User Interface Design for Analysis of Sensor Systems

Lisa Jonsson
Karin Sallhammar

2003-01-16
User Interface Design for Analysis of Sensor Systems

Examensarbete utfört vid
Linköpings Tekniska Högskola, Campus Norrköping

Lisa Jonsson
Karin Sallhammar

Handledare: Martin Gustavsson
Examinator: Ivan Rankin

Norrköping den 16 januari 2003
User Interface Design for Analysis of Sensor Systems

Författare
Authors
Lisa Jonsson and Karin Sallhammar

Sammanfattning
Abstract

In the future network-based Swedish Defence (NBD), attaining information superiority will be of great importance. This will be achieved by a network of networks where decision-makers, information- and weapon-systems are linked together. As a part of the development of NBD, we have performed a study of user interface design for a future network-based tool package for analysis of sensor systems, referred to as the C2SR-system.

This thesis was performed at Ericsson Microwave Systems AB, Sensor and Information Networks, during the autumn 2002. A pre-study concerning the requirements of usability, trustworthiness and functionality of a user interface for the C2SR-system was performed. Officers representing the future users in the NBD played an important role when gathering these requirements. Another important part of the pre-study was the evaluation of software that contains parts of the functionality necessary for the C2SR-system.

On the basis of the results from the pre-study, we have designed a user interface to the future C2SR-system. To demonstrate the most important conclusions, a prototype was implemented.

Nyckelord
Keywords
User interface, HCI, GUI, usability, trustworthiness, NBF, NBD, radar sensor systems, electronic warfare
Abstract

In the future network-based Swedish Defence (NBD), attaining information superiority will be of great importance. This will be achieved by a network of networks where decision-makers, information- and weapon-systems are linked together. As a part of the development of NBD, we have performed a study of user interface design for a future network-based tool package for analysis of sensor systems, referred to as the C²SR-system.

This thesis was performed at Ericsson Microwave Systems AB, Sensor and Information Networks, during the autumn 2002. A pre-study, concerning the requirements of usability, trustworthiness and functionality of a user interface for the C²SR-system was performed. Officers representing the future users in the NBD played an important role when gathering these requirements. Another important part of the pre-study was the evaluation of software that contain parts of the functionality necessary for the C²SR-system.

On the basis of the result from the pre-study, we have designed a user interface to the future C²SR-system. To demonstrate the most important conclusions, a prototype was implemented.
# Table of Contents

**Nomenclature** .......................................................................................................................... 1

**Chapter 1 Introduction** ........................................................................................................... 3
  1.1. Background.......................................................................................................................... 3
  1.2. Problem Overview.............................................................................................................. 4
  1.3. Objective............................................................................................................................ 4
  1.4. Thesis Outline..................................................................................................................... 4

**Chapter 2 Problem Definition** ................................................................................................ 7
  2.1. Description ........................................................................................................................ 7
  2.2. Delimitation....................................................................................................................... 10

**Chapter 3 Method** .................................................................................................................. 11
  3.1. Theory.............................................................................................................................. 11
  3.2. Approach.......................................................................................................................... 14

**Chapter 4 Usability** ................................................................................................................. 17
  4.1. Background........................................................................................................................ 17
  4.2. Questionnaire................................................................................................................... 18
  4.3. Results.............................................................................................................................. 19

**Chapter 5 Trustworthiness** ..................................................................................................... 21
  5.1. Background........................................................................................................................ 21
  5.2. Questionnaire................................................................................................................... 22
  5.3. Results.............................................................................................................................. 22

**Chapter 6 Functionality** ......................................................................................................... 25
  6.1. Background........................................................................................................................ 25
  6.2. Questionnaire................................................................................................................... 25
  6.3. Results from the Questionnaire....................................................................................... 25
  6.4. Additional Requirements................................................................................................. 26
  6.5. Task Analysis.................................................................................................................... 27

**Chapter 7 Design Requirements** ............................................................................................ 29
  7.1. Effectiveness...................................................................................................................... 29
  7.2. Efficiency.......................................................................................................................... 29
  7.3. Satisfaction....................................................................................................................... 30
  7.4. Learnability....................................................................................................................... 30
  7.5. Flexibility.......................................................................................................................... 30
Chapter 8 Existing Software ................................................................. 31
  8.1. Background ................................................................................. 31
  8.2. Eliran ......................................................................................... 31
  8.3. Wrap ......................................................................................... 33
  8.4. Contribution to the C\textsuperscript{2}SR-system .............................. 35

Chapter 9 Behavioural Design ............................................................. 37
  9.1. Scenarios and Systems ............................................................. 37
  9.2. Window Management ............................................................. 38
  9.3. Working with the User Interface .............................................. 39

Chapter 10 Information Design ......................................................... 41
  10.1. Presenting Information .......................................................... 41
  10.2. Design of Icons ..................................................................... 43

Chapter 11 Visual Design ................................................................. 45
  11.1. The Style Guide ..................................................................... 45
  11.2. Implementation of a Prototype .............................................. 45

Chapter 12 Discussion ................................................................. 47
  12.1. The Pre-study ....................................................................... 47
  12.2. Implementation ..................................................................... 47
  12.3. Future Work ......................................................................... 48

Chapter 13 Conclusions ................................................................. 51

Acknowledgements ............................................................................. 53

References .......................................................................................... 55

Appendices .......................................................................................... 59
  Appendix A - A Questionnaire ....................................................... 59
  Appendix B - Task Models ............................................................. 63
  Appendix C - Usability Checklist .................................................. 64
  Appendix D - Usability Evaluation of Eliran .................................. 66
  Appendix E - Usability Evaluation of Wrap .................................... 68
  Appendix F - The Menu Bar and the Tool Bar ............................... 70
  Appendix G - Working with the User Interface ......................... 72
  Appendix H - Design of Icons ...................................................... 82
  Appendix I - Class Description ..................................................... 83
# Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td>One-dimensional</td>
</tr>
<tr>
<td>2D</td>
<td>Two-dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-dimensional</td>
</tr>
<tr>
<td>$C^2$</td>
<td>Command &amp; Control</td>
</tr>
<tr>
<td>$C^2SR$</td>
<td>Control &amp; Command, Surveillance &amp; Reconnaissance</td>
</tr>
<tr>
<td>Demo05</td>
<td>System Demonstrator 2005</td>
</tr>
<tr>
<td>Demo06</td>
<td>System Demonstrator 2006</td>
</tr>
<tr>
<td>FSR 890</td>
<td>Flying Surveillance Radar 890 (Erieye)</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographical Information System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-Computer Interaction</td>
</tr>
<tr>
<td>HTA</td>
<td>Hierarchical Task Analysis</td>
</tr>
<tr>
<td>FOI</td>
<td>Swedish Defence Research Agency</td>
</tr>
<tr>
<td>ID</td>
<td>Identity</td>
</tr>
<tr>
<td>ISO</td>
<td>The International Organisation for Standardisation</td>
</tr>
<tr>
<td>NBD</td>
<td>Network Based Defence</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>$P_D$</td>
<td>Probability of Detection</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PS-90</td>
<td>Pulse Doppler Radar, model 90, (Swedish Defence)</td>
</tr>
<tr>
<td>S/N</td>
<td>Signal to Noise Ratio</td>
</tr>
</tbody>
</table>
Chapter 1
Introduction

1.1. Background

The Swedish security situation has changed. The threat of a major invasion is nowadays substantially reduced. However, the transformation from an industrial to an information society has created new vulnerabilities. This fact, together with the Swedish intention to participate in international conflict management, puts new demands on the Swedish Defence. As a consequence, the Parliament has decided that the platform-based defence of today must develop towards a Network Based Defence (NBD). The purpose is to create a flexible, high-quality defence, prepared to meet unknown future threats [A1, R1].

One important issue in the NBD is to obtain information superiority, i.e. to detect, estimate, decide and act faster and with a higher degree of precision than that of a possible opponent. Access to correct information at the right time and ability to estimate its quality are the most important prerequisites to obtain information superiority. This will be achieved through a network of networks where decision-makers, information- and weapon systems are linked together. Essential for the development of NBD are four co-ordinated management development activities, LedsystT, -M, -O and -P 1. LedsystT is responsible for development of the technical parts of NBD and the activity will be carried through in several phases [R1, R2].

The development of NBD is still in its first phase. “Strategy Goals 2010” is a template for the evaluation of the Swedish defence during the next ten years [R2]. A highly important milestone in this work is the development of two demonstrator systems which will be presented in 2005 and 2006, respectively. The purpose is to gather up ideas, agree on solutions and verify implementations. These two demonstrator systems are Demo05 – Battle Space Awareness and Demo06 – Command & Control. Demo05 will demonstrate how a consistent understanding of the situation between all users can be created when the individual view of the situation is based on joint information. This will form the basis for Demo06, which will demonstrate the ability to support management and performance of operations in the NBD.

1 Ledsyst is an abbreviation for Command & Control, C. The capital letters T, M, O, P represent technology, methodology, organisation and personnel, respectively.
The result of the demonstrations will form the basis for further development of the Swedish Defence towards the NBD and Strategy Goals 2010 [R1].

1.2. Problem Overview

Ericsson Microwave Systems AB, Sensors and Information Networks, has recently completed studies within the first phase of LedsystT. This phase included learning and testing activities to prepare for the development of Demo05 and Demo06. To create and retain information superiority in the NBD, an important issue is to plan and manage the use of accessible resources. Demo05 - Battle Space Awareness will therefore include a tool package for decision-support, planning and analysis of systems of sensor-, communication- and electronic warfare-systems. This tool package is referred to as the C²SR-system in this thesis. One part of the development of the C²SR-system is the design of the user interface. As the development of LedsystT now enters the next phase, there is a need for a study of user interface design for the C²SR-system [R1, R2]. This is the focus of this thesis.

The key to success when designing a user interface is to design for usability, i.e. to make it appropriate for the user to perform his tasks [L1]. Usability will therefore be an important issue to consider in this thesis. Furthermore, since the C²SR-system will be used as a decision-support, the trustworthiness of the system is another important issue.

1.3. Objective

The main objective of this thesis was to perform a study of the needs of a user interface which provides the user with the information necessary for decision support, planning and analysis of systems of sensor-, communication- and electronic warfare-systems. The requirements of usability, trustworthiness and functionality of this user interface were surveyed in a pre-study. To demonstrate important conclusions, a prototype was designed and implemented.

1.4. Thesis Outline

A problem definition is presented in Chapter 2. Here, more detailed information about the C²SR-system is given and the concepts of usability and trustworthiness of a user interface are defined and discussed. Furthermore, the delimitation of this thesis is stated. Chapter 3 presents the methods used in this thesis and our approach to the task.

In Chapter 4, 5 and 6 theory, investigations and results from a pre-study of usability, trustworthiness and functionality respectively are given. In addition

---

2 C²SR stands for Command & Control, Surveillance & Reconnaissance.
to that, Chapter 6 includes the functionality requirements set up for the C²SR-system. In Chapter 7, design requirements of the user interface, i.e. the usability and trustworthiness issues for the required functionality, are presented. In Chapter 8, the existing software is evaluated and its possible contribution to the C²SR-system is presented.

In Chapter 9, the results of the behavioural design process of the development of the prototype are presented. Chapter 10 and 11 includes the results of the information design process and the visual design process, respectively.

In Chapter 12 a discussion of the thesis work and suggestions of possible future work are presented. Chapter 13 includes conclusions.
Chapter 2
Problem Definition

2.1. Description

2.1.1. The C²SR-system in the NBD

A virtual network, “The Net”, will form the basis of the Network Based Defence. The Net is intended to provide the infrastructure necessary for technical activity systems of various kinds to co-operate in a predetermined and safe way. The Net will consist of dynamic networks of networks, where information from different kind of sources will be combined into a joint set of information. A vision of the NBD includes the users being able to use their applications, and thereby performing their tasks, regardless of terminal and physical location [A1].

A proposal has been made about the logical structure of the C²SR-system [R3]. The system should be based on calculation modules called from a detached user interface. The calculation modules request information via the Net, e.g. from databases containing geographical information and from databases providing basic data on sensor-, communication and electronic warfare systems (Figure 1). In this way the user interface will present an individual view of the situation, based on joint information.
Figure 1. The logical structure of the network.

For example, a sensor-commander in the NBD with operational or tactical tasks will be able to plan and analyse the use of sensor systems by means of the C²SR-system. Through the graphical user interface (GUI) (Figure 1), he or she can download position and status of existing sensors in a certain area from the “Sensor” module. By downloading e.g. map- and terrain data from the “GIS” module and basic data on the sensors from the “Basic Data” module, the performance of the existing sensors can be calculated. In the same way, pre-defined sensor models can be downloaded from the “Models” module to plan and analyse fictitious scenarios.

In the development of NBD, the concept of service is fundamental to describe what a system should perform [R4]. The main purpose of the C²SR-system is to give the user access to the services included in “Reconnaissance Performance” and “Reconnaissance Management”. The former includes the sub-services “sensor status”, “object status”, “sensor performance” and “operation area”, whereas the latter includes “sensor management” and “platform management” [R5]. Since these services were not fully defined at the start of this thesis, the concept of service will not be used in the report. However, when discussing functionality in this thesis, we implicitly refer to the concept of service.

Another important concept in the development of the NBD is the role-based situation awareness. The user will, when logging on the future C²SR-system, get access to the services and information that are authorised to and needed
for his particular role. By the means of these services he will then accomplish his task.

2.1.2. Existing Software

The functionality needed in the C\textsuperscript{2}SR-system already exists to a great extent in existing software. The design of this functionality could be a starting point when developing the user interface for the C\textsuperscript{2}SR-system. The calculation modules, which the system will need, can be derived from this software [R3]. Two of the software packages that are considered as of greatest interest are:

**Eliran**  
Software developed by an Israeli company, TIL Defence Systems Ltd. Eliran can calculate and display performance of sensor-, communication- and electronic warfare systems, the technical parameters of the systems and topography taken into account.

**Wrap**  
Software developed by AerotechTelub. Wrap can analyse radio- and radar-performance and propagation and supports frequency assignment. Wrap is in use within the Swedish Defence of today.

2.1.3. The User Interface

When a user is working with a computer system to solve tasks, the user interface is the link between the user and the underlying technique. To help the future users of the C\textsuperscript{2}SR-system to perform their tasks quickly and effectively, the user interface must be as usable as possible. Usability is a well-known concept nowadays, and is an important issue to consider when developing and implementing software.

A fundamental way of maximizing the usability of the C\textsuperscript{2}SR-system is by using a GUI. Nevertheless, usability through a GUI will result in refined information. Refined information means that the information from the system is interpreted and presented for the user in a way that is easier to grasp than the original information. Since the future user will use the C\textsuperscript{2}SR-system as a decision-support, it is essential that the trustworthiness of the refined information is sufficient to base decisions on. However, the extent to which the information is trustworthy must be correctly understood by the user. This is of particular importance in the NBD where incorrect decisions might have disastrous consequences.

Consequently, the GUI we are to design must be usable, and the information must be trustworthy. Hence, usability and trustworthiness in this context must be defined.

2.1.4. Definitions

The International Organisation for Standardisation (ISO) has defined usability 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”. Furthermore, *effectiveness* is defined as “the accuracy and completeness with which users achieve specified goals”, *efficiency* as “the accuracy and completeness of goals in relation to resources expended” and *satisfaction* as “the comfort and acceptability of the system” [R6].

With a definition of human-computer trust by McAllister as a starting point [R7], trustworthiness in this study is defined as *the extent to which a user is confident in and willing to act on the basis of the information and recommendations of an artificially intelligent decision aid*.

These definitions of usability and trustworthiness were interpreted on the basis of the thesis and underlie the design of the user interface.

### 2.2. Delimitation

The focus of this thesis is limited to the functionality included in reconnaissance performance rather than reconnaissance management. The user interface has been developed for a sensor-commander’s role. Assuming that all sensor-commanders will have access to the same services and information, the concept of different roles is not considered in this thesis. One of the visions of the NBD is that the users shall be able to perform their tasks regardless of terminal. However, this user interface is designed for an ordinary PC (laptop or stationary). Functionality for management of systems of sensor- and electronic warfare-systems has been considered whereas systems of communication-systems have been ignored. Nevertheless, an expansion of the user interface towards management of the complete electromagnetic spectrum is considered.

The proposal for a user interface for the C²SR-system has been implemented as a prototype. The main effort has been put into the design of the user interface. Hence, functionality of the prototype has not been given first priority in this thesis. Because of the limited time of this thesis, no usability evaluation of the prototype was performed.
Chapter 3
Method

3.1. Theory

3.1.1. User-Centred Design

A central aspect of user-centred design is the involvement of users throughout the design process. Their requirements, needs and opinions are of greatest importance to design a user interface they find useful and usable. Developing the system iterative is a way of ensuring that both the users and different kinds of expertise can get involved in the design process as needed. An iterative development means the system is designed in several steps where each step involves a user-based analysis of the problem, design and implementation of a prototype (Figure 2) [L1].

![Figure 2. An iterative system development model.](image)

3.1.2. Interaction Design

Interaction Design is another method for development of usable design. This method can be used for the first iteration of the iterative system development model. According to the principles of Interaction Design, interactive user interfaces should be developed in several phases: conceptual design, behavioural design, information design and visual design (Figure 3).
Conceptual design involves deciding what to produce. In this phase the focus is on the future users, who they are and what goals and needs they have. One should also determine what technical possibilities and limitations there are. This phase will help in giving priority to the right functions and removing the ones not needed. Behavioural design involves how the interface will function when used. How should the information be structured and what way through the structure should the user go to perform the tasks? Which tools and functionality should be available in which situations? Information design is how the functionality and information should be presented to facilitate understanding. How could the program help the user understand what happens and get an overview of the quality of the results? Visual design is about visualizing the functionality and giving the interface an appropriate graphical look [W1, W2, W3].

3.1.3. Questionnaires

Using questionnaires is a way to gather information about the users. There are several different techniques and methods for questionnaires. They can be either interviewer-administered where the interviewer asks the questions and fills in the answers in a form or self-administered where the subject reads and fills in the form without any assistance. In self-administered questionnaires the questions have to be very clear, since there is no opportunity to explain to the subject what is meant. The questions can be open or closed. In a questionnaire with open questions the respondent is free to formulate his answers himself whereas in one with closed questions the respondent is asked to select an answer from predefined choices. Open questions are good for gathering information on a broad basis because they allow the subject to answer in any way they choose, whereas closed questions limit the response to a given range of possible answers [L2].
3.1.4. Task Models

The definition of usability provided by ISO states that we should look at “specified goals” [R6]. To be able to design a usable user interface we therefore need to consider the goals that the users want to achieve. According to Norman [L3], a system satisfies the user’s goals by providing suitable tasks that can be performed in order to accomplish the goals. A task is defined as a human activity that will achieve a goal. Each task can be seen as a hierarchy of tasks and subtasks and can therefore be broken down to lower levels [L2].

To assist the users in achieving their goals, the user interface must provide the necessary tasks. The process of task analysis produces a clear understanding of what a system must do. The result of the task analysis can be summarized diagrammatically, as a task model [L2]. An example of a task model is shown in Figure 4.

![Figure 4: An example of a task model.](image)

In Figure 4 the subtasks to the main task “Wax ski” are decomposed to a desired level. For example, the task “Apply new wax” is broken down into the subtasks “Heat iron”, “Melt wax” and ”Iron ski”. A single horizontal line after a box means there are no further decompositions. An arrow indicates that there is further decomposition. For example, “Remove old wax” can be further decomposed into subtasks. These further subtasks are usually shown in another task model. The plan expresses the expected performance of the tasks and allows for alternatives. In the task model, one can see the goal and the way in which the task should be performed.

3.1.5. Style Guide

One way to ensure consistency across different parts of a user interface, or a family of systems, is to base the visual design on a set of principles and rules, a style guide. Companies usually develop a style guide to obtain consistency over all of their products and thereby simplify for the users and to achieve
product loyalty. Style guides can be as detailed as the developer considers necessary.

3.2. Approach

This thesis comprises one iteration in the iterative development process, shown in Figure 2. Firstly, an analysis of the existing software and of the users’ goals and needs is performed. Secondly, the design is considered and finally the prototype is implemented.

Interaction Design is the comprehensive method used throughout this thesis. To better fit this context the phases of Interaction Design are divided in a slightly different way (Figure 5). Conceptual Design, the first phase, still includes the users’ needs and goals and the functionality that should be included. According to the focus of this thesis, this phase is divided into three subgroups: usability, trustworthiness and functionality. The next phase, Interface Design, includes behavioural design, information design and visual design of the prototype.

![Interaction Design in this thesis.](image)

3.2.1. Conceptual Design

For the user interface to fulfil the requirements of usability, we contacted officers working with management of sensor, communication and electronic warfare systems in the platform-based defence of today. Self-administered questionnaires were sent out to five different persons. According to Nielsen [W4], even though five respondents cannot give statistical validity, it is sufficient for a reliable result when conducting usability studies. He states that additional respondents will not give further information; they will only
confirm the reached conclusions. The questionnaires included both open and closed questions about trustworthiness, the desired functionality and the future users work situation and pre-knowledge.

We carried out a careful examination of the functionality necessary in the future tool-package. The answers from the questionnaires and a study of existing software [R3], formed the basis for this work. An evaluation of the usability of the necessary functionality in the existing software was performed through a checklist consisting of general usability principles.

3.2.2. Interface Design

In this phase Behavioural Design, i.e. how the interface will behave, how the information will be structured and which tools that should be available in which situations, was first considered. The structure of the interface was designed and Task Models were constructed for the main functionality. Information Design, i.e. how the information should be presented to facilitate understanding, was then considered. The presentation of the information was designed by making outlines of the user interface before implementing.

The prototype was implemented in Java. Ericsson’s style guide [W5] was used for the visual design. Throughout the whole Interface Design phase the requirements of usability were taken into careful consideration.
Chapter 4
Usability

4.1. Background

As mentioned previously the ISO defines usability 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”. Furthermore, effectiveness is defined as “the accuracy and completeness with which users achieve specified goals”, efficiency as “the accuracy and completeness of goals in relation to resources expended” and satisfaction as “the comfort and acceptability of the system” [R6]. This definition of usability and the following definitions of the attributes effectiveness, efficiency and satisfaction, are very broad statements. To be able to use these definitions in practice, we summarized possible ways of looking at and measuring the attributes of the usability of a user interface.

4.1.1. Usability Attributes

According to the ISO definition, the effectiveness of a user interface is the accuracy and completeness with which users can perform their tasks with the user interface. Thus, effectiveness has nothing to do with time, it is rather about whether or not the task can be accomplished. To be able to measure the level of effectiveness of the user interface, one can measure how effective the accomplished task is by an examination of output quality. Since tasks performed with a user interface are often very complex, it might be difficult to judge the achieved level of effectiveness. Breaking tasks down to subtasks and then examining the subtasks individually is therefore a good approach [L2].

The ISO definition of efficiency implies the effort required from the user to accomplish the task. This can be interpreted either as the time spent in accomplishing the task or as the physical amount of effort made by the user. The time aspect could refer to either the time spent in accomplishing subtasks, the time spent looking for external information or the time spent dealing with errors. The physical aspect could be measured as the number of actions required to perform a task with the user interface [L2, L1].

According to the ISO definition, satisfaction is the comfort of the user interface experienced by the users. This attribute is probably the most difficult
to put in relation to a practical case. Measurements of the experienced satisfaction tend to be subjective but there are methods to make the measurements reasonably objective. For example, measuring the proportion of the users who experience the user interface as satisfying is one such method [L2].

As early as 1980, Schackel provided a definition of usability which included the attributes effectiveness, learnability, flexibility and attitude [L4]. Today, the concept of usability does not have exactly the same meaning as in those days. Nevertheless, two of Schackel’s attributes of usability, learnability and flexibility, are regarded as important considerations in this study. Learnability in this context means that the user interface should be easy for the user to learn so that it is possible to start using it as quickly as possible. The learnability of the user interface will most likely colour the user’s attitude towards it. As the user keeps on learning how to handle the user interface, the learning rate will be reduced. It is unlikely that all functionality of the user interface is necessary for all users; some users might even never learn to master the user interface completely [L2]. Learnability of the user interface can therefore be measured as the effort spent when learning to use the user interface for solving a pre-specified task.

By flexibility, one can intend either that the user interface can adapt to its environment and the task the user is about to perform or that different users might want to use the user interface in different ways [L2]. For example, a novice user might only want to have access to the functionality necessary for performing his task whereas an experienced user might want to have full control over all functionality. Flexibility of the user interface can therefore be interpreted as to what extent the user interface can adapt to the users’ experience and needs.

The ISO definition of usability includes “specified users”, “specified goals” and “specified context of use”. Therefore, it is important to define what these three expressions correspond to in this thesis. The specified users are the future users in the NBD. Because of the long-term development of NBD towards Strategy Goals 2010, the future users were impossible to contact. However, there are people in the Swedish Defence of today, who can be considered as equivalent to the future users. These are officers who work with sensor-, electronic warfare- and communication systems management. When referring to “the users” in this thesis, these officers are intended. To find out more about the future users, their goals and the context in which they will use the tool package, these officers were an important source of knowledge.

4.2. Questionnaire

An investigation of the users’ requirements on usability was performed. A face-to-face meeting with the users was difficult to arrange. Therefore, a
questionnaire, including questions about the users’ requirements on usability of the future user interface was constructed (Appendix A). Questions to find out more about the “specified users”, “specified goals” and “specified context of use” were also included. The questionnaire was sent to five officers in the Swedish Defence. To complement the questionnaire a meeting with one of the officers was arranged. The questionnaire is presented in Appendix A.

4.3. Results

The questions concerning effectiveness deal with the accuracy and completeness with which users can perform their tasks with the user interface and also which the desired tasks are. The answers to these questions are discussed thoroughly in Chapter 6.

The first questions on usability deal with the situation of the future users. The officers believe that the time the future users will have to accomplish the task will vary from minutes or hours to an unlimited time. Therefore, they suggest that it should be possible to change the accuracy of the calculations so that one can get an estimate of the results first and then gradually increase the accuracy of the information. The C²SR-system will be used by different users and in different situations, i.e. by officers at the headquarters or by the person in command in the field.

The other questions deal with the previous knowledge of the future users and the flexibility of the user interface. Some officers believe that the future users will have excellent tactical and operational knowledge. They will also have a high level of educational attainment and they will be used to working with computers. Other officers state that soldiers also must be able to use the system with a minimum of education. Most of the officers think it is important to be able to change the user interface with increasing experience, for example to log in on different levels with an augmenting possibility to affect details. The user interface must also be designed to avoid making mistakes, e.g. by having existing systems predefined in databases. Some of the officers think that the design of the user interface should be influenced by computer games, because this will make it easier for the future users to recognise the functionality.
Chapter 5
Trustworthiness

5.1. Background

The use of software to gather and present information will involve a refinement of the information and thus a possible loss of trustworthiness. If the users do not trust the information from the software they will not base their actions on that information and the software will lose a lot of its power. Therefore it is very important to be aware of the different items that affect trustworthiness when designing a user interface for the future system.

5.1.1. Definitions and Previous Research

Most studies in the field of trustworthiness discuss expressions like reliability instead of trustworthiness. From this study’s point of view, reliability is not essential since it depends on the system’s design, not the design of the GUI. Since we do not have any influence on the system’s design, we cannot assume that the system is perfectly reliable. Hence, we will have to keep in mind that different functions of a system are not necessarily equally competent so the focus is not on giving the users implicit faith in the system. It is inappropriate for a user to trust or distrust functions equally. Instead, the user must learn to calibrate his trust, that is, to set his trust to a level corresponding to each function’s trustworthiness [R8].

The definition of trustworthiness presented in Chapter 2 is a modified version of McAllister’s definition of human-computer trust: “The extent to which a user is confident in, and willing to act on the basis of, the recommendations, actions, and decisions of an artificially intelligent decision aid” [R7]. We have chosen to use this definition as a starting point because by mentioning both the users’ confidence in the system and their willingness to act on the system’s advice, the definition identifies the objectives of considering trustworthiness when designing the user interface. Yet the definition does not perfectly describe trustworthiness in this study because this system will not act on its own, nor will it make its own decisions. Our system will rather serve as a resource to provide the users with information and recommendations to help them make their own decisions.

Bonnie M. Muir differentiates between the kind of decision aid mentioned in McAllisters definition and the one in this study. She calls them prosthetic and
instrumental decision aids respectively [R8]. They differ in their type of output: the prosthetic decision aid produces recommended decisions and solutions whereas the instrumental decision aid produces information to be used during the user’s decision making process. In the former case, the user’s task is to use and supervise the machine, which is presumed to have superior competence to find and choose solutions. The user is really in no position to evaluate such a machine’s capability and will have difficulty calibrating his trust in it. In the latter case, the user is in control of a decision support system, which is designed as an instrument. The user uses the instrumental decision aid as one of many resources he may call upon to provide information that will help him to make decisions and solve problems [R9]. Thus, the fact that our system is instrumental is advantageous from the aspect of trustworthiness since it facilitates the user’s interpretation of the information. Hence, it becomes easier for him to calibrate his trust. We modified McAllister’s definition by removing the parts about the decision aids actions and decisions in order to adapt it to our instrumental decision aid.

*Trustworthiness in this thesis is defined as the extent to which a user is confident in, and willing to act on the basis of, the information and recommendations of an artificially intelligent decision aid.*

Results from another study regarding trustworthiness, performed at the University of Illinois, were considered as important for this study. The study investigated how people’s trust in an electronic data collection and control system changed as they got used to the system and as they experienced faults in the system. The result shows that users’ trust in a system increases when they get used to the system and that faults, even small ones which have almost no effect on the system’s performance, cause the users’ trust to drop [R10].

### 5.2. Questionnaire

A few questions about trustworthiness were also included in the questionnaire (Appendix A). The purpose of these questions was to find out what in the user interface the users think affect their trust in the presented information. The purpose was also to find out whether they have calibrated their trust in the functionality in a specific software package, WRAP, and if they would act on the basis of the information from this software.

### 5.3. Results

The users consider knowledge about the information’s origin as an important factor for experiencing the information as trustworthy. This requirement might be impossible to fulfil in some cases since some information may have security restrictions. There are differing opinions on the importance of understanding the structure of the user interface. Most users consider the accuracy of the calculations as important, even though one of them claims that
the calculations are usually more accurate than the rest of the system i.e. the properties of the equipment. Time-stamps are another item the users believe affect their trust in the system. Since the answers to many of the questions on trustworthiness differ, it is very hard to make any further conclusions from this part of the questionnaire.

The user who works with WRAP on a daily basis says that he has enough experience to react when the results are unreasonable and he usually determines where the problem occurred. To be able to trust the results from WRAP if the consequences of a failure were disastrous, he would make a few different calculations to confirm the result. He would also like to have more experience of verifying the results in field. If the results from the field correspond to those from WRAP, he would trust the results from WRAP in the future. The other users do not work with WRAP on a daily basis but they have been in contact with it. They think it is hard to say whether information from WRAP is trustworthy or not. One user states that the detection range is extremely dependent on the radar target area and height, jamming and possibly the weather. Since the enemy controls at least three of these, the user states that all sensor planning will be very insecure and no decision aids will be very trustworthy. He concludes that usability therefore is a lot more important than trustworthiness.
Chapter 6
Functionality

6.1. Background

To find out more about the sequence of work when planning and analysing the use of radar sensor systems in the Swedish Defence, officers who work with sensor-, electronic warfare- and communication systems management were contacted. The officers could say that, today, the planning takes place without advanced technical aids. When deciding the positions of the sensors, ordinary maps are used. An estimation of suitable positions is made, and the estimated positions are tried out by moving the sensors there. The achieved performance is controlled and if the result is poor, progress by trial and error continues. The officers who have worked in this way consider the sequence of work as very primitive.

6.2. Questionnaire

To gather the users’ opinions and requirements of functionality in the C²SR-system, questions about functionality were included in the questionnaire (Appendix A). A previous study of the needs and requirements for a C²SR-system [R11] played an important role when drawing up a proposal for functionality in the C²SR-system for the questionnaire.

6.3. Results from the Questionnaire

By means of the answers of the questionnaire and discussion with the officers, the users’ requirements of functionality in the C²SR-system were summarized:

**Situation Picture**
Getting the present situation picture is considered important. The situation picture must give fast and clear information on available resources, e.g. the platforms’ positions, status and equipment, and information about the possibilities of a fast transportation, e.g. by displaying road networks and terrain.

**Performance**
Calculating and displaying sensor- and electronic warfare performance is considered very important. This functionality must give the users a fast understanding of the detection range of the sensors and the consequences of real or estimated electronic warfare effects.
Communication Performance
If the radar stations cannot get in contact with the command centre, the information gained from the radar equipment is useless. Hence, calculating the radio range is indispensable when planning positioning of mobile radar sensors. It is important that communication performance can be calculated by means of the same tool that calculates sensor performance.

Optimization
Letting the tool package optimize the use of available resources, sensors or electronic warfare is considered as a desirable functionality. When optimizing the use of sensors, the tool package must be able to calculate and display the positions of available sensors in a pre-defined area (reconnaissance base), which gives optimum coverage in another pre-defined area (target area). To include the possibility to transport a certain platform to a certain location in the calculations, the optimization should be connected to the operation area.

Dynamics
A dynamic presentation of how the performance of mobile platforms changes with time is considered to be of interest by most of the users. A dynamic presentation is important for visualizing a course of events and thereby being able to act before facing the fact. However, the users want a clear presentation, hence it is important to avoid a messy dynamic presentation. The users prefer manual control of the updating frequency and the options of pausing or turning off the updating frequency.

Calculation Methods
Some of the users wish to have the possibility to choose between different methods to measure sensor performance, e.g. Signal to Noise (S/N), Probability of Detection (PD), or different kind of tracking performance parameters. They want answers to tasks such as “calculate the PD for a certain radar target area and display where the PD is 50%, 70% and 90% respectively”. On the other hand, other users prefer a fast and simple answer in the form of “coverage = yes/no”.

2D or 3D Presentation
The users are of different opinions regarding the need of a 3D presentation of the performance. Some users consider a 2D presentation sufficient, whereas other users desire an additional 3D presentation. One user in favour of 3D mentions the realistic terrain feeling that arises in a 3D view. Another user states that it is difficult to present radar range for different target heights in 2D.

6.4. Additional Requirements
Apart from the results of the questionnaire, additional requirements of functionality in the future tool package were given in advance of this study.
**Operation Area**

It must be possible to calculate the area that a platform can cover, considering fuel supply, threats, the firmness of the ground, etc. It should also be possible to calculate the best path to a certain position [R5].

**Calculation Accuracy**

The users must be able to change the accuracy of the calculations. In some cases the users might want to have a rough and fast estimate of a systems performance whereas in other cases they might want a more accurate result [R3].

**Performance on all arenas**

An important feature in the NBD is the ability to gather the effect of systems of all arenas, i.e. the ground-, sea-, air- and information-arena [R2]. Functionality for analysis of the accumulated performance of sensors attached to air-, surface-, submarine- and ground-based platforms with the same software is therefore very important.

**Registration of Data**

In LedsystT, basic data for decision-making should be saved [R2]. Hence, it must be possible to save scenarios built up in the user interface, i.e. sensor and electronic warfare systems, geographical information and calculation results.

### 6.5. Task Analysis

On the basis of the results in Chapter 6.3 – 6.4, three important goals, which the users must be able to achieve, were identified:

1 – **Simulate Scenario**

The users must be able to simulate a scenario in the region of interest. This will be achieved by positioning models of platforms with sensors or electronic warfare equipment as desired. They must also have the possibility to work with platforms from different arenas, i.e. ground-, sea- and air- based platforms.

2 – **Situation Picture**

The users must be able to download the current situation picture from the Net. The situation picture should include the positions and status of existing platforms in a certain region and, if desired, weather reports and other relevant information.

3 – **Calculate Performance**

The users must be able to calculate the performance of the sensor and electronic warfare systems in the simulated scenario as well as in the situation picture. They must have the possibility to choose method and accuracy for the calculation. The performance must be displayed in a perspicuous way for both static and dynamic platforms.

By means of the above conclusions and by analysing functionality in existing software (Chapter 6), it was possible to carry out a task analysis and then
construct task models for the three goals. The task models, enclosed in Appendix B, include the main tasks and the subtasks that the user must perform to be able to achieve these three goals. These task models were used during the behavioural design process (Chapter 9). The subtasks of these goals include most of the functionality considered necessary by the users. Design of the remaining parts will not be pursued in detail in this thesis.
Chapter 7
Design Requirements

The previous three chapters describe the conceptual design process in this thesis. The requirements of functionality in the C²SR-system were summarized in the previous chapter. This chapter contains the design requirements, i.e. the usability and trustworthiness issues for that functionality.

7.1. Effectiveness

The user interface must make it possible for the user to perform the tasks as accurately and completely as possible. There are three main tasks that the users must be able to accomplish with the user interface: “simulate a scenario”, “get the situation picture” and “calculate the performance” of sensor and electronic warfare systems. These tasks have been broken down into subtasks (Appendix B). Each of these subtasks must be designed in a way that is logical and intuitive to the users, so that their goals can be reached effectively.

7.2. Efficiency

The user interface must enable the users to accomplish the tasks in only a few minutes and with as few actions as possible. There are several measures that should be taken to fulfil these requirements. There should be drag-and-drop functionality to position the systems, i.e. the radars and jammers, on the map. There should also be a toolbar containing icons for the most frequent actions.

There must be pre-defined systems so the users do not have to adjust technical parameters. That is, a technician should define an inventory list of existing systems in advance, so that the users only can choose between the ones available. Pre-defined systems will also prevent faults that could occur if the users by mistake defined non-existent systems. The help function must be reachable from everywhere in the user interface, it should include all functionality and it should be legible.

The parameters in the calculations should be set to default values so that calculations can be performed without making any adjustments. Nevertheless, it is important that the users can easily find out what the default values are set to and that they can easily be changed. It must be possible to perform fast
calculations so that the users can get an instant result. The users should also have the option of making more accurate calculations when not limited by time. It would be useful to have a calculation choice that delivers approximate results immediately and gradually delivers more accurate results.

7.3. Satisfaction

The user interface has to be designed to satisfy the users. Therefore, their opinions have to be considered during the design process. The prototype of the C²SR-system should be evaluated by the users with respect to satisfaction.

7.4. Learnability

The user interface must be intuitive and easy to learn. This will be accomplished by following conventions and using standard window design. The terminology should be adapted to the users. The user interface should also be structured in a way that is logical to the users.

7.5. Flexibility

The user interface must be designed to satisfy users with different levels of experience. This can be achieved by setting the parameters to default values that can be used by the novice users and can easily be changed by the more experienced ones. The users must be able to set new default values. The user interface must also be adapted to the users’ different backgrounds, e.g. technical/tactical officers. Therefore, it should be possible to select the preferred level of technical detail.

7.6. Trustworthiness

For a decision aid to be trustworthy, it should provide information during the human decision making process rather than providing complete solutions. To help the users in calibrating their trust time-stamps and information about the origin of the information should be provided where this does not break any security restrictions.
Chapter 8
Existing Software

This chapter and appendices D & E contain usability evaluations of the software packages Eliran and Wrap. Note that the results reflect our personal view and that these evaluations were performed using the program versions that were available for us. This implies that some results may not be applicable for the latest version of the programs.

8.1. Background

A great extent of the functionality necessary for the C²SR-system is already implemented in existing software. According to a previous study, two of the most interesting software packages are Eliran and Wrap [R3].

To benefit from the fact that parts of the necessary functionality already exist in Eliran and Wrap, an evaluation of usability of the relevant functionality in these software packages was performed. Since the focus of this thesis is management of sensor and electronic warfare systems, the evaluation of usability was performed with respect to this functionality only. The usability of remaining functionality has been ignored. During the usability evaluation we checked whether the most important HCI design requirements were met. These requirements are summarized as a checklist in Appendix C. We also tried to perform two fundamental tasks by means of the software, i.e. to position radar sensors and jammers in the terrain and to calculate their accumulated performance.

8.2. Eliran

Eliran is a software package developed by an Israeli company, TIL Defence Systems Ltd. Eliran can calculate and display performance of active and passive sensor-, communication- and electronic warfare systems, the technical parameters of the systems and topography (but not vegetation) were taken into consideration. The result of the calculation is presented as a 2D static picture. Profiles of the terrain between two positions can be displayed. Eliran is not in use within the Swedish Defence of today; however, a few of the officers, who we have been in contact with, have superficial knowledge of this software.
The fundamental object that the user works with in Eliran is called a *project*. A project in Eliran consists of map information and one or more *scenarios* (Figure 6). A scenario includes a number of *systems*, i.e. platforms with their equipment, and the *connections* between the systems. The minor window “Project Explorer” displays information about the scenarios in the project.

![Figure 6. A scenario in Eliran. The scenario includes systems, connections and map information.](image)

### 8.2.1. Results from the Usability Evaluation of Eliran

Eliran was developed by people with excellent tactical experience [R3], which is reflected in the user interface. The sequence of work when building up scenarios with systems, i.e. position radar and jammer systems in the terrain are easy to perform by means of a “System Inventory” which contains predefined systems. There is a “Query Wizard” to guide the user through the calculation process. The user interface of Eliran is generally clear and easy to grasp for novice users. However, in spite of the educational Query Wizard, we found it complicated to perform a calculation during the evaluation. This reduced our, in other respects, very positive impression of Eliran. Another issue that we noticed: one must select an area in the map where the calculation is to be performed, instead of automatically performing the calculation in the detection range of the radars.

To sum up the result of the checklist for important HCI design requirements: even though Eliran has extensive functionality, it is easy to get an overview of the user interface. The use of windows is flexible. Existing window
conventions are followed, which means that the user can quickly adapt to the Eliran environment. The number of icons in the tool bar is limited and the icons specific for the software are well chosen. There are extensive drag-and-drop possibilities. However, handling the connections (the red arrows in Figure 6) as separate objects is not entirely satisfactory, because it makes the map view slightly messy. The complete result of the checklist is given in Appendix D.

8.3. Wrap

Wrap is a software package developed by AerotechTelub. Wrap has extensive functionality for analysis of radio- and radar- performance and propagation and supports frequency assignment. In contrast to Eliran, Wrap can include influences from vegetation in the calculations. It also supports a 3D graphical presentation of the terrain. Wrap was developed in Sweden, which means it is adapted to Swedish conditions. The software was originally designed to manage communication systems and functionality for radar sensor systems was added at a later time. Today, Wrap is used for planning the use of radio link communication systems in the Swedish Defence. Wrap can also be used for planning the use of radar sensor systems and electronic warfare, but according to the officers with whom contacts have been made, this functionality is not in use.

Similar to Eliran, the fundamental object that the user works with in Wrap is called a project. A project in Wrap may consist of stations, possible groups of stations, calculations and results. Radars and jammers are examples of stations in Wrap. An open project in Wrap includes several windows (Figures 7 & 8).
8.3.1. Results from the Usability Evaluation of Wrap

Because of the extensive functionality in Wrap and because the equipment on the platforms is not predefined, it is hard for a novice user to start using Wrap without a thorough introduction. Therefore, we only looked at already completed projects with pre-defined stations and performed calculations. With
guidance from a more experienced user, we managed to perform additional calculations with respect to the stations in the project.

To sum up the result of the checklist: the visual design of the user interface of Wrap is to a large extent satisfactory. Existing conventions are followed, which means that the users recognise the environment. Nevertheless, Wrap includes extensive functionality without any possibility to hide the parts not in use, which makes it hard to get an overview of the available choices. It is not possible to hide any of the sub-windows. As in Eliran, the detection range is displayed as a coloured area on the map. This does not provide an intuitively correct understanding of the situation since one can see the rest of the map clearly whereas the covered area is shaded. The complete result of the checklist is presented in Appendix E.

8.4. Contribution to the C²SR-system

Eliran is developed by people with great tactical knowledge and Wrap is developed by people with great knowledge of the Swedish conditions [R3]. Therefore, these software packages will serve as models when designing the user interface for the C²SR-system. The best parts of each software’s behavioural and information design will be merged and possibly re-designed to fit our user interface. However, since important functionality for the C²SR-system is missing in both Eliran and Wrap, the design of the user interface for the C²SR-system will, in some aspects, differ noticeably from the user interfaces of these existing software packages. Firstly, as things stand today, neither Eliran nor Wrap is adapted to a network-based architecture. It is not possible to download external information, e.g. from detached databases or from real or simulated systems to any of these software packages. In the NBD, the users must be able to download position and status from real or simulated systems from the Net and be able to analyse their performance by means of the user interface. Furthermore, the users will almost never have to (even though the possibility must exist) define entirely new systems, i.e. to set technical parameters of for example equipment and antennas, on their own. This can be done in advance by a technician and the data can be stored in databases. The user will then be able to, via the user interface, load pre-defined systems from databases and immediately calculate their performance [R11]. Admittedly, this proceeding is practically identical in Eliran today since pre-defined systems are stored in the “System Inventory”.

Secondly, neither Eliran nor Wrap can manage dynamic planning and analysis. Since dynamics is considered very important to include in the C²SR-system, this will make additional demands on the design of the user interface for the C²SR-system. Finally, since both Eliran and Wrap are pure planning and analysis tools, they may not be appropriate for a real-time situation. The user interface for the C²SR-system must therefore be designed for other environments than the existing software.
Chapter 9
Behavioural Design

To demonstrate the most important conclusions given in the previous chapter, a prototype was designed and implemented. In this chapter, the result of the behavioural design process of the development of the prototype is presented.

9.1. Scenarios and Systems

In both Eliran and Wrap, a modular structure is used. Eliran uses projects with one or more scenarios including systems and Wrap uses projects including stations. Due to the tactical and operational experience of the developers of these software packages, we have chosen to follow this arrangement as far as possible. Nevertheless, we wanted the structure of our user interface to be as simple and clear as possible. In our prototype, a scenario is the fundamental object that the user works with. A scenario can be opened, closed or saved. When the user downloads a situation picture, i.e. position and status of systems, from the Net, a scenario with these systems is created in the C²SR-system.

A scenario in the C²SR-system includes information about a certain region. This information is GIS data, systems positioned in the area and calculations performed with respect to the systems’ parameters and the GIS data (Figure 9).

| Scenario: |
| - GIS data |
| - Systems |
| - Calculations |

Figure 9. A scenario in the C²SR-system.

GIS data is geographical information in one or more layers. The information can be static or dynamic. Commonly used GIS layers are basic map-information: topography, vegetation, water and road networks, but it can also include e.g. traffic fluxes, population statistics or weather reports and forecasts. A system is a platform with equipment (radar sensors or jammers). Pre-defined systems are listed in a “System Inventory” so that the user can quickly simulate fictitious scenarios with a few mouse clicks. If the user wants
to use other systems than the pre-defined ones, he first has to create and add new system to the System Inventory. A *calculation* is, for example, a calculation of the systems’ performance.

Scenarios, systems and calculations each have a *status*, i.e. detailed information about the object. The status of a scenario includes the name, the co-ordinates of the region, and a possible time-stamp (for situation pictures). The status of a system is divided into *properties* for the platform (e.g. equipment), which cannot be changed, and *settings*, which the user can change by means of the user interface. Examples of settings are position and receiving/transmitting frequency. The status of a calculation includes the name of the calculation and all parameters that were considered in the calculation.

**9.2. Window Management**

The main window in the user interface for the C²SR-system contains a title bar, a menu bar, a tool bar and a desktop (Figure 10). A detailed description of the design of the menu and the tool bar is given in Appendix F.

<table>
<thead>
<tr>
<th>Title bar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu bar</td>
</tr>
<tr>
<td>Tool bar</td>
</tr>
<tr>
<td>Desktop</td>
</tr>
</tbody>
</table>

*Figure 10. Sketch of the positions of the title bar, menu bar, tool bar and desktop in the user interface*

At the desktop, sub-windows, which contain information about the opened scenario, are shown. A scenario includes five sub-windows that display information about the scenario. These five windows are ”Map Window”, ”Scenario Explorer”, ”Index Map”, ”Map Layers” and ”Time Slider” (Figure 11).
<table>
<thead>
<tr>
<th>Map Window</th>
<th>Scenario Explorer</th>
<th>Index Map</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Map Layers</td>
<td>Time Slider</td>
</tr>
</tbody>
</table>

*Figure 11. Sketch of the positions of the sub-windows that contain information about an opened scenario*

The Map Window is the main window of the five sub-windows that contain information about the scenario. The other four sub-windows can optionally be opened and closed from the View menu item; however if the user closes the Map Window, the whole scenario is closed down. The positions of the four minor windows were chosen from their frequency of use. The most frequently used is positioned on the top, next to the main window, etc. In the next chapter, “Information Design”, the information presented in the five sub-windows will be further explained.

### 9.3. Working with the User Interface

To design a user interface that makes it as effective and intuitive as possible for the users to perform their tasks, the task models constructed in Chapter 6 were considered. These task models describe all subtasks included in the three main tasks, i.e. “Simulate a Scenario”, “Get the Situation Picture” and “Calculate Performance”. All subtasks were thoroughly analysed one by one. Then outlines of possible window sequences were drawn. After all details of the complete task were considered, the windows were implemented in the prototype.

Detailed descriptions of all subtasks performed during these three main tasks, together with images of the implemented windows, are given in Appendix G.
Chapter 10
Information Design

10.1. Presenting Information

We have chosen to present the information in the prototype in mainly two different ways, on a map or as text. Some information, e.g. GIS data and relative positions of objects, is easier for the human being to grasp when presented as a 2D view than in plain text. Most of the information in the C²SR-system is of this kind. Therefore it was an obvious choice to display the information in the main sub-window, the Map Window, as a map and thereby give the users an overview of the information. Other information, e.g. the status of systems and calculations, has a text-based presentation in separate windows that can be displayed on command.

Index Map
The Index Map window displays a key map where the region of the scenario is indicated. This window will give the user an overview over the location of the scenario.

Map Layers
The Map Layers window shows the GIS data included in the scenario. The window contains checkboxes for all GIS layers in the scenario. By marking a checkbox the user displays that particular GIS layer on the map in the Map Window. It is also possible to add new GIS layers to the scenario from this window.

Map Window
This main sub-window displays a map over the region of the scenario with the GIS layers and calculation results that the user has chosen to display. In the bottom of the window the coordinates and the height of the location indicated by the mouse is displayed. Systems are also displayed on the map in the form of icons. The icons are composed of a symbol and the name of the particular system. Moving systems are displayed as icons and lines to show the paths of the systems. Results from the coverage calculations are displayed on the map as semi-transparent grey colour in each point not covered and clear in the

Figure 12. Map layers window
areas covered. This is intended to give an intuitive understanding of the actual situation. The result of a dynamic coverage calculation is operated from the Time Slider window.

**Figure 13. Map View window.**

**Time Slider**
The Time Slider window contains a ruler that is connected to a dynamic scenario and buttons to manage the ruler. The scale of the ruler is automatically adjusted to agree with the time it takes for the dynamic system to move along its path. The user has the choice of displaying the dynamic coverage continuously, in steps or as a composite coverage image.

**Figure 14. A dynamic calculation result.**  **Figure 15. The time slider.**
Scenario Explorer
The Scenario Explorer window shows an overview over the systems and calculations in the scenario in the form of a tree structure. This window allows the user to see all the systems included in the scenario and all calculations made, even the ones that are not visible in the Map Window. The Scenario Explorer is connected to the Map Window, i.e. when selecting a system in the Scenario Explorer window the same system will be marked in the Map Window. The user can view the status of the scenario, the systems and the calculations by marking the object of interest and pressing the status button. The status is displayed as text in a detached window.

![Figure 16. Scenario Explorer.](image1)
![Figure 17. The status window.](image2)

10.2. Design of Icons

For the design of military symbols, there is no homogeneous international standard. However, there is a Swedish standard for symbols, designed to as far as possible agree with the symbols that are used in international contexts. These symbols should always be used when creating situation picture maps and for computer-supported management systems [R12]. Therefore, we used these symbols when we designed the icons for the user interface (Appendix H).

All icons for systems simulated by means of the user interface are black. This is because the C²SR-system is not intended to differ between friendly and hostile systems during performance calculations. When downloading situation pictures the icons for friendly systems are blue and the icons for hostile systems are red, according to the Swedish standard. This use of three different colours intends to give the user a clear understanding of the situation and to avoid confusion of “real” systems downloaded from the Net and systems simulated in the C²SR-system.
Chapter 11
Visual Design

11.1. The Style Guide

To give the user interface a uniform graphical look, a style guide was defined. In the Ericsson group there is a project named Wanda [W5]. The Wanda project’s main task is to develop ”the Guide”, the Ericsson operation and maintenance style guide for development of GUIs. The Guide gives detailed information about all parts of the design of a GUI, for example the design of windows, menus, toolbars and wizards and the use of buttons, checkboxes, text areas and symbols. We have chosen to use this style guide for the visual design of the user interface.

11.2. Implementation of a Prototype

We chose to implement the prototype in Java, because of its extensive graphical programming support. Swing components, which are part of the Java™ Foundation Classes, were used to implement the GUI of the prototype. A description of the classes in the prototype is given in Appendix I.

Figure 18. The prototype.
Chapter 12
Discussion

12.1. The Pre-study

The study of the requirements for the user interface is, to a large extent, based on information from the questionnaire and from discussions with the users. As mentioned in Chapter 3, we have been in contact with five officers who work with sensor-, electronic warfare- and communication systems management and these officers have been considered as the users. Nielsen states that five users are sufficient for a reliable result when conducting usability studies [L5]. Other researchers have questioned this statement. Whether Nielsen is right or not, scanty answers from one of the users will reduce the reliability of the study considerably. Another insecure factor concerning the users is the long-term development of NBF. It is not evident who the future users will be and what background they will have. Therefore we might have got other answers if users with different military background were included in the study. Hence, a study including more users with different military background should be performed to verify the results.

The officers participating in the questionnaire gave similar answers to many of the questions considering usability and functionality. These answers were considered when designing the user interface. However the answers to the questions about trustworthiness differ widely and one of the users did not answer these questions at all. Therefore we could not draw any unambiguous conclusions from this part of the questionnaire. To get more reliable results concerning trustworthiness further studies must be performed.

12.2. Implementation

Even though Java has extensive support for graphical design, it has taken a much time and effort to get the GUI exactly as we wanted it. Some effort that we would have preferred to put into designing additional functionality has been put into changing details in window design, etc. It might have been better to use an application-building program where one easily can build the layout and still be able to modify the code as preferred. On the other hand, it is always hard to work with code that you have not written yourself. Therefore, modifications might still have taken long time if we had used an application-building program.
12.3. Future Work

Due to the limited time of 20 weeks for our work, this thesis comprises only a small part of the development of the future C²SR-system. Therefore, we have put together a few suggestions of possible future work on the basis of the results of this thesis.

12.3.1. Iterative Development

According to the user-centred design method, brought up in Chapter 2, an iterative system development model (Figure 19) is a good way to ensure usability of the system. In this thesis only the first part of the iterative system development model was performed. Therefore, future work with the C²SR-system should include analysis, re-design and implementation of a new prototype in additional steps.

![Figure 19. An iterative system development model.](image)

The next analysis phase involves testing and evaluation of the prototype. A usability test of the prototype should be performed by means of the future users. This test aims to validate that the product satisfies the users’ requirements. It is very important that the usability testing does not include an incorrect target group. Officers, like the ones who participated in the questionnaire in this thesis, are therefore suitable for testing the usability of the prototype. The usability test should be followed by a usability evaluation. This evaluation will then serve as a basis for design and implementation of a new prototype.

12.3.2. Implementing Functionality

After testing and evaluating the prototype, basic functionality can be implemented. First, a “map engine” that handles the map layers correctly should be implemented. Then a sensor performance calculation module can be added. One can, for instance, start with a simple free space calculation algorithm that includes the parameters of the systems, possible jamming effects and the topography. With this functionality implemented, one can simulate scenarios and calculate performance of sensors and electronic warfare in optional regions.
12.3.3. Further User Interface Design

Our implementation of a user interface for the C²SR-system only considers limited functionality. According to the results in Chapter 5, additional functionality is desired. The most important remaining parts are:

- Systems including communication equipment.
- Design of the calculation modules “Operation Area” and “Optimization”. Additional calculation modules for communication performance.
- A user interface suitable for soldiers in radar shelters, as well as a simplified user interface adapted for a PDA.
- The presentation of dynamic GIS data.
- Reconnaissance management, i.e. command and control of available resources

When possible, we have designed the user interface to prepare for this additional functionality.
Chapter 13
Conclusions

The main objective of this thesis was to perform a study of the needs of a user interface which provides the users with information necessary for decision support, planning and analysis of systems of sensor-, communication- and electronic warfare-systems.

By starting from the field of application, i.e. planning, analysis and decision-support, we discovered that there are three main tasks that the users will perform with the user interface. These tasks are to “simulate a scenario”, to “get the situation picture” and to “calculate the performance” of the systems.

We discovered that usability is one of the most important issues for the design of the user interface. Effectiveness, efficiency, satisfaction, learnability and flexibility are attributes that should be considered to attain usability. By contacting officers, representing the future users, we found out more about the users, their tasks and work environment. Important statements from them were that the time for completing the tasks will be limited, the users’ level of pre-knowledge and experience will differ considerably and the user interface must be designed to avoid mistakes. With this as a starting point we designed a GUI that fulfils the usability requirements. This was achieved by following conventions, adapting the vocabulary, making the sequence of work intuitive for the users and allowing the users to choose the level of detail for the tasks.

Trustworthiness of the information in the user interface is another issue that we have considered. Because the C²SR-system will be used as a decision aid we believe that it is of great importance that the users get the right understanding of how trustworthy the information is. We discovered that time-stamps and knowledge about the information’s origin will increase the average user’s trust. However, it is impossible to make a decision aid for analysis of sensors trustworthy because too many uncertain factors affect the performance of sensors. Therefore, usability is much more important than trustworthiness for such a system.

The prototype was implemented to visualise our design of a usable GUI for the C²SR-system.
Acknowledgements

We would like to thank all people that have been supporting us and giving us advice during this thesis. Thank you Martin Gustavsson for being our supervisor at Ericsson Microwave Systems AB and for giving us great support during our work. Special thanks to our examiner Ivan Rankin for giving us valuable feedback and guidance when writing the report. We also appreciate the officers in the Swedish Defence who have taken their time to answer the questionnaires.

Moreover, we would like to thank the other people at FT/X for giving us a nice time at Ericsson Microwave Systems AB.
References

Literature


Reports


Articles


Websites
http://www.jjg.net/ia/elements.pdf, 130902 3:24 PM.

http://designafterthought.net/eftertankar/eftertanke01.htm, 130902 11:01 AM.

http://www.useit.com/alertbox/20000319.html, 021002 4.00 PM.

http://www.lmera.ericsson.se/~wanda/, 141102 11.00 AM.


http://www.magazine.se/webbdesign/Artiklar_om_gränssnitt.html, 200801.
Appendices

Appendix A - A Questionnaire

Hej,
Vi är två examensarbetare på Ericsson Microwave System AB i Göteborg. Som en del av utvecklingen av LedsystT, jobbar vi med utformning av användargränssnitt till ett framtida nätverksbaserat programverktyg för planering, analys och beslutsstöd för system av sensor-, kommunikations- och telekriksystem.
Först och främst kommer vi inriktas oss mot planering och analys av användandet av radarsensorsystem vid inverkan av olika faktorer, t ex topografi/vegetation och telekriksystem. Användargränssnittet ska dock kunna utvidgas för analys av system inom hela det elektromagnetiska spektrumet.

Den funktionalitet som ska ingå i det framtida programverktyget finns till viss del redan i existerande program, t ex Wrap och Eliran. Vi kommer därför bygga utformningen av användargränssnittet till det framtida programverktyget på dessa program. Därför är en utvärdering av dessa program är en viktig del i vårt arbete.

Det framtida programverktyget kommer även innehålla funktionalitet som idag inte finns i något program. Eftersom ni i Försvarsmakten är de framtida användarna till ett sådant programverktyg, är det mycket viktigt att era åsikter tas hänsyn till. Era önskemål och synpunkter på funktionalitet i det framtida programverktyget är därför mycket viktiga.

Frågeformuläret innehåller 22 frågor. Du kan skriva svaren direkt i dokumentet eller på ett separat papper (frågorna är numrerade) och maila svaret till oss. Vi är vädligt tacksamma för att du tar dig tid att besvara frågorna!

Med vänliga hälsningar,
Lisa Jonsson, Lisa.Jonsson@emw.ericsson.se
Karin Sallhammar, Karin.Sallhammar@emw.ericsson.se
Ett framtida programverktyg

Även om planering och analys av radarsensorer inte är en av dina arbetsuppgifter, så tror vi att du som officer kan ha värdefulla åsikter angående ett sådant framtida programverktyg.

Funktionalitet
Detta är de tjänster vi tror är speciellt viktiga i ett framtida nätverksbaserat programverktyg för planering och analys av användandet av radarsensorer.
Kommentera de olika tjänsterna. Är det någon tjänst du anser överflödig? Motivera!
1. Inhämtning av aktuell lägesbild. Dvs grafisk presentation av sensornas nuvarande placering/läge.
2. Beräkning och grafisk presentation av sensornas täckning.
3. Beräkning och grafisk presentation av sensornas täckning vid inverkan av telekriegresurser.
5. Dynamik. Dvs kunna se hur täckningen varierar med tiden när rörliga sensorer förekommer.
6. Har du förslag på ytterligare tjänster som bör ingå i ett framtida programverktyg?

Presentation av resultat
7. Hur vill du att täckningen ska kunna beräknas och presenteras i programverktyget?
   7.1 Visa var signal/brus förhållande är högre än ett visst decibelvärde.
   7.2 Visa upptäcktsannolikheten för en given radarmålarea.
   7.3 Visa förväntad tid till upptäckt för en given radarmålarea.
   7.4 Visa följenoggrannhet
   7.5 Annat?
8. Vill du ha en 2D grafisk presentation av resultatet?
9. Vill du ha möjlighet att se en 3D grafisk presentation av resultatet? Om ja, vad tror du att en 3D presentation skulle tillföra för information som man annars kan missa i en 2D presentation?

Övrigt
10. Övriga kommentarer angående ett framtida programverktyg:
Användbarhet

För att ett användargränssnitt verklig ska vara användbart är det viktigt att hänsyn tas till de som ska använda programmet och den situation de befinner sig i. Användargränssnittet måste testas och utvecklas i många omgångar för att det ska vara anpassat till de framtida användarna i det Nätverksbaserade Försvaret (NBF). Dem har vi ingen möjlighet att få kontakt med men vi är övertygade om att ni har en mer korrekt uppfattning om vilka de kommer vara och hur deras situation kommer att se ut, än vi har. Därför har vi sammanställt några frågor som ska ge oss riktlinjer om dem med utgångspunkt i dagens situation.

Användarens situation
11. Det är viktigt för oss att veta i vilken situation ett planering/analysverktyg för radarsensorsystem kommer användas. Kommer man tex ha gott om tid eller kan det vara brådskande när man planerar utplacering av sensorer?
12. Har du någon mer kommentar som kan vara viktig angående användarens situation?

Graden nödvändig förkunskap
13. Vilka förkunskaper kan man förvänta sig att de framtida användarna har?
   13.1 Operativa / taktiska:
   13.2 Tekniska:
14. Är det viktigt att kunna utveckla programmet med “växande vana”? Dvs tror du att erfarna användare vill arbeta på ett annat sätt än nya användare?
   14.1 Har du i så fall någon idé om på vilket sätt de kommer vilja arbeta annorlunda?

Övrigt
15. Är det något mer du vill tillägga om de framtida användarna?

Trovärdighet

Allmänna frågor
16. Hur mycket tror du att följande faktorer påverkar om man uppfattar information från ett program som trovärdig? (Motivera gärna!)
   16.1 Känsla av att ha koll på vad informationen kommer ifrån.
   16.2 Känsla av att ha koll på programmets struktur.
   16.3 Informationens noggrannhet
   16.4 Annat?

Wrap-relaterade frågor
17. Vad använder du Wrap till?
18. Litar du på informationen/resultaten från Wrap? Varför?
19. Tror du att du skulle upptäcka om någon information från Wrap var fel?
20. Kontrollerar du ibland den information du fått från Wrap? När i så fall?
   20.1 Känner du i så fall dig säker på när sådan komplettering behövs?
   20.2 Brukar det du kommer fram till efter kontrollen överensstämma med den ursprungliga informationen från Wrap?

Övrigt
22. Är det OK att vi ringer upp om vi vill diskutera vidare kring något av svaren?
Appendix B - Task Models

Simulate a Scenario

**Goal:** Simulate Scenario  
**Plan:** 1,2,3

1. Choose region  
2. Add systems  
3. Set position of systems

1.1 Choose location  
1.2 Select GIS layers  
1.3 Name scenario 

2.1 Select type of system  
2.2 Name system

Get the Situation Picture

**Goal:** Situation Picture  
**Plan:** 1,2,3,4 or 1,4,2,3

1. Choose presentation  
2. Choose area  
3. Name sit. picture  
4. Select information

Calculate Performance

**Goal:** Calculate Performance  
**Plan:** Optional

1. Select systems  
2. Calculation details  
3. Name calculation

2.1 Select calculation  
2.2 Set parameters  
2.3 Set accuracy and time range
Appendix C - Usability Checklist

This checklist for important HCI-principles is a revised version of a model for evaluation of usability, carried out in a master thesis by Emelie Ekvall at Linköping University [R13]. This checklist is revised to fit an evaluation of usability for a GUI rather than a website as in Emelie Ekvall’s thesis.

Visual clarity
1. It should be easy to find relevant information [W6].
2. Information of different kinds should be clearly separated [L6].
3. When large amounts of information is displayed, it should be clearly separated into sections [L6].

Clarity
4. The terminology should be familiar to the user [L5].
5. It should be clear where and in which format to input information [L5].
6. Icons, symbols and other graphical information should be easy to recognise and understand. Conventions, if any, should be followed [L6].
7. It should be clear what to use each part of the user interface for [L6].

Consistency
8. Icons, symbols and other graphical information should be consistent [L6].
9. The same kind of information should be displayed in a consistent position, with consistent layout in consistent format [L5].
10. The input format of a certain kind of information should be consistent throughout the whole user interface [L6].
11. The method for feeding a certain kind of information into the user interface should be consistent [L6].
12. The method for selecting among different choices should be consistent [L6].
13. The way the user interface responds to a certain action should be consistent [L6].
14. Colours should be used consistently throughout the whole user interface [L6].

Informative feedback
15. It should be clear which measures the user must take to accomplish a certain task [L6].
16. Instructions and messages from the user interface should clearly indicate which measures the user should take [L6].
17. It should at every moment be clear to the user which actions that are available [W7].
18. The user interface should inform the user when a requested action is performed successfully [L6].
Structure
19. The user interface should give a correct understanding of the structure of the system [L5].
20. It should be clear to the user where in the structure he is [L5].

Adequate functions
21. Sufficient and relevant choices should be displayed in every element of a task [L6].

Flexibility and control
22. It should be easy to undo and reverse an action [L6].
23. It should be possible to change information generated automatically by the user interface [L6].
24. At information input, the user interface should automatically fill up stored information [L6].

Error handling
25. Error messages from the user interface should in a legible way inform the user of what is wrong, where the error occurred and why the error arose [A2].
26. There should be a possibility to control input before it is dealt with [L6].
27. The user interface should immediately respond on an erroneous command, e.g. when text is fed into a field for numerical values [L6].

User support
28. Help and search functions should be available from all parts of the user interface [L6].

Use of colour
29. All parts of the user interface should be readable on monochrome or poor resolution screens and for colour-blind persons [W6].
30. Colours should be used according to conventional associations, e.g. red = stop [L6].
31. Colours should be used to help increasing the clarity [L6].

Mappings
32. Natural mappings between items and their representations should be used [L2].
Appendix D - Usability Evaluation of Eliran

The numbers refer to the numbering in Appendix C.

1. OK.
2. OK.
3. OK.
4. OK.
5. OK.
6. Existing window-conventions are followed. The icons specific for ELIRAN are clear and well selected.
7. OK. Especially the pull-down main menu is well designed.
8. OK, however the symbols for radar-systems and jammer-systems are confusingly alike.
9. OK.
10. OK.
11. OK.
12. OK.
13. OK.
14. OK.
15. OK. ELIRAN gives an appropriate amount of help without being annoyingly helpful.
16. OK. The fact that messages often do not appear until a mistake has been made is satisfying from an advanced user’s point of view.
17. OK. The non-available actions are shaded in pale grey
18. OK.
19. OK. However we are not entirely satisfied with the fact that connections between stations are treated as objects.
20. OK.
21. OK. The “Query Wizard is a good example.
22. There is no undo-functionality.
23. OK.
24. OK.
25. OK.
26. OK.
27. OK. It is impossible to type letters in fields intended for numbers.
28. No, however this version of ELIRAN is a demo version.
29. OK for a monochrome or poor resolution screen, however the shading of the connections might cause problems for colour-blind people.
30. OK.
31. OK, however sometimes different colours are used without obvious reasons.
32. OK, with one important exception: when displaying coverage versus non-coverage, the covered area should be visible and the non-covered area should be shaded!
Additional comments:
+ The possibility of writing the result of a calculation to a document.
+ Extensive drag-and-drop possibilities.
+ Flexible management of windows.
+ Pre-defined systems and the possibility of adding new systems to “System Inventory”.
- No layer-functionality (one might, for instance, want to hide the connections in a scenario).
Appendix E - Usability Evaluation of Wrap

The numbers refer to the numbering in Appendix C.

1. OK if the user is experienced. Due to the extensive functionality of Wrap, a novice user working with limited tasks might find the user interface difficult to grasp.
2. OK.
3. OK. However exceptions exist, for example when creating a new station.
4. OK.
5. In most cases but not always.
6. All existing Window-conventions are followed. The icons specific for Wrap are not always obvious.
7. OK.
8. OK, however sometimes the user interface seems ”too” consistent, for example when the same symbols are used for both radars and jammers.
9. OK.
10. OK.
11. OK.
12. OK.
13. OK.
14. OK.
15. This is not clear for a novice user. However, the sequence of work is logically constructed and with a short introduction and by trial and error, the users should be able to complete his tasks on his own.
16. OK. The user interface does not give any unnecessary instructions. However, the help function is very detailed.
17. OK. The non-available actions are shaded in pale grey.
18. No. For example, when performing calculations, no message is given to indicate when the calculation is completed.
19. OK.
20. OK.
21. OK.
22. No. Most of the actions cannot be reversed, even though an undo-function exists.
23. OK.
24. OK.
25. OK.
26. OK.
27. OK.
28. OK.
29. OK. The user can choose which colours and raster to use when displaying the calculated performance.
30. OK.
31. OK. For example, when displaying calculated performance, different colours can be chosen to display different levels of the result.

32. OK, with one important exception: when displaying coverage versus non-coverage, the covered area should be visible and the non-covered area should be shaded!

Additional comments:
+ It is possible to tile or cascade different projects.
+ The extensive calculation possibilities.
- It is not possible to remove or hide windows that are included in a project.
- There is no drag-and-drop functionality between windows.
- Insufficient connections between windows. For example, when the user marks a radar object in the "List View" window, the corresponding radar symbol in the "Map" window should also be marked.
- The icon for the cursor is missing in the toolbar.
Appendix F - The Menu Bar and the Tool Bar

The Menu Bar
The menu in the prototype contains the main menu items: Scenario, View, Map, Situation Picture, Calculate and Help (Figure 20). These main menu items are chosen and positioned in the menu in a way that logically groups the functionality available in the user interface.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>View</th>
<th>Map</th>
<th>Situation Picture</th>
<th>Calculate</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Scenario</td>
<td>Zoom In</td>
<td>New</td>
<td>Performance</td>
<td>Content</td>
</tr>
<tr>
<td></td>
<td>Explorer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td>Map Layers</td>
<td>Zoom out</td>
<td>– Map</td>
<td>Operation Area</td>
<td>About</td>
</tr>
<tr>
<td>Close</td>
<td>Index Map</td>
<td>Scale</td>
<td>– Document</td>
<td>Optimization</td>
<td></td>
</tr>
<tr>
<td>Add System</td>
<td>Time Slider</td>
<td>– 1:100 000</td>
<td>Update</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>– 1:500 000</td>
<td>Distance</td>
<td>Historic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>Path Profile</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 20. The menu in the C²SR-system.

Scenario
Since a scenario is the fundamental object that the user works with, this menu item is positioned on the far left of the menu bar. This first part of the menu includes tools for planning and simulating by creating new scenarios, open saved ones, printing scenario information, and adding systems to the current scenario.

View
Here the user can view or hide the four minor windows containing information about the scenario.

Map
This part of the menu contains map tools. The user can pan, zoom in or out in the map and easily adjust the map to a certain scale. Distance and path profile between two positions on the map can be shown.

Situation Picture
From this part of the menu, the user will be able to download data of current interest from the Net. The situation picture may be presented on a map or in a document. The current situation picture can be updated. Previously saved situation pictures might also be downloaded.

Calculate
From this part of the menu, the user can reach the calculation modules available for the systems in the scenario.
Help
There is a detailed help functionality (however not implemented in the prototype) and it is accessible from all parts of the user interface.

The Tool Bar
By means of the tool bar, the most frequent actions will be easy to perform. The actions we have chosen for the tool bar are presented in Figure 21. The icons have been divided into small groups that logically belong together.

*Figure 21. The tool bar. From left to right: New, Open, Save, Print, Add System, Select, Pan, Zoom In, Zoom Out, Update Situation Picture, Quick Performance Calculation.*
Appendix G - Working with the User Interface

Simulate a Scenario

Goal: Simulate Scenario
Plan: 1, 2, 3

When simulating a new scenario the user chooses "New" from the Scenario menu item. This results in a sub window (Figure 23) that allows the user to perform subtask 1.
Figure 23. Choose region for a scenario.

1.1 The user can choose region by either selecting a rectangle on the map with the mouse or by filling in the centre coordinates. In the prototype a map of 100 km * 100 km over Gothenburg is chosen by default because map material over only this region was available to us.

1.2 The user can select the GIS layers of interest. These layers will be downloaded and possible to view/hide on the Map Window. By default a standard map including topography, water and cities is downloaded.

1.3 To give the scenario another name the user types the required name in the text field. The default name is Scenario1.
When finishing subtask 1, the desktop consists of the Map Window with the standard map, the Scenario Explorer window showing the structure of the scenario and the Map Layers window with checkboxes for all the selected GIS layers. (Figure 24) If the user wants to download more GIS layers, he or she can do so from the Map Layers window. Next step in simulating a scenario is to add systems. This is done from a sub window (Figure 25), which is opened by choosing “Add Systems” from the Scenario menu item.

Figure 24. A new scenario.

Figure 25. Add systems to a scenario.
2.1 The desired type of systems can be selected from a list of systems.
2.2 To give the system another name the user writes the wanted name in the
text field. The default name is PS-90, FSR 890 etc.

When the type of system and a name is chosen the user presses the Add
button. Then the icon of the system is visible on the map in Map View (Figure
26) and added to the Scenario Explorer tree (Figure 27). The user can add all
the systems first and then position them (change status) as desired.

3 When a system is added to the scenario the user can move it to the desired
position by dragging and dropping it on the map or by changing the
coordinates in the Status window (Figure 28). The Status window is
opened from the Scenario Explorer window. In the Status window other
settings of the system can be changed as well.

Figure 26. Systems displayed in MapView...
Figure 27. ...and in Scenario Explorer.

Figure 28. Status of a system.
Getting the Situation Picture

1. Choose presentation
2. Choose area
3. Name sit. picture
4. Select information

Goal: Situation Picture
Plan: 1,2,3,4 or 1,4,2,3

Figure 29. Task model of Situation Picture.

When the user wants to get the situation picture, he chooses “New” under the “Situation Picture” menu item. When doing so a sub-menu appears.

1. In the sub-menu the user can choose either “Map” or “Document” for the presentation of the situation picture. If the user selects “Map” a Situation Picture sub-window appears. (Figure 30)

Figure 30. Get situation picture.

2. To present the situation picture on a map the user has to choose a region. This is done by selecting an existing scenario or creating a new one. If a scenario already is opened when starting the Situation Picture task, that scenario is set to the default region from which the situation picture will be downloaded. If no scenario is open there is no default region. The user can choose a saved scenario by pressing the Browse button and selecting one. If he does not want to start from an existing scenario he can press the “New” button and the New Scenario sub-window will open. There he can define the region in the same way as in the Simulate scenario task.

3. To rename the situation picture the user writes the required name in the text field. The default name is Situation Picture_1 etc.
When the region is defined the user can select what kind of information to download. In the Situation Picture sub window there are checkboxes for two types of information, systems and dynamical GIS data. Systems include all the different types of systems and dynamical GIS data includes e.g. weather and traffic fluxes. If the user for example chooses PS-90, the positions, status and IDs of all the PS-90 radar stations that are located in that particular region will be downloaded. If the user requires to get other information, e.g. information about a PS-90 with a certain ID number, he can press the “Advanced” button.

The result of downloading a situation picture with for example PS-90s is that a scenario with these systems is created and displayed on the Map Window. (Figure 31) In the same way as in “Simulate Scenario” the Status window for the systems and the scenario can be opened from the Scenario Explorer window. The status of the scenario includes date, time and location for the download of information.

![Figure 31. A situation picture.](image-url)
Calculate Performance

**Goal:** Calculate Performance  
**Plan:** Optional

1. Select systems  
2. Calculation details  
2.1 Select calculation  
2.2 Set parameters  
2.3 Set accuracy and time range  
3. Name calculation

Figure 32. Task model of Calculate Performance.

When the user wants to calculate performance of systems he must start from an open scenario or a situation picture. He chooses “Sensor Performance” from the Calculation menu item and then a Calculation Wizard appears (Figure 33). The user can advance in the Calculation Wizard by either pressing the Next button or clicking on a box in the image to the left in the Calculation Wizard. When clicking on a box the user will advance to that particular step in the Wizard, thus the user can complete the calculation subtasks in any order.

Figure 33. The Calculation Wizard.  
Figure 34. Select systems.

1 In the first part of the Wizard all the systems included in the scenario are listed and selected (Figure 34). If the user does not want to include all the systems in the calculation he can easily deselect the ones not wanted.
2.1 In the “Select calculation” step the user selects calculation method from a list of alternatives, e.g. Signal/Noise (S/N), Probability of Detection (P_D) (Figure 35). The settings for the different calculation methods can be changed by pressing the “Detail” button for each method.

2.2 The next part of the Wizard includes parameters that can affect all the different calculations i.e. topography, ground reflection, vegetation, atmosphere and rain reflection (Figure 36). It also includes information about the target, i.e. radar target area and target height. The default values for the parameters are displayed. If the user wants to change the default value, there is a button for each of them. There are also checkboxes so the user can exclude parameters from the calculation. The target information is more commonly adjusted and therefore it is possible to change the radar target area and the target height directly in this part of the wizard.
2.3 In the next part of the Calculation Wizard the user sets the accuracy of the calculation and the time intervals (Figure 37). The default resolution is set to 1000m. When performing a calculation with a dynamic system, the period of time and the number of calculations must be taken into consideration. The dynamic calculation can either be continuous or discrete. The default value for the period of time is the time it takes for the dynamic system to move along its path. The default value for the resolution is set so there are four calculations per minute.

The last part of the Wizard (Figure 38) displays a summary of all the choices made and a text field for the name of the calculation. To change the name, the user writes the new name in the text field. The default name is S/N, P_D, etc.

By pressing the “Finish” button the calculation is performed, displayed on the map and added to the Scenario Explorer window. The Status window for the calculation displays all the calculation settings and can be opened from the Scenario Explorer window. Examples of how static and dynamic calculation results are displayed in the Map View are given in Figure 39 and Figure 40.

Figure 39. A static sensor performance calculation result.
Figure 40. Radar sensor performance affected by a flying jammer. The dynamic calculation result is operated by the time slider.
Appendix H - Design of Icons

The symbols in Figure 41 were chosen for the user interface. The box, circle and triangle represent ground-, surface-, and air-based platforms respectively. There are two different symbols for radar and jamming equipment. The two minor circles under one of the ground-based platforms indicate that the platform is mobile. The name of the system is positioned to the left of the symbol in the icon.

<table>
<thead>
<tr>
<th>![Symbol]</th>
<th>Mobile ground radar</th>
<th>![Symbol]</th>
<th>Airborne radar</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol]</td>
<td>Stationary ground radar</td>
<td>![Symbol]</td>
<td>Airborne jammer</td>
</tr>
<tr>
<td>![Symbol]</td>
<td>Sea radar</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 41. Symbols in the user interface.*
## Appendix I - Class Description

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asystem.java</td>
<td>Describes the object system.</td>
</tr>
<tr>
<td>C2SRsystem.java</td>
<td>Starts the C2SR-system prototype.</td>
</tr>
<tr>
<td>Calculation.java</td>
<td>Describes the object calculation.</td>
</tr>
<tr>
<td>CalculationStatusDialog.java</td>
<td>Displays the properties of a calculation.</td>
</tr>
<tr>
<td>CalculationWizard.java</td>
<td>The “Calculation Wizard” window.</td>
</tr>
<tr>
<td>GetSituationPicture.java</td>
<td>The “Get Situation Picture” window.</td>
</tr>
<tr>
<td>IndexMapView.java</td>
<td>The “Index Map” window.</td>
</tr>
<tr>
<td>JammerStatusDialog.java</td>
<td>The “Status” window for a jammer.</td>
</tr>
<tr>
<td>MainFrameView.java</td>
<td>The main window of the prototype.</td>
</tr>
<tr>
<td>MapLayersView.java</td>
<td>The “Map Layers” window.</td>
</tr>
<tr>
<td>MapView.java</td>
<td>The “Map View” window.</td>
</tr>
<tr>
<td>NewScenario.java</td>
<td>The “New Scenario” window.</td>
</tr>
<tr>
<td>RadarStatusDialog.java</td>
<td>The “Status” window for a radar.</td>
</tr>
<tr>
<td>ReadScenarioFromFile.java</td>
<td>Opens a “saved” scenario.</td>
</tr>
<tr>
<td>Scenario.java</td>
<td>Describes the object scenario.</td>
</tr>
<tr>
<td>ScenarioExplorerView.java</td>
<td>The “Scenario Explorer” window.</td>
</tr>
<tr>
<td>ScenarioStatusDialog.java</td>
<td>The “Status” window for a scenario.</td>
</tr>
<tr>
<td>SliderView.java</td>
<td>The “Time Slider” window.</td>
</tr>
<tr>
<td>SystemInventory.java</td>
<td>The “System Inventory” window.</td>
</tr>
</tbody>
</table>