation of AIU/UIM was evaluated by comparing the two prototypes that were developed in parallel.

*Our hypothesis was, that using a combination of AIU and UIM to complement "traditional" models, will result in a first version of the prototype that is more usable than a prototype based on results from "traditional" models only (not AIU/UIM).*

The two paper mock-ups were evaluated by three usability experts with heuristic evaluation (Nielsen & Molich, 1990). Each of the three evaluators got a short verbal introduction to the users work and were then instructed to study the two paper mock-ups and identify potential usability problems. The findings were documented in forms where Nielsen's heuristics were defined. These heuristics are:

1. Simple and natural dialogue
2. Speak the users' language
3. Minimise user memory load
4. Consistency
5. Feedback
6. Clearly marked exits
7. Shortcuts
8. Good error messages
9. Prevent errors
10. Help and documentation

### *Results*

The potential usability problems identified by the evaluators were later rated according to severity as; major problems, minor problems and problems of no relevance. Table 2 presents the number of identified problems for the design based on AIU/UIM and on traditional models only (not AIU/UIM).

| | AIU/UIM | Not AIU/UIM |
|---|---|---|
| Major problems | 3 | 11 |
| Minor problems | 16 | 9 |

*Table 2. Number of potential usability problems identified with heuristic evaluation.*

Following are examples of potential usability problems in the prototype based on traditional models (not AIU/UIM):

- The dialogue is not intuitive
- Difficult to overview the work to be performed
- Difficult to navigate in the system
- The design forces the user to switch between several screens (windows)

### *Discussion*

Table 1 shows a rather big difference between the two prototypes in number of identified potential usability problems. 11 major usability problems were identified in the prototype where AIU/UIM had not been used as compared to only 3 when AIU/UIM was used. One reason for this could be that there were different people designing the mock-ups, with different background, experiences, etc. Another probable reason for this is that the paper mock-ups differed in completeness. The prototype where AIU/UIM was used included more details which is also a probable explanation to why more minor problems were discovered in that prototype. However, minor problems, such as a missing frame, were not judged to be severe. To understand how the actual methods have effected the different design solutions, it is necessary to further analyse the usability problems that were identified during the evaluations.

In the prototype where AIU/UIM was not used, the major usability problems concerned difficulties in navigation and overviewing the tasks. In that prototype, the users communicated with each use case through a unique window. Since a user often has to communicate with many use cases when performing work, a lot of windows will occupy the screen. This will in most cases cause problems with navigation and overview. Designing a user interface that supports each use case instead of supporting the users' actual work situations will most likely result in a fragmentary interface that is not efficient for the users in their work. Traditional methods and use cases are good support for developing several parts of the system, however, they do not give enough support for the design of the user interface. Therefore it is necessary to complement traditional methods for systems analysis with methods for gathering requirements on the user interface, e.g. through AIU and UIM.

Another important result from this parallel development was that the software company decided to use our prototype as the basis for further development.

# 5. Redesign and evaluation

In this phase the prototypes were redesigned and evaluated in an iterative process. The style of the interface was evaluated separately from the content.

## 5.1. Redesign the prototype

Our paper mock-up was used as an input for the design of a new version of the prototype. To ensure that all information included in the paper mock-up would fit the limited display area, we created a screen version by using a standard drawing program, Freehand™, (Figure 6). It was possible to create this prototype without having to make any restrictions because of the limited screen area.

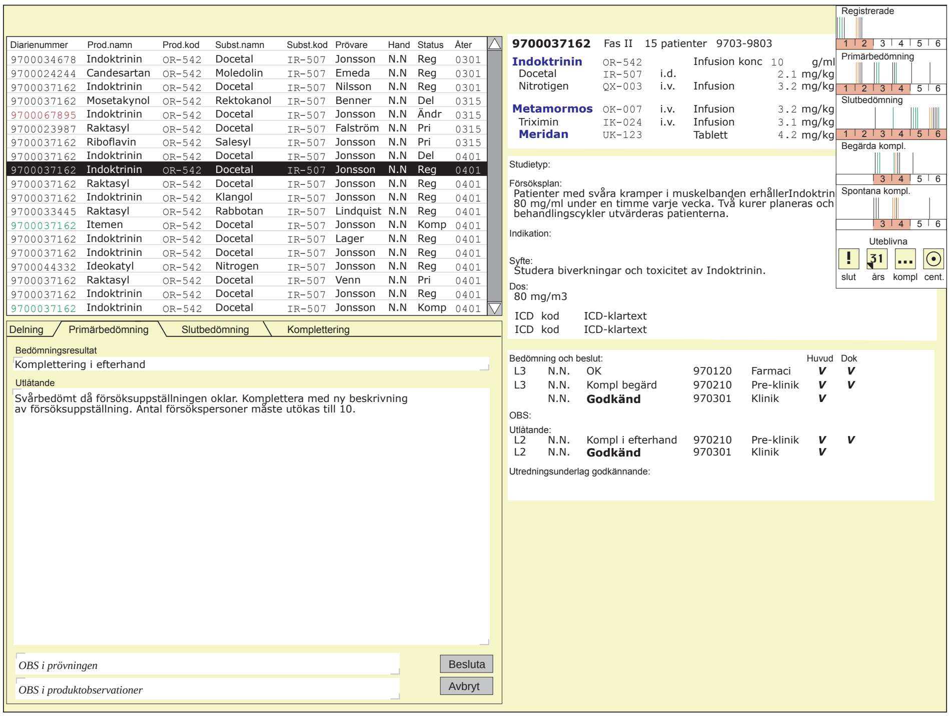


*Figure 6. The paper mock-up was first realised on the screen using a standard drawing program. (Colour plate 4).*

The result was presented to the software company. They decided implement the prototype, preserving the style, in Visual Basic™. During this process they made some changes, due to constraints in the development tool, and the limited amount of time.

## 5.2. Evaluate the style and the content of the interface separately

This prototype was then evaluated using ESC. The style of the interface was evaluated by three usability experts as defined in the method. They were introduced to the system with a short verbal description of the users' work and were then instructed to study printouts of the prototype and identify potential usability problems. They were given a time limit of one hour for the evaluation.

Each evaluator documented potential usability problems guided by the following ESC checklist:

1. Transparency
2. Orientation and navigation

3. Disposition of the screen
4. Readability
5. Discrete layout
6. Control and feedback
7. Minimise the users' system related input
8. Errors and on-line help
9. Consistency

The content of the interface was evaluated together with the potential end-users. This evaluation started with a short introduction to the new system. This was followed by a discussion based on different scenarios. During this discussion, the users identified potential problems with the system. These were documented according to the following ESC checklist.

1. Work related layout and functionality
   - Is the screen layout related to the users' work?
   - Is it obvious how the tasks can be performed?
2. Simultaneous presentation of information
   - Is all information needed when performing a task visible?
   - If not, what information is not visible?
3. Emphasise important information
   - Is the important information emphasised?
   - Is the emphasised information important?
4. Use shortcuts and default values
   - Are shortcuts used for frequent operations?
   - Is the proper information given as default values?
5. Speak the users' language
   - Is the language obvious to the user?
   - What terms are difficult to understand?

The result of the evaluation was further analysed in order to identify problems to be considered during the redesign of the prototype. Examples of such problems were:

- It should be possible to search for a clinical trial application in each workspace.
- The form for recording data from the application needs to be improved to correspond to the users' requirements.
- The workspace supporting case handling should include all needed information. The limited screen area is not optimised.
- It should be possible to interrupt ongoing work to serve incoming phone calls.
- It should be possible to handle a case that has been distributed to a colleague.

### 5.3. Redesign the prototype based on the results of the evaluation

The results of the evaluation were taken into account by the staff at the software company. They redesigned the prototype and developed the parts that had not yet been designed. This process continued for two weeks. At the end of this period the prototype of the complete system for judging clinical trials was evaluated. In figure 7 a bit map from the workspace supporting Case handling is shown.

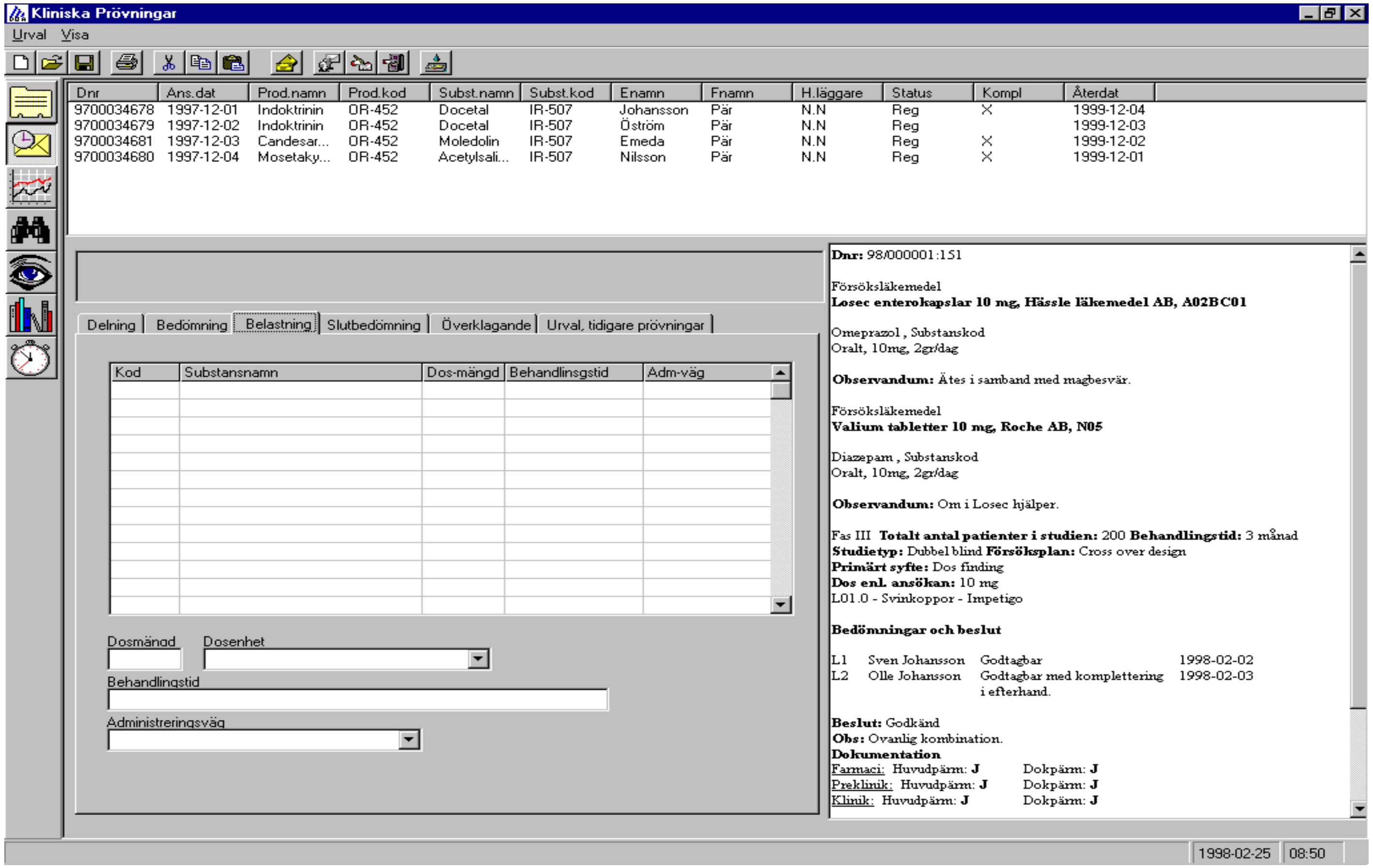


*Figure 7. The workspace supporting case handling in the second prototype.*

### 5.4. Evaluate the style and the content of the interface separately

The evaluation of the content and style was performed in the same way as before. Three usability experts evaluated the style of the interface and the content was evaluated together with the users. Following are examples of problems that were found during this process.

- Sometimes it would be valuable to see the history of an application for clinical trial.
- It should be possible to register data for several companies or clinics at the same time, without opening and closing a dialogue.
- Facilities for identifying former registration number is needed.

### 5.5. Evaluation of ESC

In this second study, the utility of ESC in design was evaluated by comparing it with Heuristic evaluation.

*Our hypothesis was that when using ESC a larger number of potential usability problems would be identified compared to using heuristic evaluation.*

In each of the two iterations three usability experts evaluated the style of the interface and the content of the interface was evaluated together with a group of users. In parallel three other usability experts evaluated the user interface with Heuristic evaluation. Heuristic evaluation was chosen because it is a commonly used method and because it is similar to ESC in the sense that it can be performed at a low cost. These evaluators got a short verbal introduction to the users work and were then instructed to study the interface and identify potential usability problems and specify these according to Nielsen's method.

The three usability experts that evaluated the style of the interface with ESC in the first evaluation performed a heuristic evaluation in the second iteration. The three usability experts that performed a heuristic evaluation in the first iteration evaluated the style of the interface using ESC in the second iteration.

#### *Results*

The total number of potential problems identified during the two iterations when using ESC was compared to the total number of potential problems identified when using Heuristic evaluation. The problems were categorised as Major or Minor problems and problems of no relevance. Table 3 shows the number of identified major and minor problems.

| | ESC | Heuristic evaluation |
|---|---|---|
| Major problems | 28 | 13 |
| Minor problems | 64 | 39 |

*Table 3. Number of potential usability problems identified when using ESC and Heuristic evaluation.*

### *Discussion*

More major and minor problems were identified when ESC was used than when Heuristic evaluation was used. One reason for this is probably that usability experts are not capable of identifying problems concerning the domain. To do this it is necessary to involve users. When using ESC the users do not have to learn a method, instead they identify problems while discussing a scenario and the identified problems are documented by an HCI expert during these sessions.

The utility of the method can be judged by comparing the problems identified in the first iteration with the prototype developed in the second iteration. The following problems were emphasised in the first iteration.

- It should be possible to search for a clinical trial application in each workspace. *These facilities were included in the second version.*

- The form for recording data from the application needs to be improved to correspond to the users' requirements. *The form had been improved but there were still some corrections to be made.*

- The workspace supporting case handling should include all needed information. The limited screen area is not optimised. *This workspace was definitely improved, however, some further improvements were needed.*

- It should be possible to interrupt ongoing work to serve incoming phone calls. *This was possible in the second version.*

- It should be possible to handle a case that has been distributed to a colleague. *This was possible in the second version.*

The problems identified in the first iteration had been corrected, or at least been improved, in the second iteration. However, since the prototype was more complete in the second iteration, some new problems were identified.

ESC has shown to be a useful tool in the design process to guide the design of the interface. If potential usability problems can be discovered early they are easier to eliminate in the final application.

## 6. Conclusions

In this paper a new approach to design has been applied. A combination of AIU and UIM was used as a complement to use cases for gathering requirements on the user interface. The design of the interface has been performed in an iterative fashion where the prototype was evaluated with ESC in each iteration. It is important to understand that specification of user requirements is an ongoing process. The models should be changed according to new insights and knowledge. In this particular case study the number of workspaces increased because new knowledge was gained about the users work. It is also easier for the users to understand how the system will look and behave as the prototypes become more and more complete, which also will give rise to changes in both the models and the prototype.

When comparing the early version of the prototype (Figure 6) with the one that was created in Visual Basic™ (Figure 7) it is evident that there are some differences. Some of these changes are a result of potential problems discovered with ESC, e.g. to much space was used in the early version for showing the list of applications. Other changes have been made because of the limited amount of time and money, e.g. only a simple version of the "bookshelf" has been realised here but a more advanced version will be implemented in the next version of the system. Deadlines and budgets are almost always major constraints in development projects.

The methods have shown to be a good support for user interface design. However, to utilise the strength of the methods fully, it is necessary to include these methods in a user centred process and involve a user interface designer with knowledge in usability. Usually, the user interface is designed by software engineers who seldom have the time, interest, or competence needed to design a usable interface. Software engineers and designers have different priorities. Software engineers are usually more interested in fulfilling the functional requirements on the system. The designer, on the other hand, is more concerned about the usability of the system. Both these priorities are of course important why it would be valuable to include a usability expert that could focus on the user interface design only and thereby have better possibilities for gaining experience and skill. We will in the future further evaluate the role of the designer in systems development. Interesting issues are, how, when, and where a user interface designer should be involved to enhance the system development process and also how design decisions should be documented and how to follow up the development of the system to ensure that the design decisions are properly considered during construction of the application.

## Acknowledgements

Thanks to the Medical Products Agency (Läkemedelsverket), Pharmasoft and Redina for good cooperation during this case study.

## References

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

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

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.

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

GOULD, J.D., & LEWIS, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM, 28* (3), 300-311.

GULLIKSEN, J., LIF, M., LIND, M., NYGREN, E., & SANDBLAD, B. (1997). Analysis of Information Utilisation. *International Journal of Human-Computer Interaction, 9* (3), 255-282.

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.

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

LIF, M. (1998a). *User Interface Modelling - adding usability to use cases.* (Rep. No. 84, CMD). Uppsala, Sweden: Uppsala University.

LIF, M. (1998b). *Evaluating Style and Content separately, during the design of the user interface.* (Rep. No. 86, CMD). Uppsala, Sweden: Uppsala University.

LIF, M., OLSSON, E. GULLIKSEN, J., & SANDBLAD, B. (1998). *Work Spaces enhance efficiency - Theories, concepts and a case study.* (Rep. No. 85, CMD). Uppsala, Sweden: Uppsala University.

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

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.

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

SUTCLIFF, A.G., & WANG, I. (1991). Integrating Human Computer Interaction with Jackson System Development, *The computer journal, 34*, 132-142.

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

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