USER INVOLVED **SOFTWARE & USER INTERFACE DESIGN** FOR THE INTERIOR LIGHTING IN THE CAR

Simon Welarius

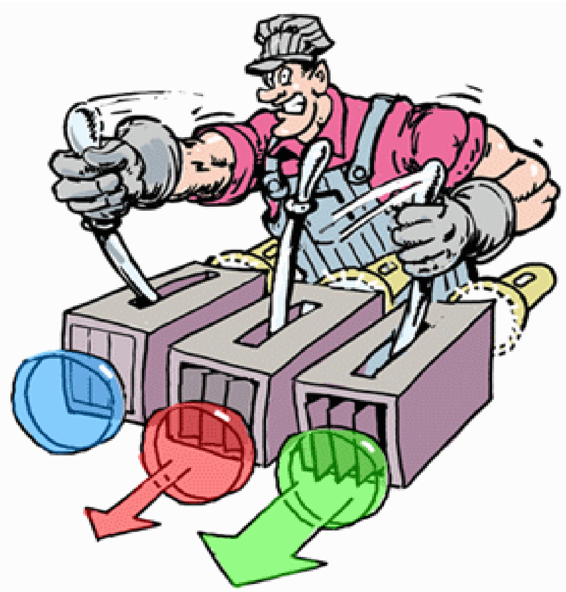
Master Thesis

Department of Interior
Saab Automobile

Division of Ergonomics and Aerosol Technology
Department of Design Sciences

ISRN LUTMDN/TMAT - 5052 - SE

USER INVOLVED



SOFTWARE & USER INTERFACE DESIGN

FOR THE INTERIOR LIGHTING IN THE CAR

SIMON WELARIUS

Table of contents

Preface

This master’s thesis was performed during 20 weeks in the autumn 2001 (2001-09-03 until 2002-01-25) within the master of electrical engineering education at Lund Institute of Technology (LTH). The interior department at Saab Automobile in Trollhättan is the orderer of this master’s thesis. The master’s thesis was shaped in co-operation with the department of Ergonomics and Aerosol Technology at LTH. The head supervisor at Saab Automobile was Christer Waldemarsson and the supervisor at LTH was Lars Hanson.

The reason for this thesis was that the interior department at Saab Automobile wanted to change the technique for the interior lighting in the car. The present technique consists of incandescent lamps with “tiny” software to control them. The wish was to change the whole system to an LED technique and this is the focus of the project “LED & Interior Lighting”. This project turned out to be very big and was therefore split into two master’s theses, this one and another one.

This thesis treats the software and the user-interface part of the project “LED & Interior Lighting”. To read about the hardware design part of “LED & Interior Lighting” see the report about hardware and test methods by Karl Flensburg. The work about the light design will be summarised in a report by Kristoffer Johnsson some months after this report is finished. The goal is to present a system with both software and hardware that are thoroughly investigated which can then be the basis for the interior light function for Saab.

I want to thank everybody that helped me to carry through this thesis and those who supported me during this time. My gorgeous wife that was indulgence with many late nights and supported with cooking etc, Rose-Marie Akselsson for the huge help with the language in the thesis, Roland Akselsson for constructive criticism, Ivan Wilkinsson, Johny Ekberg and Klas Lundgreen for much time in supporting with the software, Svante Welin from the market department on Saab Automobile for great support with the clinics, Kristoffer Jonsson light designer and instructor on Saab Automobile, Crister Waldemarsson head-instructor on Saab Automobile, Lars Hanson instructor on LTH in Lund, Karl Flensburg and Anna-Carin Dahl my fellow workers on Saab Automobile. Andreas Ericsson for the rent-free living, my family for all the prays and God for hearing the prays.

1. Introduction

<Background, motive, purpose and issue as well as explanations of expressions>

1.1 Background

The interior light functionality in cars has been neglected for a long time. Design and functionality has remained stable over a long period. The fundamental reason for this has been limited illumination techniques and poor knowledge about the possibilities and the value of light designing.
Today there exist both technique and understanding for the illumination and its possibilities. New techniques, for example LEDs, give better possibilities to design more flexible illumination equipment without producing heat but give longer lifetime, lower energy consumption and a more compact size. The techniques also give the possibility to control the characteristics of the light and together with that the identity of the car.

The technical improvements also open doors to new functions, which in turn allow more automatic functions, sensors and controls, user interfaces, logics etc.
To be able to build economical and user-friendly cars there is a demand for reliable solutions with natural logics. The illumination is a good example of a function that will increase complexity. Like all new functions it must be implemented in a user-friendly manner.

To provide a complex function and control of the interior lighting in the cars, SAAB has started the project LED & Interior lighting. In this project the system from the source of light to the user is studied. The result of this project can be used as a basis for the design of the illumination in future car models.

1.2 Purpose

Light has the capability to affect humans in both positive and negative direction. One example is when springtime comes with light, joy and other spring feelings. However, light can also cause, for example, headache if it is used in the wrong way. Tests have been made about how different light colours and light intensities affect humans. These tests have shown how humans get, for example, more creative, calmer, less stressed etc. as an effect of different kinds of light.

Today the interior lighting in Saab cars consists of incandescent lamps. The main task for these lamps is to help you see in the dark and, in general, the only demand is that they shall reach a specific intensity level and that they are placed so they do not dazzle the users.

With the LED technique it is possible to create a light with exactly the right colour and intensity. This makes it possible to increase the application area for the interior lighting. Besides helping you see in light it can increase the safety in many ways, for example make you less stressed while driving.

The new LED technique makes it possible to add many more functions than was possible with the earlier interior lighting technique (incandescent light). All these new functions create a problem: how do one control all these functions? The customer should not have to read a huge instruction book to be able to understand how to manipulate the lighting in the car. All new functions should be activated automatically and the driver should be able to decide his own light environment. Another problem is now appearing - how to design a system that runs automatically but also enables the user to decide how he wants the system to work.

Some big questions arose at the start of this project. How should all the new ideas be structured? Are there any new ideas to be added? What shape of software is possible with the resources at hand? Which point of action should be used to create a good user interface?

How should one create lighting functions and a control-panel that satisfy the driver in all conditions? A general rule in user-interface design is that it is impossible to design a user-interface that satisfies everybody. The goal, however, is that this design should not be harder to understand and control than the one that already exists in a regular car. This is usually a single switch, occasionally two or three, not more.

The aim of this thesis is to design software and a user interface for the new interior lighting in Saab cars. Both the software and the user interface are supposed to increase safety and usability for the car. Therefore the software and the user interface have to be designed in a way so that the user easily understands.

1.3 Explanations of expressions used in this report

Main Interior Light Switch: This is the main switch with which the user can control the light functions. It has three options ON, OFF or AUTOMATIC. The ON position means that all lighting fixtures in the car are always active with full intensity. The OFF position means that all lighting fixtures in the car are turned off in all conditions. The AUTOMATIC position means that the intensity of the lighting fixtures is automatically controlled. Pictures and more detailed explanation of this switch can also be found in appendix: 1.

Clinics: Clinic is the name of a group (in this report four to six persons) of people that is put together to test and investigate a product or a proposal.

State Mate: State Mate is the name of the software development tool used in this report.

FRS: Function requirement specification. This is a document used at Saab Automobile to describe new functions in the car in an early state of the development.

Concept car: This is a car built by a car enterprise in only one specimen. This car is used to show some glimpses about what the future models are going to look like.

2. Theoretic frame of reference

<Theory, method choice and other products>

2.1 Software (State Mate)

This company (SAAB) does not (when this is written) design their software themselves. Instead they describe functions by documents to the suppliers. Of course, this always causes misunderstandings and problems when testing the function before it is ready. With a tool called State Mate it is possible to make models that can both be tested during the development and also act like a complete software and be tested in a car. It is almost impossible for the suppliers to misunderstand this description. Then the suppliers can decide if they want to re-write the program or just translate it to a specific program language. Another main reason for using State Mate is that it is used by GM, of which Saab is a part, and GM has a large library of functions already made in State Mate.

State Mate is a system devise tool that is built on UML (Unified Modelling Language). It can be seen as an object orientated program language.

2.2 User interface

2.2.1 Theoretical models

2.2.1.1 The influence of the users in the development process

When a user interface is designed you can let the user have different levels of influence during the design process. For example, Löwgren (1993) has divided the influence into four levels described in the following figure (Figure 2.1):

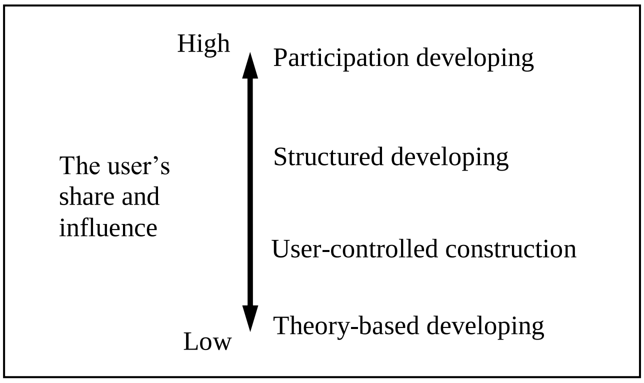


Figure 2.1: The influence of the users in the development process

Theory-based developing was the first method where human - computer interaction was taken into consideration. This implies that general psychological investigations of how the users behave with computers are used to shape the product. Facts and models that scientifically have been established are used as a basis to construct the product in the right way.
Next step in the development**, user-controlled construction**, was to make the users more involved and to find out more about their assignments and needs to be able to do analyses that could act as a basis for a prototype. The method is iterative and considered to be more practical and, therefore, it has more the character of an engineer's point of action than that of a scientific one. This is also why the progress has gone on to the following two methods.

Structured developing implies that the user partakes in a questioning where you examine his or her work and working environment to give the designer a schematic, structured understanding of the user's work.

The highest level, **participation developing** implies that the designer and the user mutually get to know each other. The designer learns how the user works and the user learns what the product could do for him or her.

2.2.1.2 Iterated user-involved designing

Bannon (1991) also claims the importance of having participation of an active user during the development process. Among other things he points at the consideration for the user. What is essential is to consider the user as an "active user", in other words, a person who wants to complete his or her ambitions to understand connections and is willing to go forward and further examine the product if it is not clear or does not completely fulfil his or her demands. If the system is not clear the user will try to understand it and if he does not understand build a picture of his own of how things work. (Bannon also adds that humans always look for logics in their environment.)

If you look at the users from this point of view you understand, according to the author, that the human-computer interaction plays a very important role in the design process. However, Bannon claims that it has been difficult to implement all the theories and the applied research that been made for this purpose so far. Therefore he means that you should do some changes in the subject human - computer interaction that can be helpful when trying to overcome the distance between theory, experiment, system construction and the work in reality. He also claims that you should pay the development process more attention. In other words, you must work with the user in every step in this process. You must also accept the iterative nature of the design and the always changing final goals, as a result of the development process.

These viewpoints consequently end up in two recommendations (according to Bannon):

- ⇨ **From "user- centralised" to "user-involved" development**
  One step in the right direction is to let the user test the product during the development process, but it still makes the user rather passive. The best thing to do would be to involve the user in the development, for example let the user join the development team. In this way the result would be a more democratic organisation change (creation of a new system) that insures that the system will fulfil the user's needs in a better way.

- ⇨ **From development of the user's requirement specification to iterative prototype development**
  Through the years it has been proved that the general way to represent the user's demands, in a functional requirement specification, often has not been enough. One has asked oneself if it depends on the method for the study or if it is a fundamental problem, that designers cannot in advance describe the user's needs and demands in a successful way by simple techniques of investigations and interviews. Bannon means that a user must have very much experience to be able to decide what a future system would look like and it will almost be impossible to produce a requirement specification that includes all future needs.

These two recommendations therefore advocate an iterative development process, where one as much as possible tries to involve the user actively in the development process.

2.2.1.3 The phase model

Sigfried (1993) describes the development process as a process divided into several phases. These phases are demarcated by documents. Each phase starts with a couple of documents and when a phase is at an end some documents are produced. When a process is determined, every produced document has to be approved, and if they are approved "they freeze" and the next phase can be started.

By putting several of these phases together you will get a "phase model" (figure 2.2). This model was developed as early as about 1950 and was at that time called a step model. After some time of badly adjusted and designed software this model introduced the following ideas:

a) Let us at first hand have a phase where the user can describe what he wants.
b) Let us think about the structure of coding before the real coding starts.
c) Since testing is difficult and time-consuming, take it into the planning from the beginning.

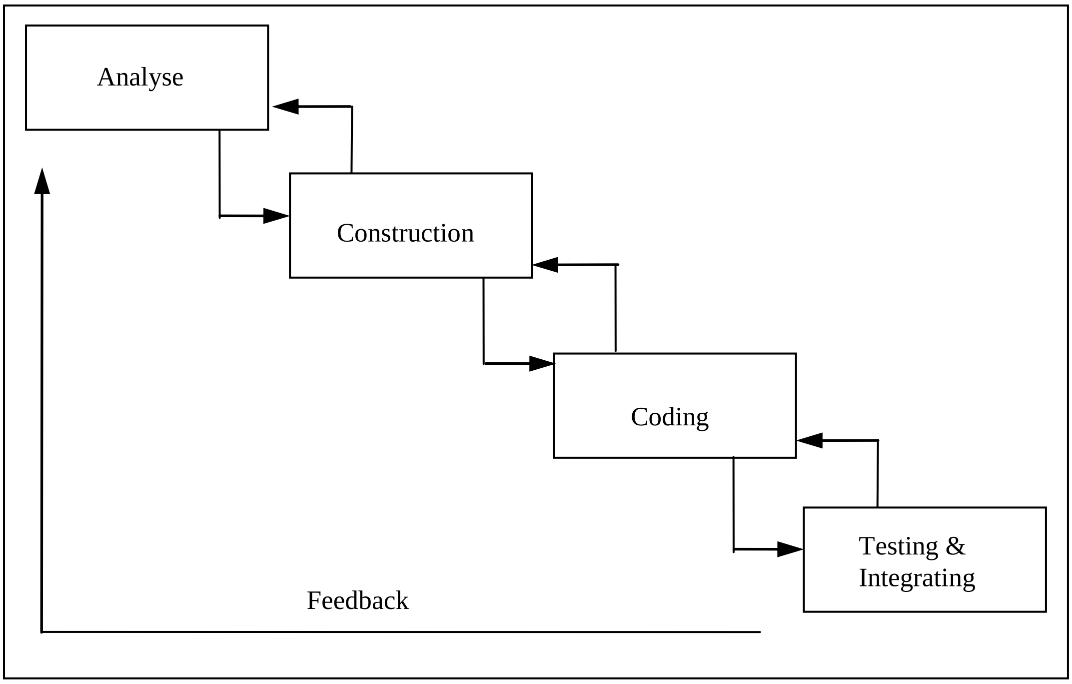


Figure 2.2: The phase model

In about 1970 this model was polished. Above all one understands that it is practically impossible to go through the different phases completely sequentially. It is a continual iteration between the phases as figure 2.2 shows. For example, when you are doing the coding you can discover defects that have their ground in the construction, that have their ground in the analyse phase etc. This feedback depends on the human nature and its restrictions; **the conclusion is that the human nature wants to work in an iterative way!** Humans do not want to and, in fact, cannot work in a predictable and sequential way. For example, humans are not shaped to follow the phase model, which would work in the best way if each phase were finished before the next one starts.

When one starts the problem-solving process, one often does not know the exact goal. More likely the human nature wants to test something and then, while learning, do adjustments and in this way the process continues.

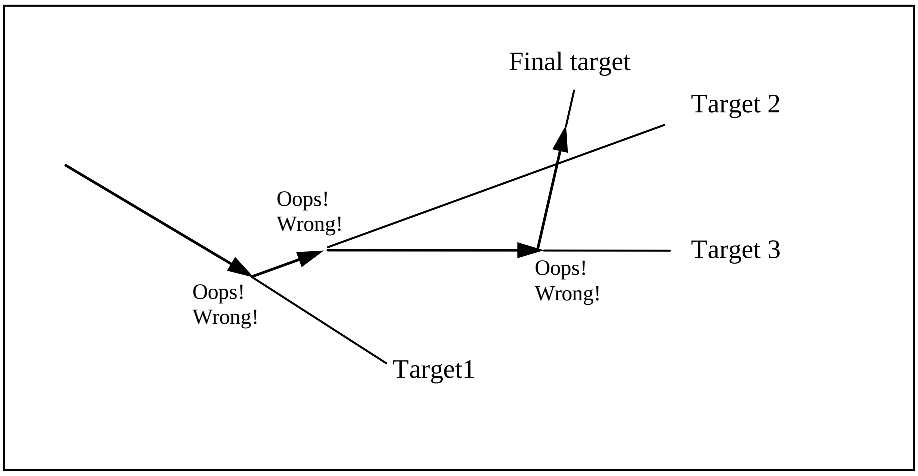


Figure 2.3: Humans prefer not to work deterministically, instead they want to create a solution by iteration.

The result of this is that **you have to have the knowledge: the human nature, presses the development process to be iterative** and by this, the programming method must consider this iteration.

2.2.1.4 Problems with reformulation of goals

R Groner, M Groner and W F Bischof (1983) write about heuristics and in connection with this about the problem with reformulation of goals. The problem-solving process should not go so far that the problem solver does not solve those problems that should be solved, but instead those that he is able to solve (but not should).

As a reaction an interim target, which is outside the problem area, gets an incorrect priority and replaces the target.

The basis of this phenomenon is that the problem solver has not got enough knowledge about the problem area. The problem solver stops to collect data or only collects data that fit his system of assumptions about the reality. This means that the problem solver will never get negative feedback; his system of assumption will get dogmatic.

2.2.1.5 Problems with the user-centralised program development process

Heinbokel, Sonnentag, Frese, Stolte and Brodbeck (1996) wrote an article about the problems that could occur in connection with the user-centralised program development process. Initially the two conceptions, **user-orientated development** and **user-participated development,** are separated and defined. The first conception mostly implies an understanding from the software developer towards the users, to have a positive idea of the user. The second conception implies that you organisationally place the user or the user representatives in the program development team. The authors say that the purpose of their article is to point out, that the question no longer is about choosing between involving the users in the development process or not. Instead the article wants to highlight and develop a realistic understanding of the problems that occur in connection with a user-centralised program development.

The main argument for user-centralised program development is very convincing. The program developers in general have very little knowledge about the area for which they will develop a program. Because of this they are unable to create useful tools for the user. Further arguments are that only the users know what is best for them and what they need. Because of that, useful software can only be made with the participation of the users. Consequently the functionality and the usefulness depend on an information flow from the user to the program developer. Since this flow is often complicated and the program designers often have not enough time, motivation or knowledge about the needs of the users, it appears unnatural to involve the users in the development process to secure this information flow.

This appears to be very contradictory when user-involved program development professionally is used poorly. What is this due to? One reason is that the user involved disturbs the development process. Further, designers complain that users do not have enough knowledge about computers and have not the ability to describe their own assignments. On the other hand the users often are not satisfied, because they feel that they have very little influence. Because of this the users think that they can reduce their involving in the construction part. The article further points out that there does not exist any established successful methods for user-centralised program development. Additionally there is no direct proof that a user-centralised program development guarantees a good result.

The investigations made in the article were performed by interviews of a couple of persons involved in different program development projects. To make the investigation practicable, the point of action was changed a little. Instead of investigating how the usability depends on the level of user-participation, tests were done to see if a smooth program development process is negatively influenced by a user-centralised point of action or not.

In the summary it is established that a clear pattern can be seen: The user-involving and the user-orientated development were negatively related to the program development process and the product quality. **User-involving development** was related to few successful results, fewer innovations, less flexibility and lower team efficiency. **User-orientated development** was related to high stress factors, fewer successful results and low team efficiency.

What is this result due to? The authors give several aspects of interpreting the result. Of these aspects the following are the most likely:

The investigation is built only on the developer's point of view
The variables of the level on a successful result were given by the developer themselves. All the data that the investigation were built on came from interviews with the members of the development team, and their points of view do not necessarily show the reality. The weakness of the investigation is that only the developers' subjective theories about the influence from the users in the development process have been taken into consideration. However, the interviews with the users have not been made to test if the program development has given a successful result.

User-involving and user- orientated development lead to more objective problems
The projects with the users involved have to take the blame of several problems related to user - program developer relations. At first hand the user showed ability to develop more sophisticated ideas during the program development for the solution of the programming task. The developers were not capable of predicting these ideas, which are often difficult to take into consideration in a later state of the program development process. This and several other factors make more objective problems occur, when user-orientated development and development with users involved are used. There are several interests that have to be taken care of, more relations to meditate on. Therefore the smooth working procedure and development process are a little "endangered" in a project when you start to involve user members, which is of great practical importance to the developers and their directors.

The authors, however, end the article by proclaiming that the results implicate that opinions that user-centralised program developing always should be positive, must change. The authors do not mean that user-centralised program developing from beginning to end is wrong but that the problem that has occurred during the investigation is a by-product of a user-centralised point of action. **User centralised-program development is perhaps a more uncomfortable way of developing software but despite that it is the best choice from the user's point of view.**

2.2.1.6 Integration of usability evaluation in the program development process

Hammond and Mc Manus (1991) write that the economic thinking in companies and organisations becomes more and more concentrated on benchmarking, and the demands on flexibility and capability to be able to answer to changing markets and needs increase. This appears as an understanding that the user-friendly availability is the key to increased competitiveness and better products. In consequence more and more companies and organisations have started to use tools for the development of human - computer interaction. In other words, one has realised that user-relied questions are not just of cursive art but, on the contrary, must be taken seriously.

Hammond and McManus investigate how a large-scale enterprise uses evaluations of user-friendly ability as a tool to change the organisation culture and to be sure that principles of user-friendly ability will be integrated in the daily software development process. It should be added that the enterprise has proportionally many persons involved in new programs and systems.
The enterprise investigated in the article uses a standardised development model, very similar to the phase model mentioned earlier, where each phase is iterative. This means that each phase could demand evaluation cycles and tests before it satisfies the user's demands and needs. Further back evaluation/testing was done only when all the tasks were "done". At this point the developer got a user-perspective for the first time in the process. This caused problems with adapting the software according to the user's demands and needs because it was/is difficult to make changes at this stage.

Inside the enterprise in the investigation, intern consultants are used who help to shape these evaluations at the end of each phase. The consultants gather software development teams and the right users (in number and position). After that the evaluations are done, with the software development team present when the users test the application. The users are encouraged to tell about their thoughts while they try to figure out how they should work, or if they are not able to solve the problems that they meet at the different states of the program. The users learn to express the problems they meet and the misunderstandings they reveal when they try to fulfil their tasks in the new system.

The opinions are collected by the development team and decisions are made which, how and in what order they should be treated. The changes are made iterative until the product works satisfyingly.

What makes this method of operation successful is that sharp evaluation goals are set and the members in the development team are involved in the development process. This will in the development team build a strong feeling of participation in questions that have a connection to user-friendly ability. This will then encourage to bigger and more constant changes in the attitude towards user-centralised software developing.

Additionally the users will be given an important role in the development process and they appreciate to be a part in the development of the product that they will later use. This will also lead to feelings of participation **that help the user to accept the new system or the new product that will be introduced.**

2.2.2 Ideas of model choice

The software development complexity was not at first hand on the coding or data-technical level but more in the analytical or the construction area. The usability and the functionality were the big area to handle. This implies that a close connection between the designer and the user is necessary to get a satisfying result. In this case the designer is also a user and that will simplify some things. However, a major reason for testing on several users is that a non-technical person may think and understand things differently than a technician. Also, when you work with a system for a long time you lose the impression that the system gives to a first -time user. However, the users also proved to have different opinions about what the system should look like, which also speaks for a close connection between the designer and the users. By involving the users in the development process and continually consider their opinions there will be a chance for them to see their opinions in future solutions and the results may have a better chance to satisfy more people than just the designer. It is always tempting for a designer to only consider his own wishes and to create, from his point of view, a perfect system but as discussed above it is not the best method.

These arguments spoke for a very user-centralised development process.

2.2.3 Ideas from other products

When designing a new product it could be useful to look at already existing products to see different possibilities, re-use good solutions and maybe take the advantage of using experiences from well-known products. It may also show ideas that are not working and so avoid some bad ideas. Not only car solutions should be taken into consideration.

3. Design of function, software and the user interface

<Mode of procedure>

3.1 Introduction

As explained earlier this part of the project "LED & Interior Lighting" consists of functions and software design and also a design of a user interface. All these parts were done with the usability as the central point of action.

The first thing that was done was to reflect on what should be included in the system, and when this part was ready we went on to think about how this should be controlled by the user. Considering the light designers' knowledge and together with that relevant notions plus that the system had been very complex, the idea was to protect the user against too many choices. Because of this we wanted the system to have as many automatic functions as possible, which, if right designed, would be positive from a user-friendly point of view. However, when structuring the automatic functions the user always has to be taken into account and the functions have to be well structured to be useful.

In the beginning an FRS (Function Requirement Specification) was made which is a document that is used on SAAB. **The FRS describes the function of the system in words, pictures and state charts. This document can be found in appendix: 1**. At the same time as the function was structured the next step was prepared. The question was to find out how the software should be made to work with software from other functions in the car. Therefore contacts with other departments had to be built up. It proved to be exactly the right moment of doing this work, because the Body electronic department at Saab was just starting a project that would introduce a new way to create software for Saab-cars. Body Electronic functions would also include the interior light function. In this way our project was included in this pilot project, which proved to be a perfect way of making the model.

At the same time clinics were to be established and research at Saab began to find out how this could be done. This project became a pilot project and a try to introduce this way of working at Saab. It was decided that some proposals should be made and with help from the market department clinics were made. The different proposals were later to be tested/evaluated in the clinics.

3.2 Function creation

First we decided which functions would be included and which of them the user should be able to manipulate.

The work was now divided into two parts. One part was started with the programming of the functions that should be automatic, and did so create a basis for the software. The other part was to develop the user interface that will give the user a direct conversation with the system.

3.2.1 Software development

3.2.1.1 Programming

To develop the software, the tool "State Mate" was used. Saab has not made their software themselves before; instead they have explained the functions to subcontractors who develop the software. Just in time with our project Saab began a pilot project to test the tool "State Mate". This was a perfect opportunity for our projects to work together (two pilot projects). This would also increase the chances to make our ideas a reality in a future car.

3.2.1.2 The automatic functions

The automatic functions for the interior light have several tasks. In almost every car today the automatic functions for interior lights are the same. These functions are (good and) well known but in this project there will be added so many new functions for the light that it could cause trouble for the user to understand all the new functions. The following things must be thought of when implementing new automatic functions:

- It should be easy to understand the logics of the automatic functions.

- The automatic functions should facilitate the applications and increase the safety in every situation when using the car in darkness.
- They should not take away the possibility for the user to decide the environment himself.

When the design of the new automatic functions started, the already existing automatic functions in the interior lighting for the car was attended to. To increase the understanding of the automatic functions as many as possible of the old automatic functions should be kept. Discussions with the light designer were held to decide what the wishes were.

When the framework for the automatic functions was decided, function structuring began. The first document that was created to describe the function was the FRS (Function Requirement Specification); this is a standard document used at SAAB and it was a good start to make a good structure of the work. **The FRS (could be found in appendix 1)** is a description of parts and functions in words and pictures as well as in state charts. In the following work the FRS was used as a basis and a reference. When the FRS was ready the modelling (software building) in State Mate began. This framework took form later under the modelling and got more and more clear and new opportunities but also obstacles turned up. All along the work with the model in State Mate the user was involved and the functions were discussed.

The work with the structure of the functions and the modelling later seemed to be the big and the most time-demanding part of this project. Due to the important work with these things, other parts in the project were somewhat neglected.

3.2.2 User interface

3.2.2.1 The development process

The method used in this thesis is summarised in figure 3.1. The figure shows a very user-centralised designing method where the user opinions and suggestions are taken into consideration at several levels of the designing process. The idea is to involve the users in the designing as much as possible without making them designers, which would make them think in a wrong way. The time to discuss with the users is also limited and it is important to get as much as possible out of these meetings with the users. Therefore some preparing was made before the meetings. To facilitate and to show the idea with the user interface for the new interior light system some proposals were designed, mostly in order to create something to build the discussion on.

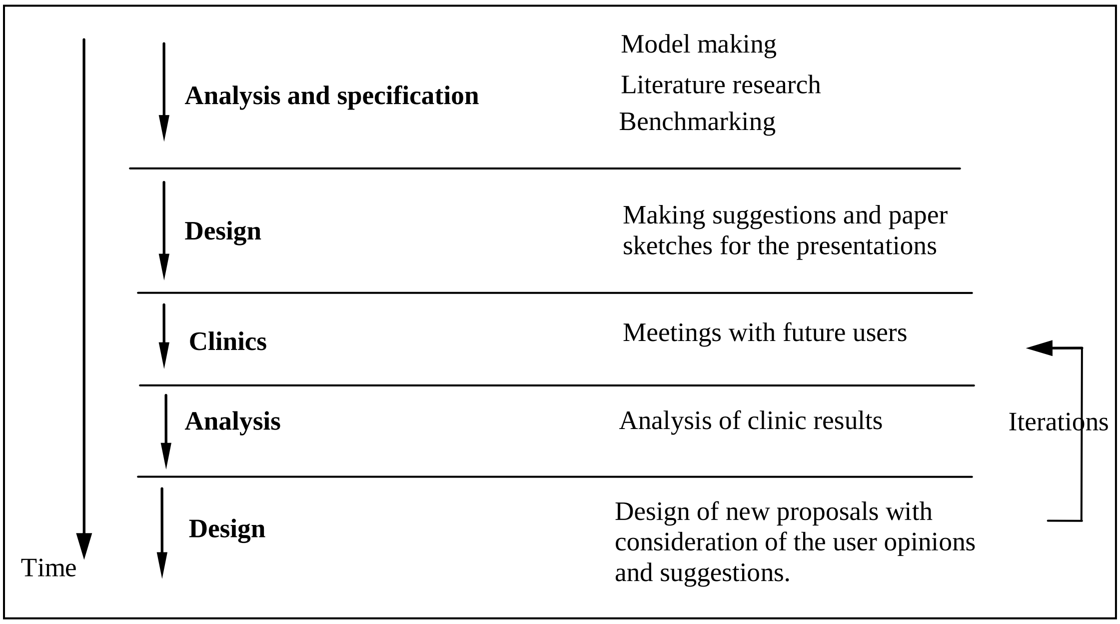


Figure 3.1: Overview of the user interface development process (the scale is not linear)

3.2.2.1.1 Proposals

A major reason for making proposals before the meetings with future users (clinics) is that they should help the clinic group to think in the right directions and, hopefully, come up with their own suggestions.

Six proposals were made and the thought behind these was that as many different kinds of user interfaces as possible should be included in these proposals.

When the proposals were being designed there were, of course, many things to be taken into consideration. In order to make good proposals, literature about user interface design was studied. This literature was later along the work always in mind. Competing cars and even other things such as cell phones and different kinds of computers were studied and concepts of their user interfaces were included in the proposals. This was done in order to get hold of and examine the today already existing good user interface design when testing the proposals on coming customers. To make the work more interesting even new ideas of our own were included in the proposals.

These proposals were tested in two groups with possible future users (the clinics). The proposals were explained on paper to the clinics. The reason for just making paper sketches for the first meetings was that the users would not get the idea that the development progress had been in process for a longer time and to give the users courage to give their opinions.

3.2.2.1.2 Clinics

These clinics on sub-system level proved to be a relatively new way of working at Saab. It seemed, in the beginning, to be rather difficult to find good non-technical clinic groups. The groups contained four to six persons and the discussions lasted for about two hours.

One of the groups had only female members and the other group had only male members. These different kinds of groups gave many different kinds of view that later were useful when the design of the user interface should be decided.

3.2.2.1.2.1 Preparing

Before the tests I made, as earlier explained, six proposals of different user interfaces were described on overheads. An important part of the preparing was to make pictures and find a way to explain the different proposals in a "fair way". Every proposal has to give the clinic group a right picture of how the ending product would work with the proposal. Every proposal also has to be explained in an impartial way to get the right answers. At the same time as the proposals should be understandable they have to give opportunities for new thoughts and suggestions from the clinic group.

The first try was to put an article in the newspaper "Saab today" to get non-technical people who were Saab employees so that we could expose the project and even re-use the groups.

We decided to start with an already existing group at Saab. This group is called "Saab's female reference group". This is a group of non-technical women at Saab who, on short notice, could be used as a reference group.

People from the market department in Göteborg constituted the second clinic group. They were also Saab employees with non-technical background. When the meeting later took place it proved to be a clinic with only male members. This was not bad because now we had one male and one female clinic group and so we could get both male and female opinions.

3.2.2.1.2.2 Performance

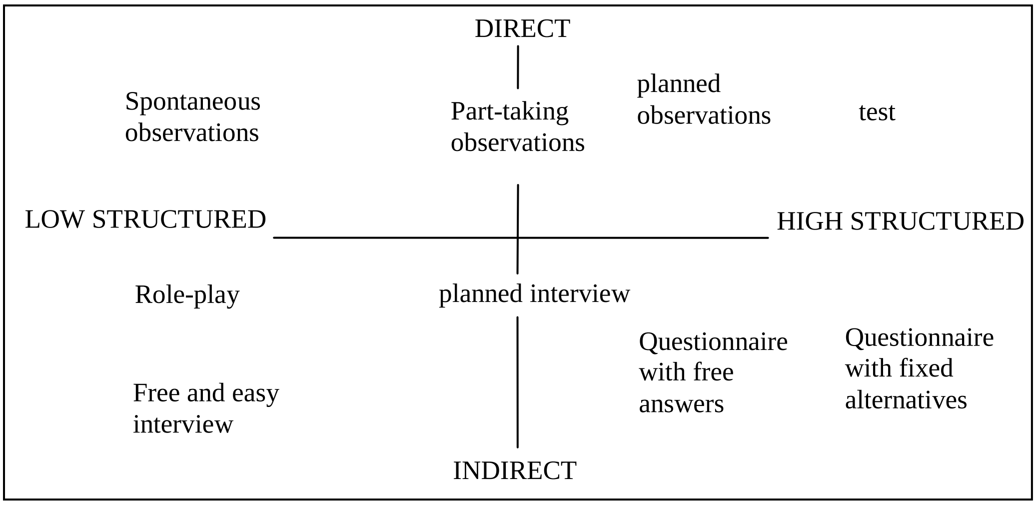


Figure 3.2: Interview methodologies

Figure 3.2 shows that there is a continuum of interviewing styles, ranging from direct to indirect interviews. The style that was used was indirect and had mostly the form of “Free and easy”. The idea was to make the clinics think themselves and hopefully come up with their own ideas. The big part of the interviews was made as a discussion about different kinds of user interfaces. Parts of the interviews could, however, be seen as highly structured because questionnaires about already existing solutions were given to the group members.

The tests began with an explanation of the purpose of this kind of tests and how the test would be done. After that the project was presented. The test then went on with a presentation of each one of the proposals and after every presented proposal there was a discussion and a documentation of what the clinic thought was good and bad things about the proposal. The documentation consisted of a video-camera to tape all reactions and notes on a big scratch-pad so that the group could see that we understood their opinions right. The group members also got questionnaires with questions to answer and space to express their specific opinions in their own words. They got one questionnaire for each proposal and then also one all-embracing form where they could compare and grade each proposal. The questionnaires are found in appendix 7.

3.2.2.1.3 Analyses

After the meetings with the clinic groups the work went on with making summaries of the results. The thought was that the positive thing in every one of the six proposals should be taken care of. This would shape two or three new proposals and, of course, the negative things that occurred on the meetings should be avoided as much as possible in the new proposals.

Therefore two lists were made, one with positive and one with negative points. The two most popular proposals were chosen and made a basis for two new proposals. The two old proposals were completely rebuilt with the knowledge that came out from the clinics. Positive things from the other four old proposals were included in the new ones as much as possible.

4. Results

<Results for every part that was described under the headline “Method” >

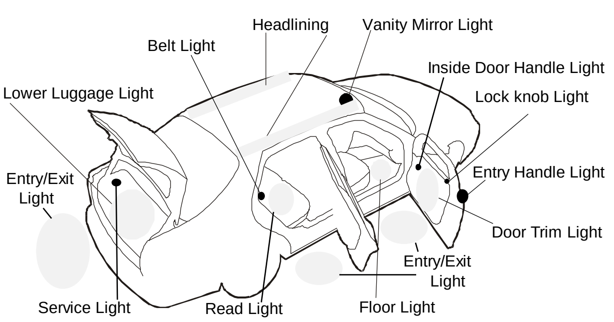
4.1 Complete system – requirement specification

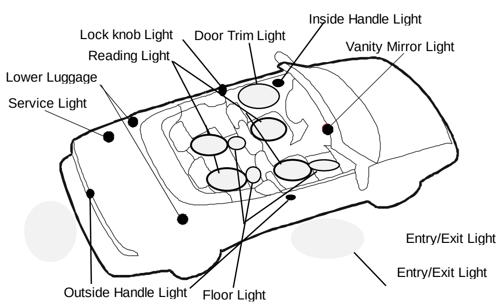
The interior lighting consists of several lighting fixtures placed in many places in the car interior. Depending on the position of the lighting fixture it should be able to change colour and intensity. The lighting fixture should have a function that facilitates the use of the car, increases the premium feeling in the car, embellishes the interior and makes the trip and the loading of the car safer.

To simplify the design, every lighting fixture has been given a name after its position and/or its function.

- Headlining - provides an impressive illumination of the front, the middle and the rear cabin areas and be located to support the function (The rear area is meant for the luggage in a wagon).
- Floor illumination - gives an impressive illumination of the footwell areas, front and rear. Placed not to dazzle the driver.
- Read Light - placed not to be seen by the driver and to give an inviting illumination of every specific seat when entering the car. May also be switched on separately and used as a reading lamp.
- Door Trim Light - placed to bring out colours of the door trim and to increase the standard of the driver’s environment.
- Inside Door Handle Light - illuminates the door handle when the belt is not on and the user is supposed to step out of the car.
- Lock knob Light - illuminates the lock knob, placed not to disturb.
- Belt Light - illuminates the belt connectors, placed not to disturb.
- Lower Luggage Light - illuminates the area under the luggage-blind in a wagon and the whole luggage in a sedan.
- Service Light - placed in the luggage to be used when serving the car, switched off and on separately.
- Tailgate Light - placed to effectively illuminate the rear outside of the car in a wagon when the luggage door is opened.
- Entry Handle Light - illuminates the entry handle in doors and the handle of the luggage when loading or approaching the car.
- Entry Light - illuminates areas outside around the doors and the luggage when loading the car both when the doors are open and closed, placed to support the function.
- Storage Light Closed/Open - provides illumination in the storage, effectively placed.
- Vanity Mirror Light - illumination shall be effectively placed, preferably adjacent to the mirror.
- Instrument Panel Light - provides illumination of all controls in the car.

The approximate positions of each lighting fixture in the car could, for different car models, be seen in the three following pictures:





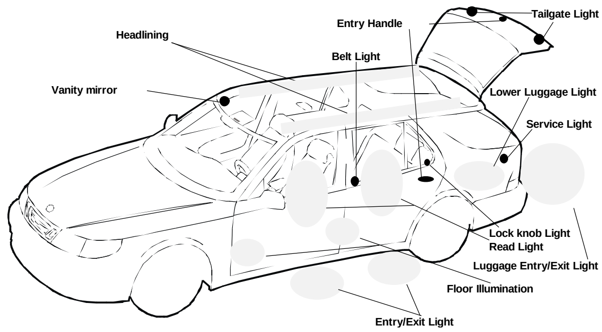


Figure 4.1: Approximately positions for the lighting fixtures in different car models.

These lighting fixtures are in their turn divided into parts depending on the position in the car, for example, headlining is divided into: driver’s headlining, front passenger’s headlining, rear right passenger’s headlining, rear left passenger’s headlining and luggage compartment headlining (the last one only in wagons).

Some of the lighting fixtures should be able to change light colour and some of them could always have a pre-set light colour. This depends on the specific lighting fixture function and can be read about in appendix 5 or found in the table in appendix 4.

Every state the car goes through, for example: “a door is open” or “the engine is started”, is divided into groups so that every state could have its specific illumination to serve the functionality, for safety and so on.

The different states which the car goes through is named from A to K and can be described as follows:

A to J are only in action when the Main Interior Light Switch is in Automatic position.

A:	Car is un-alarmed, no door open, no seat activated, and the engine not turned on.
B:	One or several doors open, no seat activated and engine not turned on.
C:	One or several doors open, one or several seats activated and engine not turned on.
D:	No door open, one or several seats activated and the engine not turned on.
E:	No door open, one or several seats activated and the engine turned on.
F:	No door open, engine turned on and car is moving forwards.
G:	No door open, engine turned on and car is moving backwards.
H:	Accident sensor activated, or Main switch is in On position and car is standing still.
I:	One or several doors open, engine turned on and car is moving forwards or backwards.
J:	One or several doors open, engine turned on and car not moving.
K:	Main switch in On position and car moving.

If the Main Interior Light Switch is in Off position all lights are turned off no matter what state the car is in.

The states A-K are explained in appendix 3 in a state chart. In appendix 2 you can find a table describing how each lighting fixture works through all the different states A-K. Each of the states A-K is in their turn divided into smaller groups, for example in the case of state B, which door is open and so on. How these smaller states are divided is not explained here because it would be too long and complex but it could be found in the model made in State Mate. This is the technical explanation of the interior lights function. If you want to know how the user is supposed to perceive the interior lights function you can see in appendix 5, which is the description by the light designer. This means that it has to be a very big understanding between the light designer and the software developer.

4.2 User interface

4.2.1 Market research

4.2.1.1 Car solutions

Of course both good and bad solutions can be found.

Figure 4.2: This picture (from a Mercedes Benz) is an example of a huge set of buttons at first sight it could frighten anybody.

Another good example of a huge cockpit-like set of buttons is found in the car model, Alfa Romeo 164, which was built in about 1985. In that time it was considered exclusive to have many controls as it gave an impression of many functions. Now we think we know better, because it is better to have few functions easy to understand how to use than lots of functions that we do not understand.

Figure 4.3: Two pictures from the same car (a Lexus) show a way to remove the first frightening impression that a huge set of buttons could create. In the picture to the left all buttons and the display are uncovered but in the picture to the right buttons for functions not in use are covered.

Figure 4.4: An idea to increase safety when controlling functions while driving could be to put the controls on the wheel as in this example (a Volvo concept car).

Figure 4.5: Mercedes has come up with a perfect and understandable system to adjust the seats. The buttons could relatively easy (by the user) be connected to the functions (in this case the electrical seat adjustments) that they are meant to control.

Figure 4.6: Two solutions from BMW. In the right picture you can see the right arrow pointing to a small display that has the structure of a menu with different levels. This menu can be chosen from the controller (that the left arrow points at) that can be screwed and pushed. The left picture shows a similar idea but this solution contains a big display and the controller cannot just be screwed and pushed but also dragged in all different directions. Both pictures show High-Tec solutions (which BMW is known for). The functionalities are good but it can, in bad cases, take a long time to learn them.

Figure 4.7: This picture shows a concept car from Mitsubishi and is a good summary of what many concept cars (cars with ideas of the future) from different companies are heading for. In this picture (in the right, lower part of the picture) the control is a huge handle. In many new cars on the market you will find many buttons and controls but in the concept cars there is one or just a few controllers and these are in many cases very big. Another example of this type of ideas in concept cars for the future is the Mercedes pictures below:

Figure 4.8: Mercedes concept car

All these examples are just a few of all those that had been under consideration during the development process.

4.2.1.2 Other technical things

Figure 4.9: Other product functions that had been considered are, among other products, the mobile telephones, mostly because they are well known and used by many people. They also include many functions in a small area and to be understandable they must be well structured. In telephones there are often also many voice functions included such as voice controls or voice instructions or even conversations with data-voices.

4.2.2 The functions

What the user should be able to do turned out to be as follows. The first decision for the user to make was the choice that also the old interior light contained and consists of the Main interior light switch (this switch and function are earlier explained in the introduction of this report). The next choice would be between different light environments and they are as follows (these light environments are all decided by the light designer and therefore only briefly explained here):

- **"Standard interior light":** This is a standard environment with light colours and light intensities.
- **"Nordic interior light":** This is an environment with a cold impression that would be preferable in a hot climate.
- **"South interior light":** This is an environment with a warm impression that would be preferable in a cold climate.

- **"User's choice of interior light":** This is a possibility for the user to choose his very own light environment with different light colours and light intensities for **every** different position in the car.

This is the basis for what the user will be able to control with the user interface. The big work is, as you can understand, to design the user interface for the user adjustments in the user choice of environment.

4.2.3 The proposals

The proposals in the first loop were as follows:

4.2.3.1 Small display

This proposal is a one-row display that can be manipulated by just four buttons. The idea is that this system should require very little space and not have so many frightening buttons. A user who also uses the very well known mobile phone "Ericsson" should easily understand the function because the idea is very much the same. With that in mind this will also test a system that several people recognise. Describing pictures can be found in appendix 6.

4.2.3.2 Specific keyboard

A specific keyboard tests the old push buttons and analogue screw buttons. It also tests the fact that a specific function has its own keyboard. The third thing about this suggestion is an idea of hiding all the buttons that are not in use at the moment. This will add the advantage of many controls but take away the frightening factor that a huge set of buttons creates. This will also give the user a chance to focus just on the function he wants to manipulate at the moment. The last thing that this system was created to test was the idea of pictures that give hints about the functions. Describing pictures can be found in appendix 6.

4.2.3.3 Icons and mouse

The suggestion with icons and a mouse tests the idea of likeness to a home PC. This system was supposed to be understood by everyone who ever used a PC. It would also test what a user thinks about collecting all the other different functions in the car (such as radio, climate control etc.) at the same place and create a kind of tree that you could go through to find what you are looking for. Describing pictures can be found in appendix 6.

4.2.3.4 Multi-control

This system is built on the idea that it should be possible to control all the functions in the car by one single multi-control. This controller could then be placed and shaped in a perfect way from an ergonomic point of view. This also includes a try to create a system that should be understood by a first-time user without needing a handbook but also be quick to use by an everyday user. It contains, besides the control, a display with the chosen selections and with the guiding text explaining how to control each function. Describing pictures can be found in appendix 6.

4.2.3.5 Speaking system

This proposal is an attempt to create a system that can be used safely while driving. It also tests what the user thinks about a voice talking to him or her and it should be able to manipulate different functions directly by buttons on the wheel. This suggestion also leads to a discussion about systems controlled by the user's voice. Describing pictures can be found in appendix 6.

4.2.3.6 Press display

This proposal was to test what a possible user thinks about a touch display in a car. Describing pictures can be found in appendix 6.

4.2.4 The clinics

Two clinics were done and the results could be seen in 4.2.4.1 and 4.2.4.2.

4.2.4.1 First clinic

This was the clinic with just female members. The notes from this meeting are summarised in figure 4.10 and the judgement of each suggestion that was made by each group member can be found in appendix 7. It can also be added that this meeting went on surprisingly well according to what was planned. The result from this meeting and the opinions from the group discussion can be seen in the following figure:

Concept	Assessment	1	2	3	4	5	6	7
Simple display	Positive	Few buttons	Short simple text	Likeness to the cell phone "Ericsson"	Many choices	Possibility to save ones own personal choices		
	Negative	Placement	Security when adjusting while driving	Slow to make adjustments	Mixed language between text and buttons	Bad name Nordic= hot?	Too many choices	
Specific keyboard	Positive	Few button- pushes	Easiness	Easy to understand and find the right function	No instruction necessary	Distinct.	Able to be hidden	Passenger has a possibility to adjust the functions
	Negative	Lock-function is missing	Something comes out. Does not look ready or space is missing	Can not change angles	Hard to read for the passenger			
Icons and mouse	Positive	Mouse could be a joystick	Likeliness to an PC-environment	Share place with other function	Close to driver			
	Negative	Long way to decision	Difficult for older people without PC skills	Where would the mouse be placed	Not safe to adjust wile driving	Escape button is missing	Bad likeness between the different pictures	
Multicontrol	Positive	Easy to use	Sort instruction on the display is an advantage	Shares place with other functions	Big display easy to read	Easy to reach by the driver	Can be made elegant	
	Negative	Takes big place	Long way to go before adjustment is ready	Analogue scale is to prefer	Old fashion	Gives a complicated impression	Escape button is missing	Information about the temporary adjustment is missing
Speaking system	Positive	Easy to reach	Safe to use when driving	Demands small place	Easy to understand, user friendly			
	Negative	Stressful	Nagging and boring. Long way to decision	This function should not be centrally placed	Needs language adapting			
Press display	Positive	Easy to use	No extra buttons.	Available for the passenger	Clean flat surface, easy to clean	Combination with other functions	The arrow to drag is good	
	Negative		Symbols would be easier to understand than text	Could be messy on the display	Escape button is missing	Headline telling you about your position in the system is missing	The arrow to drag is not good. + and - buttons would be better	

Figure 4.10: Results from the first clinic

The judgement by giving each proposal a grade showed that among these proposals the "Specific keyboard" got the highest grade by this group and on second place was "Point display". The whole list can be seen in appendix 8. The proposals were given grades from different points of view and the different grades can also be found in appendix 8.

4.2.4.2 Second clinic

This clinic contained just male members. The notes from this meeting are summarised in figure 4.3 and the judgement of each suggestion that was made by each group member can be found in appendix 7. This clinic was much harder to handle. The members had difficulties to find positive things in the proposals and had also a very strong opinion about how a user interface should act and work. Of course, their opinions were useful but if only this clinic had been used the result would have been misleading. The result from this meeting and the opinions from the group discussion can be seen in the following table:

Simple display	**Positive**	Likeness to the cell phone “Ericsson”			COMMENTS	Good function that can be combined with other functions but it should have a screw-button instead (easier and more secure in traffic)	Difficult to handle physically. The precision buttons are not good. Demands time to get used to it and demands too much attention when driving				
	Negative	Complicated with + and -, prefer as screw-button	Demands too much attention	Demands too much space							
Specific key-board	**Positive**	Direct choices	Simple logic	Effective	The picture is easy to understand	COMMENTS	Serve function good and is easy to understand but ugly = looks like a toaster	Takes too much attention from the eyes when driving	SUGGESTIONS	More clarity	Combine with touch display
	Negative	Demands much attention when used	A button that can adjust everything at the same time is missing	Different versions in different countries	Not premium						
Icons and mouse	**Positive**	Direct choices (quicker)	Likeness to an PC- environment				COMMENTS	The precision is not good	Dangerous, rotten and bad precision		
	Negative	Difficult to find in the system	Demands concentration.	Bad precision, difficult to manipulate the arrow	Difficult to handle with gloves on	Step function and support for the hand are missing					
Multi-control	**Positive**	The guys liked this kind of idea!			COMMENTS	Finally a screw-button but not good enough . Too many buttons	Too many buttons, bad precision . Add steps	SUGGESTIONS	Only screw and press button in the same button and add escape button	Add support for the hand	
	Negative	Too many buttons	Scroller is not good especially when the car is accelerating	The screw-button is too big and dominating							
Speak-ing system	**Positive**	Speaking system	Can be used while driving. Easier with a voice				COMMENTS	The sequences makes it annoying			
	Negative	Takes much time	Reverse button is missing	The loops get too long	Waiting time for the choices	The system is not intuitive					
Press-display	**Positive**	Easy to us	Fast	No sequences	Can be combined with other functions	COMMENTS	Hard to use when driving on a jumpy road	SUGGESTIONS	Pull function instead of dragging an arrow	Escape button to quickly return to the top level	Combine with a voice. Help function after a pre-set time
	Negative	Is not OK to use when driving	Hard to reach	The display will get messy							

Figure 4.11: Results from the second clinic

The judgement by giving each proposal a grade showed that among these proposals the “Multi-control” got the highest grade by this group and on second place was “Specific keyboard”. The whole list can be seen in appendix 8. The proposals were given grades from different points of view and the different grades can also be found in appendix 8.

4.2.4.3 General suggestions from both clinics

- Different solutions for different car models, for example, differences between a luxury equipped car and a standard equipped car.
- Share memory function with, for example, the seat adjustment and also with other personal settings.
- Fast response from the buttons or controls is important.
- More clearness.
- Possibility to read the display in strong sunlight.
- Helping voice after, for example, 10 seconds with no action.
- A demo show of the light functions.
- Possibility to create a personal light environment at home on the computer and later bring the personal settings to the car.
- Good to combine the user interface for the light function with other functions in the car.
- If the function shares, for example, a display with other functions the light function should not be at the top level.
- Different opinions about if the passenger should be able to reach the functions. The suggestion is that it should be possible for the driver to change the passenger’s possibilities to reach the functions.
- Possibility to make some adjustments from the trunk.
- The personal settings of the light environment could be memorised by every personal remote control for the car, so that each user always get his/her personal setting when arriving to the car.
- Possibility to pre-set if the entering light is activated when unlocking the car.
- Possibility to light up the entering light with the remote control. This function could be usable if children are playing around the car or if some burglar is sneaking around the car.

4.2.5 New proposals (second loop)

The three new proposals for the second loop were as follows:

4.2.5.1 Combination of “Point display” and “Specific keyboard”

This suggestion has the “Point display” and the “Specific keyboard” as base alternatives. The thought is to take the idea of the “Specific keyboard” but place the buttons on a display that can be hidden in the instrument panel.

4.2.5.2 Combination of “Multi-control” and “Speaking system”

This is an attempt to develop a new and better “Multi-control” from all the new inputs from the clinics. It will also be combined with a helping voice. The thought is to fulfil the wish of the second group about a perfect multi-control liked by the user.

4.2.5.3 Voice controlled system

The voice-controlled system tests how the users react on using their voice to control the functions. This could create a possibility to make some of the adjustments when driving. When bigger settings are made you can stop the car. This also tests if the user is ready to dispense with bigger adjustments while driving just to make it more secure in traffic.

4.3 Programming

This part of the thesis could be hard to understand for persons who are not familiar with programming. Therefore this part is written for a person interested in software and interested in using State Mate for software development. The following part is a short and a little more simplified explanation of the software than the document in appendix 9, where the software is completely explained.

The state charts in appendix 3 and appendix 1 and the table in appendix 2 were the basis for the design of the software. All the software was, as earlier explained, made with the tool “State Mate”.

The software for the interior lighting is a part of the whole software that manipulates all the functions in the car. Our part of the software (made in State Mate) is therefore called “Interior Lights”. The connection with the other functions is shown in figure 4.12.

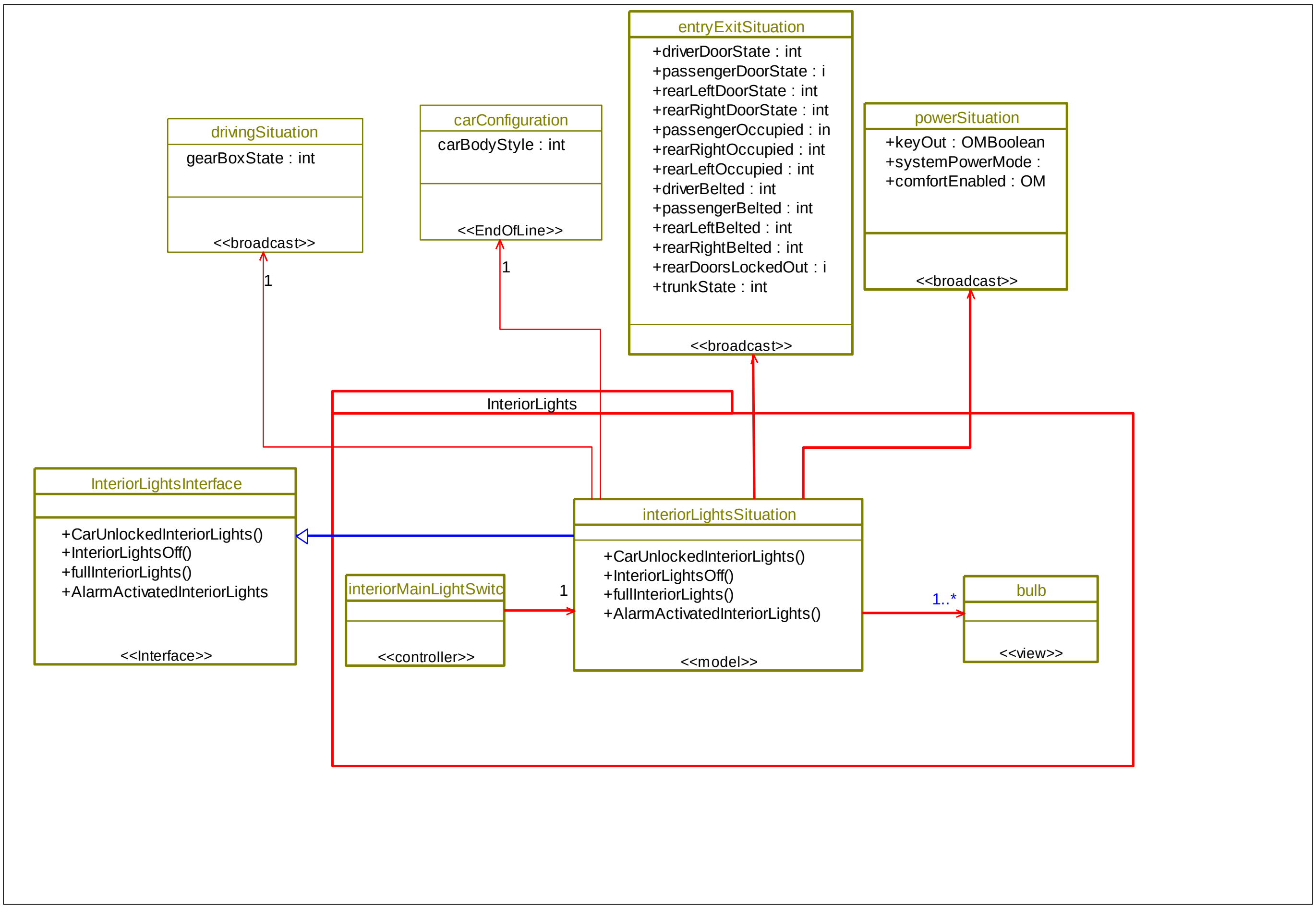


Figure 4.12: High level model of State Mate Interior_Lights

As you can see (if you have some knowledge about object-orientated programming) this function (Interior Lights) has not got any output that will be input for other functions, but on the other hand Interior Light has many inputs that come from other functions. On a high level the Interior Light function can be viewed, as shown in figure 4.13.

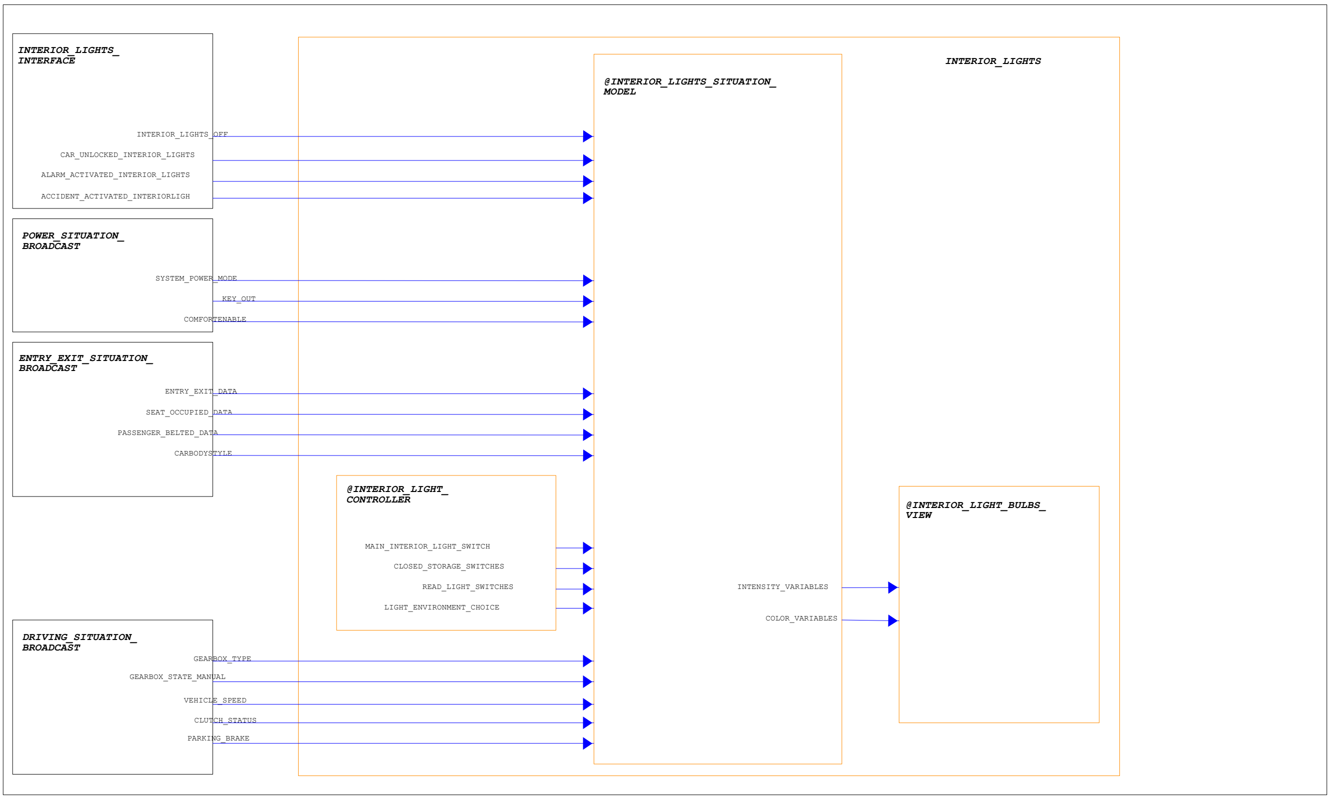


Figure 4.13: State Mate chart Interior_Lights

In the box “INTERIOR_LIGHTS_SITUATION_MODEL”, all the data from the user and other functions is considered and handled in the right way to create the right functions. This part stands for the main part in the Interior light functions. The controller “INTERIOR_LIGHT_CONTROLLER”, contains the interface towards the user and handles the data to make them understandable to the program. The view “INTERIOR_LIGHT_BULBS_VIEW” is the last handle of the data in the program. In the view the variables of light intensity and light colour are rebuilt to fit the output signal. The view also takes care of making the change in light -intensity and -colour of a tuning type, in other words, so the light does not just switch for example intensity in one second but with a grading delay so that you cannot notice a distinction between two different environments.

In figure 4.14 you can see the high level construction of the Interior Light in State Mate. The box “LIGHT_ENV_MANAGER” can be seen as a processor for the Interior Light function. This “box” decides among other things, how and when the different light environments A-K (earlier described and shown in appendix 2) should be active, one at a time. It also handles specific actions such as if the user switches on the reading light. The specific light environment (in appendix 2) is described and created in each of the boxes “LIGHTENVIRONMENT_A” - “LIGHTENVIRONMENT_K”. In other words, when the “LIGHT_ENV_MANAGER” box has decided which environment A-K should be active, then the chosen box “LIGHTENVIRONMENT_A” - “LIGHTENVIRONMENT_K” describes what this specific light environment looks like.

The box “DOOR_AND_PASSENGER_STATE” takes in and handles information about the door states and the passenger states, in other words, which door is open and in which place in the car a passenger is placed. It also handles the alternatives of manual or automatic gear box and contains the situations alternatives of wagon, sedan, two doors coupe or convertible.

The box “INIT_ALL_VARIABLES” handles, as you can predict, the initialising of all the variables that handle light intensity and light colour.

The box “NO_ACTION_SITUATION” describes the condition that acts as energy saving condition in the case when the light is on but the user is not present.

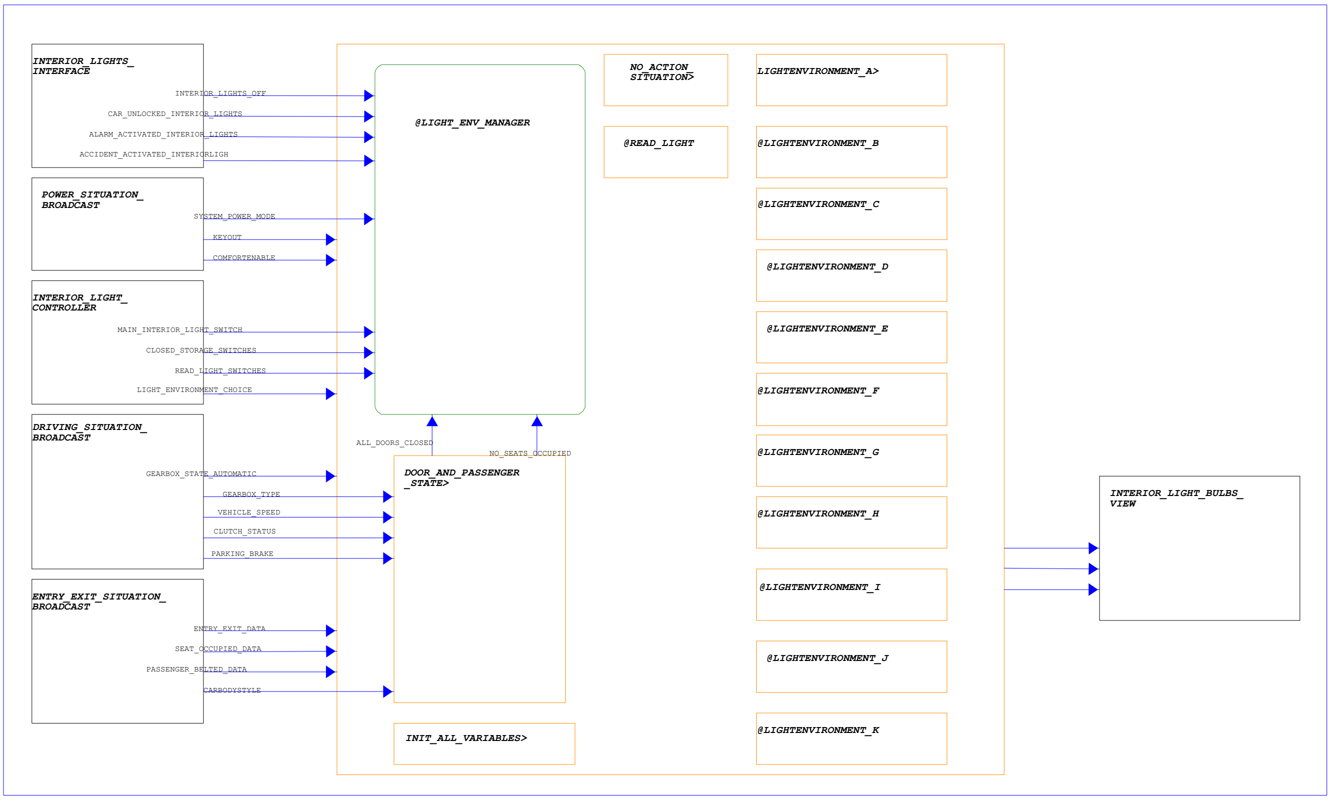


Figure 4.14: State Mate chart Interior_Lights

The contents of the box “LIGHT_ENV_MANAGER” can be seen in figure 4.15. To get a good structure the “processor” is divided into groups (states).

The first state “INTERIOR_LIGHTS_INACTIVE” is in general a description of the condition when the car is not used and the alarm function is active.

Next state “INTERIOR_LIGHTS_ACTIVE” is a state that handles the regular use of the car. This state is also divided into under-states that handle three big areas when the car is used “APPROACH_AND ENTER_CAR” “MOTOR_STARTED” “CAR_MOVING_ENV”. These three boxes handle the areas that are described by each name.

The other boxes “ENV_H_ACCIDENT” “MAIN_SWITCH_ON” AND “BOXES_VANITY_AND_READLIGHT” are in the following order handling the situations of car involved in accident, Main switch is by the user set to On and the user uses vanity mirror or reading light or opens a closed box applied to the car.

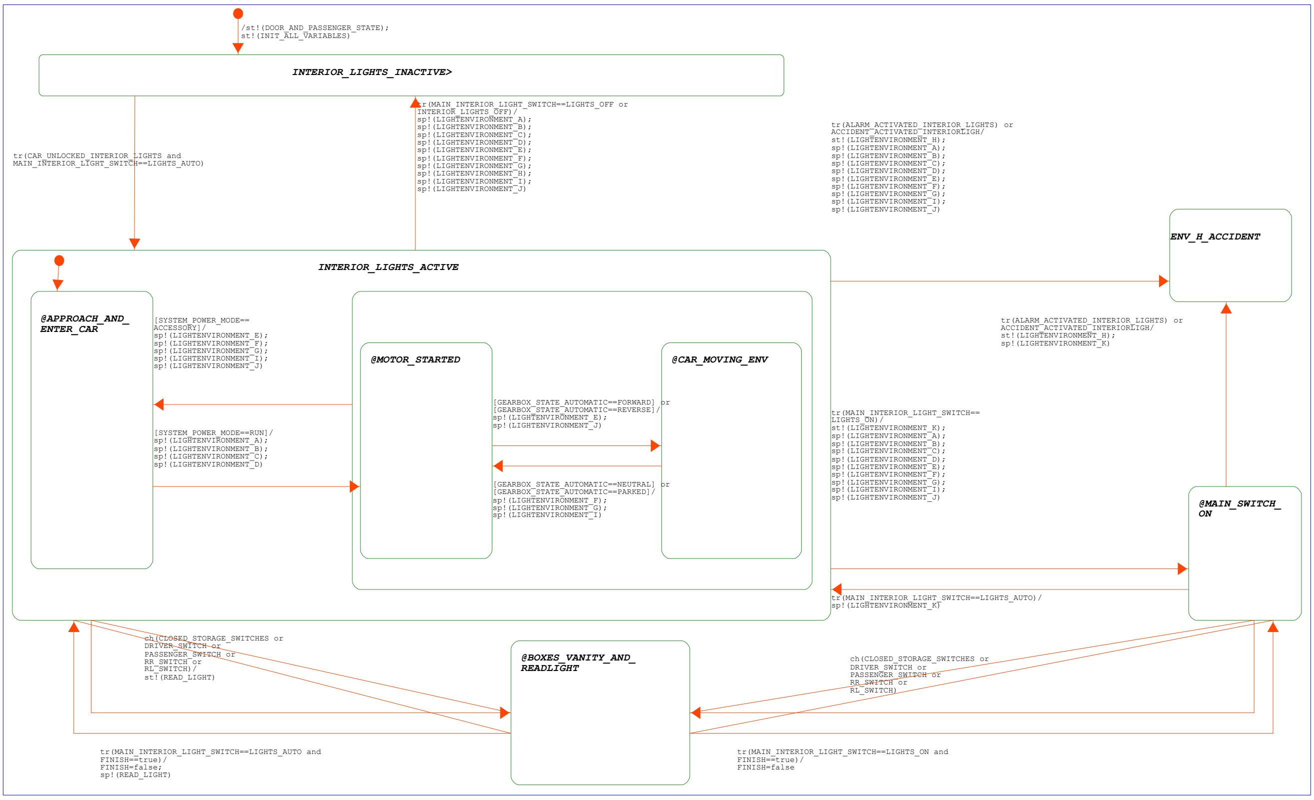


Figure 4.15: State Mate chart Light_Env_Manager

Each of the boxes in figure 4.7 of course has its own content. For example the box "APPROACH_AN_ENTER_CAR" has the content that is viewed in figure 4.8. Here is the final decision of which environment A-K (Appendix Y) should be active at just this specific moment. Each of the boxes in figure 4.7 has a content similar to the one in figure 4.8. This content can also be found in appendix 9 (the State Mate model).

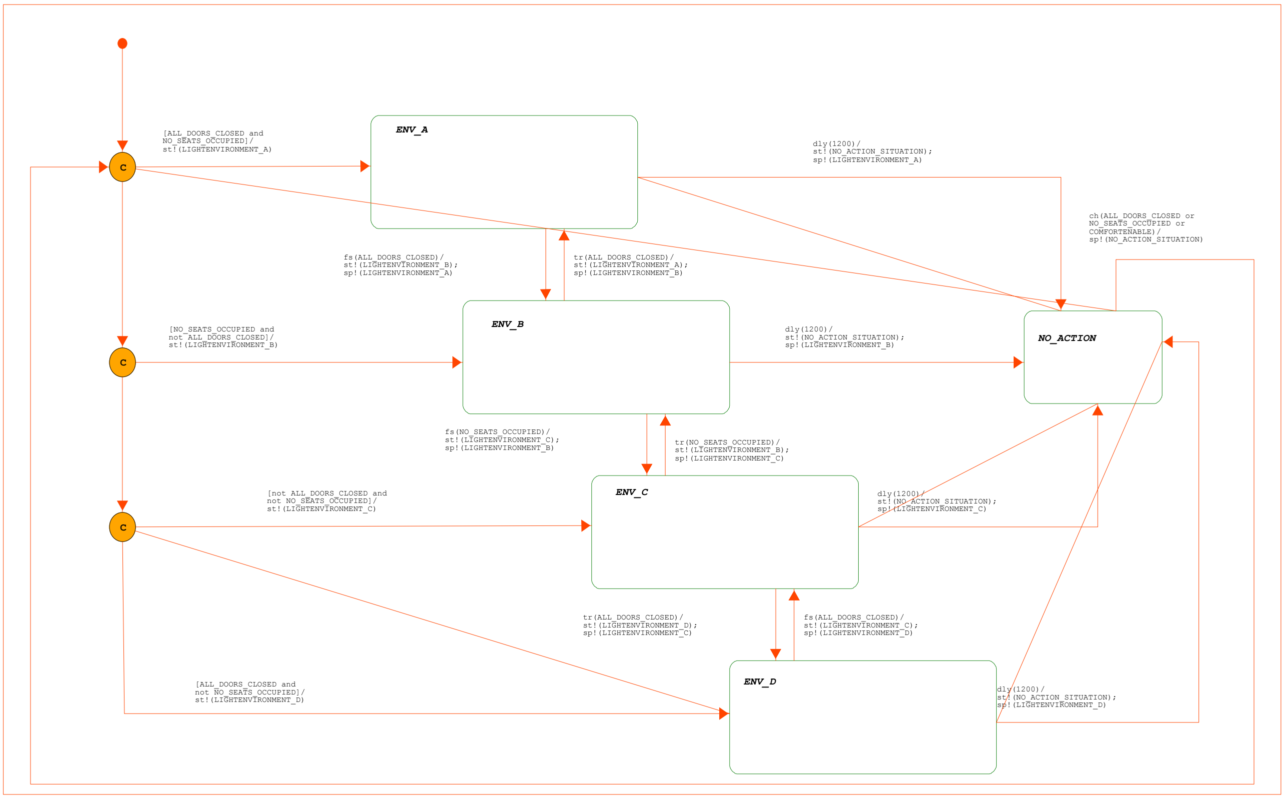


Figure 4.16: State Mate chart Approach_And_Enter_Car

All codes that handle the flow between the “boxes” and the code that is behind every box is done in the program language C. But that code has not been included in this report because it would be too extensive. Moreover, this code is not of interest in this report. What has been explained above is only the model, to show the structure. For the controller and the view you are referred to the model made in State Mate. If you are interested to see the whole model including the code etc. see the model in State Mate.

Abbreviations useful when studying the model are found in Appendix 9.

5. Discussion

<Comments to the results, answers to the issues and my own conclusions and opinions>

In the introduction of this thesis the idea of this project was presented. The idea contained a good working function for the interior light also described with a working software and a user interface that would be a helping hand for the user to be able to make some of his own choices about the light environment in the car.

The software should be well structured and useful for future interior light design. The functions should be designed in a way that increases security, facilitates the use of the car, and gives an exclusive, welcoming and comfortable look. It should also be able to act as a helpful tool for the light designer to, proceeding from the new lighting fixtures, design a good light environment by testing different light colours and intensities on each lighting fixture in the car. The idea was also that the light designer should be able to save his adjustments in the software without having any knowledge about the tool State Mate. You could say that the software should be able to support the light designer to create good light environments at all different states.

All the new functions that were to be implemented could create a problem to the user. This problem had to be resolved by a well-designed user interface. The idea was that this system should facilitate for the user not to add any problems when trying to understand the functions or how to control them, which was not an easy problem to solve. Often added functions create new problems to understand the system.

Because of the wideness of this project the complete development process was not finished. However, the function was completely structured and the framework of the software was almost complete with the exception of some small bugs. The model is also able to read a text file containing inputs about variables such as light intensity and light colour written by the light designer and given the output that the light designer prefers. The user interface development process has created useful data for future user interface design, but the development process intended and structured in the introduction was not finished.

5.1 Automatic functions

The automatic functions were the comprehensive problem that was to be solved. All the new functions that were implemented had to get a good structure in order to be understood by all the different persons who were involved in this project. The functions also had to have a structure that increased the user-friendly ability.

In my opinion the function has a structure that serves the development process well. However, if time had not been a problem the structure for the development process could have been increased even more. To be able to get a comprehensive view of this complex system it had to be divided into well-defined groups. This is the standard way of solving problems that often proves to be a good way of action.

The automatic functions were unfortunately not tested in a clinic because of lack of time. The opinion of the respondents (the designers) was, of course, that these functions were structured in such a way that it increased the security, facilitated the use of the car, and gave an exclusive, welcoming and comfortable look to the design. In this case the designer (the respondent) is also a user of this kind of product, which facilitates the design and gives a better credit to a good product. However, you cannot rely on only one user to get a good product and the opinion is far from always the same between different users. In this case it can also be important to look at other functions in Saab cars. This is important because there is a wish to form all views to keep the Saab image in the car and this also includes functions. Saab is especially known for cars with special and smart solutions and this has also been considered in the design and the structure of the functions. However, the project is not far away from being able to make clinics of the automatic functions. Thanks to the model made in State Mate the next step in the development process should be to connect the model to an existing test-rig (a test-rig that is used by Saab to test all kinds of different body electronic functions) where you could test how the functions work in a realistic manner. If bad functions should be exposed later when a test is made in the test-rig it is not a big problem to change the functions by using the tool State Mate.

5.2 Software

In my opinion the software is the most successful part of the project because the tool State Mate proved to be a very useful tool. This project and the model are, hopefully, a good help when the time is coming to make the decision whether the product should be used or not in a car in production. I mean that the tool State Mate is a very good tool to use when developing as well as testing a product with the user involved, because it is easy to change in the model of the functions during all the development process and even when a system is completed. It is also very easy to simulate the functions and to find bugs and exactly where the bugs appear. The simulation can also be made in a very illustrating way both on the data-screen with simple pictures and in a test-rig. In other words, the model both simplifies the development process for the function designer and the light designer and makes it easy for the clinics to understand what the designer means. It also simplifies the conversations between the user and the designer.

5.3 User interface

The user-centralised development process was in this project facilitated by the fact that the designer is also a user of the product. However, both technical and non-technical users use the product. This was a good reason for involving also non-technical users in the development process, because technical and non-technical users see things from different points of view. Therefore it was important to consider both sides.

The experience that the designer had was that this kind of development process is both fruitful and funny. Different people proved to have very different opinions and ways of looking at things. Some ideas or opinions from the clinics could be implemented in the product right away but other ideas or opinions proved to fight against each other.

An example of different opinions was that the female clinic wanted a simple and understandable user interface that is good for a first-time user or when not used so often. The male clinic had opposite opinions and wanted a high tech “multi-control” function that maybe would be difficult for a first-time user or when not used so often but would be liked by an everyday user.

An example of different kinds of looking at things is a comment from the female group that “the interface should be easy to clean”.

It is impossible to create a system that is liked by every user. A step in the right direction could be if you could create one system for male users and one for female users. In the future maybe it could be possible to create two software systems that could be changed between each other. However, at this state the best thing to do is to do a good compromise between opinions.

A Swedish saying is “too many cooks spoil the broth” and if you do not handle the ideas and the opinions from the clinics in the right way that would be the case. After all, in the end the decision is up to the designer to, proceeding from all the opinions and ideas, design a good product.

The experience from this project is also in harmony with the standpoint of Heinbokel (1996), that the user-centralised developing is perhaps a more uncomfortable way of action (it demands more time) but the best way from the user’s point of view. Nevertheless the respondents’ opinion is that a user-centralised point of action is also very interesting and funny.

5.4 Work to be continued

The idea was that, after this analysis, two new proposals should be tested that were made out from the results given from the analysis of the first six proposals. A new analysis should later be made of the new test, which would give one new proposal that should be tested, then modified after test-analysis and this would be the resulting user interface. Due to lack of time no more tests were made than the two first ones, proceeding from the six proposals.

References:

Articles:

Hammond, J. & Mc Manus, B. (1991). Integrating usability evaluation with systems design. Human aspects in Computing. Vol. 2: 1, pp. 733-7.

Heinbokel, T., Sonnentag, S.,Frese, M., Stolte, W. & Brodbeck, F. C. (1996). Don´t underestimate the problems of user centredness in software development projects – there are many. Behaviour and Information Technology. Vol. 15: 4, pp. 226-36.

Books:

Groner, R., Groner, M. & Bischof, W. F. (1983). Methods of Heuristics. Hilldale, NJ: Lawrence Erlbaum Associates.

Kuutti, K. & Bannon, L.J. (1991). Some Confusions at the interface: re – conceptualising the "interface" problem.

Löwgren, J. (1993). Human – Computer Interaction. Student literature.

Sigfried, S. (1993). *Understandning object-oriented engineering*: *a practical approach*. Stockholm: Industrilitteratur.

Wilson, J.R. & Corlett, E.N. (1990). Evaluation of Human Work. A practical ergonomics methodology. London: Taylor & Francis.