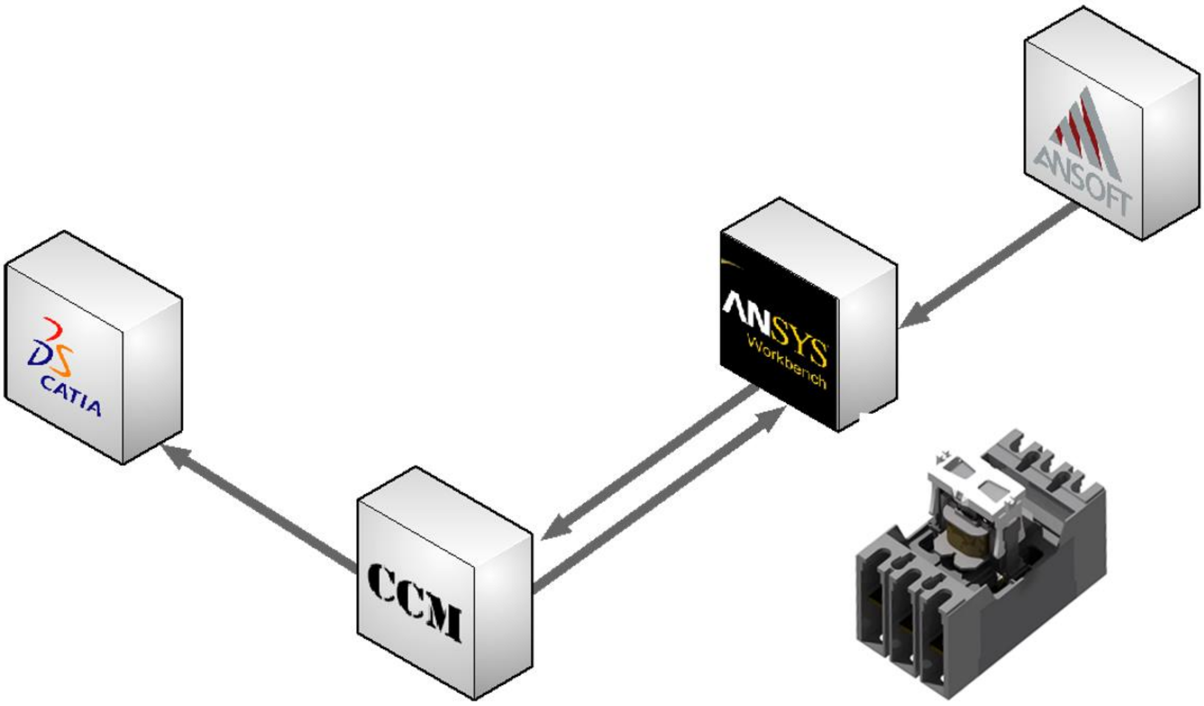
CHALMERS
UNIVERSITY OF TECHNOLOGY



Modelling and Simulation of a Product Family of Contactors using Configurable Components

Master thesis in product development

DAVID RENBORG

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Modelling and Simulation of a Product Family of Contactors
using Configurable Components
DAVID RENBORG

PPUX60 – Master's Thesis no 2016:06
Department of Product and Production Development
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover:
The cover illustration graphically portrays the PDM architecture, which is the core of the automated generation process for the product family of contactors.

Printed by
Chalmers Reposervice
Gothenburg, Sweden 2016

Modelling and Simulation of a Product Family of Contactors
using Configurable Components
David Renborg
Department of Product and Production Development
Chalmers University of Technology

Abstract

The purpose of this master thesis was to provide ABB with a product platform development solution based on the methodology of configurable components. The considered platform was limited version of ABB's product family of AF-contactors. The product family is divided into series, where each series handle a certain range of units. In the case of this master thesis, three contactor series were in focus. The currents included in these series were: 400A, 490A, 570A, 630A, 800A, and 900A.

This master thesis provides evidence that a platform of AF-contactor can be described as configurable components by utilising the methodology through the provided CCM software. The research model that has been used is derived from the synergies of combining theory and empiricism. Technical knowledge and support regarding contactors have been acquired from engineers at ABB, and regarding the methodology of configurable components it has been acquired from literature and scientists at Chalmers University of Technology.

The result of the thesis is a PDM architecture that automatically can generate multiple contactor variants in order to create a platform. The configurable component model function as a core, controlling external software which perform simulations on each variant. The generation of all variants included in the scope had a duration of six hours and resulted in 1000 variants with attributes comparable to contactors in production today at ABB.

Finally, for further development, it is suggested to continue with implementing screening methods in the generation process to sort out the non-feasible variants and study possible algorithms for optimising the contactor variants included in each series

Keywords: configurable components, platform design, contactor, CCM, Chalmers, ABB, COPE Sweden, product development

Acknowledgements

This master thesis has been carried out by a student at Chalmers University of Technology with a bachelor in Automation and Mechatronics and a master in Product Development. The project has been both interesting and educational. It has been a very exciting period and I am grateful that I was given this chance to really work in the frontline with a methodology that I think will be very beneficial for product developers in the future. The project has extended my understanding of both platform design and product development processes. It has also given me the opportunity to meet new interesting people.

First of all, I would like to thank professor emeritus Hans Johannesson at Chalmers University of Technology and Peter Edholm at PE Geometry AB / COPE Sweden AB for giving me the opportunity to perform a 60 HP master thesis, especially Hans, who also functioned as a sounding board when I encountered obstacles in designing the platform.

Secondly, I would like to thank Magnus Andersson at PE Geometry AB / COPE Sweden AB. He always made time when I needed to discuss the CCM software and was very responsive if there was any need of change in order to enhance my model.

I also would like to thank the following people:

The engineers at ABB, Lars Andersson, Gunnar Johansson, Alfred Nilsson, and Jonas Larsson. I could not have achieved this without your support and knowledge.

Göran Brännare and his crew at Chalmers University of Technology, after staring at my model for some time I could always take a break and have some laughs with you.

Jonas Landahl, PhD student at Chalmers University of Technology, for enjoyable lunches and CCM discussions.

Finally, I would like to thank my father, Dag Renborg, for his support and that he tirelessly has been insisting that I must complete what I have started.

David Renborg, Gothenburg, June 2016

Abbreviations

A	– Ampère
C	– Constraint
CC	– Configurable Component
CCM	– Configurable Component Modeller
CI	– Control Interface
COPE	– COPE Sweden AB
CS	– Composite Set
DR	– Design Rationale
DS	– Design Solution
EF-M tree	– Enhanced Function-Means tree
F-M tree	– Function-Means tree
FR	– Functional Requirement
Icb	– Is constrained by
Ipmb	– Is partly met by
IS	– Interface Set
Isb	– Is solved by
Iw	– Interacts with
PDM	– Product Data Management
Rf	– Requires function
VDP	– Variant Definition Parameter

Table of Contents

1. Introduction

This chapter provides the background to the master thesis, the purpose of the thesis, a description of the case, and a description of the problem to solve. The actors and stakeholders of the thesis are also presented, as well as the delimitations.

The master thesis was based on an on-going project at Chalmers University of Technology (Chalmers). The purpose of which is to establish efficient methods and processes for product platform development that focuses on virtual verification and has its origin in object oriented platform systems development.

1.1 Background

Platform based development focuses on commonality and the reuse of parts when developing products. A more comprehensive definition of a platform is that it is a "collection of assets that are shared by a set of products" (Robertson & Ulrich, 1998). It provides a number of benefits regarding the development process and the manufacturing process, e.g. the company is only required to develop certain parts of the product in order to acquire a new variant. This way the company obtains the possibilities of both customisation and economies of scale (Johannesson, 2014). However, the usage of platforms may limit the further development if there is a usage of already designed parts. Another extreme is that, by using the already designed parts in another context, the need of modification may emerge. That could result in a chain of modifications and then the benefits of the economic scale may be lost (Johannesson, 2014).

A way of treating the mentioned disadvantages is to extend the platform thinking and take the design information, meaning not only the design of the parts but the design rationales and design histories into consideration as well. A design rationale includes the design decisions as well as the justification of those decisions and the design history of the recorded design process. Hence, by using this extended view of a platform an approach based on system thinking and object orientation is derived. The approach, based on so called Configurable Components (CC) is reported in the literature, e.g. (Johannesson, 2014).

In order to utilise this platform view a software tool has been created, the Configurable Component Modeller (CCM). This software is intended to function as a core for CC based development of platforms and product families. It is the CCM software that functions as the user interface and contains the CCs as well as their interrelationships. However, to be able to perform development processes and to establish a product lifecycle management (PLM) environment other software tools have to be included as well, e.g. computer aided design (CAD) and product data management (PDM) tools (Johannesson, 2014).

In this master thesis the product of interest is a contactor. The contactor will both serve as a base for the product platform development solution provided to ABB and as a case to facilitate the development of the CCM software. The contactor, presented in Figure 1.1, is an electrical unit which is used to switch currents on and off and is controlled by a microcomputer. It consists of a number of subsystems, the main ones are: electromagnet, electronic control system, chassis for contact parts, chassis for operation parts, spring system, contact system, contact bridge and arc control device. A contactor is used in multiple applications where the current in the circuits must be controlled and the applications often have shifting loads. This can result in sudden changes of the current which lead to that the contactor must be able to withstand a current overload during a limited time; both factors are depending on the current it is designed for. The contactors' lifetime is limited by the lifetime of the contact plates[1].

Figure 1.1 – AF contactor (ABB, 2016)

1.2 Purpose

The purpose of this master thesis was to provide ABB with a product platform development solution and to further develop the utilisation of the CCM software, with respect to the contactors belonging to ABB. Included in the purpose is also to support as large parts as possible regarding the development process. In order to achieve that, the already existing CCM-model of the contactor had to be completely described in the CCM environment and thus enable the process of creating the series of contactors. Models of the contactor were created and further developed in Catia V5 (Catia), Ansys Simplorer (Simplorer), and Ansys Maxwell (Maxwell), in order to enable simulations and analyses of the contactors' attributes. The results and data of the analyses were integrated with the CCM environment.

In order to achieve the purpose, two main objectives were set:

- Platform preparation: Complete the models of the contactor, with respect to the different software.
- Platform execution: Utilise the CCM-model to create series of contactor variants, as well as ensure easy handling and interaction.

[1] Gunnar Johansson (Senior Principle Engineer, Control Products, ABB AB), meeting 140626

1.3 Case Description

In order to facilitate the usage of the CCM software, Chalmers has created a platform case in collaboration with ABB through the Wingquist Laboratory VINN Excellence Centre. The case was concretised by the development of ABB’s contactors. The contactors within ABB’s AF family are based on standard motor sizes, each motor size corresponds to a specific current, and the range of the family is between 25A and 2650A.

The AF product family is divided into series, where each series handle a certain range of units. These series are based on the size of the electromagnet and the chassis, seen in Figure 1.2, and the highest current of that series dimensions the electromagnet used for the particular series. In the case of this master thesis three contactor families were created, the currents included in these three families were: 400A, 490A, 570A, 630A, 800A, and 900A.

Figure 1.2 The conceptual AF product family

1.4 Project Scope and Approach

The scope of the project was broken down into different tasks to achieve a more manageable overview of the thesis. This work was performed with an object oriented approach, meaning that the physical parts was analysed on a higher level of abstraction. The work was initialised by mapping the contactor, in order to acquire the answers to the following:

- What functions does a contactor have and how are the functions related to each other?
- How should the functions be broken down into a function-means tree?
- What parts does a contactor consist of?
- How does a contactor behave, with respect to mechanics and electronics?

When these questions were answered the most important information regarding contactors had been gathered. Based on this knowledge the models were created, leading to the part of the thesis that focused on the software and how the different software could interact with each other. In order to examine the interaction possibilities between the software the following questions were answered:

- What input parameters does the different software need in order to perform their tasks and what output parameters does the different software generate?
- How should the contactor be described as a CCM model?
- What rules to configure variants must be defined?
- How should the flow for the designing of a contactor be defined?

Thus, by having answered the questions stated above the two objectives mentioned in section 1.2 was completed and thereby the aim of this master thesis was fulfilled.

1.5 Delimitations

In this master thesis, the development of the utilisation of the CCM software has been constrained to treat the development of three series of the AF contactor. In order to be able to use the CCM software as a core, the surrounding software had to provide relevant outputs that could be utilised by it.

Due to the pre-existing contactor system the platform development that has been performed was highly tainted by the already existing solutions. Therefore, it was more oriented on identifying functions to the solutions as well as mapping the system with respect to the theoretical framework.

The contactor was not examined on a detailed behavioural level; only the characteristics of the contactor were examined. Attributes regarding the contactor was constrained to one of the magnet types that already was in use today and the width of the electromagnet was limited to from 0.08 to 0.14 m. The model of the contactor is also limited in the perspective of breaking technique, which is disregarded in this thesis.

Regarding the CCM software and the model, functions such as interface and interactions between objects have been unavailable. Due to the limitation of time, this master thesis did not focus on the optimisation of the CCM software.

2. Theoretical Framework

In this section, theory of which the master thesis is based on will be presented. First will platforms be described, what a platform is and how it can be designed. Then will the concepts of configurable components, and what configure-to-order is, be described. Finally, theory regarding the contactor will be presented and the analysis of the physical contactor.

2.1 Product Platforms

As mentioned in section 1.1, a platform can be defined as “a collection of assets that are shared by a set of products” (Robertson & Ulrich, 1998). Hence, by having a core product and applying a platform strategy with shared assets, e.g. modularity or scalability, a company can acquire a more flexible portfolio of products to meet customer needs and market changes (Johannesson, et al., 2016).

By looking upon the platform in terms of knowledge and information instead of physical components the product can be described on a higher level. This unlocks a design process based on functions and solutions, meaning what the product is supposed to fulfil and how it can accomplish that functionality (Edholm, et al., 2016). Using this functional view on a product platform systems emerge, and through further analyses subsystems that are similar to each other can be identified. These subsystems encourage reuse of functions when designing more variants in the product platform family.

2.2 Configurable Components

The concept of CCs forms a structured technique of encapsulating generic information defined in systems and subsystems, where each CC corresponds to a system (Gedell, et al., 2011). Thus, it leads to making flexible interfaces usable as a substitute for rigid system architecture when modelling a platform (Michaelis, 2013). With the purpose of facilitate that way of platform modelling a number of principals and constructs should be followed (Gedell, et al., 2011).

In order to provide the CC with its functionality, to fulfil the conceptual description, certain features are required. These features are adopted from Gedell (2011) and described by Michaelis (2013) as function-means (F-M) trees, control interface (CI), composition set (CS), interface set (IS) and variant definition parameters (VDP).

2.2.1 Enhanced Function-Means Tree

Depending on the desired level of detail, the initial system can be broken down into subsystems and each subsystem can itself be derived further until the adequate level is acquired (Gedell, et al., 2011). The information that is contained in the system is only used within the CC by default and is represented by a F-M tree. It consists of the elements functional requirements (FR), design solutions (DS) and constraints (C), (Michaelis, 2013). The difference between a function-means tree and an enhanced function-means (EF-M) tree is mainly that the EF-M tree also describes the relationships between functions, solutions, and constraints, adopted from Schachinger and Johannesson (2000) and described by Michaelis (2013).

The first element, FRs, define the desired function of the CCs on the system or subsystem level. The FR is built up by parameters in order to recognise the functionality. Thus, these parameters describe the bandwidth of *the functional properties* of the system. (PE Geometry, 2015)

The second element, DSs, are the corresponding solutions to the FRs and concludes the design decisions, each FR has a specific DS coupled to it. Similar to the FR, the DS is built up by parameters to provide substance to the solution. These parameters describe the bandwidth of *the physical characteristics* of the system. (PE Geometry, 2015) The DS can both be in the shape of low-level details as well as a more abstract concept, or anything in between (Gedell, et al., 2011).

The cardinality between FRs and DSs is defined as

$$FR: DS = 1: 1$$

and

$$DS: FR = 1: n, n > 1, n \in R$$

Meaning that a FR can only have one DS coupled for a specific variant, meanwhile a DS can cause several FRs when decomposing the system further (Johannesson & Claesson, 2005). Although, there is an exception, a number of DSs can be coupled with the same FR in a case where the active DS for a specific variant is decided by the bandwidth of the FR (Levandowski, 2014).

The third element, Cs, addresses requirements that do not have any impact on the function (Johannesson, 2014). However, they have impact on the number of feasible solutions for the system (PE Geometry, 2015) and are also built up by parameters defining its bandwidth. Cs can influence DSs on different levels, meaning that one C can cover all DSs in the system as well as one particular DS.

These three elements form the *design rationale* (DR) of the system; the reasoning and justifications behind the solutions (Johannesson, 2014).

Also presented in the DR are the labelling of the relationships in the EF-M tree (Schachinger & Johannesson, 2000). These are the following: *is solved by*, describing which DS that solves a certain FR; *requires function*, describing which FR a certain DS cause; *is constrained by*, describes which C that constrains a certain DS; *is partly met by*, describes a DS that partly fulfils a C; *is influenced by*¸ describes a DS that affects but is not the solution for a FR; *interacts with*, describes a DS that has impact on another DS, this one is strongly connected to the *is influenced by* relationship and vice versa. A descriptive illustration of the relationships is presented in Figure 2.1.

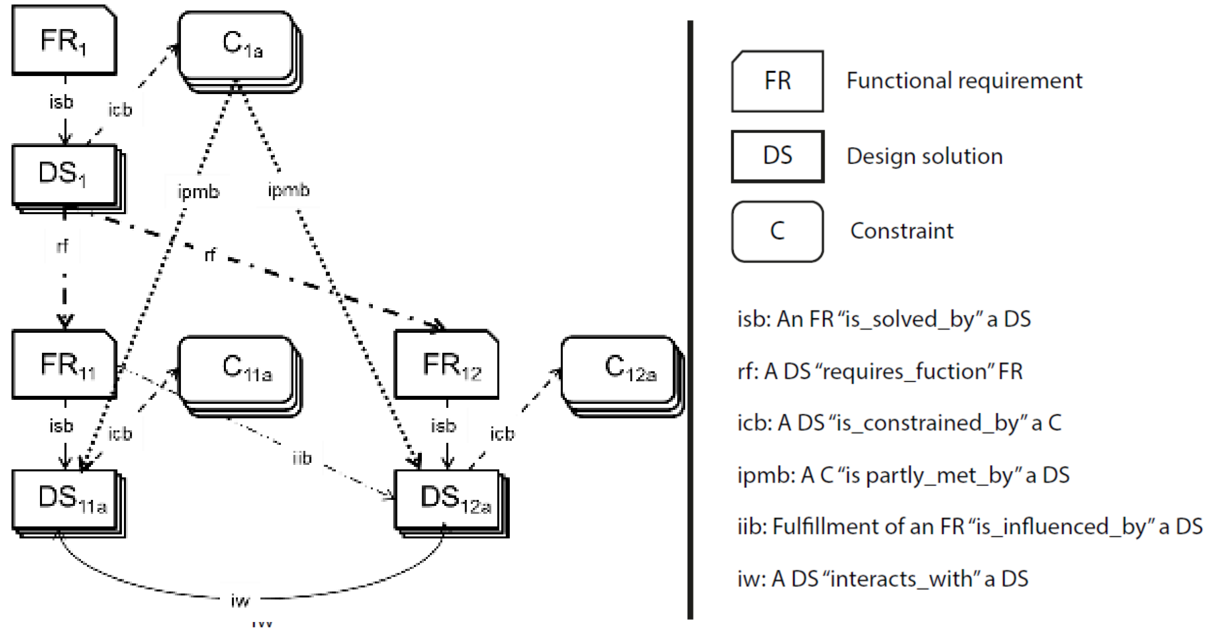


Figure 2.1 Fundamentals of the design rationale. Adopted and altered from (Levandowski, 2014)

2.2.2 Control Interface

The CI of a CC allows the user or a parenting CC to set the parameters values of the CC. It enables the possibility to create a link between the global parameters and internal parameters with respect to a specific CC.

2.2.3 Composition Set

The CS determines the other CCs used to further define and realise the functionality of the considered CC. These CCs represent subsystems to the main system.

2.2.4 Interface Set

An interface (IF) is a specific design solution that interacts with its counterpart in another CC (Michaelis, 2013), these interactions can have both physical and non-physical representations.

2.2.5 Variant Definition Parameters

As a result of the generic system, a CC can develop into different variants depending on the parameter values; it can be set both in the DSs and as input to the CC. Each way resulting in a unique variant (Gedell, et al., 2011).

2.3 Platform Preparation and Platform Execution

When designing a product platform along with implementing the concept of configurable components the process can be split up into two parts, platform preparation and execution (Johannesson, et al., 2016). Accordingly, the platform preparation is about building the model with the desired product information as well as the bandwidth of the platform. The platform execution on the other hand is about facilitating the information when creating variants bound by a bandwidth. In turn, both parts comprise a number of steps, see Figure 2.2.

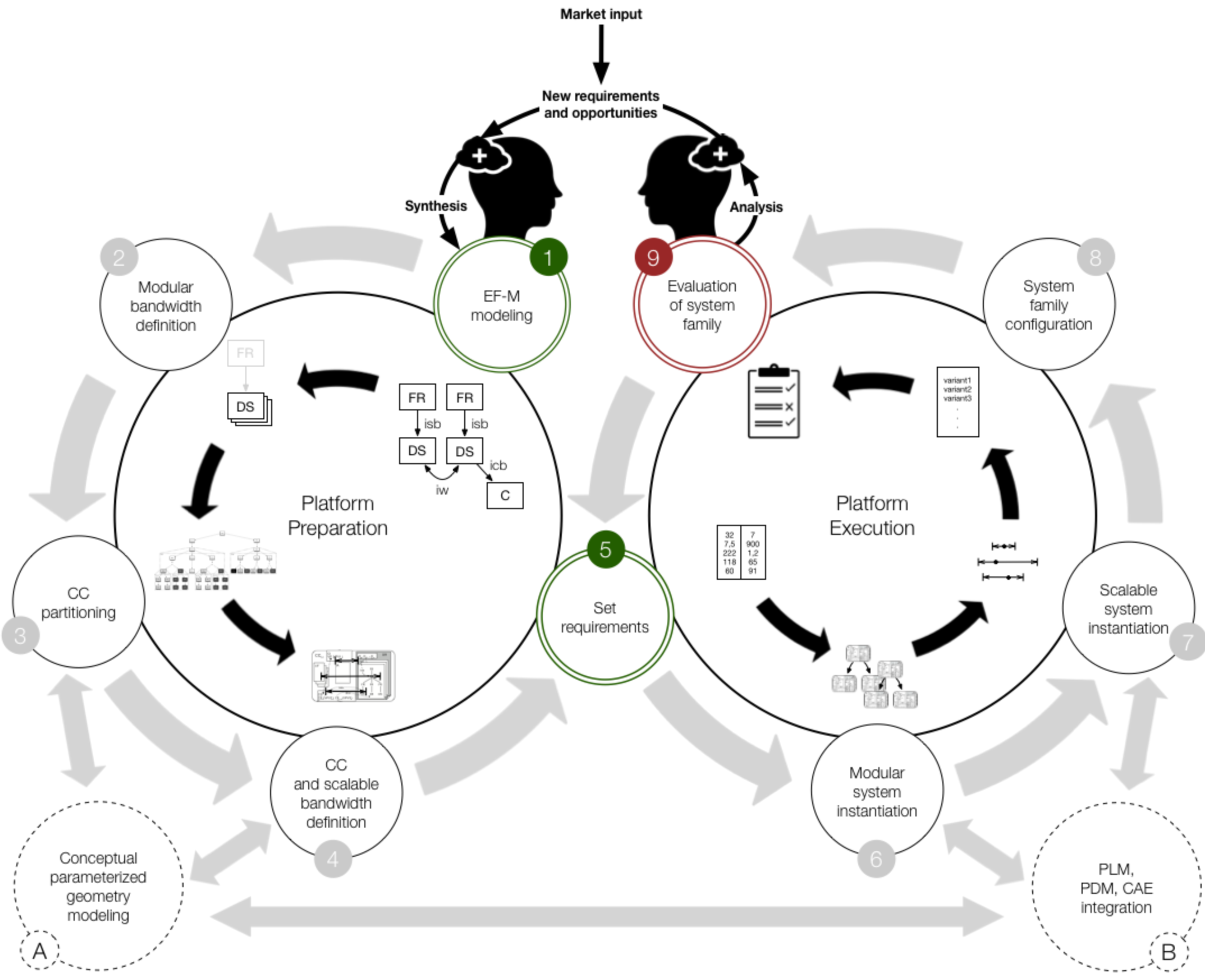


Figure 2.2 The platform preparation and execution processes (Johannesson, et al., 2016)

The *platform preparation* process begins with the creation of the EF-M tree. The desired top function is defined and further broken down into additional functions, each function paired with one single design solution. When the EF-M tree is completed, the system of the platform is described.

The second step is to set the modular bandwidth of the system, in order to retain the solution space within demanded limits. These demands are derived, for the platform, from e.g. the intended market. This step also includes expanding the EF-M tree to encompass alternative DSs for FRs and to map how the design solutions relate to each other, their interactions (Johannesson, et al., 2016).

The third step is to partition the system, in other words, identify clusters of FR - DR pairs that can be used in multiple sections of the EF-M tree (Johannesson, et al., 2016). These clusters are transformed into CC objects. Also, to be identified, are subsystems that are suitable for transformation, into CC objects, in order to simplify the complete system's structure.

The final step of the platform preparation process is to describe the CC objects internally and the external behaviour by setting up parameters (Johannesson, et al., 2016). In other words, specifying how the FRs, DSs and Cs shall act as well as their design rules - internally. Along with, how the CC objects shall behave in relation to each other, by describing the CI and CS for each object – externally. Also, the scalable bandwidth is further described.

The *platform execution* process begins with concretising the parameters, e.g. giving them values. The customer needs are also revised, which can result in new requirements that need to be met and adaption of constraints.

The next two steps are to instantiate the modular and the scalable system. Through the modular instantiation, the CC objects' relationships are activated along with the internal alternative DSs; this offers an architectural view of the platform. Meanwhile, the scalable instantiation provides alternative variants based on the collection of parameters (Johannesson, et al., 2016).

The fourth step configures the platform family, consisting of the variants, based on the parameter values using the two instantiated systems (Johannesson, et al., 2016). This step creates the unique variants that are to be evaluated in the final step.

2.4 Configure-to-order

When designing a product platform mass customisation and the level of personalisation are in focus. There are several approaches regarding how to handle the mass customisation, depending on the requested level of personalisation (Johannesson, et al., 2016). The approach treated in this thesis is configure-to-order, which has a high level of personalisation.

This is a customer needs oriented approach where specified variants and their configurations are in focus. In the development process both system architects and designers work together to analyse and translate the customer requirements into terms of parameter values (Johannesson, et al., 2016). This is reflected through all phases of the process; the conceptual, system and detailed development phases.

2.5 Contactors

A contactor is essentially a relay that is computerised in order to be able to automatically shut and break an electrical circuit (All about circuits, 2014). They are used in multiple applications today and vary in size.

The incoming current, i, defines the counterforce needed to be able to exceed the repelling force F_r between the contact plates, which is roughly described by the following equation:

$$F_r \approx i^2$$

The counterforce, in turn, defines the size of the electromagnet, since the electromagnet has to generate enough force to exceed the counterforce in order to close the circuit. A graph was provided by ABB, further describing the relationship between the current and the necessary forces, see Figure 2.3.

The forces are defined by the following equations, expressed in newton [N]:

$$F0 = 0.047 * i + 10 \qquad F2 = F1 + 0.531 * 10^{-3} * i^2$$
$$F1 = 1.1 * F0 \qquad F3 = F1 + 1.083 * 10^{-3} * i^2$$

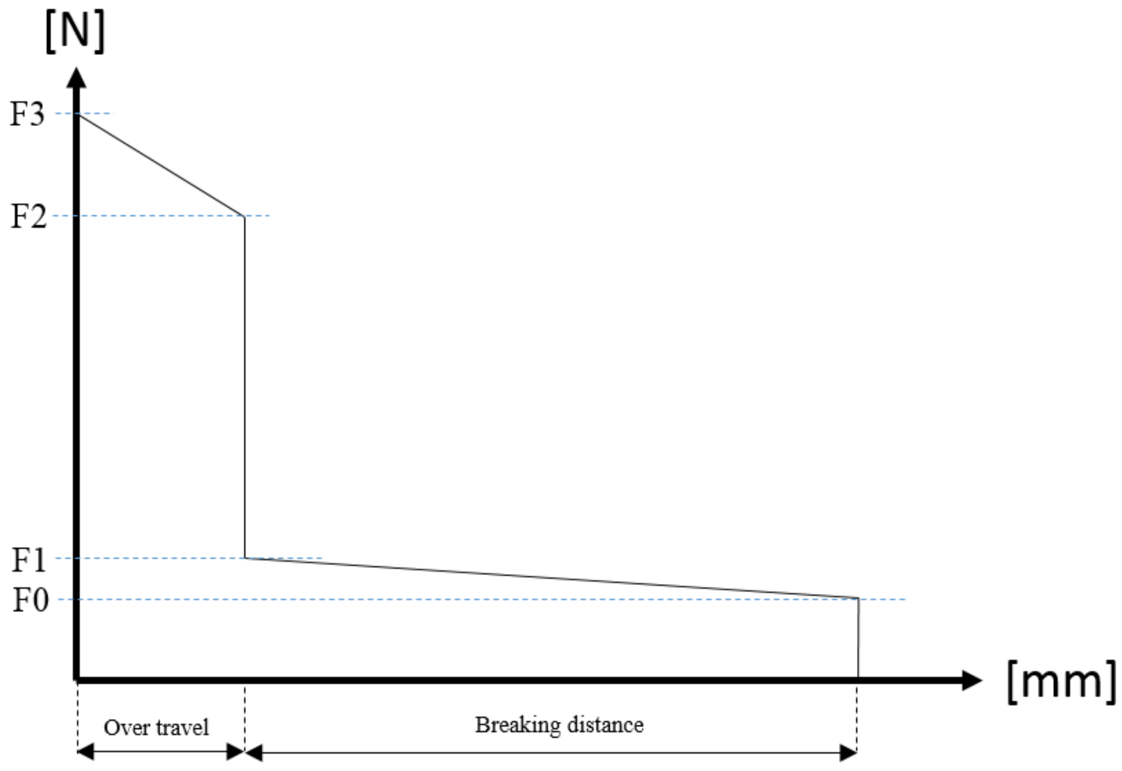


Figure 2.3 Counterforce curve

Also described in the figure is the relationship between the forces and the stroke, which can be split up into over travel and breaking distance. These two distances, expressed in millimetres [mm] are defined by the following equations:

$$Over\ travel = 0.0047 * i + 2$$

$$Breaking\ distance = 0.0159 * i + 2$$

Nevertheless, these two distances are also related to the contact springs and the return springs, further described in the following section. On behalf of the return spring it must have the length necessary to be able to achieve enough distance between the contact plates, securing that the circuit is broken, this distance is the breaking distance. For the contact springs the over travel defines the distance they need to have in order to generate the force necessary to be able to resist the repelling force.

The contactor in focus in this assignment is a 3-pole contactor that belongs to ABB's AF family; it has an electronically controlled coil with built-in surge suppression (ABB, 2014).

2.6 Analysis of Physical Parts of the Contactor

The contactor was disassembled in order to acquire a greater knowledge of how it works and how the parts behave in relation to each other. The internal parts were analysed and described on a system level. A system can include several physical parts.

2.6.1 Contactor System

The contactor system is the main system on the top level of the hierarchical structure of the complete contactor. The physical parts it consists of is described through the systems presented below. Three main functions for the system were identified as:

- **Control circuit,** how to shut and break the circuit as well as transmit current.
- **Operate contactor,** when to shut and break the circuit as well as transmit current.
- **Hold and position parts,** to ensure that the parts can be facilitated and that the contactor follows size standards.

2.6.2 Chassis System

The chassis system consists of two main parts made of plastic, defined as chassis contact and chassis operate in this thesis. The system's function is to hold and position the parts of the contactor. Chassis contact is the part that is assembled with the external application and where the contact plates, which are connected to the external circuit, are mounted. Chassis operate are mounted on top of the other part and it encloses the movable parts of the contactor. Thus, there are two identified functions:

- **Position and hold moving parts**
- **Position and hold parts connected to the main circuit**

2.6.3 Electromagnet System

The electromagnet system consists of four parts and has a number of interfaces to other systems. A moving iron armature which is assembled in the contact bridge, an iron core with a geometrical interface to one of the chassis parts, a bobbin made of a plastic that is used as a frame for the copper thread to be wrapped around in order to create a coil. The bobbin is assembled to the chassis and the ends of the copper thread are connected to the electronic controlling device.

This system provides the centre mechanism of the contactor, generate a movement and hold force that makes it possible to shut and keep the circuit closed as well as break the circuit. Thus, by energising the coil the core becomes magnetic and the moving armature is pulled towards the core. Consequently, to be able to achieve that, the identified main functions of the system is to:

- **Generate a magnetic field intensity (H-field)**
- **Generate a magnetic induction (B-field)**

2.6.4 Contact System

The contact system consists of twelve parts and has several interfaces to other systems. There are six outer copper contact plates to which the cables of the circuit are connected, one plate for each incoming cable and one for each outgoing cable, theses plates are assembled to the chassis parts. Between the outer contact plate pairs there are an inner copper contact plate that is assembled on an insertion, which is attached to the contact bridge.

This system's objective is to transmit the current from the incoming cables to the outgoing cables and work along with the other systems to enable the control of the circuit. In order for the system to achieve that two functions were identified:

- **Receive and Emit** the current of the circuit
- **Bridge** the current of the circuit from the contact plates connected to the incoming cables to the ones connected to the outgoing cables

2.6.5 Contact Bridge System

The contact bridge system is an integrated subsystem of the contactor system. There are several relationships with the surrounding parts it consists of that must be taken into account. The contact bridge system consists of three physical parts; a *lower part*, where the inner contact plates of the contact system is mounted as well as positioned and the electromagnet's core is placed; an *upper part*, where the electromagnet's moving anchor is placed; a *pin*, that fix the moving anchor.

The identified main purpose of the contact bridge system is to shut the circuit by transmitting the linear movement generated by the electromagnet to the contact system; this transmission is handled by the three physical parts earlier described and the internal relationships between the parts. For the system to be able to achieve that, two functions for the system was identified:

- **Receive** the movement from the electromagnet
- **Deliver** the movement to the contact system

2.6.6 Return Spring System

The purpose of the return spring system is to push the contact bridge upwards when the coil is inactive; this is executed by two return springs. That manoeuvre breaks the main circuit and the force that must be handled has it origin in the masses of the movable parts of the contactor. The main function of the return spring system was identified as:

- **Generate force** to enable movement

2.6.7 Contact Spring System

The purposes of the contact springs are to press the inner contact plates onto the outer contact plates when the coil is active. This in order to ensure that there is contact between the contact plates and thereby ensuring that the main circuit is shut. The force that must be handled is the repelling force that occurs between the contact plates. This contactor type has three connections, that each uses a contact spring, making it a total of three contact springs. The main function of the contact spring system was identified as:

- **Generate force** to enable pressure

3. Methodology

This chapter presents the procedures that have been used during the master thesis in order to acquire the result, the materials used to acquire the data and creating a PLM environment. The procedures are defined by the two phases, platform preparation and platform execution, and the performed steps are based on the problem definition

3.1 Research Methodology

The thesis has followed a research model derived from the synergies of combining theory and empiricism, the model is presented in Figure 3.1. The starting points were firstly defined, in this thesis the problem base was to design a platform family for ABB's contactors and the theory base was the concept of CCs as well as the platform preparation and platform execution phases. The work performed followed the two paths and was evaluated according to the model.

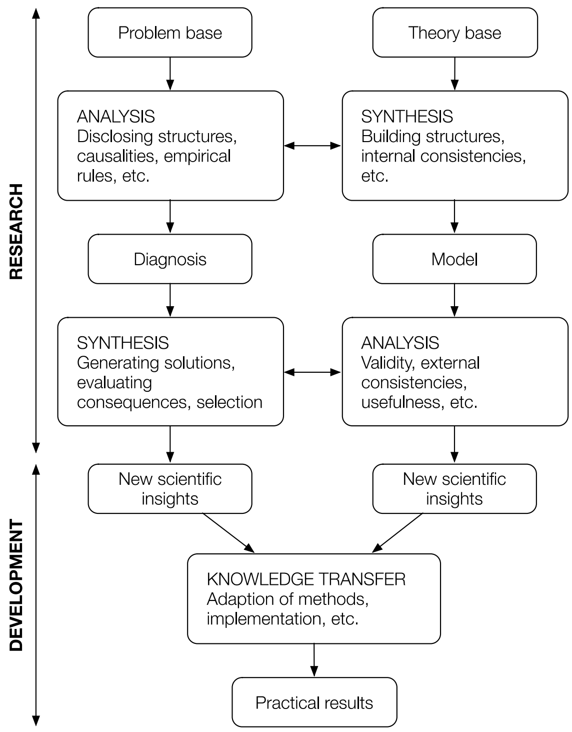


Figure 3.1 Synergy of combining empiricism and theory (Michaelis, 2013)

The analysis of the problem base consisted of studying the contactor in terms of functions and relations between functions; the syntheses with the theory base involved mapping the findings in an EF-M tree and begin to design a model aligned with the CC methodology.

During the second synthesis-analysis interaction, the theorised CC model was analysed with focus on findings of re-usable subsystems and translation of those into CC objects. The syntheses took its form as studying the outcome from the CC model and examine it against data from the actual contactor extracted from engineers at ABB.

The merger of the two paths resulted in a model that generates virtual contactors with data that correspond to real contactors. The full process is described in detail in section 3.3 and section 3.4.

3.2 IT Software Tools and Models

In order to complete this master thesis there were a number of IT tools used; the non-standard tools are presented with a brief description of what it is and what it has been used for. Examples of standard tools are Microsoft Office, Google Drive, etc.

3.2.1 Configurable Component Modeller

CCM is a software developed by COPE Sweden AB. The purpose of it is to enable product platform development based on the theory of configurable components. In this thesis it functions as the core in the PDM structure. The main user interface is seen in Figure 3.2.

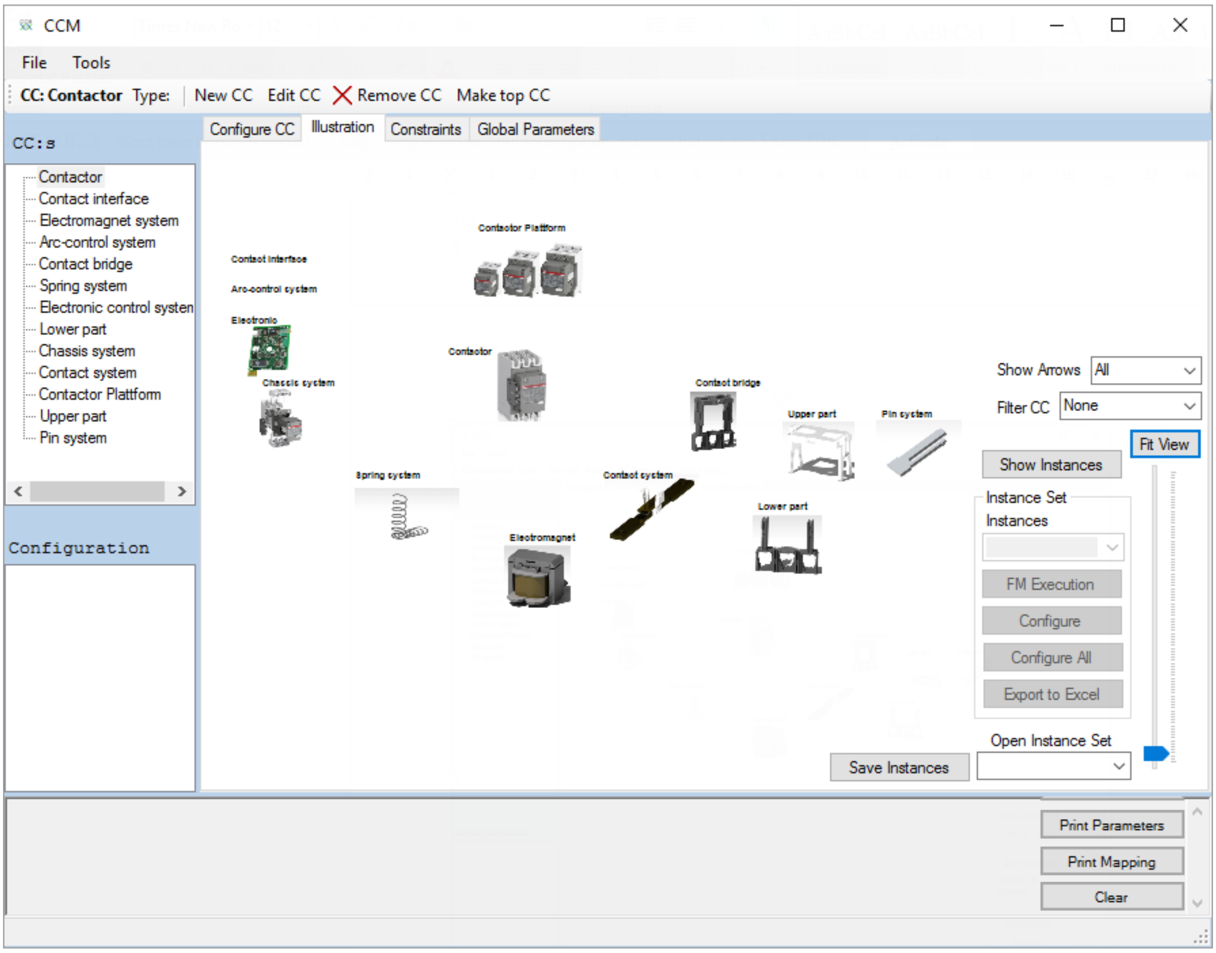


Figure 3.2 The CCM software

3.2.2 Ansys Maxwell

Maxwell is developed by Ansys, a major company providing a wide range of IT tools focusing on simulation in the field of engineering. This specific software focuses on simulation regarding electromagnetic fields, using the finite element method (Ansys Inc., 2016).

In this thesis Maxwell was used to perform simulations of the electromagnet and its static behaviour in pre-defined situations, this in order to acquire data for the dynamic simulations. The model was created in close collaboration with engineers from ABB.

3.2.3 Ansys Simplorer

Simplorer is also developed by Ansys. This software focuses on simulation with cross-domain systems (Ansys Inc., 2016), e.g. systems with both electrical and mechanical building blocks.

Simplorer was used, in this thesis, for performing dynamic simulations of the contactor system based on the electromagnet unit from Maxwell and the surrounding functions. The model was built in close collaboration with engineers from ABB.

3.2.4 Catia V5

Catia is developed by Dassault Systèmes, a large company focusing on virtual modelling IT tools and PLM IT tools. Catia is a CAD software where the user is able to create advanced 3D models (Dassault Systèmes, 2016).

In this thesis it was used to visualise the parts of the contactor. Its functionality with parameterisation and Excel reading capability was also used in order to enhance experience of the product design concepts of the contactor and communicate with CCM.

3.2.5 Visual Basic Script

VBScript is developed by Microsoft. It is a programming language that can be used for multiple environments and it uses the Windows Script standard to communicate with applications (Microsoft, 2016).

The written code was used as a key component in this thesis to achieve an automated generation process for contactor variants. The script handles the input data extracted from CCM to Simplorer, executes the simulation and exports the output for CCM to read. The full VBScript is provided in Appendix A – Visual basic script.

3.3 Platform Preparation

In order to acquire a platform where contactor variants could be generated, the current contactor and its functions were analysed. Then the result was translated according to the CC framework and a model describing the contactor was created in CCM. However, to acquire a reliable model the data and the relationships simulation models and constraints were obtained from ABB.

3.3.1 Analyses of the Physical Contactor and the Existing CCM Model

The first two steps that were taken were to perform analyses of the physical contactor and the existing CCM model acquired from COPE. This was crucial for the understanding of the product, the CCM software, and the relationship between them. The analysis of the physical contactor consisted of disassembling it while examining the parts and how they were connected to each other. Along with meetings and discussions with experts at ABB, this lead to an understanding of contactors, the purpose and the function of its subcomponents. That enabled a higher understanding of the existing CCM model which was to be analysed next.

The existing model consisted of eleven CC objects, seen in Figure 3.3. The statuses of the objects varied, a number of them were elaborated in greater extent but there were several

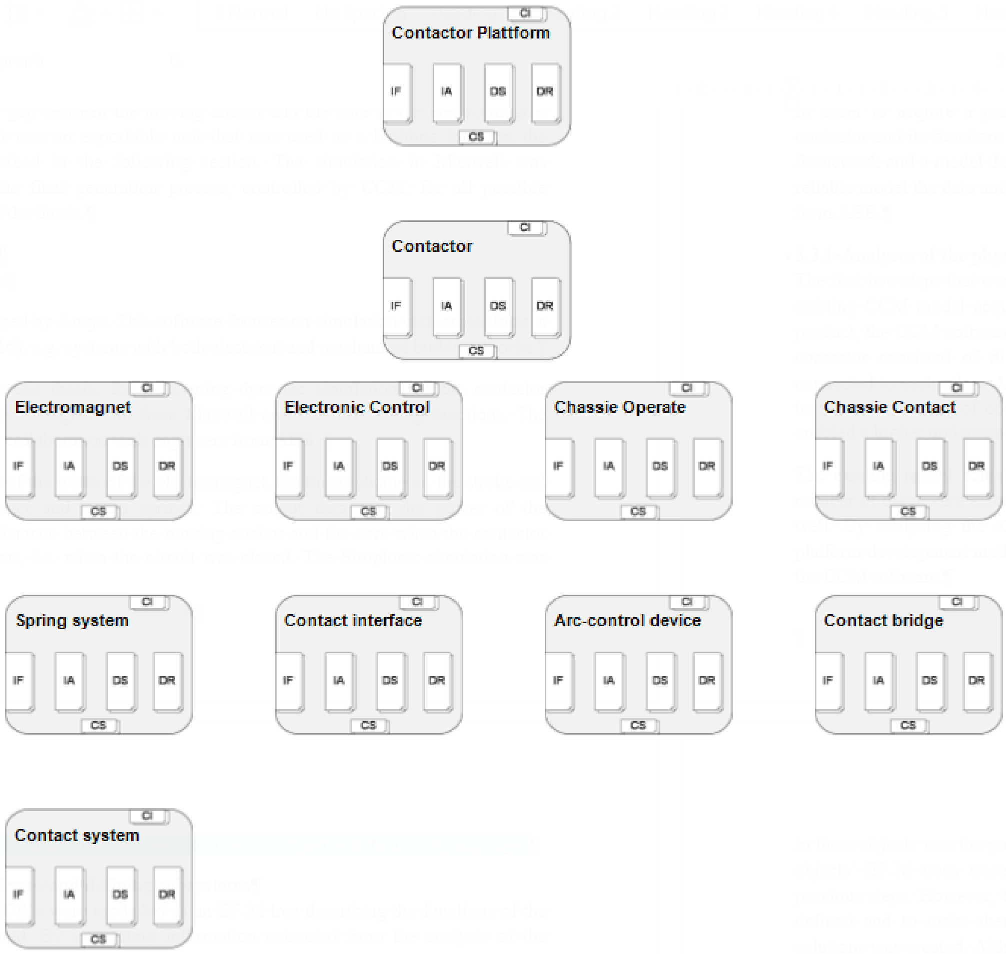


Figure 3.3 The existing CCM model

undefined objects as well. By analysing the elaborated objects, the understanding of the fundamentals of the platform development methodology was improved as well as the understanding of how to use the CCM software.

3.3.2 Creation of EF-M Tree and Defining Subsystems

The elaborated objects' DR's were modelled as an EF-M tree describing the functions of the contactor that was created. By integrating information extracted from the analysis of the physical contactor the limited existing EF-M tree could be extended to cover all functions of interest of the contactor. This step of the procedure was an iterative process, meaning that the EF-M tree was altered many times throughout the project. Hence, the logics must work all the way to the lower levels of the hierarchy and due to the continuous information obtainment new logics regarding functions were discovered continuously as well.

When the EF-M tree had been created the identification of suitable subsystems was performed, see example in Figure 3.4. These subsystems were strongly influenced by the physical parts of the contactor, but defined on a higher abstraction level.

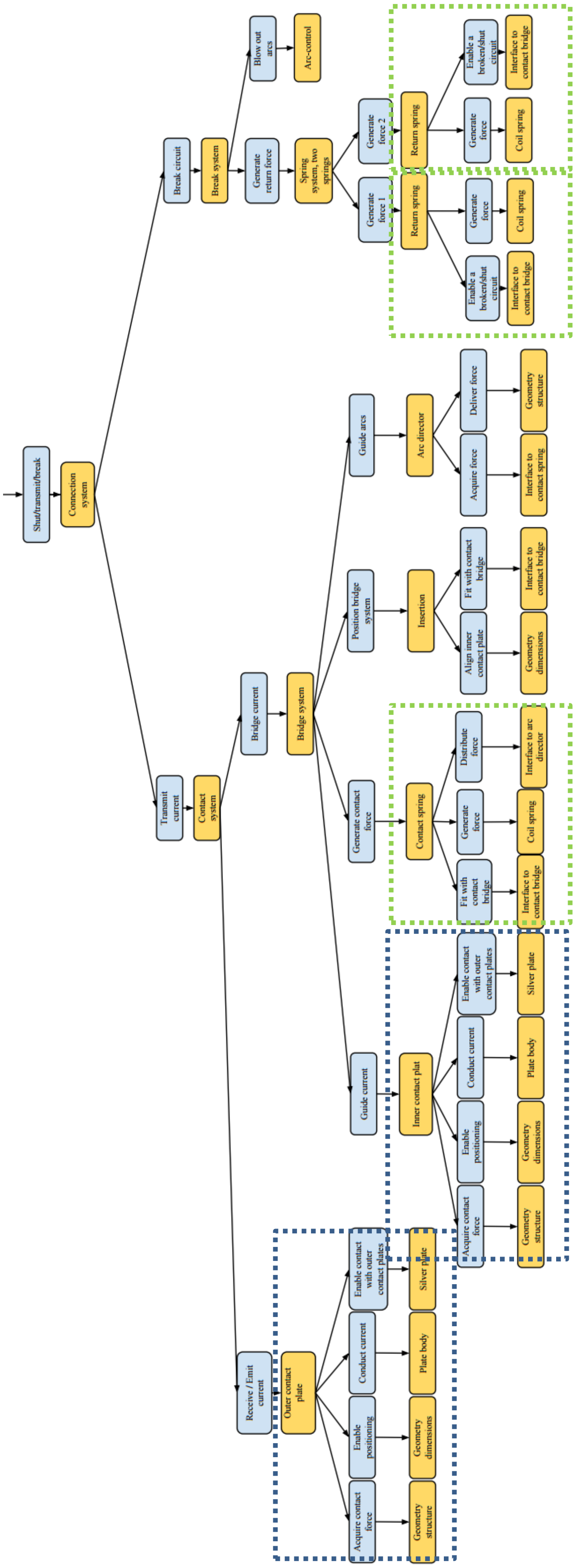


Figure 3.4 Partitioning of the EF-M tree

3.3.3 Defining CC Objects and Relationships in CCM

The subsystems defined in the previous step was translated into CC objects in CCM, included in these objects was the partial EF-M tree describing the subsystem. Most of the existing CC objects' EF-M trees were altered in order to better suit the logics generated during the previous steps. However, by creating the EF-M tree in the object, the design solutions were defined and to make them functional, parameters describing the attributes of the design solutions were created. Although, in a number of cases additional design solutions were added in order to acquire full functionality in the objects.

When the objects had been defined internally the next step was to set the communication between the objects. This was performed by firstly identifying what input parameters the object required to function, i.e. defining the control interface of each object. When those parameters had been defined they were examined regarding if they were to be set by a parenting object or manually by the user. In those cases, where a parenting object set the parameter, corresponding transfer parameters were created in the composition set of the parenting object. By now the CCM model could be instantiated, the instantiated model can be seen in Figure 3.5.

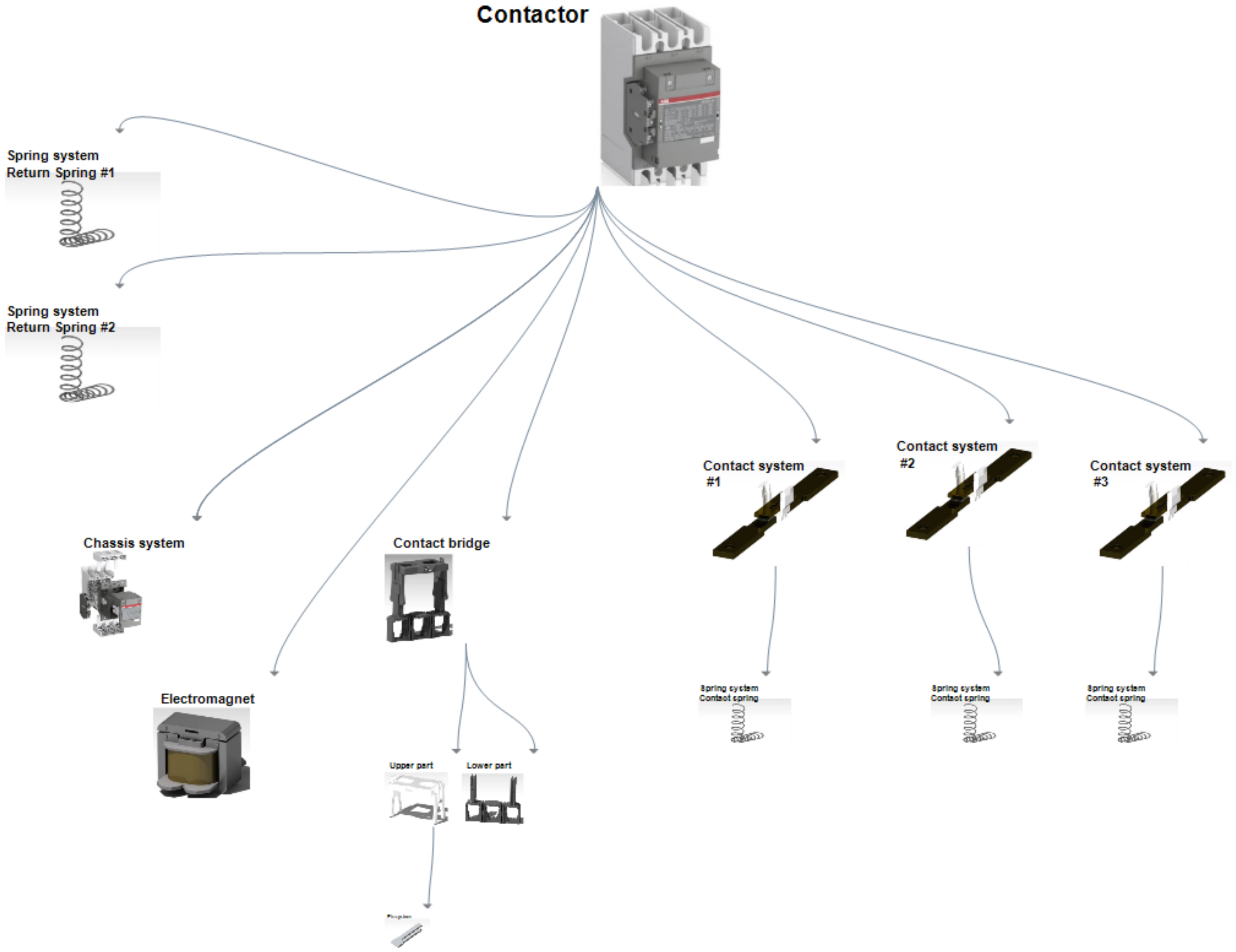


Figure 3.5 Instantiated CCM model

3.3.4 Obtainment of Models and Constraints from ABB

The final step of the platform preparation phase was to acquire models and data regarding the other software, i.e. Maxwell and Simplorer. ABB performed the simulation in Maxwell and exported the data in a file compatible with Simplorer. This file contained the static characteristics of the electromagnet for the sizes in focus in this project.

Along with that file, a Simplorer model describing the dynamic system was obtained. That model was completed by importing the file containing the electromagnet’s characteristics. By manually setting the contactor’s magnet’s width, the thread diameter, and the stroke, along with the spring characteristics, the fundamental electro mechanics of the contactor could be simulated. These models were later on recreated to fully understand the models.

In parallel with the steps above, parameterised CAD models, describing the systems modelled in CCM, were created in order to acquire an adaptable visual presentation of the contactor. The final assembly is seen in Figure 3.6.

Hence, at this point all the building blocks necessary to generate a contactor variant had been obtained or created. Therefore, the project could move into the next phase.

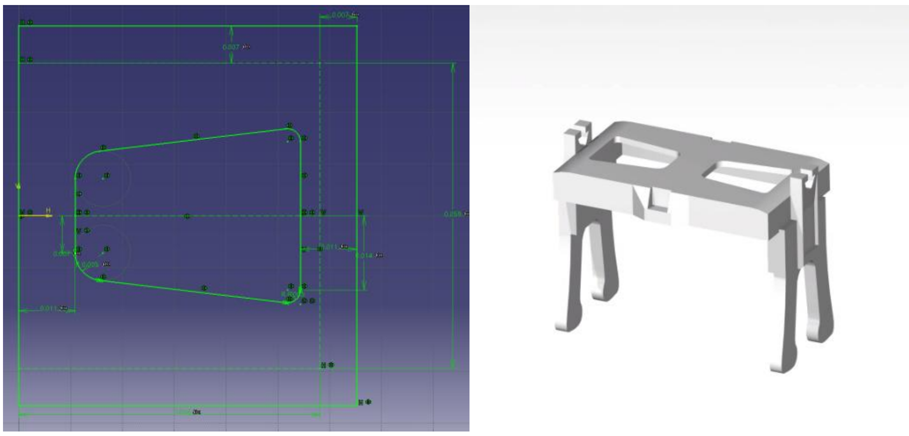


Figure 3.6 Parameterised CAD model of upper part

3.4 Platform Execution

With the purpose of generating contactor variants with attributes that correspond to the actual ones developed by ABB, the process of how to acquire the data generated by the simulations from the external software was required to be identified. In order for this to be possible it was required to set up a PDM structure based around CCM, which can be seen in Figure 3.7.

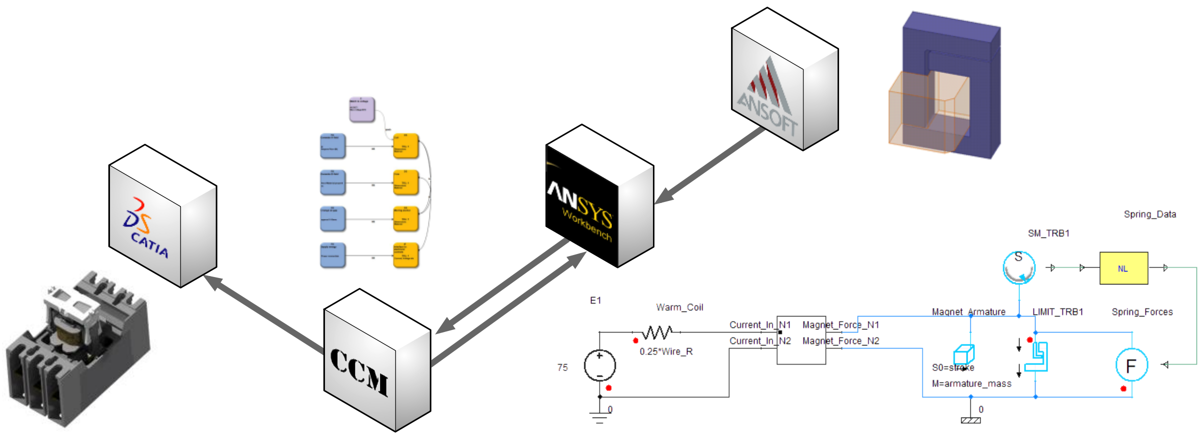


Figure 3.7 The PDM architecture (Edholm, et al., 2016)

3.4.1 Identification of the Generation Process

The initialising step was to define the process in which a single contactor variant could be generated. This was achieved through analysing the required inputs to Simplorer and the possible outputs from Simplorer. The analysis resulted in a process consisting of four stages: the first stage was to generate a variant in CCM and printing the parameters of interest to an Excel sheet. The second stage was to manually set the input parameters in Simplorer and execute a simulation, the third stage was to manually read the resulting parameters and set the corresponding parameters in the CCM model. The final stage was to generate a new variant in CCM with the same basic input parameters along with the ones with values from Simplorer. This provided, after some alternations in the different models, a contactor variant with the correct attributes.

3.4.2 Automation of Generation Process

When the generation process had been evaluated and confirmed to be functioning with a satisfying result the next step was to achieve an automated process. This meant by only setting the current, the electromagnets width, the coil’s thread diameter, and then press configure in CCM, a complete variant should be generated. In order to be able to achieve that, certain keys were identified. These keys were: An Excel file that converted the spring characteristics into a format that Simplorer could handle, a Visual Basic Script that imported parameters to Simplorer, ran the simulation and exported the data, and adaptation of CCM enabling it to read from files[2].

3.4.3 Generation of Multiple Contactor Variants

The final step was to once again analyse the generation process and identify how CCM's communication flow should be altered in order to handle generations of multiple variants. This was important in order to be able to generate the full product family of contactors. This analysis was an iterative process, accompanied by the one of the developers of CCM. The process consisted of the following steps: generate one variant, examine the generation flow, alter CCM, generate a new variant and study the consequence of the alternation. After a number of alternations CCM was automatically able to generate multiple variants, based on the input parameters, with attributes consistent with ABB's contactors.

3.5 PDM Architecture

The PDM structure, previously mentioned, was identified by analysing the input, result, and purpose of each software individually. Thus, this provided feedback necessary for understanding how the implementation and connection with CCM and the CCM model must be formed.

3.5.1 Maxwell

The input data for the simulations were the width of the electromagnet, the currents included in the scope and the air gap between the moving anchor and the core of the electromagnet (the stroke). The output was an exportable unit that was used as a building block in the Simplorer model, described in the following section.

The simulation in Maxwell was performed outside of the final generation process, controlled by CCM, for all possible scenarios in the scope of the thesis. The model used for the Maxwell simulation is seen in Figure 3.8. As can be seen in the figure, one fourth of the electromagnet system is used in the simulations, this was to decrease the duration of the simulations.

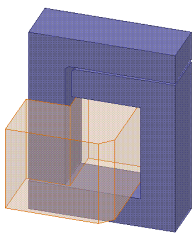


Figure 3.8 The Maxwell model

[2] This was performed by the programmer of the CCM software, Magnus Andersson

3.5.2 Simplorer

The input data consisted of the width of the electromagnet, the thread diameter, the stroke and the forces for the contact and return springs. The output data was the power of the electromagnet and the distance between the moving anchor and the core when the contactor was in its activated status, i.e. when the circuit was closed.

The Simplorer simulation was included in the final generation process; as described in section 3.4. Unlike the Maxwell simulation the simulations performed in Simplorer were performed for each unique contactor variant every generation loop. The Simplorer model is presented in Figure 3.9

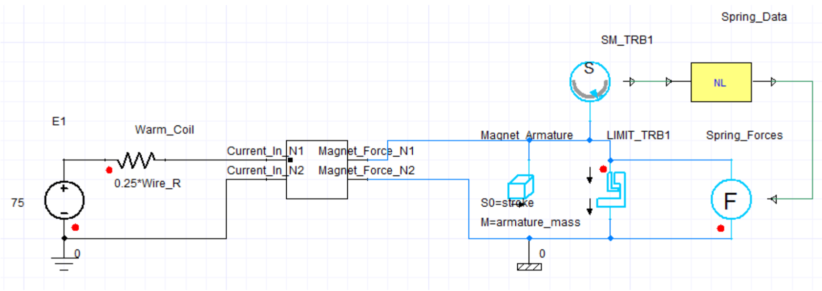


Figure 3.9 The Simplorer model

3.5.3 CAD Model

The input data was the width of the electromagnet and width of the chassis; the output was the visualisation of the generated contactor variant.

The CAD model that is used in this CCM project consist of 39 parts that are assembled together in order to create a model of a contactor, presented in Figure 3.10. These parts are scalable and individually parameterised, which enables them to be controlled by different excel documents when it is enabled in the CCM software. However, at this moment the model is controlled by a single parameter. The assembly acquires that parameter from an excel document generated by the contact bridge object of the CCM model. The purpose of the model is to be able to visualise the concepts generated by the CCM software.

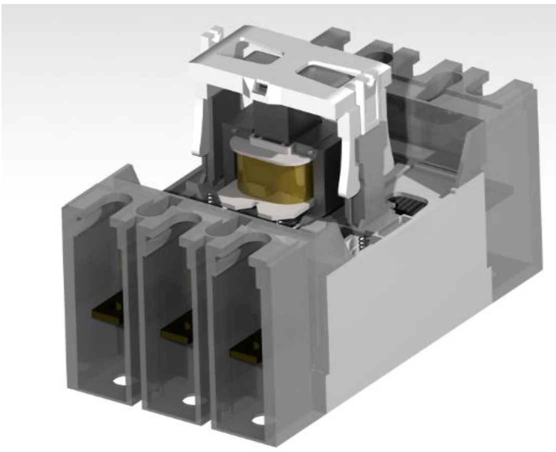


Figure 3.10 The contactor assembly CAD model

4. Results

This section presents the results that were acquired during the execution of the procedures described in previous sections. Firstly, the CCM model is described, along with how the EF-M tree was integrated in the models DR. Then the results from the Simplorer, Maxwell and Catia models are presented. Finally, the resulting generation process is described, accompanied by the result from the platform execution phase.

4.1 CCM Model

Throughout the creation of the EF-M tree the contactor was studied as a complete product and from that it was derived what functions it needed. The information extracted from that study was then combined with the analysis performed in section 2.6. The result from that is presented in this section.

By combining the results presented in the two previous sections the modelling in CCM was possible.

The main systems that were identified during the system breakdown as suitable for being facilitated as CC objects were the following systems:

- Contactor system
- Contact system
- Electromagnet system
- Contact bridge system
- Upper part system
- Pin system
- Lower part system
- Return spring system
- Contact spring system
- Chassis system

The two spring systems were realised by one CC object. This resulted in a model consisting of nine CC objects that when instantiated defines the limited version of a contactor, seen in Figure 4.1.

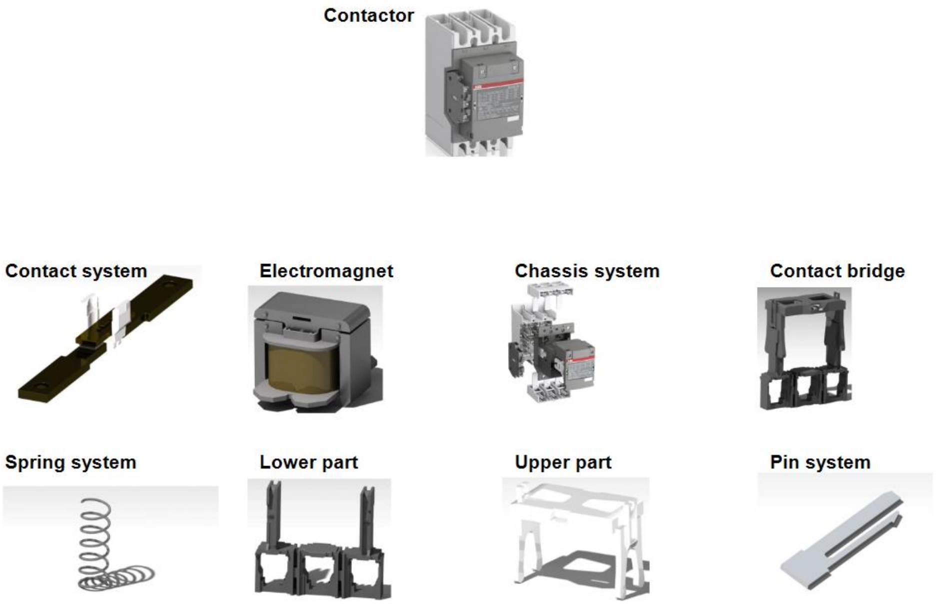


Figure 4.1 The nine objects of the CCM model

4.1.1 Contactor System Object

The contactor system object is the master object of the model and also, to some extent, the user interface. It is this object to which the user set the parameters describing the contactor variant or the variant range of interest and it is this object which presents the result of the generated variant(s).

Control Interface

The input parameters to this object are the current of the circuit, the number of connections, the number of return springs, the electromagnet's width, the coil's thread diameter, and the maximum current of the intended application. Based on the inputs, the force curve equations are calculated, derived from the circuits current. The external output of this object was presented in an Excel sheet, the chosen data were the total cost of the contactor, chassis size, current of the circuit, both size and position status of the electromagnet, power, and thread diameter.

Design Rationale

The design rational, which can be seen in Figure 4.2 The contactor system's design rationale, fulfils the requirement ‘control circuit. The DR is derived from the following functions:

(a) Control Circuit

In order to control the circuit, the DS for this function was a connection system that enables the control. The requirements of that system were that it had to be able to shut the circuit, break the circuit, and transmit the current. A suitable solution for shutting the circuit was identified as the *contact bridge system* and to the requirement transmit the current the *contact* *system* was chosen. The last requirement, break the circuit, was solved by a breaking system. This system was further defined by that it must blow out the arcs that occurs when the circuit is broken and that there is a need of a return movement due to that the inactive state of the contactor is with a broken circuit. The first requirement was solved by an arc control system and the latter one was solved by a *return spring system*.

(b) Operate the Contactor

In order to operate the contactor there must be an operating system, which was chosen as the DS for this requirement. This operating system must generate a movement and be able to keep it static when in position as well as achieve a certain force, which was solved by an *electromagnet system* solution. It must also handle that movement and force, the solution to that requirement was an electronic control system.

(c) Hold and Position Parts

For the contactor to be able to hold and position parts, in order to ensure that the parts can be facilitated and that the contactor follows size standards the *chassis system* was identified as the solution.

Composite Set

Because of its position as a core object this CC communicates with a number of other objects through the CS interface. The relationships to the child objects are controlled by composite rules and parameter sets, parameters are both transferred and acquired. The objects used are the contact bridge system, springs system, electromagnet system, contact system, chassis system and arc-control system. The last object has not been derived further due to exclusion in the scope.

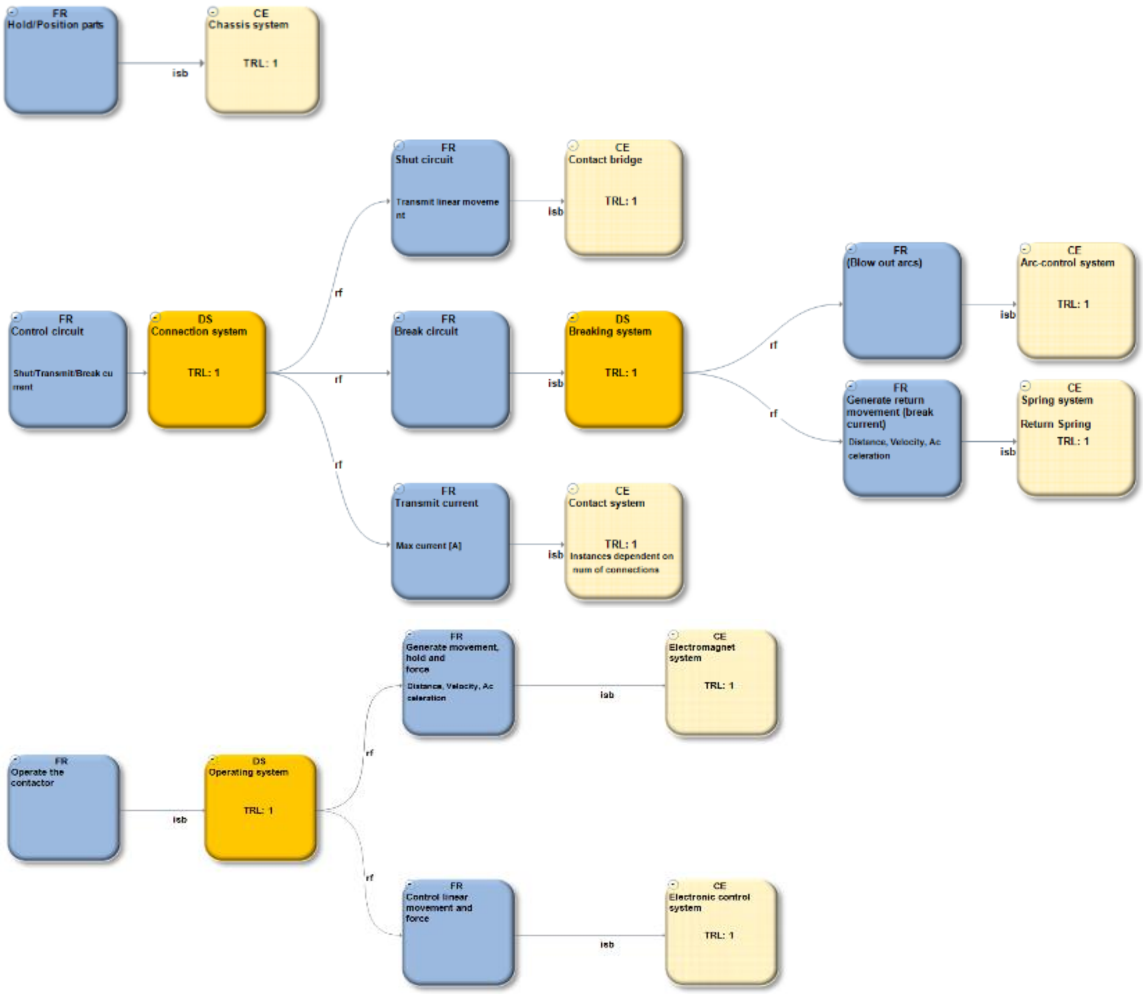


Figure 4.2 The contactor system's design rationale

4.1.2 Chassis System Object

The chassis system object is a child to the contactor system object. This object enables for the software to determine which chassis size that is suitable for the set up.

Control Interface

The input parameters acquired through the control interface is the maximum current and the width of the electromagnet. The output of this object is the size of the chassis and it is transferred back to the contactor object.

Design Rationale

The design rationale for the object describes how it fulfils the FR 'hold and positon parts', it is presented in Figure 4.3. As can be seen in the figure, there is an interaction between the two chassis parts and both parts are constrained by a standard dimensioning. The constraint checks the value of the maximum current parameter and sets the parameter defining the dimension of the chassis. It alters between three values, which in the end decides the series the contactor will belong to. The DR is derived from the following functions:

(a) Hold and Position Moving Parts

In order to fulfil this requirement, the defined part chassis operate was transformed into a DS. This DS must position the main parts of the contact bridge system as well as position the core and bobbin of the electromagnet system. That was solved by several interfaces matching each FR. It also led to the requirement that it must fit all necessary parts which was solved by that the chassis operate must have certain geometry dimensions.

(b) Hold and Position Main Circuit Parts

The solution to this requirement was the transformed part chassis contact, similar to chassis operate it has several interfaces to other systems. Hence, it must position the return springs, position outer contact plates, and guide the lower part of the contact bridge system. That is solved by the interfaces between the DS and the other systems. Furthermore, it must fit the parts, which is solved by geometry dimensions.

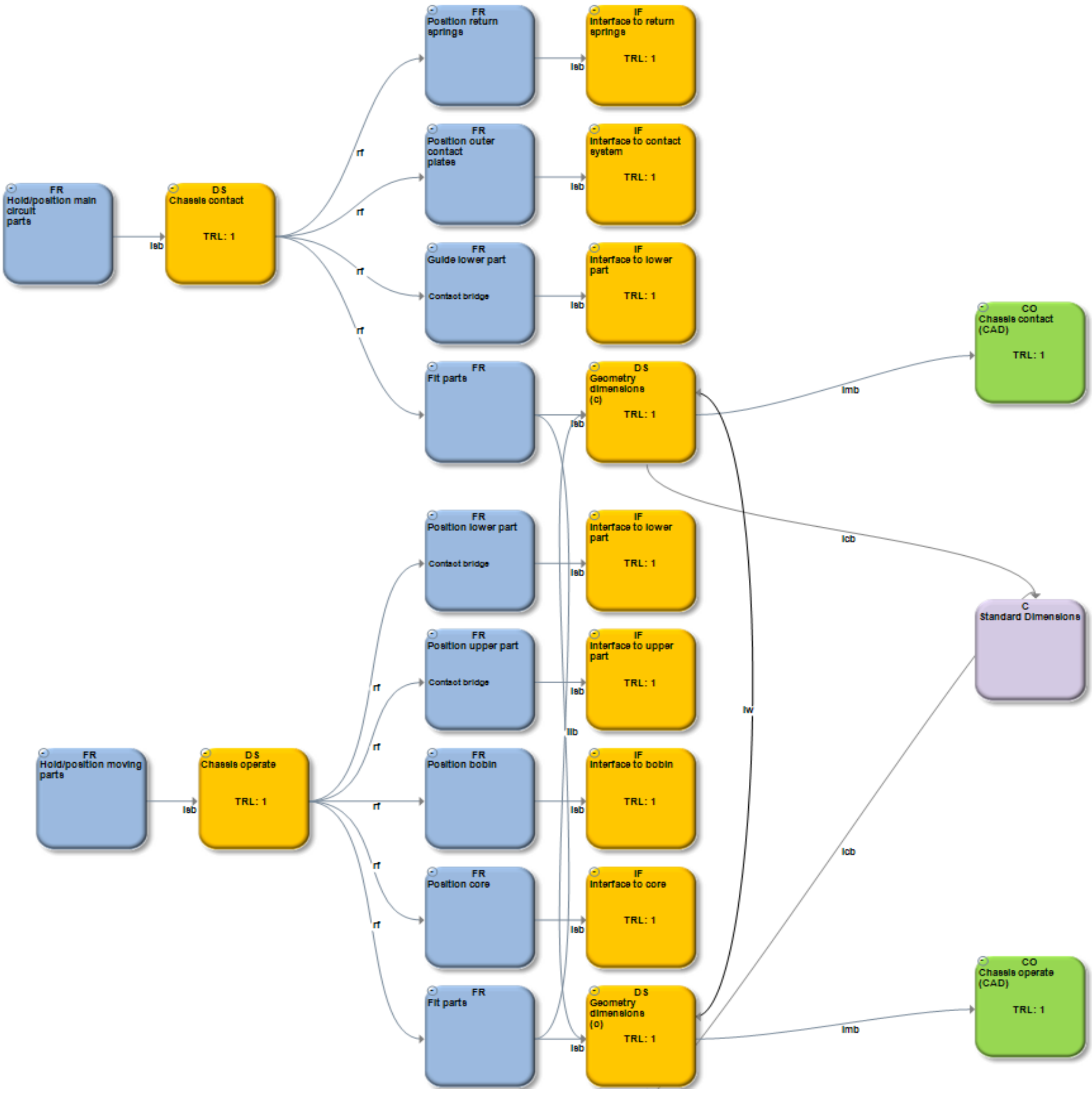


Figure 4.3 The chassis system's design rationale

4.1.3 Electromagnet System Object

The electromagnet system object is a child object to the contactor system object. This object handles the connection to the external simulation performed in Simplorer.

Control Interface

The inputs to this object are the force curve equations (F1, F3, over travel, breaking distance), current, electromagnet width, thread diameter, and maximum current. There are also a number of parameters that needs to be set manually regarding performance, dimensions and material properties. The performance related parameters are: the airs permeability, fill factor, activation current, resistivity, the magnet cores permeability, and thread isolation. The dimension related parameters are the play and bobin distances. The material properties that needs to be set are the material, its density and its price. The outputs from this object are the cost and power of the electromagnet along with the status reports regarding the position and size of the electromagnet. The outputs are transferred to the contactor system object.

Design Rationale

The design rational, which can be seen in Figure 4.4 The Electromagnet system's design rationale, fulfils the requirement 'generate movement and hold force'. The DR is derived from the functions identified in the analysis of the contactor.

(a) Generate a Magnetic Field Intensity (H-field)

In order to acquire the necessary H-field the electromagnet system must have a coil system. For that coil system to function as wanted there must be a winding, it must acquire energy and it must be positioned correctly. Consequently, to acquire a winding there must be a geometry structure and to position the coil there must be geometry dimensions. The solution to the requirement of acquiring energy was an interface to electronic control (not treated in this thesis)

(b) Generate a Magnetic Induction (B-field)

For the electromagnet system to generate a B-field the chosen design solution is a magnet system. That second system acquired the requirements change air gap and dimensions the magnet. The first requirement acquired a moving system, solved by system with a moving armature. The second requirement is solved by defining the geometry dimensions of the magnet system consisting of the core and the armature. Consequently, these must be positioned. That is solved by an interface to the contact bridge for the armature and to the chassis operates for the core.

Simplorer Connection

The connection to Simplorer can be seen as three steps, briefly mentioned in section 3.2.5. The first step is to export the necessary parameters (electromagnet width, thread diameter and stroke) to excel. The second step is to check that the parameters have values, this is performed by an AND-operation in CCM and it returns a Boolean (true or false). If the Boolean is true CCM activates the script. The script imports the parameters to Simplorer, runs the simulation and exports the result to a CSV-file that is created for each simulation loop. The third and final step is for CCM to read and delete the CSV-file, the file must be deleted each generation loop because otherwise CCM will acquire the same values the next generation loop. Now the simulated data can be used in the CCM-model.

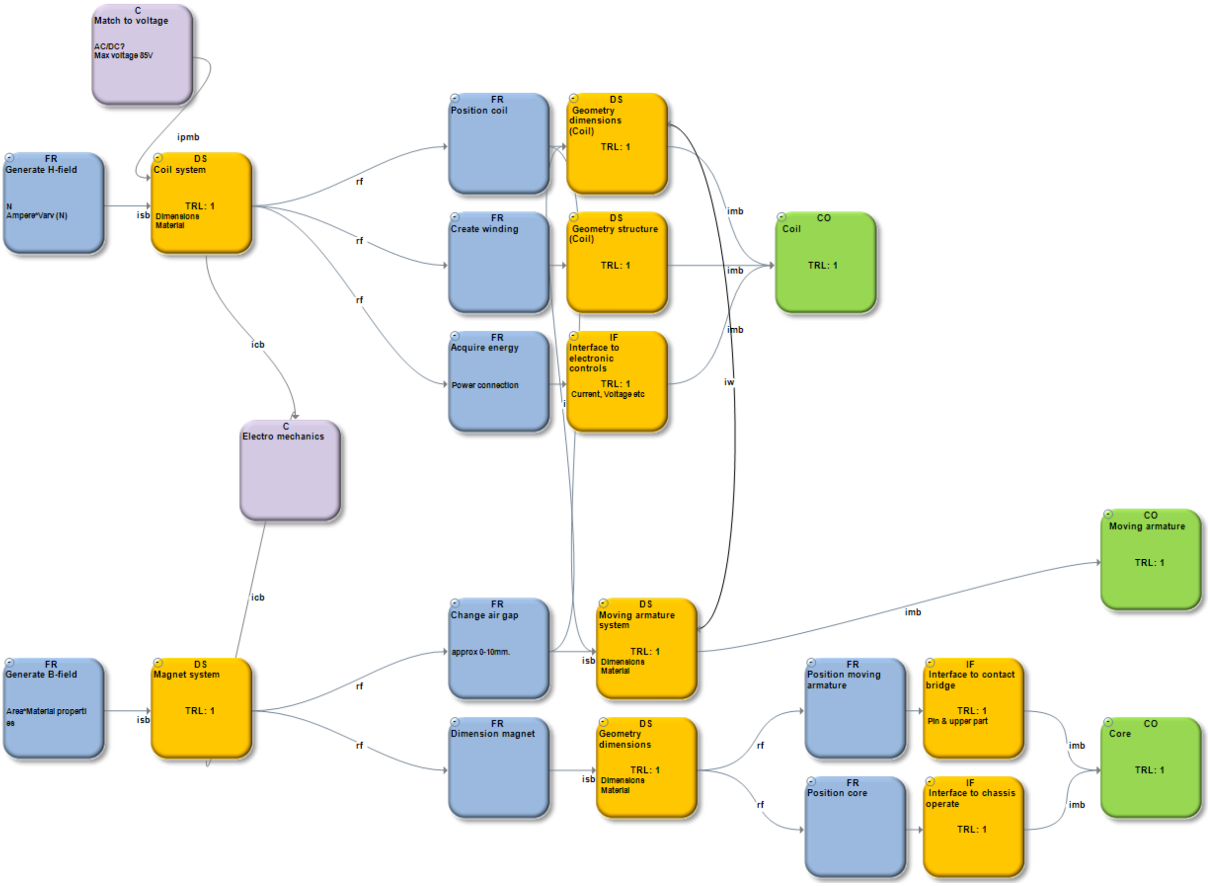


Figure 4.4 The Electromagnet system's design rationale

4.1.4 Contact System Object

The contact system object is a child object to the contactor system object, in this model three objects are called upon when it is instantiated. Hence, the number of contact systems are dependent on the number of connections for the circuit and in this thesis the contactor is defined for three phase applications.

Control Interface

The input parameters from the parent object are the current, force curve equations (F2, F3 and over travel), electromagnet width, and maximum current. There are also a number of parameters that must be set manually by the user in this object, such as: density, resistivity and price of materials (copper and silver), as well as the maximum accepted deflection for the contact plates. The output of this object is the material cost and it is transferred back to the contactor system object.

Design Rationale

The design rational, which can be seen in Figure 4.5 The contact system's design rationale, fulfils the requirement ‘control circuit’. The DR is derived from the functions identified in the analysis of the contactor.

(a) Receive and Emit Current

For the contact system to fulfil this requirement it must achieve a connection to the external circuit, this was solved by an outer contact plate system. The plates must enable a connection to the circuit, enable contact with the inner contact plates and be able to conduct the current. That is acquired by describing the geometry dimensions for the first requirement, for the second requirement, applying a silver plate, and for the latter requirement by defining a plate body. The plates must be positioned as well; this is solved by an interface to chassis contact.

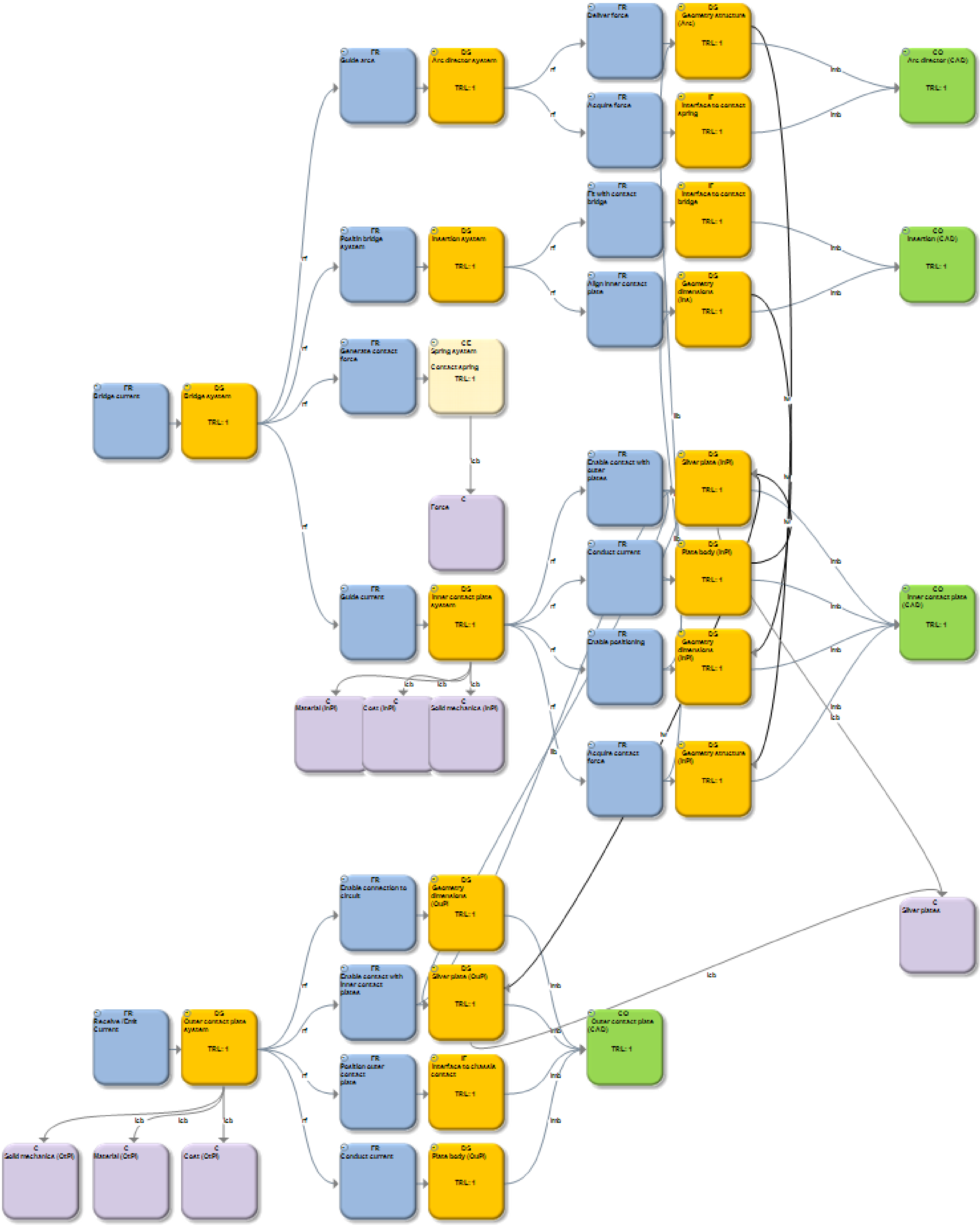


Figure 4.5 The contact system's design rationale

(b) Bridge Current

In order to be able to bridge the current a bridge system was used as a solution. Additionally, the bridge system is required to guide the current, generate a contact force, acquire a position, and guide the arcs that appear when the circuit is broken.

To guide the current an inner contact plate system is used as the solution. That added the requirements: acquire contact force which was solved by a geometry structure, enable positioning which was solved by geometry dimensions, conduct current which was solved by a plate body, and enable contact with outer plates which was solved by silver plates.

To generate the contact force a *contact spring system* was used as a solution. In order to position the bridge system an insertion system was found as the solution, with the additional requirements fit with contact bridge system and align inner contact plate. These two requirements were solved by an interface to contact bridge and geometry dimensions for the latter. The requirement to guide arcs is solved by an arc director system, adding the requirements deliver force which is solved by a geometry structure and acquire force which is solved by an interface to the contact spring system.

4.1.5 Contact Bridge System Object

The contact bridge system object is a child object to the contactor system object and has three lower level objects.

Control Interface

The input to this object is the current of the contactor, number of connections, and force curve equation F3. This makes it possible to create a number of different sizes of the contact bridge. There are outputs transferred to the two child objects, sizes of the electromagnet and the force generated by it. There is also an output that are transferred to the contactor system object, the material cost.

Design Rationale

The object’s design rationale (DR) is describing the solution to the FR ‘Shut circuit’ which is what the CC object is the solution to, the DR is seen in Figure 4.6 The contact bridge system's design rationale. The constraints existing in this object treats cost, dimension of the parts and how it can be manufactured, as can be seen in the figure.

(a) Receive Movement

For the contact bridge system to receive movement from the moving anchor an *upper part system* was specified as the solution. The following requirements were that the system should control the upper movement and hold the moving anchor meanwhile being constrained by cost and manufacturing.

To enable the control of the upper movement a geometry structure was the solution, which in turn has to be able to guide the movement which was solved by an interface to the chassis and constrained by friction, connect with the lower part which was solved by an interface to the lower part and constrained by solid mechanics, being stabilised which was solved by geometrical dimensions and fit a pin system which was solved by a holding system geometry and also constrained by solid mechanics.

To be able to hold the moving anchor and facilitate the generated movement a holding system was the chosen solution. The requirements of that system were to fit the moving anchor which was solved by geometry dimensions and fix the moving anchor which was solved by a *pin system* constrained by material and cost.

The pin system in turn had to be compatible with the moving anchor which was solved by an interface to the electromagnet and be able to handle the force which was solved by a geometrical structure constrained by solid mechanicals.

(b) Deliver Movement

In order for the contact bridge to deliver the movement the solution was identified as a *lower part system,* this system was constrained by cost and manufacturing. The requirements on that system were identified as fitting the core of the electromagnet which was solved by a geometry width, position the contact system which was solved by an interface to the contact system, position return springs which was solved by an interface to return springs and to control the movement which was solved by a geometry structure.

The geometry structure needed to connect with the upper part which was solved by an interface to the upper part, stabilise for a load which was solved by geometry dimensions that was constrained by solid mechanics, and to guide the movement which was solved by guiding tracks.

The guiding tracks added the requirements guide upper movement which was solved by an interface to the operational part of the chassis and guide lower movement which was solved by an interface to the contact part of the chassis, constrained by friction.

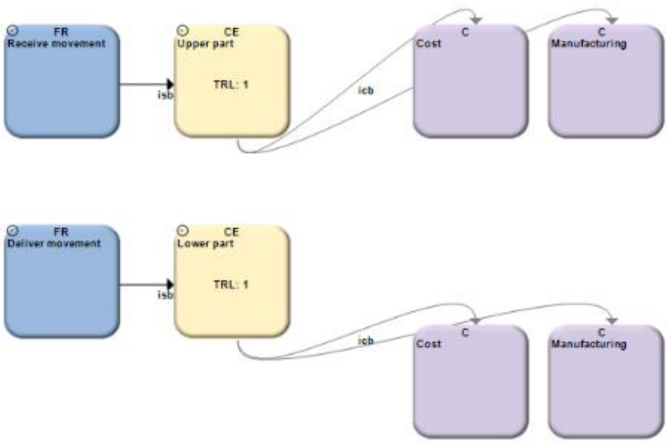


Figure 4.6 The contact bridge system's design rationale

4.1.6 Upper Part System Object

The upper part system object acquires inputs from its parent object and from the user.

Control Interface

The upper part system object acquires input from the contact bridge system object, the pin system and the user. The user must set the elastic moduli for the material of choice which enables the possibility to examine how different materials affect the structure.

Design rationale

The object's DR describes the solution to the FR receive movement and it is presented Figure 4.7. This DR has several more FR's and DS's than the one for the contact bridge system and as can be seen in the figure a number of them interact with others. The interaction between the 'Interface to lower part' and the 'Geometry dimensions' is a one-way interaction where the connection between the upper and lower part affects how the geometry dimensions of the pillars will be set. The interaction between the 'Pin system' and the 'Holding system geometry' is a one-way interaction as well, the width of the pin is apprehended from the pin object and then it defines the dimensions of gap where the pin is located in the upper part.

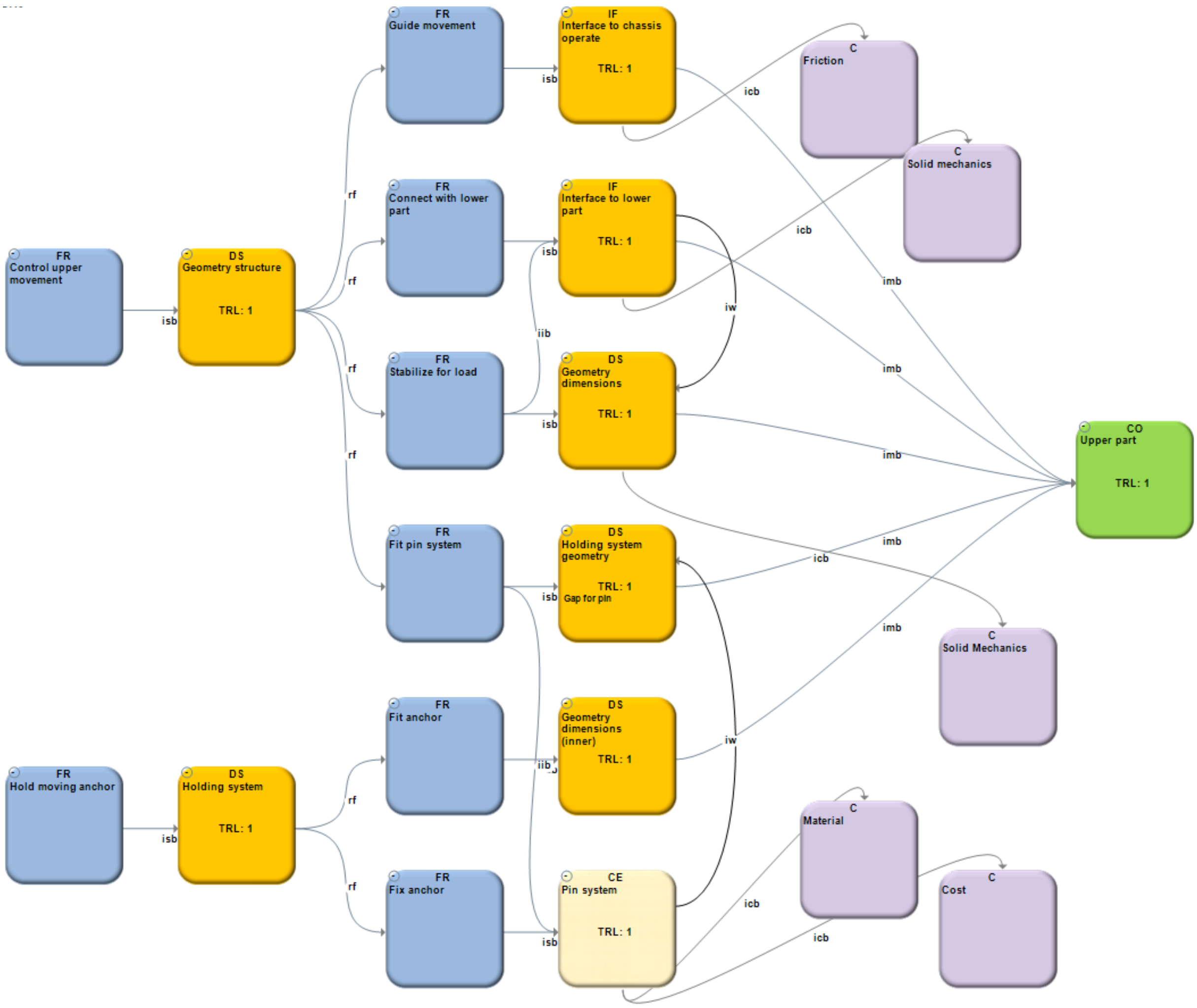


Figure 4.7 The upper part system's design rationale

The intention was that the object should be visualised in Catia by an interaction between parameters in the component 'Upper part'; therefore, all the DS's are connected to the component. However, due to that it was not possible as earlier mentioned it has no function in this model. The constraints in this model are mainly based on mechanical functions such as the deflection for the top of the part and the maximum breaking force the pillars can handle, all derived from the force of the electromagnet. Another constraint is the friction between the chassis and the upper part is low enough.

4.1.7 Lower part System Object

The lower part system object acquires inputs from its parent object and from the user.

Control Interface

The inputs are the width, height, core height, and maximum force of the electromagnet as well as the elastic moduli from the user. Since the object does not have any children there are no outputs.

Design Rationale

The object's DR describes the solution to the FR 'Deliver movement' and it is presented in Figure 4.8. The DR is similar to the one for the upper part object with the exception of more

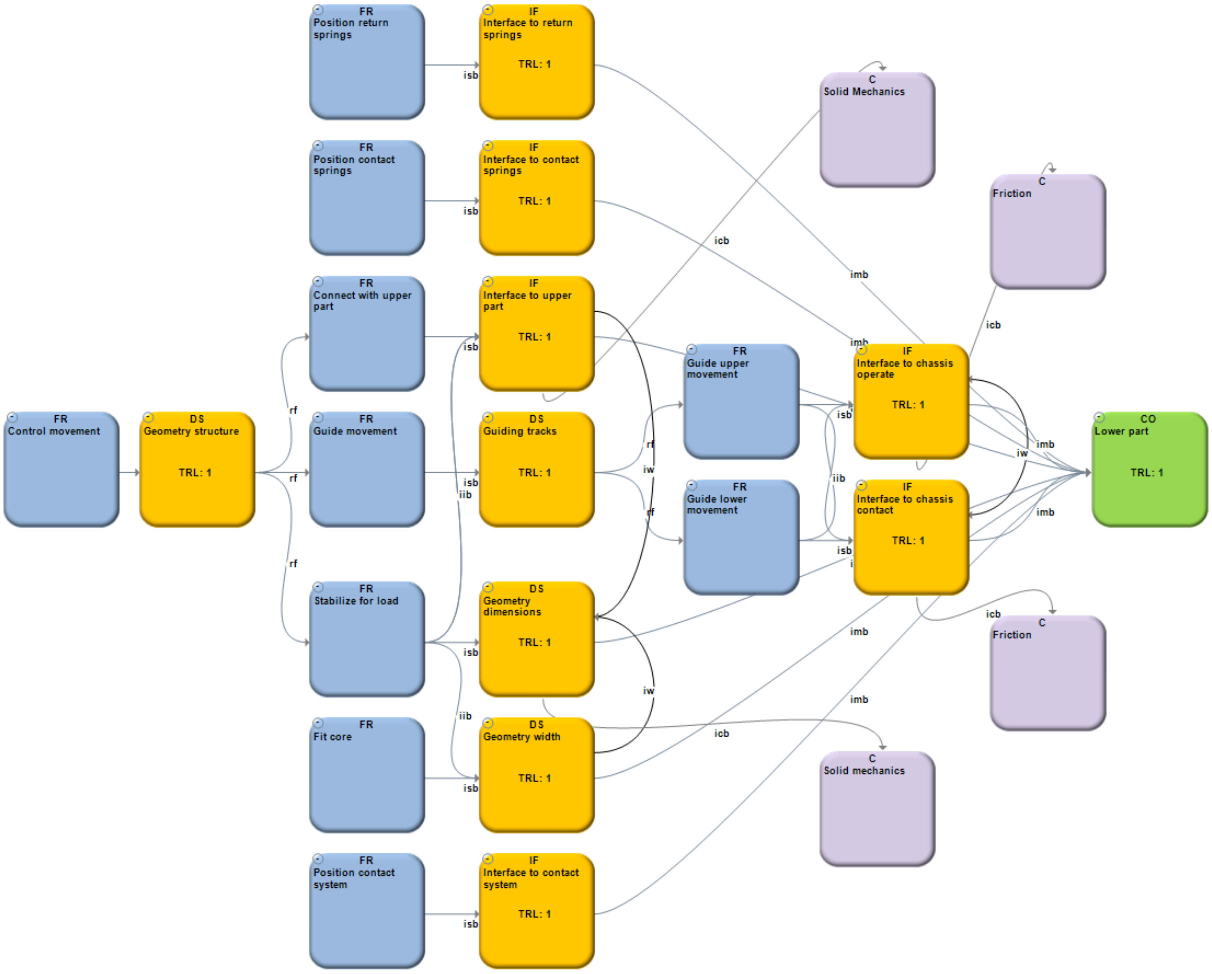


Figure 4.8 The lower part system's design rationale

interfaces to other systems due to that this object is more integrated. The low-level DS's and interfaces are collected in a lower part component that was supposed to be the link to Catia. There are also similar constraints as in the upper part object. The calculations performed in this object are executed in the DS 'Geometry dimensions' and treats the maximal breaking force for the pillars of the lower part.

4.1.8 Pin System Object

The pin system object is the child of the upper part object and acquires inputs from that object and the user.

Control Interface

The inputs are the force and width of the electromagnet as well as the elastic moduli. The output of this object is the width of the pin and it is transferred to the upper part object.

Design Rationale

The object's DR describes the solution to the FR 'Fix anchor' and is presented in Figure 4.9. The DR has two FR's. One treats the handling of the force from the electromagnet and is solved by a geometry structure, the other requirement is to fit with the moving anchor and the solution to that is an interface with the electromagnet. This DR is visualized by the component 'Pin'. The constraints on this object treats solid. The calculations performed in this object are focused on the deflection that is generated by the electromagnet. In this calculation the system is simplified to be look upon as a beam with a distributed force and from that the deflection is computed.

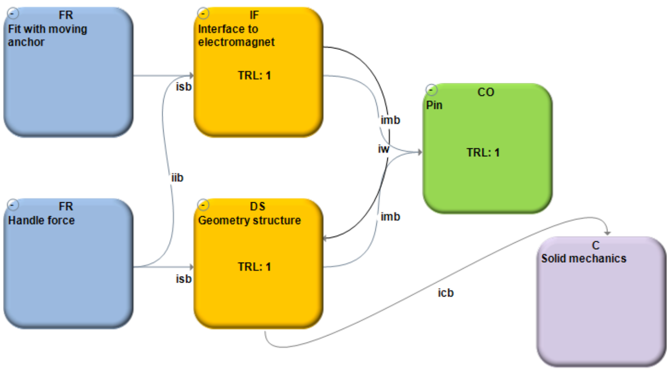


Figure 4.9 The pin system's design rationale

4.1.9 Spring System Object

When further analysed it became clear that it was possible to generalise the two spring systems' DR's when looked upon on a system level. This resulted in a single spring system object using different parameters to acquire the two spring dimensions.

Consequently, this object has two parent objects. When instantiated by the contactor object, one of the parent objects, it is referred to as a return spring and instantiated by the contact system, the other parent object, it is referred to as a contact spring.

Control Interface

The inputs to this object from the parent objects are the force on the spring in active mode, the force on the spring in passive mode, the max diameter, the breaking distance and the operating distance.

Design Rationale

When looked upon as separate systems the two springs were defined as:

- Return spring system - generate force
  For the system to generate the force necessary the solution was a coil spring, constrained to be linear. Additional requirements were fit with chassis system, solved by an interface to chassis system, and distribute force, solved by an interface to contact bridge system.
- Contact spring - generate force
  For the system to generate the force necessary the solution was a coil spring constrained to be linear, like for the return spring system. Additional requirements were fit with the contact bridge, solved by an interface to contact bridge system, and distribute force, solved by an interface to contact system.

The generic definition of the spring system object is presented by three functions and can be seen in Figure 4.10 The spring system's design rationale.

- **Generate force** to enable movement
- **Position A-end** to be able to store and release energy
- **Position B-end** to be able to distribute the force to the intended target

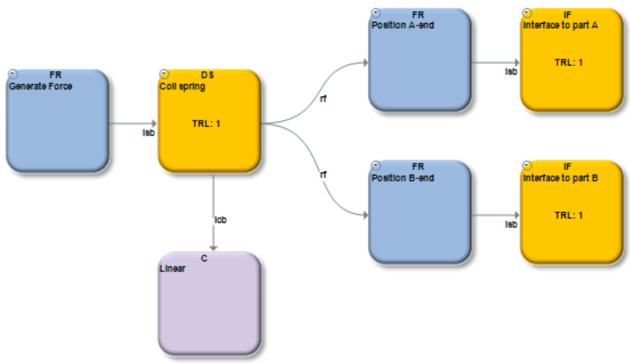


Figure 4.10 The spring system's design rationale

4.2 The Generation Process

The process to generate a contactor variant is described by the following steps:

1. The user set the parameters in CCM and decides which ones to vary.
2. CCM begins its calculations and sends the spring characteristics, the electromagnet's width, the coil thread's diameter, and the contactor's stroke to Simplorer.
3. Simplorer executes a simulation based on the acquired parameters and exports the armature's minimal position as well as the power of the specific electromagnet set.
4. CCM reads the data and checks that the armature fulfilled to shut the circuit.
5. CCM performs the remaining calculations and presents a result consisting of the value of the chosen output parameters.

When generating multiple variants this process is iterated one time for each variant.

4.3 Platform Execution

The platform model was executed by using the generation process described in the previous section and setting the bandwidth for the desired variants. Hence, by setting the bandwidth for all parameters according to the scope, all possible variants were generated. The bandwidth included from 400 A to 900 A for the current, 0.5 mm to 1.5 mm for the thread diameter, and 90 mm to 130 mm for the electromagnets width.

The automated generation process had a duration of six hours until it was finished, the result consisted of 1000 contactor variants. CCM presented the variants in the form of an Excel list at the end of the process. Although, they are still accessible in the software in order to provide visualisation of any desired variant. Nevertheless, each variant in the Excel list was defined by unique values for the chosen output parameters, the output parameters are presented in the top row in Table 1.

Table 1 Example of generated variants

Concept	totalCost [kr]	chassisSize	Powercost [WKr]	Current [A]	Electromagnet Width [m]	Electromagnet PosStatus	Electromagnet SizeStatus	Electromagnet Power [W]	F0 [N]	F2 [N]	F1 [N]	F3 [N]	Stroke [mm]	Thread-Diameter [m]
Concept 900	462	Series 8	6750000	900	0.08	OK	The magnet fits	14600	52.3	488	57.5	935	22.5	0.0015
Concept 899	462	Series 8	5170000	900	0.08	OK	The magnet fits	11200	52.3	488	57.5	935	22.5	0.0014
Concept 920	474	Series 8	4510000	900	0.09	OK	The magnet fits	9510	52.3	488	57.5	935	22.5	0.0015
Concept 898	462	Series 8	4100000	900	0.08	OK	The magnet fits	8880	52.3	488	57.5	935	22.5	0.00132
Concept 919	474	Series 8	3430000	900	0.09	OK	The magnet fits	7240	52.3	488	57.5	935	22.5	0.0014
Concept 897	462	Series 8	3320000	900	0.08	OK	The magnet fits	7180	52.3	488	57.5	935	22.5	0.00125

The contactor variants were evaluated based on the circuits intended current and their powercost, essentially the material cost multiplied with the power generated by the electromagnet. A high powercost meant that the variant was both costly to manufacture and had a high power, in other words both expensive and energy-consuming for the customer. As a result of the large extent of generated variants, a numerical analysis was discarded in favour for a visual analysis for each current.

When the material cost and the power was plotted against the concepts, after being sorted by the powercost in descending order, an interesting interaction between the two parameters' curves appeared in the graph. What can be seen in Figure 4.11 is that, meanwhile the cost is fluctuating in a rather tight span, the power has a more exponential curve. After analysing these to curves further the conclusion was that where the curves meet the optimal contactor variant should be found, see the highlighted area in the figure. Optimal in this case is defined as a contactor that satisfies the customer needs the most. Hence, a contactor that has a high power is costly for the customer during its life time in terms of energy consumption. The most optimal variants can be found in the area where the power is low and the cost has its local minimums, highlighted by the dashed lines in the figure. Although, it must be secured that the variants can handle the overloads that may appear in the intended application.

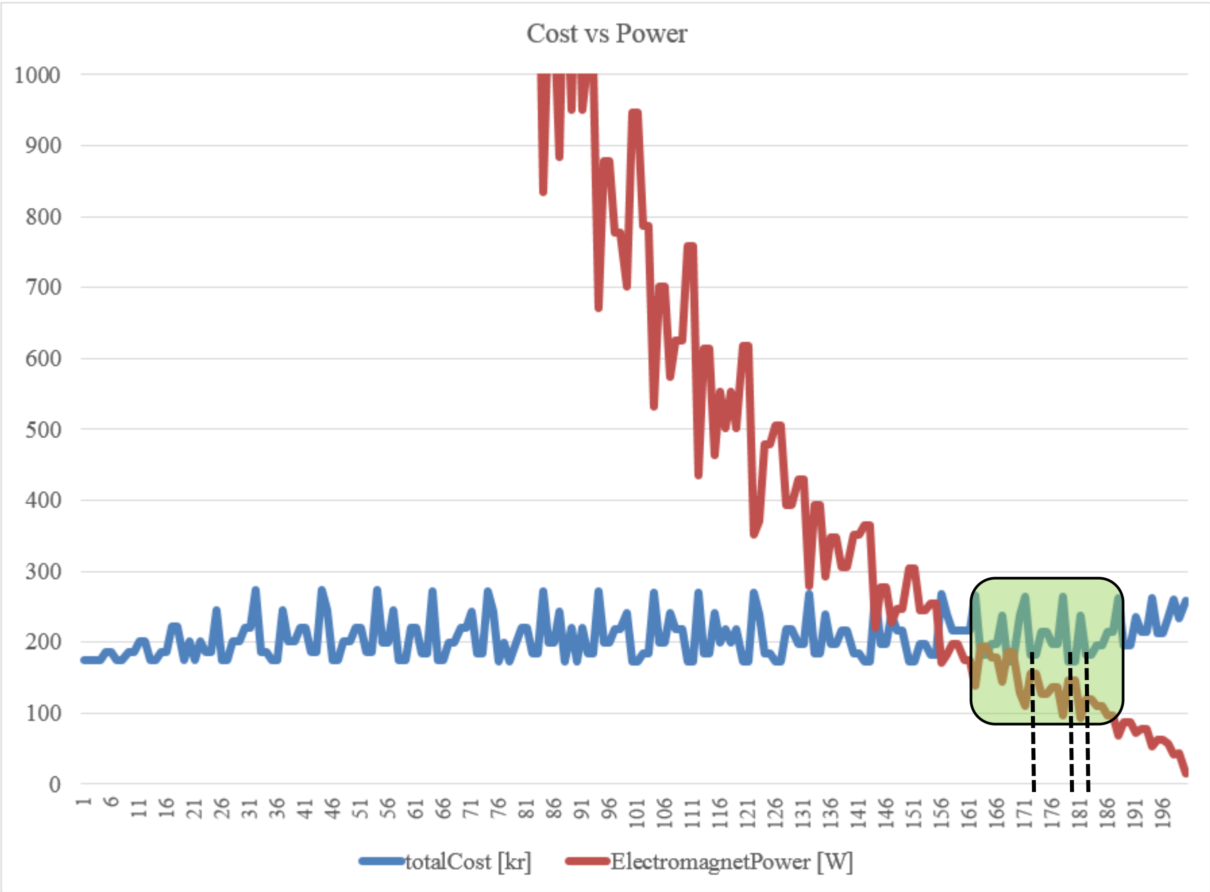


Figure 4.11 Trade-off curve

An important note is that the generation process did not have a screening process, as a consequence both feasible and non-feasible variants were included in the result. Therefore, when further studying the variants, in order to find the most suitable variant for the current, the top candidates must be analysed numerically as well.

5. Discussion

This section discusses the different building blocks that contributed to the master thesis. Firstly, the IT tools and models are discussed. Secondly the results are discussed in general, followed by focusing on the CCM model and the generation process.

5.1 IT Tools and Models

In order to provide an as useable final solution as possible for ABB and to enable ease of support, the software tools were chosen based on what ABB already uses. Although, one exception is the CAD software, due to greater knowledge and skills in Catia it was decided to use that tool instead of CREO which is used by ABB. This may have some contribution to ABB's decision not to implement CCM in their product development process, due to that they would have to allocate resources for creating parameterised CAD models and connect them to the CCM model.

Regarding the models obtained from ABB, they were not studied on a more detailed level than acquiring a solid understanding of their function and performance. If greater knowledge would have been acquired perhaps the simulations could have been expanded to include further data to design a less limited contactor.

5.1.1 CCM Software

During the duration of the project the CCM software was continuously developed according to its own agenda, meaning that a few functions desirable for this project were not prioritised or mature enough for implementation. Examples of functions of that kind are the interaction and interface functionality. Hence, from the projects point of view it would have been most desirable to have communication between CC objects that were not strictly related in the hierarchy but had geometrical relationships that influenced each other.

However, many functions necessary for this project were implemented as the programmer was very responsive. Those times when there was not a distinct solution, a work-around was provided, with excellent results.

5.2 Results

The achieved result was the outcome of a, as previously mentioned, rather limited contactor, due to the earlier stated delimitations, e.g. disregarding breaking technique. Therefore, it is not fair to compare a real contactor, which is in production today, one-to-one with the variants generated through CCM in this master thesis. On the other hand, what can be assumed is that if the current CCM model would have been further developed the result would converge towards a one-to-one relationship. Thus, the tendencies which can be concluded show that the result provided in this thesis are reliable on a conceptual level with a produced contactor.

5.2.1 CCM Model

The statement that the model's result would converge towards the results of real contactor has its founding in that during the thesis there has been a number of reconciliations with engineers from ABB. These have strongly influenced the information included in the model. Consequently, this model has only proved its reliability for ABBs AF contactors, other types and brands have not been studied in this thesis.

Another aspect is that the development process has not been based on a non-existing product, meaning that already existing solutions have been chosen and that have tainted the creation of the EF-M tree as well as the choosing of suitable subsystems.

Regarding the cost and solid mechanic calculations, these have been approximated and simplified in order to provide rough data but not prolong the development of the model more than necessary. Hence, these simplifications have impact on the reliability of the modelled on a detailed level but not as much on a conceptual level.

5.2.2 Generation Process

Regarding the generation of all variants, when comparing the time, it took to generate the variants in CCM versus the time it takes for engineers at ABB to acquire the same amount of data today, it must be put in the right context. Since ABB already has the core data regarding their contactors they do not need to take time to create models. However, if they were to adopt the methodology of configurable components they would have to relocate time and resources to create the desired product platform and to further develop the model to achieve a higher grade of reliability. Although, this resource allocation can be seen as an investment, because in the long run the development process probably would become a lot quicker when creating product families of AF contactors.

When the 1000 variants were evaluated they were ranked after the powercost, and the optimal contactor variants were to be found were curves for the cost and the power intersected. The question raised is, "*what is the definition of the optimal contactor?*" That has not been further studied in this master thesis but would be interesting to investigate and could enhance the model additionally.

6. Conclusion

The conclusion presented in this section are based on the discussion in the previous section in comparison to the stated purpose of the master thesis.

The purpose of this master thesis was to provide ABB with a product platform development solution and to further develop the utilisation of the CCM software. Regarding the solution to ABB, with the limitations in mind, this have been fulfilled. It is possible to generate variants of the AF-contactor with attributes that resemble to ABB's contactors of today. Regarding the utilisation of the CCM software, it has now been proven possible to automatically generate multiple variants of the contactor. This includes external software providing simulated data during the execution loops, as well as visualisation of the variants.

Consequently, it can be stated that this master thesis provides significant evidence that configurable components and the CCM software can be used to make the platform development process more efficient and dynamic.

7. Further Development

The result of this master thesis has a number of possible options for further development, in order to become a more finalised solution. The options can be described by the following focus areas: optimisation, expanding the CCM model, and CCM functionality.

7.1 Optimisation

As previously mentioned, there are not any screening methods included in the generation process. Consequently, all variants within the bandwidths are included in the result. A suggestion is to separate the variants based on if they are feasible or non-feasible. Hence, this would provide a quantity of data that is more easily analysed. This can be executed in a number of ways, e.g. having a parameter to check the specific status parameters. Thus, then the user can, through the Excel sheet, use filters to exclude the variants that are not feasible. Another way of doing that could be to let the CCM software read that parameter and then the exclusion is controlled by the software.

Another optimisation option, on the product level, is to include more external analyses of the contactor. An example of this could be to include FEM calculations on the CAD models, in order to optimise material use and geometrical structures.

7.2 Expanding the CCM model

The contactor model contained limited information regarding contactors, in order to become more useful for ABB it would have been beneficial to expand the model to include more information, e.g. the breaking system, electronic control, etc., until the full contactor system is described.

7.3 CCM functionality

For the model to be further utilised in its current status there are several suggestion of improvements for the CCM software.

One suggestion is to enable for CCM to write from different objects to the same Excel sheet, this would provide further possibilities to control the CAD parts from each object that contain one or more components.

Another suggestion is to define and enable the functions for interfaces and interactions between objects. By implementing those functions, the CCM model probably would have had a faster execution time, and it would have opened up the model for a more dynamic design process in terms of geometrical structures.

The final suggestion is to create a dedicated user interface, where the user can set all object specific parameters from one form. This would ease for the user to keep track of constants and enable a more rapid process for applying for example different material parameters to objects.

References

ABB, 2014. *AF 3-pole contactors.* [Online]
Available at: http://new.abb.com/low-voltage/products/motor-protection/3-pole-contactors-and-overload-relays-for-motor-starting/afcontactors
[Accessed 26 November 2014].

ABB, 2016. [Online]
Available at: http://www07.abb.com/api/ir/getimage/79f380f6-3f1e-4a75-bcf4-0040f337c7b7/2
[Accessed 22 05 2016].

All about circuits, 2014. *Contactors.* [Online]
Available at: http://www.allaboutcircuits.com/vol_4/chpt_5/2.html
[Accessed 26 November 2014].

Ansys Inc., 2016. *Ansys Maxwell.* [Online]
Available at: http://www.ansys.com/Products/Electronics/ANSYS-Maxwell
[Accessed 14 May 2016].

Ansys Inc., 2016. *Ansys Simplorer.* [Online]
Available at: http://www.ansys.com/Products/Systems/ANSYS-Simplorer
[Accessed 14 May 2016].

Dassault Systèmes, 2016. *Catia.* [Online]
Available at: http://www.3ds.com/products-services/catia
[Accessed 14 May 2016].

Edholm, P. et al., 2016. Platform Systems Engineering Design: Software Tool and Industrial Case Studies. *Work in progress,* Issue Product and Production development, Chalmers University of Technology.

Gedell, S., Michaelis, M. T. & Johannesson, H., 2011. Integrated Model for Co-Development of Products and Production Systems - A Systems Theory Approach. *Concurrent Engineering,* 19(2), pp. 139-136.

Johanneson, H., 2014. Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms. In: Sparkston, ed. *Advances in Product Family and Product Platform Design: Methods & Applications.* New York: Springer Science + Business Media, pp. 119-148.

Johannesson, H., 2014. Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms. In: T. W. Simpson,, J. Jiao, Z. Siddique & K. Hölttä-Otto, eds. *Advances in Product Family and Product Platform Design.* New York: Springer Science+Busniess Media, pp. 119-149.

Johannesson, H. & Claesson, A., 2005. System product platform design: a combined function-means and parametric modeling approach. *Journal of Engineering Design,* 16(1), pp. 25-43.

Johannesson, H., Edholm, P. & Andersson, M., 2014. *Object Oriented Platform Systems Engineering,* Gothenburg: s.n.

Johannesson, H., Landahl, J., Levandowski, C. & Raudberget, D., 2016. Platform Systems Engineering Design: Theory and Methodology. *Work in Progress,* Issue Product and Production Development, Chalmers University of Technology.

Levandowski, C. E., 2014. *Platform Lifecycle Support using Set-Based Concurrent Engineering,* Gothenburg: Department of Product and Production Development .

Michaelis, M. T., 2013. *Co-Development of Products and Manufacturing Systems Using Integrated Platform Models,* Gothenburg: Department of Product and Production Development.

Microsoft, 2016. *VBScript.* [Online]
Available at: https://msdn.microsoft.com/en-us/library/t0aew7h6.aspx
[Accessed 14 May 2016].

PE Geometry, 2015. *Design Rationale and EFM Modeling Theory v1.0.* Gothenburg: s.n.

Robertson, D. & Ulrich, K., 1998. Planning for Product Platforms. *Sloan Management Review,* 15 July, pp. 19-31.

Schachinger, P. & Johannesson, H., 2000. Computer modelling of design specifications. *Journal of Engineering Design,* 11(4), pp. 317-329.

Appendix A – Visual basic script

```
‘ --------------------------------------------------------------------------------
‘ A script importing parameters to Simplorer from an Excel document,
‘ executing a simulation and exporting the result to a CSV-file.
‘ Created by David Renborg
‘ --------------------------------------------------------------------------------
Dim emWidth
Dim strokeL
Dim threadD

Set objExcel = CreateObject("Excel.Application")
Set objWorkbook = objExcel.Workbooks.Open ("C:\CCM\Excel\toSimplorer.xlsx")

emWidth = objExcel.Cells(2, 1).Text
strokeL = objExcel.Cells(2, 3).Text
threadD = objExcel.Cells(2, 2).Text

objExcel.Quit

Dim oAnsoftApp
Dim oDesktop
Dim oProject
Dim oDesign
Dim oEditor
Dim oModule
Set oAnsoftApp = CreateObject("AnsoftSimplorer.SimplorerScript")
Set oDesktop = oAnsoftApp.GetAppDesktop()
oDesktop.RestoreWindow
Set oProject = oDesktop.SetActiveProject("PPUX60_150305")
Set oDesign = oProject.SetActiveDesign("Simplorer1")
oDesign.ChangeProperty Array("NAME:AllTabs", Array("NAME:LocalVariableTab", Array("NAME:PropServers", "Instance:0;Simplorer1"), Array("NAME:ChangedProps", Array("NAME:width", "Value:=", emWidth+"meter"), Array("NAME:stroke", "Value:=", strokeL+"mm"), Array("NAME:wire_diameter", "Value:=", threadD+"meter"))))
oDesign.Analyze "TR"
Set oModule = oDesign.GetModule("ReportSetup")
oModule.ExportToFile "Output Data", "C:/CCM/Excel/PPUX60go.csv"
```

Appendix B – Variables, constants and equations

Variables

width | *thread diameter* | *stroke*

Constants

bobin = 0.03 m | *fill factor* = 0.9 | *resistivity* = 1.67E-8

play = 0.0025 m | *wire insulation* = 2.5E-5

Equations

$base = 0.116667 * width$ $side = 0.116667 * width$ $top = 0.116667 * width$

$length = \frac{width}{3}$ $depth = 0.7333333 * width$ $piston = \frac{width}{4}$

$section\ width = \frac{width}{2} - side - bobin - (\frac{piston}{2} + bobin)$

$section\ height = depth - top - base - side - 2 * bobin$

$section\ area = section\ width * section\ height$

$inner\ length = 2 * (piston + length + 2 * bobin)$

$outer\ length = inner\ length + 2 * \pi * section\ width$

$$average\ length = \frac{(inner\ length + outer\ length)}{2}$$

$$n_{turns} = \frac{fill\ factor * section\ area}{(\pi * (thread\ diamter + 2 * wire\ insulation) * \frac{(thread\ diamter + 2 * wire\ insulation)}{4}}$$

$$wire\ resistance = resistivity * average\ length * \frac{n_{turns}}{\pi * thread\ diamter * \frac{thread\ diameter}{4}}$$