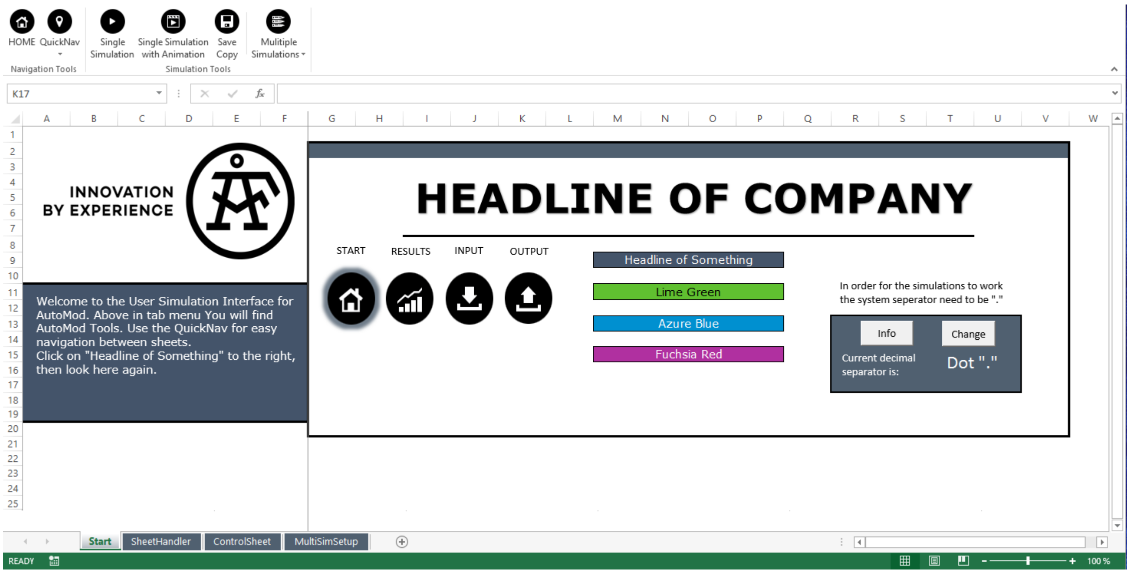
CHALMERS
UNIVERSITY OF TECHNOLOGY



Welcome to the User Simulation Interface for AutoMod. Above in tab menu You will find AutoMod Tools. Use the QuickNav for easy navigation between sheets.
Click on "Headline of Something" to the right, then look here again.

In order for the simulations to work the system seperator need to be "."

Current decimal separator is: Dot "."

Generic User Interface For Simulation

Master's thesis in Production Engineering

ROBERT MALMGREN
LEO ADELBÄCK

Department of Product and Production development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's thesis 2015

Generic User Interface For Simulation

ROBERT MALMGREN
LEO ADELBÄCK

Department of Product and Production development
Chalmers University of Technology
Gothenburg, Sweden 2015

Generic User Interface For Simulation
ROBERT MALMGREN
LEO ADELBÄCK

Supervisor: Linus Bördin, ÅF Logistics & Manufacturing devlopment
Examiner: Cecilia Berlin, Product and Production development

Master's Thesis 2015
Department of Product and Production development
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Simulation interface for AutoMod in Excel created with Visual Basic.

Typeset in LaTeX
Gothenburg, Sweden 2015

Generic Simulation Interface
ROBERT MALMGREN
LEO ADELBÄCK
Department of Product and Production devlopment
Chalmers University of Technology

Abstract

This thesis investigated how to create a new user friendly generic simulation interface for production simulations to replace an old interface solution. Several interviews were conducted to establish the interface user's needs. The interview results were processed with the affinity method. The affinity method revealed the most important user needs for the interface. The needs were translated into requirements that were used to create the generic simulation interface. The interface was developed in Excel and written in visual basic. User needs and literature on the subject of usability led to improvements to the new interface compared to the old. The new features include multiple simulations with storing statistical data of each run, a flexible navigation system, standard colour schemes for distinguishing information types. Collecting the needs from the users is key to creating a successful interface. The new interface created in this project considers many usability aspects found in literature. The aspects considered were those brought up by the user's needs.

Keywords: Production, Simulation, Visual Basic, AutoMod

Acknowledgements

We would like to thank our examiner Cecilia Berlin for guiding us through this thesis. Always giving excellent feedback and more importantly showing she cares for our results.

A special thanks to our tutor Liang Gong who helped us find literature at the beginning of our project. His recommendations really helped us navigate our project in the desired direction.

We also thank everyone from ÅF that we worked with for this thesis.
For lending us their time and always showing us support. Specially our supervisor Linus Bördin for teaching us many new things and for excellent discussions.

We also would like thank the other thesis workers at ÅF, Gustav Polhem and Rozle Penca. They helped us out with excellent feedback and support. Working closely next to them has been a real enjoyment!

Thanks to our friends and families for supporting us through our education. Without them the time spent at the University would not have been possible.

Robert Malmgren & Leo Adelbäck, Gothenburg, August 2015

Contents

List of Figures

List of Tables

1
Introduction

Simulating production systems is becoming more and more common. The possibility to simulate in a virtual environment as much as possible before testing it physically, in both product development and production development can lead to cost savings and more reliable systems. Changes can be made much easier virtually and running tests is much simpler. Several solutions can be tested and evaluated before having to implement them. Existing production systems can be modeled and then simulated, and future investments can be better investigated through simulation. Managing a simulation can be trough an interface, this project is about creating such an interface. This project uses a general software development method and literature on usability to build the interface.

1.1 Project Description

This project's main goal is to have a working communication between Microsoft Excel and AutoMod[1] without using ActiveX[2]. It is also to build generic interface to use as a template when building new client and model specific interfaces. The consultancy company ÅF in Gothenburg Sweden develop simulation models in AutoMod for their clients. Currently they develop new specific interfaces in Microsoft Excel for each client with respective model, where ActiveX functions are used for the communication between the model and the interface. ActiveX functions are used to start AutoMod and run the simulation from Excel, so there is no need to start AutoMod manually. AutoMod has recently been released as a 64 bit version where these ActiveX functions no longer exist. To minimize the time for ÅF employees to develop models with specific interfaces, this project will aim to build a generic interface, for example sheets and matrices are generated generically by the interface with user-specified parameters such as tab names and size of arrays..

1.2 Purpose

This project's purpose is to develop a generic user interface (generic-UI) for Automod simulation models that will work with the latest AutoMod version. Therefore the main goal of this project, is to have a working communication with 64-bit Au-

[1] AutoMod is a logistics and production flow simulation software.

[2] ActiveX is communication platform used in programming for sharing information between applications.

toMod and Excel. The interface will be developed in Excel with the purpose to provide simulation models with input data, present output data and the possibility to start and stop the models. A new communication solution between model and the generic-UI will be developed since the current solution will not work with the 64-bit AutoMod version. The development of the interface will include cognitive aspects and user friendliness for both modelers at ÅF and their clients.

1.3 Limitations

The generic-UI will be developed to work as a general template for any client interface. In this project the testing in the development process will mainly be on a model of Company A's production and to satisfy their requirements of an interface. This project is a standalone and not a project to develop a production model for a company which will be referred to as Company A, but to cooperate with another project that develops a production model of Company A's factory in Russia. The project time span is 12th of january 2015 until 12th of july 2015. Working 40 hours a week.

1.4 Risks

Since part of this project is to run tests on Company As production model there a risk that this model might not be completed in time or insufficient for the tests. If this is the case other already existing models can be used for the testing of the generic-UI. Further development after this project of the generic-UI can be necessary to ensure that it works well enough for use on the Company A-model.
This project involves extensive work regarding gathering information to generate a concept that fulfills all the needs from the different parties that will use the generic-UI. There is a risk that the concept work will overshadow the coding process. Even if the coding will not be completed the project aims to develop enough conceptual designs in order for future project groups to continue the work.

1.5 Research questions

Some questions that this project aims to answer to aid the development of the generic-UI are shown below.
These are likely to be changed as the knowledge base increases in the project group over time.

- How does the current interface solution work?
- What is laborious about the current interface?
- What are the characteristics of a good user interface?
- How can a user interface be cognitively supportive for the development process of client specific interfaces?
- How should data be presented that is relevant for the clients and to satisfy their needs.

2
Theory

Designing an human machine interface (HMI) involves three disciplinarians in this project, design with usability, programing and simulation. There are several methodologies and theories on how to do this. This section is about the theories and methodologies used for HMI design.

2.1 Relations in the development process

The relation between the developer and the user is important to consider in a development process. The role of the developer in this project is to develop a generic-UI for AutoMod modelers at ÅF. The generic-UI should serve as a tool for the AutoMod modelers when they develop user interfaces for their clients. Hence the importance to gather requirements from both the modelers and their clients in the development process of the generic-UI. The relations in the development process is illustrated in figure 2.1.

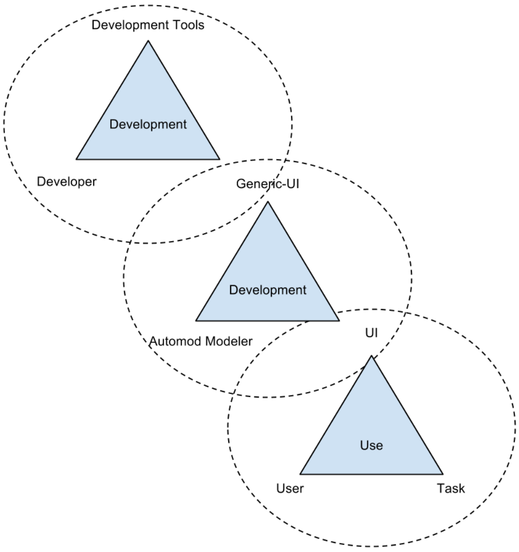


Figure 2.1: Relations in the development process

2.2 The Work Cycle

This project needs a model that uses the feedback from users early. This is so that changes for the better can be made during the development. Therefore an iterative work cycle seen in figure 2.2 is best suited for this thesis. It is based on the models from Hull [11] and the methods in [2]. It is a general interpretation of how software development is applied.

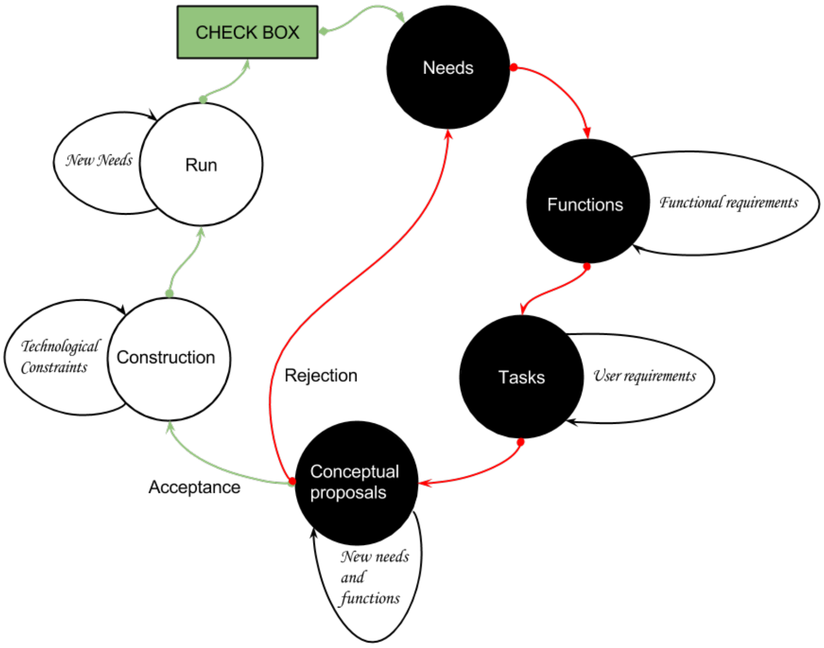


Figure 2.2: The work cycle is based on the chapters 1-6 of [2]

The model seen in figure 2.2 is an iterative process that helps this project to focus on the early stages of development and not jump to solutions. The red line represents the first iterative process used to generate a good concept that will fulfill the needs of the user. The green line is the second process when the actual system is constructed. The first thing to consider is the needs. This step is what lays the foundation, all the needs of the user should be brought forth. Not all needs will be found the first time, hence the iterative process.

Collecting needs in software development it is preferably done with interviews and observations of the users, how they use current solutions or alternatives. Also some sort of methodology is needed to organize and find the needs that are important. The Affinity- Interrelations method is used for this, it will be explained in detail later on in section 2.4. The needs are then translated into functions, how these functions will work will be stated by the functional requirements.

This could be on how a function fulfills its purpose, it may be necessary to do it in a certain way and not just fulfill the function. Simple example of this is the paint on a car. It makes the car a certain colour, but if it has to be applied in a certain way in order for the paint to protect the car. The paint's purpose is to protect the car but from a user perspective it also fulfills the purpose of the car looking as the user wants.

These functions should come together to fulfill tasks that the user wants to do, user requirements naturally come with the tasks. The tasks need to be done, but the the users' will have requirements on how they should be done. This could be that car paint should not be too reflective so sun rays do not reflect of the hood of the car and blinds the driver. Then concepts can be proposed with all the requirements fulfilled. With the conceptual proposals, more needs will come. Needs that could not be foreseen before, or it could be needs that emerge with the new system. Especially if the concept is user-tested, in which case could lead to more ideas and needs from the users as they test the system. That is what rejection means in the work cycle, the concept is in some way not complete or lacking in some way so the users' all needs are not met. A new iteration is then started to meet the new needs. These iterations can be done several times, but one must consider the time span of the project as well.

When finally the concept is of high enough quality to be accepted, the construction can begin. More unseen technological constraints might be first discovered then, as the systems starts to take its final shape. And finally the system can be started and tried out, where new needs might appear yet again. These things can not always be foreseen until before a complete concept is in front of the user and developer. The final stage is the check if the system does the job it was set out to do or how it changed during the project. Documenting how it changed would be useful for future projects, or if one wishes to continue the work with the system in future.

2.3 Interviews and Observations

Interviews is useful for the items listed below from [17, p.24]. These items are considered to be relevant for this project. Therefore this is the main data collection method as well as observations of the users using the current interface for seeing what the users do and not what they say they do, there can be a difference.

- Gather facts, attitudes, and opinions.
- Gather data on topics where the interviewer is relatively certain that the relevant issues have been identified, but still provide users with 24 Interview Techniques for UX Practitioners the opportunity to raise new issues that are important to them through open-ended questions.
- Gather data when you cannot observe behavior directly because of timing, hazards, privacy, or other factors. You might, for example, use a structured interview approach to gather data on the usability of Army command and control systems.
- Understand user goals.
- Gather information about tasks, task flow, and work artifacts such as job aids, forms, best practices documents, workflow diagrams, signs, equipment, photographs, and posters.
- Gather data on complex issues where probing and clarification of answers are required.

Since the interviewees are relatively sure about the current issues and it is their opinions about the usability of the current interface that needs to be collected, semi-structured interviews fits is a good choice.
Creating an interview guide before the interviews is intended to keep the interview going and collect as much information as possible. This method had been used by both thesis workers before so therefore a good way of collecting the data needed. Furthermore it is having an interview guide is part of the interview technique in [17]. It also the intention for more open questions and doing a semi-structured approach to keep things interesting for the interviewee, trying to keep a good discussion about the subject. According to [17, p.26] there are several strengths of semi-structured interviews:

- May uncover previously unknown issues (in contrast to a structured interview.
- Address complex topics through probes and clarification.
- Ensure that particular points are covered with each participant and also allow users and interviewers to raise additional concerns and issues.
- Provide a mechanism for redirecting conversations that digress too far from the main topic.
- Provide some flexibility for interviewers and also allows some broad comparisons across interviews.
- Require less training time than unstructured interviews because the interviewer has a set of specific questions available as a starting point.
- Can be conducted by an outside consulting company because there is a base set of questions (although you should research any external interviewing organization to ensure that they are competent at interviewing, data analysis, and interpretation).

The project participants were not fully knowledgeable about the current interface solution, so giving the the interviewees more freedom would mean that the interviewers would gain some more knowledge after the first interviews. The next interviews could then be adjusted to be more about the issues of the current interface.
The interviews can be complimented by observations, as observations show what people actually do and not what they say or think they do. These observations are very simple. The project participants will observe the modelers at ÅF when they create interfaces.

2.4 The Affinity- Interrelationship Method

The affinity- interrelationship method or AIM is a tool for finding relevant information on a subject. A detailed description of this can be found in [1]. The method can be described as 10 steps. In short one can say the method is to formulate a question/issue you want good answers to. This question/issue is then written in the upper left corner of a whiteboard, which will work as an AIM-map when all the steps are done. The answers to the question/issue are then written on post-its. They are then sorted and categorized, to be able to find the best answer and most relevant categories to the main question/issue. The entire process should be lead by the AIM-leader which is appointed at the beginning of the AIM-session. The result of the method is an AIM-map which consists of post-its with names of categories on them. These categories are also ranked in importance. This will then lead to that the project team will know what is the most important things to work with.

2.5 PACT - People, Activities, Context and Technology Analysis

PACT (People, Activities, Contexts, Technologies) is a tool for analysing four elements, people, activities, contexts and technologies for an interactive system. These four elements are explained in more detail below in this section. As mentioned in [2] designers need to understand the variety inherent in all these elements.

People, they differ both physically and psychologically. Starting with the physical, people are physically different in many ways, such as height, weight and variety in the senses (sight, hearing, touch, smell and taste). Colour blindness (commonly inability to distinguish between red and green) would affect a person's ability to use and interactive systems that require this ability to see the difference between red and green.

People also vary in many ways psychologically. For example a designer should design websites for people with bad spatial ability since not everyone has a good one. Having clear directions and explanations makes it easier for everyone to understand. One important aspect of designing is the mental models different people have created. A simplified example of this could be where one sees a sign another sees a button. Provided it beeing a button it was apparently not clear enough, and that made some people create a false mental model.

So clearly, it is important for a designer to ensure that sufficient information is provided in the interface to enable users to form correct and useful mental models of the system. This can be achieved by having a clear, logical and consistent conceptual design [2], since this will be easier to communicate to people. Some interesting concluding points about mental models from [2, p.33]

- Mental models are incomplete. People will understand some parts of a system better than others.
- People can 'run' (or try out) their models when required, but often with limited accuracy.
- Mental models are unstable – people forget details.
- Mental models do not have firm boundaries: similar devices and operations get confused with one another.
- Mental models are unscientific, exhibiting 'superstitious' behaviour.
- Mental models are parsimonious. People are willing to undertake additional physical operations to minimize mental effort, e.g. people will switch off the device and start again rather than trying to recover from an error.

There are still more things to consider when designing for people, this could be why the user uses the system. The motivation and goals they have using the same system may vary. Some might be very keen on using it while some just want to get it over with, just be done with it. The levels of knowledge also influence how they use the system. Some might have extensive knowledge of all the function while another might just have about the basic functions. Microsoft Excel is a good example. Experienced user might use macros while the average “at home users” only use the formulas.

The next part in PACT is the activities. In [2] these aspects of activities are divided into these five aspects, Temporal, Cooperation, Complexity, Safety-critical and The nature of the content.

The temporal covers the frequency of the activities. So that in mind it would be preferable for the frequent activities to be easy to do. The less frequent should be focused on reminding the user how to do them. The user might know the frequent by heart but not the infrequent ones, so they need to remind them in some way of how to do it. Or preferably be so intuitive that one can do them without instructions. Furthermore activities may require to be carried out at the same time as another, or work together with another. The different activities need to have sort of communication and awareness of each other to avoid problems that occur when things are not coordinated; some activities might be dependent on another, meaning that it would be not be carried out correctly if certain activities are not done first. For example you need to use a base layer of paint before you can apply the final coating, in order for it to look good and last long.

Speaking of this “painting task”, its rather simple and can be done through a step by step guide. Activities with more complexity might need a different approach to achieve its goal. This type of activity often comes with a more vague description of the tasks needed to be done or if all. If one for example needs to develop a way of traveling into space, and nothing more. It sure opens up a lot of possibilities and it is more about collecting data and information and no real defined step by step process. This designer need to consider, how to present way of doing the tasks needed for the activities.

This can also be a safety issue, some activities might have serious consequences if done incorrectly. This could be hospital related equipment or airplane technologies. Designers must know which activities are “safety critical” and provide the right tools for the user to ideally never make mistakes in these cases, or just dampen the effects of mistakes.

Designers should consider when a users make mistakes and how to handle them, this goes for all activities. Finally to consider about activities is The nature of the content. If the the activity needs a lot of input, so a keyboard is needed. Could also be that just a few buttons that is needed or just a barcode scanner. It all depends on the activity what supportive technology will be required.

Another thing to consider is the context which the activity takes place. To understand what is meant by context it can be divided into three areas could be three different types, organizational context, social context and physical circumstances. As an example in we could have sending an email. This is often done from a computer but it could be from a phone as well. And in this case the context can vary from an office to home or from a buss. These are all different contexts this activity take place in. And furthermore it can be work-related or personal. It might be worth considering that other emails should not show directly if the user does not want other people to see their emails, since the user could be in a public place.

Now to the final part of PACT, technology. In order to choose the best solutions one must have all the options available. Designers must be fully aware of the latest technologies that they can use for their interactive system. For example if a system needs a display and some limited input, it might be best to choose a touchscreen

instead of using a screen with a keyboard. Furthermore one could argue that a screen is easier to keep clean than a keyboard, especially in dusty environments. And of course a steering wheel is a better choice than a keyboard for controlling a car.
As a final note PACT is useful for generating user personas to aid in the development process. Or it could be used as research tool for the development process; it helps designers understand all the aspects that the systems deals with.

2.6 Designing Interactive Systems

Preece et al. (2002) describe interactive design as the design process of interactive products which support people in their everyday and working lives. The design work ultimately creates a user experience that improves the way people work, communicate and interact. In [2] Benyon refer the term design to both the creative process of specifying something new and the representations that are produced during the process. The core concerns for a designer of interactive system is accordingly to [2]: design, technology, people, activities and context. The field of design include many different design practices that can be utilized when designing interactive systems and the designer will have to decide how it should be done. Designers need to have knowledge about different technological possibilities that interactive system will be made up by. The term technology include information relevant to the design process regarding products, devices, components and software systems that are primarily concerned with processing information. In interactive design people is referred to the users of the interactive system or who it will have an impact on. Activity is concerned about what people want to do or achieve with the interactive system and those activities take place in some context.

2.7 The Design process

An HMI (Human Machine Interface) is designed to work with humans. So according to [2] it is essential to be humancentred in order to understand the needs of the users. Some bullet points in being humancentred is [2, p.14]):

- Thinking about what people want to do rather than what the technology can do
- Designing new ways to connect people with people
- Involving people in the design process
- Designing for diversity.

2.8 The Requirements

Requirements are something that the system, or product, must have or must be able to do [15]. When designing interactive systems these the requirements should come from the users. But getting to the stage of having all the requirements is no easy task; especially if they are based on user opinions, ambitions and stories. It is more or less up to the designers to guess what the users need, since new requirements will emerge during development that could not be foreseen. What to call this "requirements activity" have been under much debate, according to [2]. Here are four of them [2, p.147]:

- Requirements gathering, picking up requirements with little interaction between designer and stakeholders,
- Requirements generation, a more creative process. Could be brainstorming sessions, which would de-emphasize links to current practice.
- Requirements elicitation, some interaction between stakeholders and designers.
- Requirements engineering, a very formal approach, often used in software development.

Requirements is generated with the people who are going to use it, or will be affected by the system. It is still difficult to get all details through interviews; it's often hard for people to describe everyday of their work. Also it is common that people often say they work in one way, but they actually work in another. Therefore also observations of existing systems can complement this. They might say they are doing it like "this" now but they would rather have done like "that". These are things that they might not think of thing during an interview. It is stated in [6, p.251]:

"*Participants may provide a response that they believe is socially desirable or more acceptable rather than the truth. This is known as social desirability. Similarly, a participant may describe the way things are supposed to happen rather than the way things actually happen.*"

Once the user needs has been documented it can turn into system requirements. Generally requirements are divided into two types, functional and nonfunctional. The functional is what the system must have and will not work without. The nonfunctional are things that concern the quality, such as aesthetics, performance, security and legal restrictions etc.

2.9 Prioritizing Requirements

Having all these requirements can be a handful and should be systematized in some way. This section intend to give an understanding on why it must be done and how it could be done. First of all it will make easier for designer to prioritize the work. Specially when working within a time limit, it is always good to have one's priorities set. Requirements can be prioritized with "MoSCoW rules" [2], which are:

- Must have - self explanatory, without these the system will not work or be useless. This are the absolute minimum.
- Should have - These are still of high priority but the system will work without them.
- Could have - of lesser importance, can be left out.
- Want to have but Won´t have this time round - will not be used in this version.

Similar prioritization is seen in Firesmith [8]:

- Essential requirements that must be included in the system (i.e., the actual requirements).
- Useful capabilities that would reduce system effectiveness if left out.
- Desirable capabilities that make the system more desirable to certain stakeholders.

There are other ways to name the priorities, but will not be covered in this thesis. It is more important to understanding how to choose the important ones as “must haves” or “Essential requirements” as above. But to actually prioritize them it means as [8, p.37] says:

“*1. Prioritization by implementation order. Prioritizing requirements is the requirements task of determining the implementation order of the requirements in an incremental and iterative development cycle.*”

“*2. Prioritization by importance. Prioritizing requirements is determining the order of importance to some stakeholder or class of stakeholders of the requirements along one or more dimensions (e.g., personal preference, business value, cost of implementation, and risk).*”

Form the statements above one can realize that some requirements may be more important than others but they must be implemented in a certain order. And that they could just be important to the system.

2.10 Usability

Preece et al. [14] outline usability as goals of interaction design and which purpose is to ensure that interactive products are easy to learn, effective to use, and enjoyable from the user’s perspective. High usability is achieved by ensuring that the system is effective, efficient, safe, useful, learnable and memorable. An effective system include suitable functions and information which will enable the system to do what it is supposed to do and a system that are efficient ensures that amount of effort to accomplish a task is appropriate accordingly to the task to be done. Systems that are easy to learn and easy to remember how to use are addressed learnable and memorable. Usefulness refers to that the system does what is expected of it. Safe system protects the user while operating in the system from dangerous conditions and undesirable situations. To help the designer to achieve these goals during the design process, [2], presents 12 design principles that can be used to evaluate or critique prototype design ideas:

Table 2.1: This table lists the 12 design principles.

Visibility	It is important to make things visible to ensure that people know what functions are available and what the system is currently doing. Visible function are more likely to be used while out of sight functions are hard to find and and to know how to be used. It is easier to recognize things than recall them.
Consistency	The element of interface design should be consistent to ease learning and use of the interface. Both design features and way of working should be consistent. Inconsistent could be useful to point out that something is of importance.
Familiarity	When it possible it is beneficial to use language and symbols that the intended users are familiar with. When introducing concepts which the user is unfamiliar to, a suitable metaphor can be used to help relate the new knowledge from an already familiar subject or field.
Affordance	Refers to an attribute of an object and how these attributes relates to how the object can be used. For instance if an object looks like a button people will assume that it is possible to press it and that an event will occur accordingly to how it looks.
Navigation	Concerns with the support that allows the user to move around parts of the system such as maps, directional signs and information signs.
Control	The user should know who or what is in control and the system should allow the user to take control. The ensure the sense of control should the user be informed of the relationship between what the system does and that is happen externally. A clear and logical mapping between the controls and the effect they have improves the control.
Feedback	Sending information back to the user when an action is performed. To constantly keep the user updated of what the effect their action had is important to preserve the feeling of control.
Recovery	The user should be able to quickly and efficiently recover from actions, mistakes and errors.
Constrains	Refers to determining ways of restricting the user to make certain actions at a certain time. Constrains are important to prevent users of doing inappropriate things which can have unexpected or unwanted results.
Flexibility	Refers to the possibility to do things in many ways and be able to change how things look or behave so it suites the individual user. People have different levels of experience and interests in what the system can do and flexibility aims to please all possible users.
Style	Concerns about attractive and stylish design.
Conviviality	The users should experience the system as polite, friendly and pleasant.

2.11 Experience Design

According to Green et al. [9] people are more than just “users” and human characteristics such as hopes, fears, dreams, aspirations, tastes and personalities influence the relationship between people and products. When searching for quality the aspect of pleasurability is important to consider alongside safety, wellbeing, comfort and ease of use. Green et al. [9] suggest that other factors than the well recognized usability factors is needed to describe pleasurability of design. These factors are concerned with aesthetics elements of products such as form, colour, and tactile properties.

In [14] they explain user experience as how interaction of the system feels like to the users. They state that setting up primarily objectives is an important part of the process of designing an interactive system. They discuss two different types of objectives, “user experience goals” and “usability goals”. The terms differ by how they can be met and through what means. User experience goals refer to explaining the quality of the user experience while usability goals main concern is meeting specific criteria such as effectiveness, efficiency, safety, utility, learnability and memorability. Below is a list examples of user experience goals from [14]:

- Satisfying
- Enjoyable
- Fun
- Entertaining
- Helpful
- Motivation
- Aesthetically pleasing
- Supportive of creativity rewarding
- Emotionally fulfilling

The user experience goals are what really makes people “like” or “dislike” a system. It should be a designer’s goal to make the users enjoy their use of the system, otherwise they might end up looking for another.

2.12 Interface Design

Benyon [2] state that a user interface consist of everything in the system that the user can interact with. People’s interaction with a user interface can be physically, perceptually or conceptually. Physically interaction can be through pressing button, touching the screen or other direct contact with the device. People can interact with what they can see, touch or hear which is referred to perceptual interaction. People can for example react upon information that appears on the screen.

The designer must choose suitable labeling and size on buttons for the user to be able to understand what it will do. Conceptually interaction goes through what people already know and how to do it. Conceptually interaction is connected to the user’s individual “mental model” and trough it can access information about how the system works. The user may know that a specific function exist through have been using a similar device.

The design work of a user interface is about manipulating these three aspect of interaction to create an experience that enables the user to make the best use of the system. Icons are commonly used to represent features and functions in a software application and they are regarded as an powerful tool to help people recognize the features or functions they want to access. The following list of bullet points from William Horton's checklist cover a range of aspect of what is useful to consider to avoid common mistakes made when designing or choosing icons:

- Understandable
- Familiar
- Unambiguous
- Memorable
- Informative
- Distinct
- Attractive
- Legible
- Compact
- Coherent
- Extensible

Menus are an useful tool for organizing commands in [16] it is stated that an overview will greatly improve the user's ability to navigate. This can be achieved with a menu since most of the content can be accessed. Furthermore in [16, p.43] it is stated that:

"*Broad, shallow hierarchies offer optimal navigation time, particularly when item locations are highly predictable or when users have existing spatial knowledge of the interface.*"

This supports the use of menus since they can be made similar to menus the user is already familiar with, like Internet navigation. An interface with a lot of menus for storing commands are often called menu-driven interfaces. Designers should be aware that menus should be simple and easy to use and avoid the pitfall of complex and difficult to navigate menus. Menus is commonly used for structuring information and is often serving as the main tool for navigation.

In [4] they argue that gestalt principles can be applied to human computer interaction to improve the user involvement and promote interaction. Gestalt theory derived from the field of psychology and it tries to explain how humans make sense of their perception and cognition. Gestalt theory is often described through sets of gestalt laws which illustrates the theories into more practical examples, Chang [5]. In [2] they highlight the laws of proximity, similarity, continuity and closure as useful for the application of designing visual interfaces.

- *The law of proximity*: objects that are close in space or time are likely to be perceived together. Figure 3 shows two groups of objects one will be perceived as ordered in columns and one in rows. When organizing buttons can the idea of proximity be used. Buttons with same or similar intention put together and special buttons with unique intention by themselves can help distinguish the two different intentions.

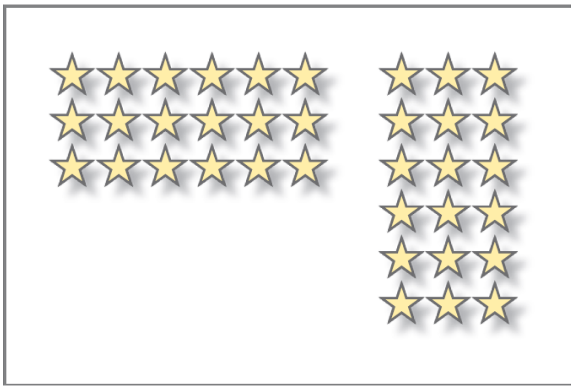


Figure 2.3: Proximity

- *The law of similarity*: refers to that object that are similar tend to be preserved as being grouped. In figure 4 are the circles and diamonds grouped together as rows of objects. When sorting objects the similarity of the object icons will be of great importance.

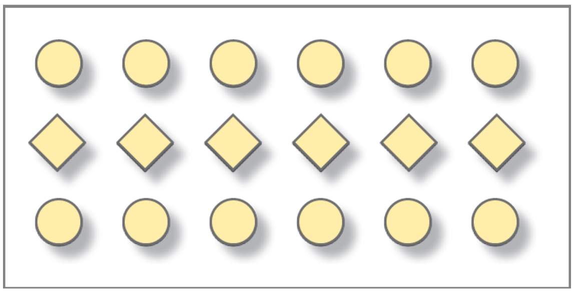


Figure 2.4: Similarity

- *The law of continuity*: It is easier to perceive smooth, continuous pattern than disjointed, interrupted ones. An example of this is illustrated in figure 5 where the picture tend to be seen as a continuous curve rather than five half circles which the curve is made of.

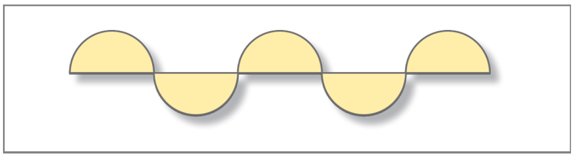


Figure 2.5: Continuity

- *The law of closure*: Closed figures are easier to perceive than open figures. Humans tend to make up for missing information by themselves if some is missing. Figure 6 can either be seen as a Maltese cross or four separate triangles.

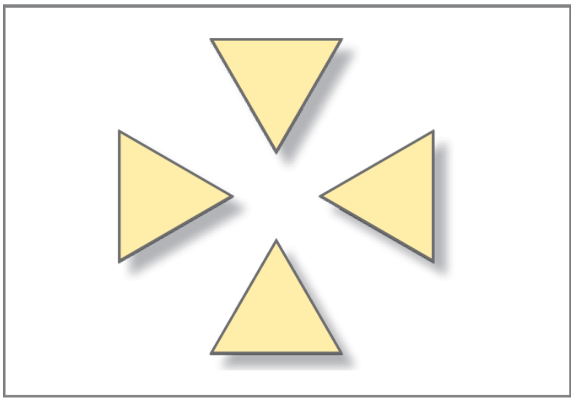


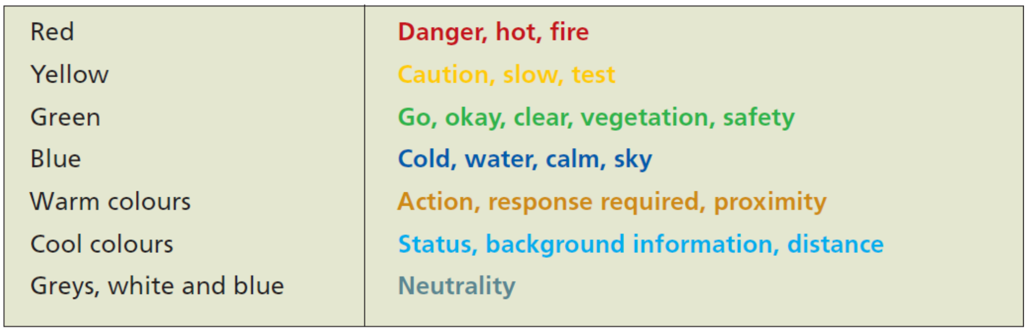
Figure 2.6: Closure

It is important to emphasize to design for memory and attention. There are several design principles to think of when designing interactive system. The working memory have capacity limitation of how many “items” of information it can work with. There are varying opinions about the number of this capacity limitation, George Millers guideline of working memory limitation is to 7 ± 2 items while recent work of Nelson Cowan argue that it is only 4 ± 1 [7]. Never the less it is agreed that it is important to restrict number of items in menus in a HMI due to the working memory storage capacity. When designing interfaces is it common to group and hide information under larger units in order to reduce the memory load, this is called chunking. The memory has also time limitations especially the working memory, therefore it is important to let information persist a longer time to help the user. It have been found to be preferable to design for recognition rather than recall since it is usually quicker and easier for the user. Recognition refers to our ability to "recognize" an event or piece of information that is familiar, while recall denotes the process of accessing related data from the memory.

Designing with colours is a hard task and there is many guidelines from the literature [12]. One well known set of guidelines is Aaron Marcus’s design rules [2, p.344].

- Rule 1 Use a maximum of 5 ± 2 colours.
- Rule 2 Use foveal (central) and peripheral colours appropriately.
- Rule 3 Use a colour area that exhibits a minimum shift in colour and/or size if the colour area changes in size.
- Rule 4 Do not use simultaneous high-chroma, spectral colours.
- Rule 5 Use familiar, consistent colour codings with appropriate references.

When designing with colours it is important to consider cultural differences according to Benyon [2]. People with different cultural backgrounds may perceive certain colours differently. In figure 2.7 from [13, p.84] is some is some common western colour conventions presented.



Red	Danger, hot, fire
Yellow	Caution, slow, test
Green	Go, okay, clear, vegetation, safety
Blue	Cold, water, calm, sky
Warm colours	Action, response required, proximity
Cool colours	Status, background information, distance
Greys, white and blue	Neutrality

Figure 2.7: Western colour conventions

However Johannesen [12] discuss some specific problems and gaps in colour guidelines in user-computer interface design. Design guidelines are often straightforward rules or laws derived from knowledge of the human perceptual system in a more “compact form” to make it practical to use. Johannesen et al. give a warning message that the use of guidelines can be misleading without the right context of use [12]. The guidelines tend to “miss underlying perceptual issues” due to that they don’t consider the detail level of the design problem.
For example Aaron Marcus’s guideline "Use a maximum of five plus or minus two colours”. The task what the designer want to accomplish nor the circumstances given. It could be problematic according to Johannesen et al. [12] since how colours are used will affect the working memory in different ways. For example if colour is used to make distinctions among visual forms, it does not have a meaning to be remembered thus don’t put much load on the working memory. Johannesen et al. suggest a redirection of the guidelines to be based on difficulties and common problems regarding the field of use and not purely of knowledge of the human perceptual system [12].

3
Method - Work Process

This section will describe the methods used in this thesis and why these methods were chosen. This project includes interviews, literature study and finally programming and development of an actual interface. The interviews will be done with a number of different experts in the fields of HMI, statistics and simulation models to create a specification of requirements for the interface. The information from the interviews and literature study was further analysed in the affinity- interrelationship method in order to select the most important aspects of the interface.
This section is on how the methods were used in this project. Bringing up notable modifications to the methods in order to fit this project. This project implemented the work cycle from section 2.2, figure 2.2. Following this process makes for easier user centered design as the users needs are the driving factors for the design work. All the needs come from the users and they were collected through interviews, as the following section will explain.

3.1 Data Collection and Interviews

In this project several interviews was done in the first weeks along with a literature study about interface design. The literature gave the project participants some basic knowledge on the subject of interfaces and how they are developed. This would make for easier interviewing with the modelers. The interviews had the goal of giving an good understanding of the current interface, its current problems and what needs the users have. The interviews were held in closed rooms between the project participants and the modelers from ÅF. All interviews were recorded and the questions were transcribed directly after. The transcription consisting of the questions with summarized answers, this summary is in appendix A.

3.2 The Affinity- Interrelationship Method

After the interview data was collected the project participants started the creation of the affinity diagram and the interrelationship digraph. The method was used as described in section 2.4. The summarized result is in appendix B. Following the 10 steps of the method with minor modifications that will be explained as they come. First off though, only two participants was used and not the recommended amount of 4-8. This was done i closed room with only the two project participants. The main question used for the method was: What are problems with the current user Interface? This question was formulated in this way find what issues the user interface had so these could be made in a better way. All the answers came from the interviews. Finding common themes that answered the main question. Firstly sorting out direct problems like it is suggested in the method [1], then creating titles and grouping them. In step 9 from [1] when it came to voting, the project participants invited three users to also vote. None of the voters could see what the others had voted on. Nor where there anyone in the room when the users voted, that could affect their vote. This way the importance of the issues where based on what the users thought.

3.3 PACT- Analysis

PACT - analysis according to [2] is good for understanding the current situation, seeing where possible improvement can be made or envisioning future situation. This chapter present a PACT-analysis from observations of the current way of developing customized interfaces at ÅF. Interviews and brainstorming sessions with modeler at ÅF provided further knowledge of the four PACT elements people, activities, context and technologies.

3.3.1 People

The people who will use the generic-UI can be categorized into two different groups, the modelers and the end customers. The modelers are the employees at ÅF who build the simulation model for the end customers and present its results through the generic-UI.

The modeler team consist of a small group of people with high knowledge in simulation, programing and with similar technical educational background. They are experienced users of computer programs and are used to handling huge amounts of data but their experience in developing user interfaces vary. The design experience among the modelers is considered to be rather low and they are relying on personal opinions and sort of “gut feeling” when designing the visual appearance of the interface for the end customer. There is also a difference in experience among the modelers to use of Excel as a presentation tool.

The characteristics of a simulation project tend to vary a lot between projects, for example the goal with the project, resources, length of the project. The end customer could be people from a large variety of businesses and industries, from mining industries to hospitals. This means that the customers with such different backgrounds will create different mental models on how simulation works. As an example some might see a factory as a place with a lot people building things in a workshop, while another sees only robots working in a line. The modelers at ÅF have different models of how the interface works. Some of them are used to macros in Excel and can create their own. Some do not even look at them, using only the basic functions in Excel accessed in the Excel menus.

3.3.2 Activities

What activities the people what to do with the generic-UI will depend on the role of the user (modeler or end customer). The modeler activities can be categories into three main activities connecting AutoMod model with generic-UI, design the visuals aspect of generic-UI and simulation analysis. Connecting Automod model with generic-UI, the modelers needs to have a connection between the AutoMod and Excel. This connection ensures that input and output data can be communicated between the programs To do this they need to do several activities:

- Construct sheets with input data.
- Construct sheet with output data.
- Send the input data to AutoMod.
- Run simulation from the interface.
- Collect the output data from AutoMod.

These are the basic activities that outline the communication between Excel and AutoMod. Design the visuals aspect of generic-UI , the modelers are also concerned about the visual appearance of the interface. The activities involved is:

- Designing of sheets, data matrices and graphical presentations.
- Create interface navigation.
- (Colour, formatting, structure, navigation etc)

Simulation analysis, is concerned about making sense of the simulation. Simulation projects have certain goals to fulfill and present to the end customer. The analysis can include comparison of multiple runs with different random seed and input parameter or only single scenario run depending on the project goals. This work is summarized into following activities:

- Create graphs of the output data.
- Analyze the simulation data.
- Establish conclusions from the simulation data.

End customers, this activity depends on project goals and roles. The end customer's needs can vary and with that also the activities. Their ambitions with the simulation can be from just one decision needing to be made, all the way to using it as an continuous improvement tool. The activities that will meet all companies needs are the following:

- Learn how to use the simulation interface
- Run Simulation.
- Run multiple simulations in one go. (rarely)
- Manipulate input data.
- Read output data.
- Analyze the simulation data.
- Establish conclusions from the simulation data.

These activities are what customers will want to do with the interface. There might be special cases where something more might be needed. It will be up to the modelers then to include the extra functions for the interface to do these extra activities.

Companies have some kind of organization hierarchy, depending on the user's position in this hierarchy they will do different activities. A production engineer might want to look at the cycle time changes and how it affects stops etc, while an executive officer just want to see the costs of an investment.

3.3.3 Context

There are some aspects of the context that are worth considering, the social and organizational contexts. The physical circumstances under which the activities take place does not affect how the interface will be used. It will be in an office environment using laptops and mostly with connected extra displays. So there is no need to consider that the physical circumstances will hinder the work in such a way that the interface must be designed with this in consideration. However since the system will be used by people who might not have been using it before, even if help from experienced user will be available it is still viable to consider help functions within the interface. Keeping sheet designs to the customers own layout (to the extent it is possible) will allow them to fill in new input data without wondering where it goes. Furthermore the overall design will follow the ÅF's design directives, such as fonts, buttons and colours. Since that is what is expected to be used in the social context at ÅF. The interface will be in all English since there is a chance of users that do not speak swedish. To motivate this further, all users will know some English. However if the users does not know English, the modelers will be able to make adjustments to account for this.

3.3.4 Technologies

Excel is a more or less standard in industries, it is the interface format the modeler are used to. Furthermore the input data often come in Excel documents from the customer. Then it can be directly copied into the input sheets in the interface. So Excel is more or less a requirement. Windows is also the standard operating system and that is what the modelers use when building AutoMod models. AutoMod is also windows based and the interface must be able to run AutoMod at the same time.

3.4 Requirements Specification

The requirements were based on the results from the affinity method. How the users voted was the deciding factor on how the requirements should be prioritised. The full requirements specification can be found in appendix C. Most of the results from the AIM method was problems and things that could be made better. How these requirements was made is will now be explained with an example. So for example, a topic from the AIM method was “lack of visual/design standards”. This was made into the requirements:

- Modeler should know where to place matrices.
- Modeler shall use correct colours.
- Aid in Sheet design.

These requirements came from interpretation of the interviews and discussion with the users. Many desirable requirements could be directly taken from the AIM map in the first level grouping. There were some requirements that had to made before anything else, these were some technical requirements that made the interface possible.

- Run 64-bit models in AutoMod 12.6 from the Excel-interface.
- Run 32-bit models.
- Run multiple simulations.

Without these functions the interface could not be used at all, because from discussion with the users these were essential for the interface to fulfill its main purpose, running multiple simulations. More requirements where also established from the interviews. But these were not needed for the main function of the interface, however they were needed in order to enhance the user experience.

3.5 Design and Concept Generation

It was decided to start with the main technical requirements as mentioned in section 3.4 to get an interface with just the core functions first, and after that start working on the user experience. Since the timespan of this project was short, having a functional interface was prioritized. The core requirements of the interface was translated into functions to fulfill them. To establish what solutions to use for each function was evaluated in a morphological matrix. Each function of the system had several ways of fulfilling them. In some cases it was just a matter of testing them and seeing which was fastest.

- Start AutoMod from Excel and run a selected model.
- Handshake between Excel and AutoMod: that AutoMod is running.
- Read Input from Excel to AutoMod.
- Write output data from AutoMod to Excel.

4
Results

4.1 Problem-description of the old user interfaces

The focus in the development process of user interfaces has mainly been on technical solutions and constructions rather than developing towards a user centered design. From the interviews has it been revealed that the modelers often design the interface after the model rather than the customer. Consequently a large amount of data both input and output have been presented without any real thought of good presentation and the reaction from the customer have been a “drowning in data” sensation. The sheer amount of data in the matrices is overwhelming for most people. The common approach to simulation has been to focus on what data the customer can supply, what data they want, then building the model after that. All the output-data have been discussed with the customer and together with the modelers they come up with how the data should be presented. Not using any real guidelines to adjust the output-data for the goals set for the project. It is true that the modeler and customer think they know what they want and need, but without any real guidelines for data selection and presentation it might not be the best for the goals of the project. The main problems with the ÅF’s current user interface is the following:

- Unique customers.
- No guidelines for this type of project
- Overwhelming data
- Bad navigation
- Project goals (weak?, unused?)
- Insufficient/low accuracy on presentation
- Problematic UI realisation
- Time consuming
- Technical problems (technical limits)
- Design for model and modeler rather than for end user

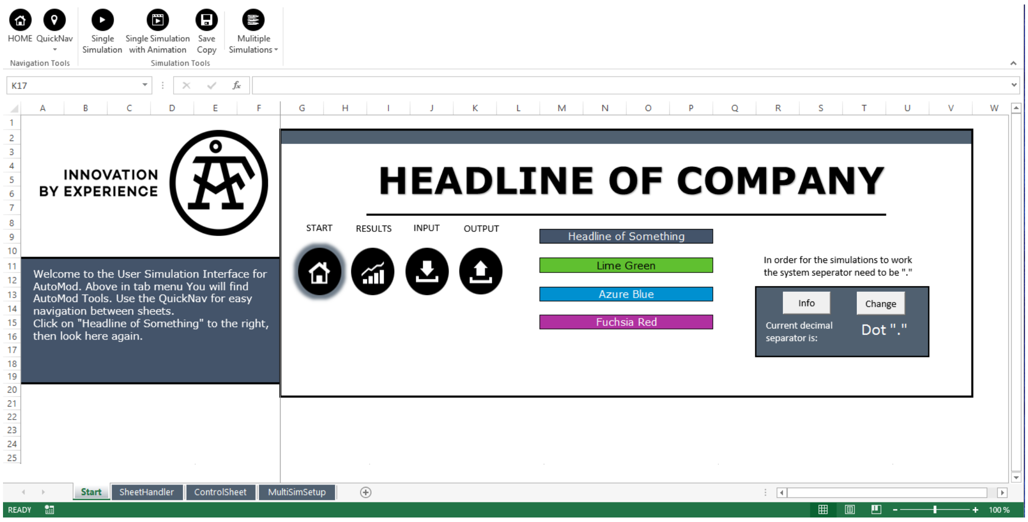
The modelers are aware of these problems. They just lack the tools to deal with them. The effects they would like to achieve are:

- Clear guidelines (work method)
- Present what the customer wants
- Easy navigation
- Accuracy, precision
- Easy realisation
- Standardization

- High level customization

4.2 The Generic User Interface

The Generic User Interface was made in Microsoft Excel. The code for was written in visual basic. The start page of the generic-UI can be seen in figure 4.1.



Welcome to the User Simulation Interface for AutoMod. Above in tab menu You will find AutoMod Tools. Use the QuickNav for easy navigation between sheets.
Click on "Headline of Something" to the right, then look here again.

In order for the simulations to work the system seperator need to be "."

Current decimal separator is: Dot "."

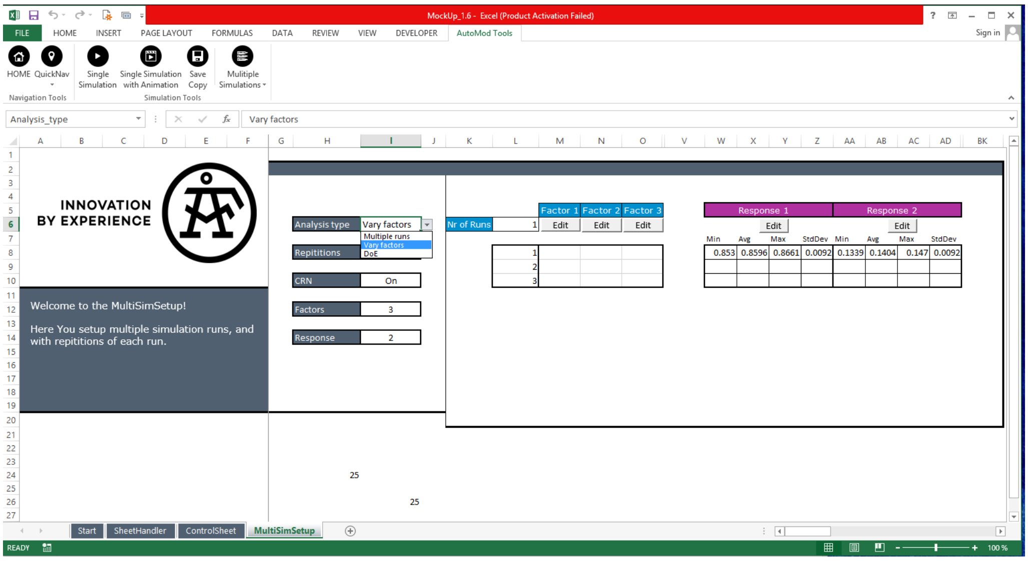
Figure 4.1: The start sheet displays some general information and features a small tutorial if the user follows the instructions in the text box.

The interface is a Excel document with a custom ribbon called AutoMod Tools. The user can start AutoMod simulation from this ribbon. More details on the ribbon in section 5.3.3. The generic-UI is a template featuring functions which aid the user to organize sheets. It can also aid the user in running multiple simulations with changing input data as well as monitoring some statistical data on the output. The capabilities of the interface have been summarized to the list below. More detailed descriptions follow in this section.

- Start simulations in AutoMod. (Can also start multiple simulations after each other, see Multiple Simulations)
- Provide analysis tools for several simulations (see Multiple Simulations)
- Communicate input to AutoMod.
- Collect output from AutoMod.
- Sort a large number of Excel sheets and divide them into groups. (see navigation)
- Navigating between the groups with a navigation tool called QuickNav (see navigation).
- Give a standard colour scheme for modeler sheets, input sheets, output sheets and result sheets.
- Provide help texts with explanations.

4.2.1 The Generic User Interface: Multiple Simulations

Multiple simulations is used to analyze the simulation model. There are three analysis types to choose from, “Multiple runs”, “Vary factors” and “DoE”. This is selected through the drop down menu shown in figure 4.2.



Analysis_type | Vary factors

Welcome to the MultiSimSetup!

Here You setup multiple simulation runs, and with repititions of each run.

Analysis type	Vary factors
Repititions	
CRN	On
Factors	3
Response	2

Multiple runs
Vary factors
DoE

Nr of Runs	1	Factor 1	Factor 2	Factor 3
		Edit	Edit	Edit
	1			
	2			
	3			

Response 1				Response 2			
Edit				Edit			
Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
0.853	0.8596	0.8661	0.0092	0.1339	0.1404	0.147	0.0092

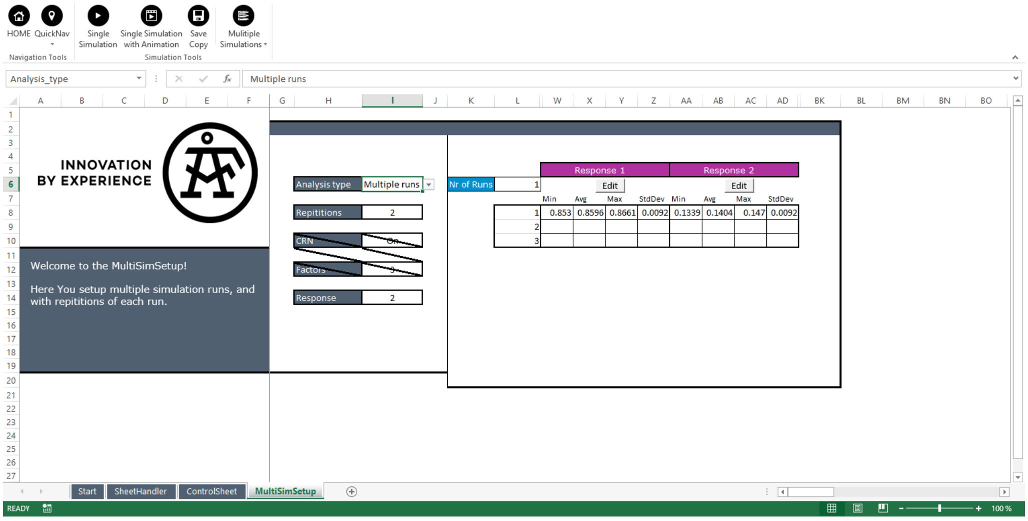
25

25

Figure 4.2: The user can select analysis type through a dropdown menu.

“Multiple runs” will run simulation without changing any input data. “Vary factors” will change the input data between runs. The user can choose to do several repetitions with the same input data before changing the input data. “DOE” stand for design of experiments and was still in development during the time of this report. Some small preparations were made for future work implementing DoE.

If the user chooses “Multiple runs” the factors will not be available for editing. As seen in figure 4.3. Only Responses are available for editing. The responses will be explained more later.



Multiple runs

Welcome to the MultiSimSetup!

Here You setup multiple simulation runs, and with repititions of each run.

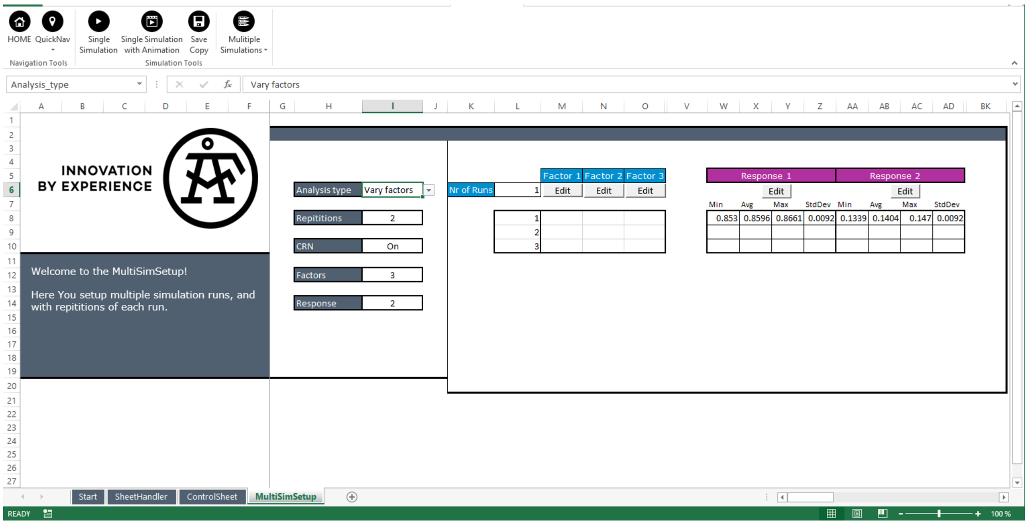
Analysis type	Multiple runs
Repititions	2
~~CRN~~	~~On~~
~~Factors~~	~~3~~
Response	2

Nr of Runs 1

	Response 1				Response 2			
	Edit				Edit			
	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
1	0.853	0.8596	0.8661	0.0092	0.1339	0.1404	0.147	0.0092
2								
3								

Figure 4.3: Because Multiple runs is selected the factors are not shown since they will not be needed. The simulation will do repetitions with the default input.

When “Vary factor” is selected the edit function becomes available. The next figure 4.4 shows this.



Vary factors

Welcome to the MultiSimSetup!

Here You setup multiple simulation runs, and with repititions of each run.

Analysis type	Vary factors
Repititions	2
CRN	On
Factors	3
Response	2

Nr of Runs 1

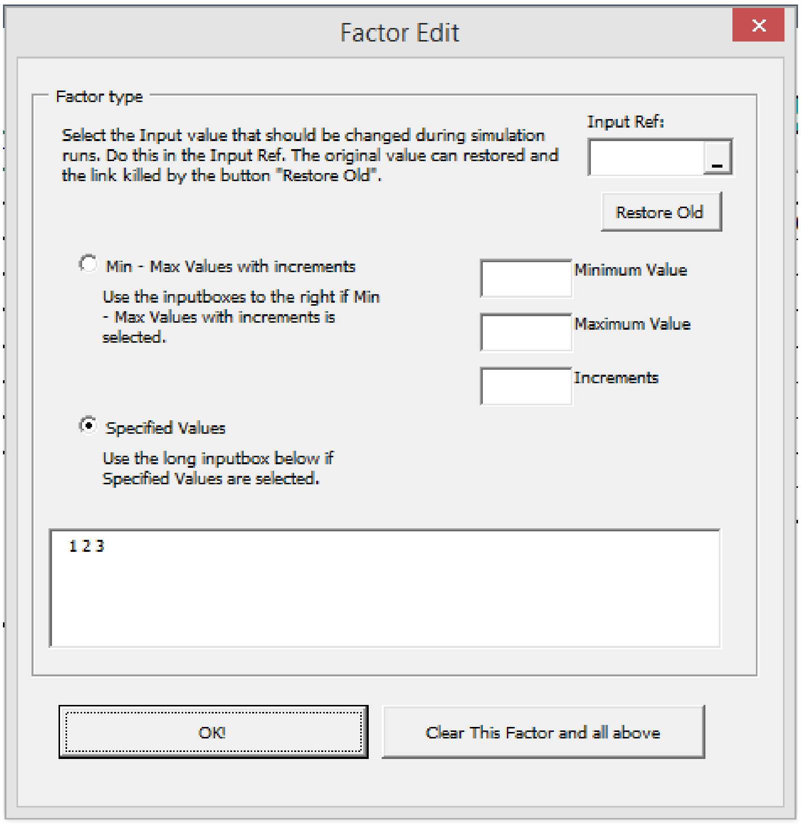
	Factor 1	Factor 2	Factor 3
	Edit	Edit	Edit
1			
2			
3			

Response 1				Response 2			
Edit				Edit			
Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
0.853	0.8596	0.8661	0.0092	0.1339	0.1404	0.147	0.0092

Figure 4.4: Since Vary factors is selected in Analysis type, the factors will be available for editing.

Pressing the “Edit” button under any factor will result in that factor being edited in the factor-edit form. The form is seen below in figure 4.5.

In the Factor Edit the Input Ref refers to a specific cell in an input sheet, which the user selects. This cell will get a new value between each run. Thus changing the input



Select the Input value that should be changed during simulation runs. Do this in the Input Ref. The original value can restored and the link killed by the button "Restore Old".

○ Min - Max Values with increments

Use the inputboxes to the right if Min - Max Values with increments is selected.

● Specified Values

Use the long inputbox below if Specified Values are selected.

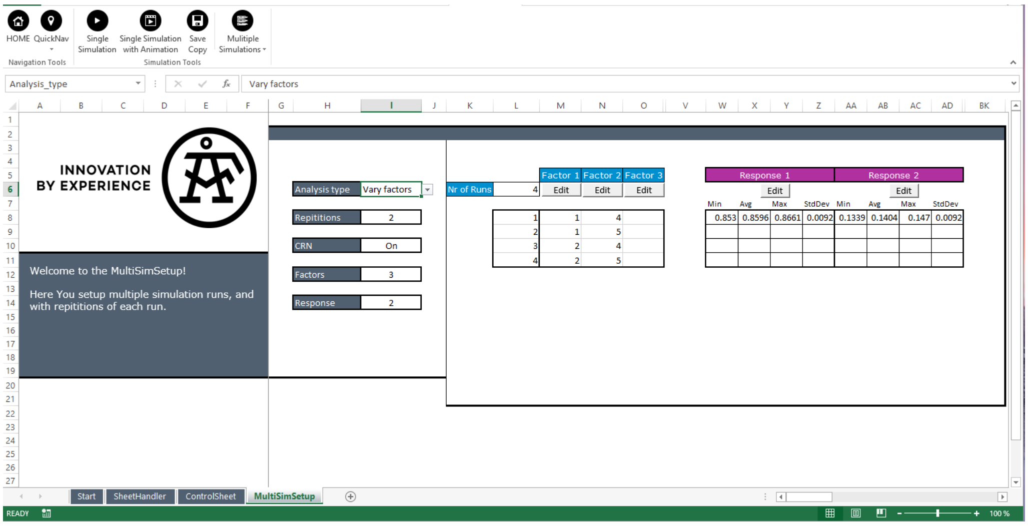
1 2 3

Figure 4.5: Factor Edit form, here the user has connected a cell from an input sheet. As seen in "Input Ref"

for the model. The changing values can either be a minimum to a maximum with a given increment or it can be specified values. When more than one factor is entered all combinations of these factors will automatically generated by the interface. An example of this is shown in figure 4.6.

As mentioned before responses can be selected for each analysis type. Selecting an response works similar to selecting a cell for the input. A the small form figure 4.7 is used for selecting a response.

The user can select any value from an output sheet to monitor. For each run the minimum value, maximum value, average value and the standard deviation will be displayed under each response. Each run is also logged and sorted in folders. The interface will save output sheets that the user has specified. If several repetitions is done with the same input a summarization can be created with the average values for each output sheet.



Analysis_type | Vary factors

INNOVATION BY EXPERIENCE

Welcome to the MultiSimSetup!

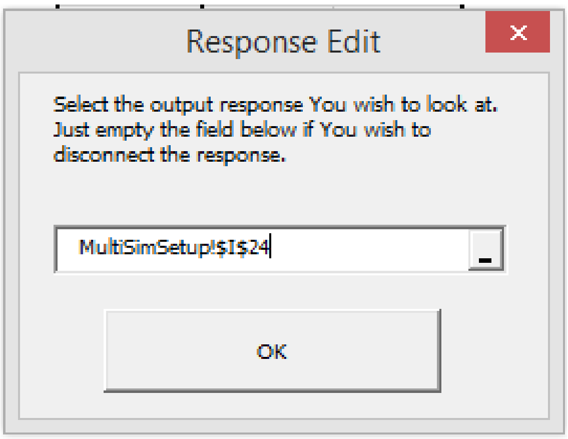
Here You setup multiple simulation runs, and with repititions of each run.

Analysis type	Vary factors
Repititions	2
CRN	On
Factors	3
Response	2

Nr of Runs 4	Factor 1	Factor 2	Factor 3
	Edit	Edit	Edit
1	1	4	
2	1	5	
3	2	4	
4	2	5	

Response 1				Response 2			
Edit				Edit			
Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
0.853	0.8596	0.8661	0.0092	0.1339	0.1404	0.147	0.0092

Figure 4.6: In this example Factor 1 was given the values 1 and 2 and the Factor 2 was given 4 and 5. The system will then generate all combinations possible with these numbers. As seen there is a total of 6 possible combinations.



Response Edit

Select the output response You wish to look at. Just empty the field below if You wish to disconnect the response.

MultiSimSetup!I24

Figure 4.7: This form is used for selecting a response form any output sheet.

4.2.2 The Generic User Interface: Navigation

The generic-UI uses a ribbon to help navigate around in the Excel based environment. The ribbon shown in figure 4.8 is equipped with shortcut icons for the navigation options and for the three important features: Single Simulation, Single Simulation with Animation and Multiple Simulation. There are six navigation options available: “Home”, “Results”, “Input”, “Output”, “UI Options” and “Show All Sheets”. The buttons are selected among hundreds of icons from a visual guidelines document provided by ÅF with Horton’s checklist from section 2.11 in mind. For example the picture on the "home" button illustrates a house. The picture is a commonly used picture associated with starting page and is therefore memorable, understandable, familiar as well as informative. The “Home” icon positioned to the far left for a easy access to the homepage of the generic-UI while the rest of the navigation options is gathered under the “QuickNav” dropdown menu see figure 4.9.

The navigation options “Input”, “Output” and “Results” each have its own homepage shown in figure 4.10. Sheets which including setup necessary for handling the generic UI is gathered under the “UI Option”. Selecting one of the navigation option will exclude all sheets but the ones related to the option chosen with the exception of “Show All Sheets” which will show all sheets. These sheets also illustrates on how familiarity and recognition is applied to the generic-UI. The start pages have the same layout with information and buttons located in the same manner. As seen the start pages also have different colours. Each sheet type has its own colour, input is blue, output is purple and results are green. This creates recognition for the user allowing them to see just by the colour which sheet type they are currently looking at. Also as suggested in section 2.12 the generic-UI uses fewer colours than the maximum amount suggest by the literature. This gives the modeler at ÅF room for implementing a few more colours of their own if need be.

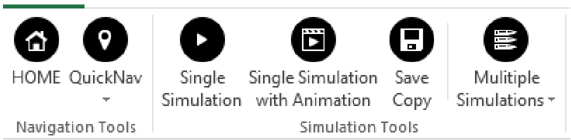


Figure 4.8: Ribbon containing shortcuts for navigation and simulation tools.

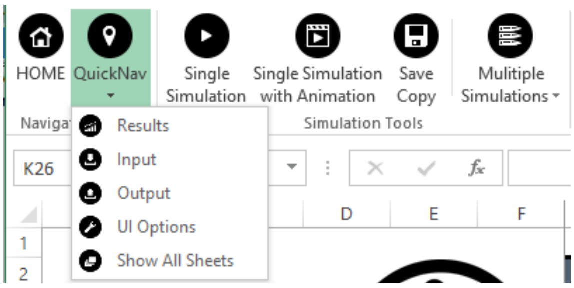
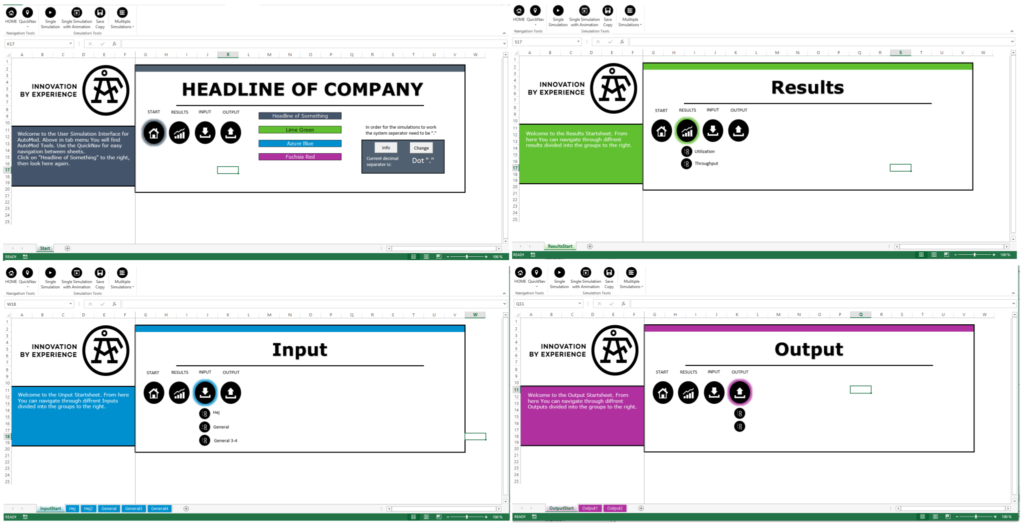


Figure 4.9: QuickNav dropdown menu for handling navigation options.

The user is able to specify what sheets that should be shown when selecting one of the quick navigation options. This categorization of sheets is be done by sorting them manually in the sheet “SheetHandler” show in figure 4.11. There is one category for each of the navigation options; input, output, result and UI options. Under each category is the name of the sheets that belong to that category chosen and automatically connected to the corresponding quick navigation option. Figure 4.12 illustrate the connection between “SheetHandler” and the navigation options.

There is also navigation buttons on each startpage “Home”, “Input”, “Output” and “Results” (see figure 4.13). Selecting one of the buttons will navigate the user to the desired location based on the categorisation of the sheets in the generic-UI described above. As seen they utilize the gestalt laws from section 2.12. The main categories



Welcome to the User Simulation Interface for AutoMod. Above in tab menu You will find AutoMod Tools. Use the QuickNav for easy navigation between sheets. Click on "Headline of Something" to the right, then look here again.

In order for the simulations to work the system seperator need to be "."

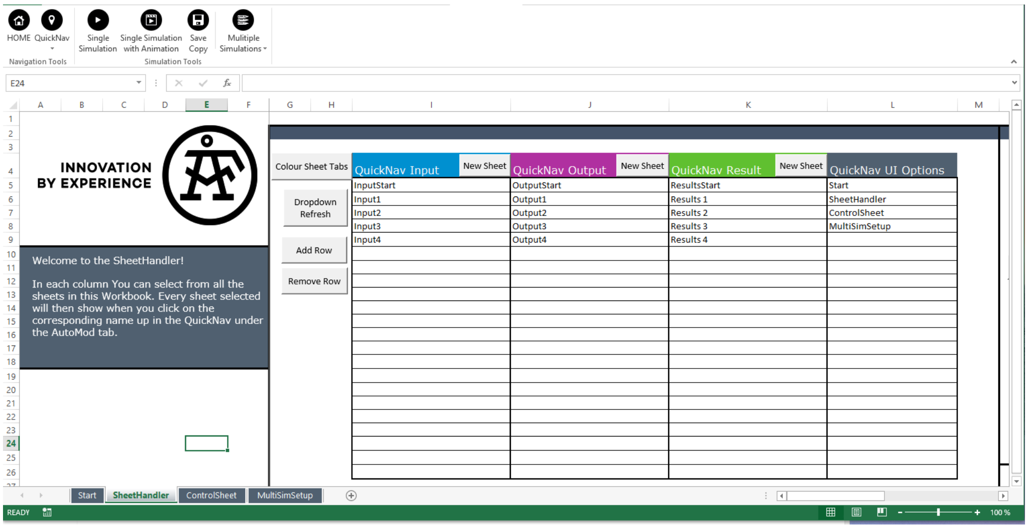
Current decimal separator is: Dot "."

Welcome to the Results Startsheet. From here You can navigate through diffrent results divided into the groups to the right.

Welcome to the Unput Startsheet. From here You can navigate through diffrent Inputs divided into the groups to the right.

Welcome to the Output Startsheet. From here You can navigate through diffrent Outputs divided into the groups to the right.

Figure 4.10: Start pages for the navigation options “Home”, “Results”, “Inputs” and “Output”



Welcome to the SheetHandler!

In each column You can select from all the sheets in this Workbook. Every sheet selected will then show when you click on the corresponding name up in the QuickNav under the AutoMod tab.

QuickNav Input	QuickNav Output	QuickNav Result	QuickNav UI Options
InputStart	OutputStart	ResultsStart	Start
Input1	Output1	Results 1	SheetHandler
Input2	Output2	Results 2	ControlSheet
Input3	Output3	Results 3	MultiSimSetup
Input4	Output4	Results 4	

Figure 4.11: SheetHandler enables the user to specify what sheets that should be shown.

as input, output and results are grouped at one level whereas each subcategory is located underneath the corresponding main category. Thus grouping them together in a proximity. The law of similarity is applied by having the main groups located in one row and with same icon size. The subcategories are located in the same manner for each subcategory group.

The user have the possibility to add navigation buttons additional to the main buttons of “Input”, “Output” and “Results”. The purpose of these optional buttons is to group the sheets of the main category in order to create an easy access to each of the new groups of sheets. Figure 4.14 illustrate an example how the results sheets can be divided further into two additional groups of Utilization and Throughput. The user is able to specify what sheets that should be shown when selecting one of

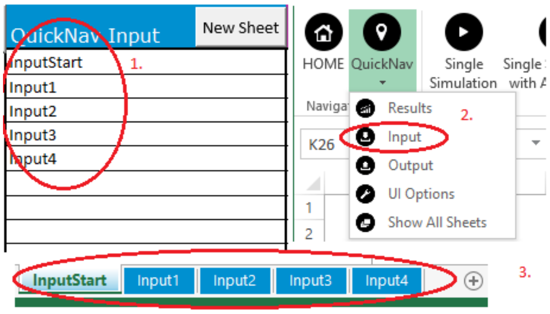


Figure 4.12: Illustrate how the user is able to define what sheets should be shown through categorization of the sheet in the “SheetHandler”. Name of the sheets is chosen under Quicknav Input (1). When the user select the QuickNav Option “Input” (2) will the sheet (3) specified be shown.

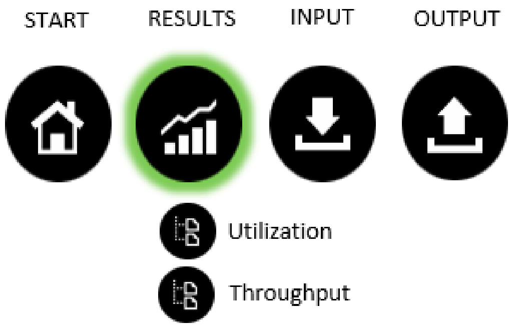


Figure 4.13: Start page main navigation buttons. The glowing green around the result icon message the current location.

the new additional buttons in the “SheetHandler” in the same manner as with the main buttons.

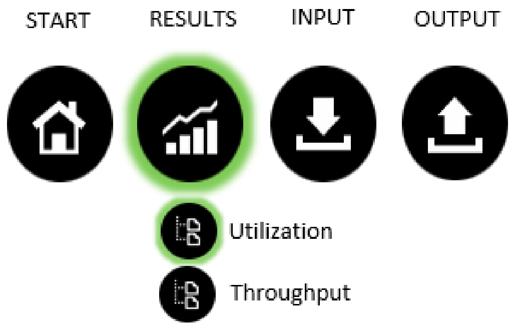


Figure 4.14: Two additional navigation buttons has been added, "Utilization" and "Throughput". The green glow around the “Utilization” button reminds the user of which sheets are currently shown

4.2.3 The Generic User Interface: Communication between Excel and AutoMod

The new communication is done with text files between the interface in Excel and AutoMod. The interface handles the text files through VBA macros and through additional code in the AutoMod model. A simplified flowchart over the communication is shown in figure 4.15. The interface creates text files and write input data to them according to what the user have specified in the input sheets. When all input is stored in text files, the main program will start the model through Windows command prompt. The model will read the input data from the text files and when finished simulating the model, write output from the model to text files. Before closing the model AutoMod will create a text file with the name “RunDone.txt”. While the model is running, the generic-UI will be waiting and constantly checking if the “RunDone.txt” exists, which works as checking if model is still running. When model have stopped will the generic-UI will find the “RunDone.txt” file, if it hasn’t been found the user will be informed that the model couldn’t finished or have been paused. If the text file have been created the generic-UI can instead proceed and read the output from the text files and write them into the output sheets accordingly to how the user have specified them. The communication is independent whether it is a 32-bit or a 64-bit model of AutoMod since it excludes the use of ActiveX and is done through Excel VBA code, use of text files and AutoMod model code.

In the Control Sheet in figure 4.16 the user can specify reading and writing ranges for input and output. Shown In figure 4.17 is an example on how the setup is done. In the first column, “Sheet Name”, is the name of the sheet containing the input selected. In the second column, “Textfile Name”, is the name of the text file where the input data should be written. The third and fourth column , “Starting Input Cell” and “Ending Input Cell”, determines the size and location of the matrix that should be written to the text file. And in the last column “Writing Method” the user can chose between two different writing methods. The first one “Write whole matrixes” takes the whole matrix formed by the ending and starting cell while the second “Write only unprotected cell” skips protected cells and writes only unprotected cells to the text file. The first row In the example shown in figure 10 forms together the writing information needed: Write the matrix B12 to P19 from the sheet Input1 to the text file test1.txt. The second rows writing information will instead be: Write all unprotected cells from B11 to ZZ (ZZ = the very last cell in sheet) from the sheet General to the text file general.txt. The design of the Control Sheet is something that the modeler are familiarized with since they have been using a similar setup matrix in the old interface. It is mentioned in the interviews that this way of doing reading and writing setup has been working well in the past and that it is a helpful tool. Familiarity is one of the design principles mention in section 2.10 and may help the modeler with using and leaning the new interface.

In addition to be able to define writing and reading ranges is the model name, path to the model, path to input text files and path to output text file specified in the control sheet, see figure 4.18. This information is needed for the generic-UI to establish the communication to the model. To make the options more visible has the laws of

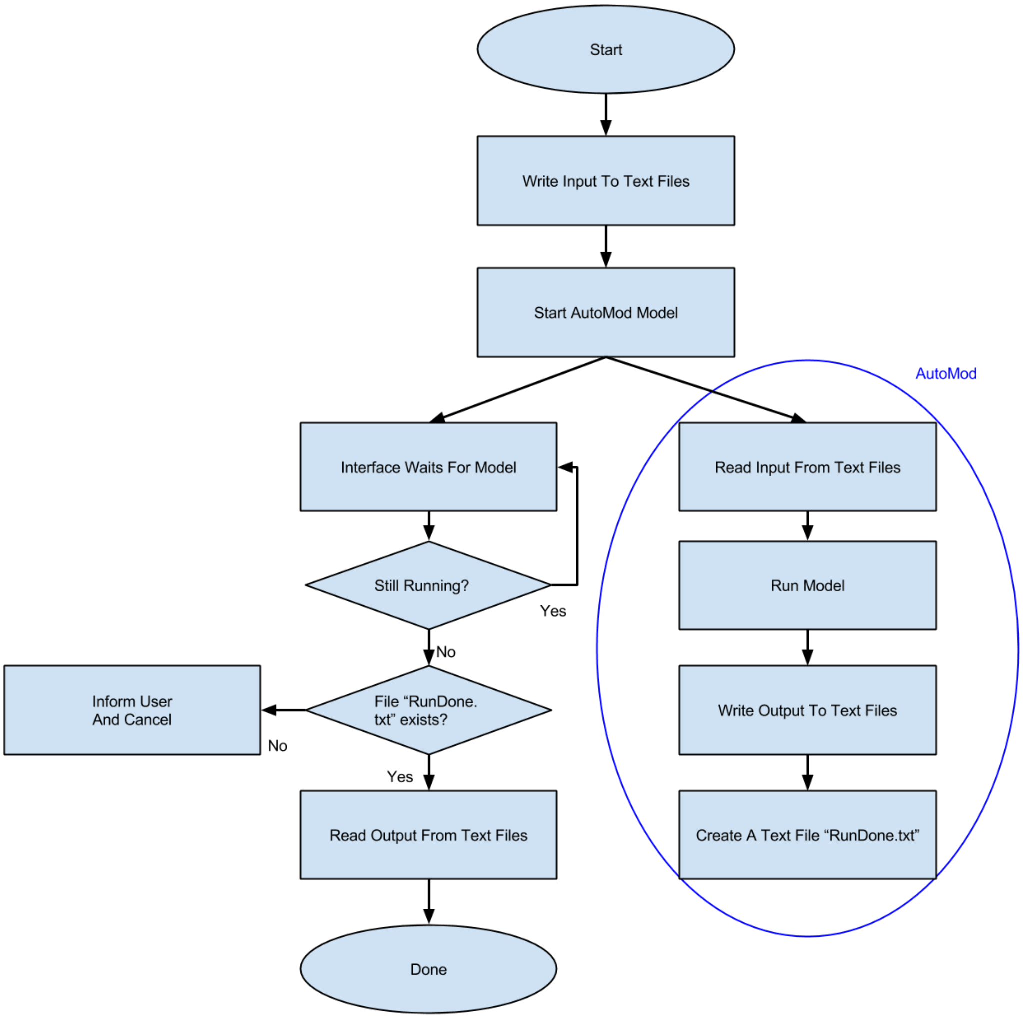
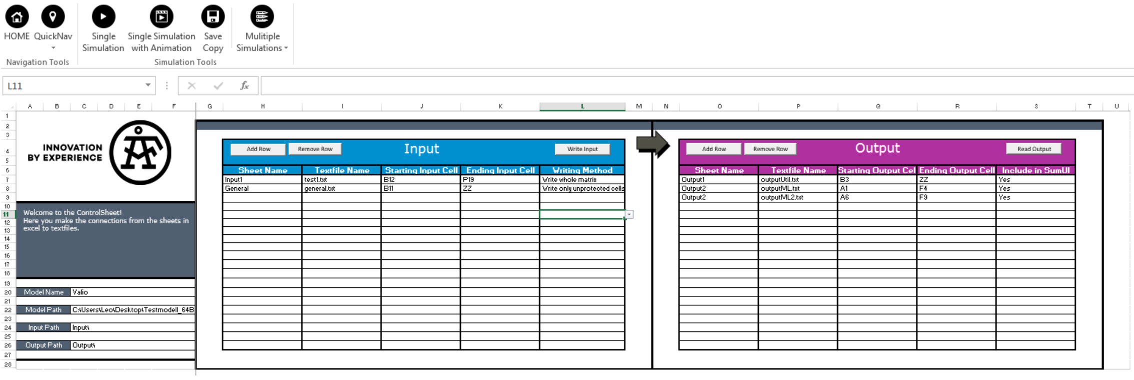

Start
Write Input To Text Files
Start AutoMod Model
AutoMod
Interface Waits For Model
Read Input From Text Files
Still Running?
Yes
Run Model
No
Inform User
And Cancel
File “RunDone.
txt” exists?
Write Output To Text Files
No
Yes
Read Output From Text Files
Create A Text File “RunDone.txt”
Done

Figure 4.15: Flowchart showing the procedures of running a model from the generic-UI.



Welcome to the ControlSheet!
Here you make the connections from the sheets in excel to textfiles.

Model Name	Valio
Model Path	C:\Users\Leo\Desktop\Testmodell_64B
Input Path	Input\
Output Path	Output\

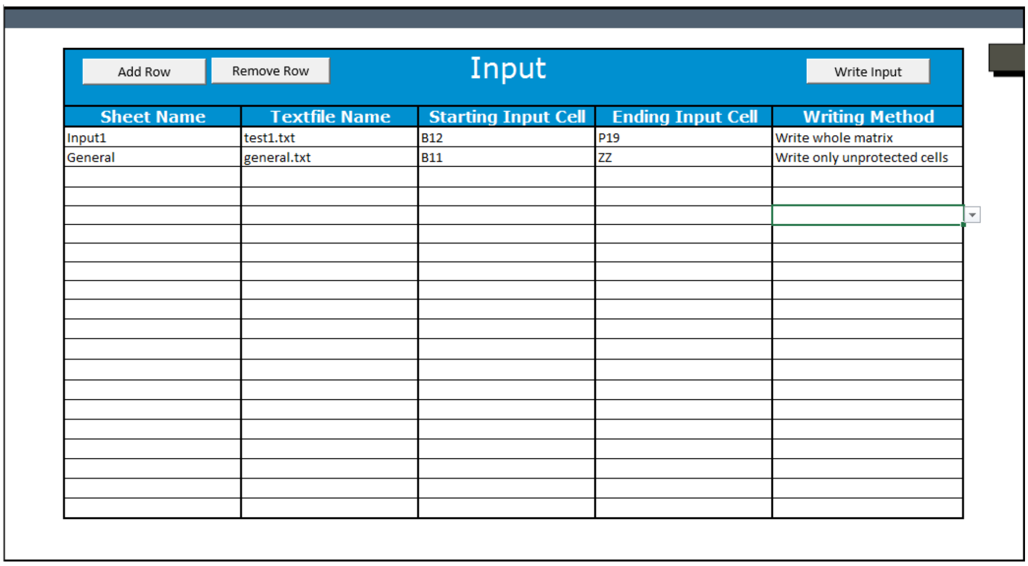
Input

Sheet Name	Textfile Name	Starting Input Cell	Ending Input Cell	Writing Method
Input1	test1.txt	B12	P19	Write whole matrix
General	general.txt	B11	ZZ	Write only unprotected cells

Output

Sheet Name	Textfile Name	Starting Output Cel	Ending Output Cell	Include in SumUI
Output1	outputUtil.txt	B3	ZZ	Yes
Output2	outputML.txt	A1	F4	Yes
Output2	outputML2.txt	A6	F9	Yes

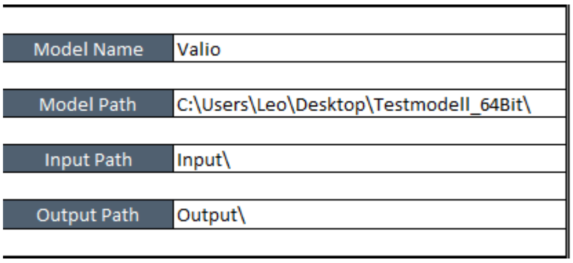
Figure 4.16: In the Control Sheet can the user defined reading and writing ranges for input and output



Sheet Name	Textfile Name	Starting Input Cell	Ending Input Cell	Writing Method
Input1	test1.txt	B12	P19	Write whole matrix
General	general.txt	B11	ZZ	Write only unprotected cells

Figure 4.17: The user specifies what sheet to write from, to what text file and the starting and ending cell of the data to write and finally which writing method to use.

similarity and proximity been considered. The four options with same purposes are grouped together and separated from other setup made in Control Sheet. The field where the name of a option is presented and the field where a option is changed are distinguished by the background colours slate gray and white. The two fields are organized next to each other to reveal their relationship. Consistency is important to easy for learning accordingly to the literature and the same design features for making a setup can therefore be found in other places in the interface for example the setup for multiple simulation see 4.2.



Model Name	Valio
Model Path	C:\Users\Leo\Desktop\Testmodell_64Bit\
Input Path	Input\
Output Path	Output\

Figure 4.18: The user specifies the name of the model, path to the model, path for input text files and path or output text file in order to connect the Generic-UI to the model.

5
Discussion

This thesis was mainly intended to solve the communication problem between Excel and the new version of AutoMod. At first it did not seem that easy to achieve. We had no real experience with programming this type of program. But after guidance from a contractor to ÅF it was not a major issue anymore. The finished communication solution was completed shortly after. The new communication solution is independent from ActiveX which was the main requirement from the modelers in order to be able to run large models. Other requirement from the modelers was to ensure that the communication is robust. Error handling is included in the communication to ensure that errors is avoided when can and that the user is informed when an error has occurred. The testing of the communication have been done through repeated trial and error scenarios and should preferably be tested further.

Because of the time limitations of this project there was no room for extensive testing of the new generic-UI. A head to head comparison seen in table 5.1 was used to support the argument that the new interface is an improvement.

Table 5.1: A head to head comparison of the old interface with the new interface developed in this project

Functionality	OLD	NEW
Standardization	No	Yes
Navigation Tool	No	Yes
Support 64-bit AutoMod	No	Yes
Standard Colour Coding	No	Yes
MultiSim	Yes	Yes
Setup for MultiSim	No	Yes
Analysis Tools	No	Yes
Automatic Graph Generator	No	No
Automatic Matrix Generator	No	No

There are several features that the users desired that are met by the new generic-UI but are not featured in the old interface. These also means that further testing is needed to validate the new generic-UI.

5.1 Software Development Method: The Work Cycle

The software development method in section 2.2 was used to develop the generic-UI. Perhaps it could have been done differently. However this was not considered since the literature provided a method that had been proven. Since this was not a conventional software, being locked to Excel, this software development method might not be the optimal. However the time limit for this project prevented us from building a new development method or doing further research on the subject.

5.2 Interviews and Affinity Interrelations Method

The literature clearly stated that the best way to develop interactive software such as the generic-UI is gather the needs through the users. Either by observations or interviews, as long as the needs come from them. By looking at different sources for software development, it often looked the same. The general approach is to gather information from the users, translate them into requirements, create the functions needed and then finally code it. It is a good way of working since the user knows what is wrong about the interface they used before. Also since the users probably have a lot of thoughts of how things could be done instead they could have solutions to the problems of the current interface.

One must also consider the ethical aspects of conducting interviews. The interviewees should feel that there will be consequences of they speak their mind. Informing of confidentiality in the beginning of the interview can help the interviewee feel more secure in telling the truth. Knowing that no one but the people present at the interview will know who is the source of the information gathered in the interview.

There was no real issue regarding collecting the needs from the users. However translating them into requirements where not as straight forward as we expected. We used methodologies for this but on placing the requirements on the correct level so to say was very challenging. Since the requirements should be translated into functionality in the system they must be translated to functions. The problem was that we ended up with doing too many requirements and it was not easy to work with. It was too time consuming to establish all the requirements on the level we first set out on. We knew from our affinity diagram what the problems were with the current interface. The problem was to interpret these into requirements that fulfill all the needs and at the same time be easily translated into functions that we could develop. The AIM is an effective tool for narrowing down many needs to a few sentences or themes. It also provides a prioritizing of these themes so the project can focus on the things most important. This is specially useful when on a narrow time span as this project. Not needing to attend all the needs and just being able to focus on the most important things in general saves alot of time but it comes with some drawbacks. By narrowing down all the needs one does also loose some information. Considering all the issues found in the AIM might have yielded a better generic-UI since then all issues would have been dealt with in detail. As of this project only the general issues have been dealt with and not all the details regarding these issues.

5.3 Usability

Since there was a lot of literature on what makes a good interface it was for us to sort out the things that was relevant to our project and that which we thought was possible for us to implement. Many of the problems with the old interface was related to how everything was presented. Therefore the first feature we worked on was the navigation. It was meant to organize all the sheets so the user would not be shocked by shear amount of sheets when opening the Excel document that is the interface. Since interaction can be conceptual, meaning that the user can recognize things that they are familiar with. Therefore we wanted the navigation be similar to a web page, since it is something that all working with interface are familiar with. As said in section 2.12 menus are good for navigation, which also supported our decision. Furthermore of the guidelines from section 2.10, familiarity, which states the importance of the user to recognize things they do not need to learn again. So they can focus their attention on learning the functions that are new to them. One guideline also states that consistency is important so the user can remember and learn easier. This is also achieved with having the navigation look the same for all sheets in the interface. To further strengthen the consistency a colour coding system was implemented. It is colours for different types of data. As seen in section 4.2 input is azure blue and output is fuhsia red, having these colour codes will always know what data or type of sheet they are dealing with. The buttons and matrices are also made with the gestalt laws from section 2.12 in mind. For example framing in buttons so one knows to what they belong, which corresponds to the law of proximity.

The work could be continued with programming the mathematics needed for DoE analysis. While doing this it would be appropriate to develop auto generating graphs from the results from the DoE analysis.

Many of the design decisions were made relatively quickly since time was limited. It was known from the start decisions had to be made so that development and coding could be done within the time span. Things might have been better if more time was given to development. Designing the whole system first before the first line of code is written. This was initially the plan as seen in the work cycle from section 2.2 But we had no experience with visual basic so the decision was made start coding early in order to learn the language.

The project also set out to answer the questions: How can a user interface be cognitively supportive for the development process of client specific interfaces? How should data be presented that is relevant for the clients and to satisfy their needs? These are not fully met as intended. However with the new possibilities with organize the sheets the clients will have an easier time finding what they are looking for. The modelers at ÅF can now sort out the relevant data relatively easy and give the client a button to access the data they want. The navigation helps to distinguish output data (raw data from the model) and results (compiled output data) which have been a problem in the past accordingly to the interviews. The positive effects is that the results sheets are a lot cleaner without the overwhelming amount of raw data and it is easier to interpret the results and for the modeler to include explanations of the results.

In section 4.1 we present what the modelers would like the generic-UI to help with. Certain desired effects have been harder to attain with the generic-UI. For the generic-UI to be supportive for accuracy and precision of what is presented is very difficult as every “to be developed UI” in the generic-UI have very different objectives and goals. One could argue that the multisim tool support accuracy and precision since the modeler can make advanced analysis with the help of it and easily present it in the UI. To establish a better work method and general guidelines could possible help with the accuracy and precision further. But due to time limitation and urgent technical problems have the main focus in this project been focusing on dealing with them as well as giving the modeler basic support in presenting their results rather than developing an extensive work method. Another wanted result from the generic-UI was easy realisation. The help with standardization of colours and visual design and the sheet handling system for navigation support the realisation for the the modelers to some degree. However would it be preferable to include further assistance for the modelers when it comes to visualize data since this was regarded somewhat problematic from the interview. This could be tools for creating standardized graphs or diagrams.

5.4 Sustainability Aspects

Simulation can be directly linked to sustainability. This project contributes to improving simulation as shown in table 5.1. Simulation can be used for improvement work, analysing an improvement before implementing it physically. This allows companies to test changes in their manufacturing systems without needing to stop them and loose potential sales. With simulation they can do more tests and more improvements and test them virtually. Once implemented physically they can lower costs and emissions. Thus making a smaller environmental footprint. This project’s contribution however small on a global scale can still help simulation by improving it for ÅF where this project was held at. Increased simulation of manufacturing systems can help companies with their sustainability work and by doing so at ÅF a contribution has been made.

6
Conclusion

The project set out to analyse a current interface for simulation at ÅF. Interviews and observations resulted in what was problematic about the interface, see section 4.1. The data from the interviews were analysed with the affinity interrelations method (AIM). The sorted data form AIM was used to generate requirements for the new generic-UI. These results from the interviews was used to create the requirements specification for the new generic user interface. Below is a reminder of the research questions for this project.

- How does the current interface solution work?
- What is laborious about the current interface?
- What are the characteristics of a good user interface?
- How can a user interface be cognitively supportive for the development process of client specific interfaces?
- How should data be presented that is relevant for the clients and to satisfy their needs.

The first two questions were answered by the interviews conducted in this project. During the interviews the interviewees also explained for the project participants how they used the interface and how it worked. The third question's answer came for an literature study of usability. The theory gathered about usability is in chapter 2. The fourth question has been addressed by creating a generic-UI which has cognitive functions built in. So the modeler at ÅF does not have to create these functions, such as the navigation from section 4.2.2. The fifth and final research question is answered by providing the modeler the ability to create dedicated result sheets that can be accessed through the navigation system. Providing the end user to quickly open the interface and look at the results or other things of interest.

More functions can be developed to give more cognitive support such as graph- and matrix generators. They where not developed due to time limitations of this project.

Bibliography

[1] Alänge, S. (2009) 'The Affinity-Interrelationship Method AIM'. Göteborg : Chalmers University of Technology

[2] Benyon, D. (2010) 'Designing Interactive Systems', Second Edition edn., : Pearson

[3] Bligård L. (2015) 'Utvecklingsprocessen ur ett människamaskinperspektiv ACD3-processen', : Unpublished

[4] Boyd-Brent J., Fraher R., (2010) 'Gestalt Theory, Engagement and Interaction', CHI 2010: Work-in-Progress (Spotlight on Posters Days 1 & 2), April 12–13,

[5] Chang, D., Dooley, L. and Tuovinen, J.E. (2002). 'Gestalt Theory in Visual Screen Design - A New Look at an Old Subject.' Available at: `http://crpit.com/confpapers/CRPITV8Chang.pdf` (Accessed: 5th May 2015)

[6] Courage, C., Baxter, K (2005), 'Understanding Your Users: A Practical Guide to User Requirements Methods', Tools, and Techniques

[7] Cowan, N. (2000) 'The magical number 4 in short-term memory: A reconsideration of mental storage capacity', BEHAVIORAL AND BRAIN SCIENCES, (24), pp. 87-185.

[8] Firesmith, D. (2004), 'Prioritizing Requirements', JOURNAL OF OBJECT TECHNOLOGY, 3(8), p.35 : ETH Zurich

[9] Green, W.S., Jordan, P.W. (2002) 'Pleasure with products: Beyond Usability', Taylor & Francis, London.

[10] Horton, W. (1997) 'Designing icons and visual symbols', Available at: `http://old.sigchi.org/chi97/proceedings/tutorial/wh.htm#U4` (Accessed: 5th May 2015)

[11] Hull, E., Jackson, K., Dick, J. (2005) 'Requirements Engineering', 2nd edn., : Springer

[12] Johannesen L, Potter S. S., Woods D. D., (1992) 'The Sophistry of Guidelines: Revisiting Recipes for Color Use in Human-Computer Interface Design', Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 4(36), pp. 418-422.

[13] Marcus, A. (1992) 'Graphic Design for Electronic Documents and User Interfaces'. ACM Press, New York.

[14] Preece, J., Rogers, Y. and Sharp, H. (2002) 'Interaction design: Beyond Human Computer Interaction', John Wiley & Sons, Inc., New York.

[15] Robertsson. S, Robertson, J (1999) 'Mastering the requirements process' Harlow : Addison-Wesley

[16] Scarr, J. , Cockburn, A. , Gutwin C. (2012) 'Supporting and Exploiting Spatial Memory in User Interfaces', Foundations and Trends in Human-Computer

Interaction, 6(1), pp. 1 -84 [Online]. Available at: `http://joey.scarr.co.nz/pdf/spatialreview.pdf` (Accessed: 19th May 2015)

[17] Wilson, C. (2014) ‘Interview Techniques for Ux Practitioners’, : Elsevier

A

Raw Interview Data

This appendix have all the answers collected from the interviews in this project. The interviews where in Swedish and all the following data is also in Swedish.
MÅL med intervju: Kartläggning av behov ur utvecklarna på ÅFs perspektiv. Nedan är frågor som skall ta reda på behoven som utvecklarna på ÅF och deras kunder har. För att kunna ta fram en mall för utvecklingen av ett gränssnitt ur ett behovs perspektiv.
Vad förväntas av mallen? För att hjälpa oss att kartlägga arbetsgången för att ta fram ett kundgränssnitt skulle vi vilja be dig att gå igenom din arbetsgång när du tar fram ett gränssnitt för en färdig modell. Tack. Varsågod att börja. (Fråga först om vi får lov att filma dem. Filma när utvecklaren går igenom arbetsgången i det nuvarande gränssnittet)
T.ex (fast mer i detalj):

- Skapandet av flikar
- Uppdelning av data
- Körning av modell
- Inskrivning av data
- Utskrivning av data
- Sammanställning av data
- Matriser

Hur väljs indata/utdata?
ÅF employee 2:
Vad behöver modellen och vad modellen ger Indata
Jobbar med inputflikar parallellt med kodning av modellen.
Börja med att göra inputfliken.
Skriver in indatan i fliken (ibland finns datan i en databas-flik och länkas på ett strukturerat in i indata-fliken)
Låser alla celler och sen låser upp celler som innehåller indatan.
Designar indatatabellen efter eget tycke och så att den skrivs in "rätt" i textfelen för autmodkodningen. (färger etc)
Skapar textfilen och kopplar den till autmodkodningen samt excel-fliken som innehåller inputdatan. Skriver in vart läsningen ska börja och sluta i fliken.
Skriver över datan till automod och kontrollerar så den används som tänk i modellen.
Utdata:
Välj den data som ska skrivas ut från AutoMod (Enligt syftet med projektet)
Skapa textfil
Koda utskrivningen av data i AutoMod till textfil
Kolla så datan är strukturerar i textfilen

Skapar en utdata-flik.
Kopplad den i control-sheet, Definierar var utskrivningen börjar och slutar.
Designar utdata-tabellen efter eget tycke. (Färgkodning, etc)
Kör modellen och kolla så utdatan hamnar där den är tänkt i utdata-tabellen.
Presenterar utdatan
ÅF employee 1:

I ÅF employee 1s projekt valde de tillsammans med kunden hur det skulle se ut. Men de hade ingen tanke. Syftet med deras projekt säger vad de vill få ut på utdatan, tex capacity. Efter kartlagt projektet sitter man med kunden för att välja. Hade varit lättare med grafer om det sett likadant ut varje gång. Men det ser ju inte likadant ut för olika kunder. Indata varierar mellan projekt. Hade varit bättre om man kunde bara välja de olika fälten. Bestämma om det ska vara låsta, färgade osv. ÅF employee 1 byggde upp sin indata till sitt exjobb bara från scratch. De lista bara vad de behövde. De tänkte bara på vilka siffror de behövde. Tänka på att de tex inte mata in "mean time to failure", utan det var ett KPI. Så man har kolla på vad som skall matas in.
ÅF employee 3:

Man får oftast färdiga "mallar" med indata som man helst ska behålla formatet. Utdatan kommer man fram till med kunden.
Grafer, hur väljer ni vilka som passar?
ÅF employee 3:
Det kan hända att de får lägga till saker för de kommer på senare att de ville ha det. De ger ofta annars färdiga indata. Så bygger de indata flikarna efter indatan som fås av kunden.
Vilka är enligt dig de grundläggande behov/funktioner som mallen måste ha?
ÅF employee 2:
Skriva in och utdata. Start, stoppa modellen.
Formatering av saker, som att skapa en ny flik. Se till att det är rätt färger, bakgrund, antal rader, kolumner osv. Kan nog tillägga att det skall fungera på både svenskt och engelskt excel. **ÅF employee 1:** Hon tänker på resultatdelen. Kunna

visa resultatet på ett smidigt sätt. Kunna generera grafer snabbt och enkelt. Hon känner att hon "är dålig" på grafer i excel, bråkar med var axlar ska va, vad de olika färgerna betyder. Få hjälp med färger osv. Hon tycker det skulle vara bra om färger för förutbestämda (likt för varje projekt) tex grönt för running, blått för idle, rött för down. Bästa scenario trycka på en knapp och så får man en graf. Kunna köra multisim med stora modeller. Kunna titta på graferna och förstå vad de säger, så de kan ta rätt beslut snabbt. Bra grafer med andra ord.
ÅF employee 3:
Finns inga. Kan bli svårt med autogenereade tabeller. Inte skriva över formatering när man länkar till data. Man kan ha en sammanställningsflik.

Vem använder gränssnittet hos kunden?
ÅF employee 2:
Bara fabian som använder det. För att hjälpa ÅF employee 2

ÅF employee 1:
Vet ej. Vet inte hur ofta de använder det som förbättringsverktyg. Linus har en kund som gör detta dock. De flesta använder det bara som ett beslutsunderlag vid investeringar tex.
ÅF employee 3:
De vill ha möjlighet att ändra på indata men de burkar inte göra det.
Vem beställer gränssnittet?
ÅF employee 2:
Chefen över PLC. **ÅF employee 1:**

Nån slags projektledare. Nån logistikchef, produktionschef, systemchef. chef chef
Vem förhandlar ni med när ni utvecklar gränssnittet?
ÅF employee 2:
Chefen över PLC. Han pratar inte med någon. Han får ritningar osv för att kunna skapa modellen. Sen anpassar han den efter hans eget behov. **ÅF employee 1:**

Va inte så delaktiga i hennes projekt, kunden var kunniga i simulering och indata och vad som skulle stimuleras var redan speificerat från kunden. Men just utvecklingen av gränssnittet var de inte alls med i.
Vad är bra med nuvarande mall? Några positiva berättelser?
ÅF employee 2:
Finns väl många saker som är bra. Han hade sett massa gränssnitt innan så var det inte så svårt att fatta. Men SKF's gränssnitt ser inte bra ut tex. **ÅF employee 1**

De som är bra är det som är förberett, I control sheet tex. Det är som är lite mall är bra. Man kan tex skriva in vilka filer man ska skriva till och vilka filer man kan läsa från dessutom bestämma vilken sheet man ska läsa/skriva till/från och mellan vilka celler.
Vad är dåligt/mindre bra med nuvarande mall? Några negativa berättelser?
ÅF employee 2:
Alla nya popups, som hamnar i bakgrunden så han måste klicka upp alla sina fönster. Jäääääääääte störigt. Detta händer när man "kör modellen med animation".
ÅF employee 1

Finns ingen struktur på hur det ska se ut. Att man får komma på allt själv. Det blir problem när man glömmer lägga in saker. Att man inte kan enkelt gå in och hitta saker som ska ändras eller läggas till. Alla funktioner fungerar inte för alla modeller. Multisim. Och att köra modellen från ecxel med stora modeller fungerar inte utan man måste skriva till och från textfiler för hand.
Vilka krav/behov tror du det finns från kunden sida på gränssnittet? Vilken data som presenteras? Hur det presenteras?
ÅF employee 2:
De måste kunna förstå vad som händer (tolka de) och ta del av den utdatan. Inte "onödiga saker". Man måste anpassa också det efter kundens kunskaper. Vem som skall använda det, om de nu ska det. Tex med SKF, väldigt mycket indata och utdata. Väldigt svårt att förstå. De hade en som förstod gränssnittet men han jobbar här nu (Pär)

Är kunden delaktig i utvecklingen av deras gränssnitt?/Hur är de delaktiga?
ÅF employee 2
Vissa, de som kan något. De borde vara. **ÅF employee 1:**

De är inte med så mycket. Det beror på hur simuleringskunniga de är. De vill förmodligen bara ha en knapp att trycka på så ska resultatet visa sig. Men om de ska utföra något experiment måste de kunna förstå indatan. Det måste vara anpassat till dem så de förstår. Hur är de vana att se deras data? (Sladdljud) Kommer de med för mycket synpunkter kanske hon måste anpassa inläsningen av indatan.
Hur kan kunden involveras (ytterligare)?
ÅF employee 2:
Kunden vet vilka in och utdata de vill ha, ibland. Beroende på kunskap. Det finns ren modellteknisk data som de inte behöver vara delaktiga i. **ÅF employee 1:**

Involvera dem mer om de ska använda det mycket som förbättringsverktyg/planeringsverktyg. De förlitar sig nog på oss att vi vet hur man gör det användarvänligt för dem.
Hur kan mallen sen ut för att underlätta ditt arbete?
ÅF employee 2:
Kommer upp fönster som frågar efter saker man ska mata. Så man slipper skapa det själv.(vart tabellen ska komma, namngivning, direkt koppling vart läsningen börjar och slutar, slippa skapa textfiler, etc) Men det måste finnas möjlighet för att ändra manuellt om något extra behövs (lägga till rader och kolumner, ändra läsnings referenspunkter etc). **ÅF employee 1**

Är det inte ni som ska komma på det? Men ÅF employee 1 vet inte. Känner att hon är för dåligt insatt. **ÅF employee 3:**

Standardisera utdata flikarna kan göras.
Vad är vad tar lång tid och är omständligt?
ÅF employee 1
Alla jäkla tabeller och grafer. Vi köper det säger Leo. **ÅF employee 3:**

Att veta var man ska läsa indata, start och stopp celler. Själv låsa och låsa upp celler.
Hur skulle den ultimata mallen se ut/fungera? No limits in technology
ÅF employee 2:
Kunna fylla i smidigt vad man vill skapa. En smart lösning till checkboxar (med true och false), de är jobbiga att göra just nu. Alltså ett verktyg för att välja inställningar till modellen. **ÅF employee 1:**

Det här med tabeller. Bara skriva in vad hon vill ha för tabell. Bara skriva in hur många artiklarnummer man har tex sen autogenerera tabeller. men dessutom kunna hantera det manuellt. Det ska va lätt att ändra eller lägga till om något blev galet eller om man kommer på något nytt. **Hur ser det ultimata gränssnittet**

ut för kunden?
ÅF employee 1:
Hon tror det är väldigt viktigt att klara göra vad all rådata faktiskt betyder. I

tabeller med massa utdata säger inte så mycket och det är viktigt för både kunden och oss att presentera det på något sätt (grafer) som visar innebörden av utdatan.
tekniska lösningar - extra diskussionspunkter ActiveX? Hur var det med C-lösningen?
Finns det några alternativ förutom Excel? (förmodligen bara Linus som vet)
Mer specifika aspekter att samla in data om, är listade nedan:
Systemmål av template Definiera indatavalen Skrivning av indata till modell * Skapa inmatningsfunktionen av indata Definiera utdatavalen Läsning av utdata från modell * Skapa utmatningsfunktionen av utdata Hantering av utdata Starta modellen Stoppa modellen
Systemmål av ett kundgränssnitt Val av indata Körning av modell Multisimulering Val av utdata Val av presentation
Tekniska lösningar som används idag
Excel - Hantering av indata/utdata - Presentation ActiveX - Kommunikation mellan Excel och Automod
Tekniska lösningar som kan andvändas i framtiden
C-lösning som användes förr

B

Affinity Summary

Index (1) is most important (2) second, (3) third etc.

Technical Issues
Technical Issues that must be resolved (1)
Minor bugs and annoying technical issues

Result Presentation
Understanding the results quickly (2)
Knowing what to present to customer
Making graphs understandable

Lack of standards and guidelines
Lack of design/visual standards (3)
There is no sheet standards (4)
Lack Of guidelines for usability

Information Structure
Its hard to find the sheet one is looking for (4)
The sheets are not self explanatory (3)
Too much info on a single sheet

Developer Toolbox
Time saving tools for the developer or time consuming activities
Creating graphs

Project Variations
The project differs a lot from each other
The project can be very dynamic
Customer involvement and knowledge varies
the indata must be adapted for the customer

C

Requirements Specification

Must have requirements

Table C.1: This is the requirements on the Generic User Interface. The corresponding functions are related to all of the requirements listed to their left and not only the ones they are in line with.

Requirement	Corresponding Functions
Simulate large models that require 64-bit. Simulate 32-bit models. Run multisim.	Start AutoMod from Excel and run a selected model. Handshake between Excel and AutoMod, that AutoMod is running and when the simulation in AutoMod is done. Read Input from excel to AutoMod. Write output data from AutoMod to excel.
Easy setup for multisim.	-
Handle data intuitive in excel, easy to define reading and writing matrices	-
Only read relevant data between excel and automod.	-
Have a all dev-tools in a "toolbox", easy access to devtools	-
Hide irrelative data/information.	Selecting to only show input sheets.
Only show relevant data/information.	Selecting to only show output sheets.
Intuitive navigation.	-
Error handler.	-
Table generator for general use.	-
Make general/common graphs easily.	-
Be able to edit tables/graphs/matrices manually	-
Help with color scheme	-
Help with visualization standards (units, fonts, structure, labels, explanations/comments)	-

Should have requirements
Simulation progress display
Easy to make checkboxes

Would be nice to have requirements
following are requirements directly from the modelers at ÅF.

ÅF employee 1
Autogenerara Tabeller efter val av kolumner, rader osv
Kunna manuellt ändra på tabeller, smidigt sätt.
Autogenera Grafer, efter val av axlar osv
Hjälpa till med felsökning.
Kunna ändra grafer manuellt på ett smidigt sätt.
Graferna skall kunna användas som beslutsunderlag.
Kunna anpassas efter kundens kunskapsnivå.
Tydliga grafer.
Ge hjäp med färgval.
Kunna köra multisim med stora modeller.
Hjälpa till med användarvänlighet.

ÅF employee 2
Skriva in- och utdata
Färdiga formateringar på flikar
Fungera på olika verstioner av excel
Kunna skapa lättförståliga grafer som kunden kan tolka
Kunna anpassa graferna från krav 4 så de är efter kundens kunskap
Inte behöva öppna upp de popupfönster man behöver.
Inte flyttas till nytt fönster när man inte vill det (tex när man skriver indata eller utdata så hoppar excel-dokumentet till den senaste aktiva fliken som skrivits från eller till, bättre om det stannar kvar i control sheet)
Kunna ändra i tabeller/indata manuellt.
Kunna skapa chechboxar enkelt (kunna skapa kontrollpaneler är kanske bättre, inte bara checkboxar utan fler typer av “knappar” för att styra modellen
Kunna enkelt skapa flikar där man kan ange vart tabellen skall komma, namngivning, direkt koppling var läsning börjar och slutar.
Kunna skapa snabblänkar

ÅF employee 3
Kunna använda färdig indata som kommer från kunden.
Inte “sabba” formateringen när man länkar data.
Förenkla skapadet av grafer, tex har man 90 dagar vill man fördela det på 90 staplar. Alltså slippa ändra på dessa villkor manuellt.
Kunna lägga till saker som kunden kommer på senare under projektets gång.
Veta vad kunden behöver om de inte själva vet det.
Mall/hjälp med presentation av utdata.
Standardisera för olika industrier
Förenkla låsning/upplåsning av celler.