THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Development of Interdisciplinary Platforms Using System Objects

JONAS LANDAHL

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2016

Development of Interdisciplinary Platforms
Using System Objects
JONAS LANDAHL

ISSN 1652-9243
Report no 102

Department of Product and Production Development
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
The cover illustration is a depiction of the strategic move that manufacturing companies need to address to achieve mass customization of complex products.
Line work made by Christoffer Löfberg.

Printed by
Chalmers Reproservice
Gothenburg, Sweden 2016

PREFACE

Spring, early 2000. Night shift at a vehicle supplier.

An assembly line for vehicle seats was nearly up and running. A new contract with a prominent vehicle manufacturer made it possible for the vehicle supplier to increase its turnover significantly. They had the potential to benefit from high production volumes, although they could only reuse few product components from former projects developed in collaboration with other vehicle manufacturers.

The workers in the production plant consisted of people of different ages, with different backgrounds and experiences. Most of them wanted nothing but receiving their paychecks each month to support their livelihood and make a decent living. I was 20 years old, had only slight experience of production, none of assembling vehicles. I had little responsibility, but was determined to undertake my tasks and do a decent job. After assembly, the vehicle seats were inspected for quality flaws, then packaged and sent to the vehicle manufacturer, where the seats were mounted into the vehicle. My job was to detect quality flaws before the seats were sent to the vehicle manufacturer.

Conveyor belts, lifting tools and screwdrivers were installed, the assembly sequences were set and the first production series was initiated. In this production ramp-up phase, a flaw suddenly became apparent. The seat did not fit into the vehicle. The diameter of a hole in the frame of the seat, with the function of holding the seat in the vehicle, was too small for a rod to enter the hole. At this point there was no time for flaws. Scraping over hundreds of manufactured seat frames, redesigning and remanufacturing of the seat frames would include high cost and, foremost, delay of the complete assembly line at the vehicle manufacturer. Such delay is tremendously costly. A production plant with hundreds of paid workers, installment of thousands of machines and large buildings, all in all, produce high cost every second.

After digging into the cause of the problem, a late design change at the vehicle manufacturer was found to give rise to a misconception. The diameter of the rod in the vehicle was changed, while the hole in the frame was never adjusted to the new specification. Thus, the design change was poorly communicated to the design team at the supplier. To manage the new situation, manual modification of already manufactured seat frames was decided upon, to prevent costly delays. During a couple of days, my work tasks were changed.

A few colleagues and I, were situated in an empty industrial building with a few large tables, drilling machines, and a bunch of containers comprising around 30 seat frames in each. The working procedure looked something like this: I went to a container, took a frame from the racking, carried it to the table and placed it on top of it. I grabbed the drilling machine and aimed towards the hole. Metal splinters were pouring as the diameter of the hole was expanding. I turned the frame 180 degrees and repeated the drilling on the other side of the frame. Thereafter, I went to return the modified frame into a new racking, and again passed the first racking to pick up another frame to drill.

During my work, presented in this thesis, I have been working on upcoming enablers to prevent late design flaws that drive design rework, especially design rework that causes high development cost and potential delays of introducing the product on the market. In this thesis the underlying industrial problems, the scientific mission undertaken and the proposed and validated solutions, based on industrial collaborations, are presented as means of increasing the awareness of these enablers for practitioners in academia, as well as in industry.

ABSTRACT

Conceptually "form follows function", as stated by Louis Sullivan. Physically however, form follows producibility.

There is an erroneous notion that many customers can be provided with customized products, produced without any battle. In reality however, behind the walls of manufacturing companies, there is a constant struggle to combine the functionality and performance of complex products, while taking knowledge from manufacturing into account.

Thankfully, there are a few shortcuts upon which manufacturing companies can capitalize. One of them is *design reuse*, i.e. there are elements of existing products and processes that can be reused in designing new products. Such elements do not have to be physical items. They can, for example, be laws of physics, mechanical properties, or known limitations of a manufacturing process. The use of *platforms* has proven efficient for design reuse. Platforms can accommodate product families, rather than a single product. By using platforms, common elements can be shared among the products in a family. Such design reuse has the potential of reducing development cost and make the development process more efficient.

However, when developing a family of products, the evaluation of performance and producibility of multiple design alternatives becomes even more challenging, compared to that of a single design. Thus, to overcome this challenge, a platform development methodology is proposed to support the creation and evaluation of multiple design alternatives. First, we need to gather and store knowledge about similar products. Secondly, we need to be able to generate a family of design alternatives, and to somehow rank the family of alternatives by their feasibility. Thereafter, the inferior alternatives can be eliminated, and the good alternatives can form the basis for further development.

Based on several industrial studies, the methodology has been developed to support modeling of known designs, generate several new alternatives, evaluate them and eliminate the bad ones. The methodology provides design engineers and system architects with the methods and tools needed to make credible design decisions early in platform development. The methodology builds on existing theoretical models, methods and tools, and describes platform system objects that support design reuse. The methodology serves three development levels, reflecting the level of design detail of the product family: 1) *functional level;* functions and alternative ways of solving them, 2) *system level;* system objects with design parameters, and 3) *detailed level;* conceptual 3D shapes. The three levels can be used iteratively, as the platform matures throughout the development process.

The aim is to support design engineers and system architects in developing platforms for the early phases of development, and provide them with the basis for harmonizing between product performance and manufacturing capabilities, partly to reduce late and costly design modifications due to inferior producibility of products, but also to be prepared for future demands by accommodating the needs of a range of customers rather than a single customer.

Keywords: product development, production development, mass customization, platform-based development, design reuse, concurrent engineering, interdisciplinary development, producibility, systems engineering, function-means modeling, flexible, adaptable, configurable, reconfigurable, product architecture, interchange of information.

ACKNOWLEDGEMENTS

The work presented in this thesis was carried out at the Department of Product and Production Development, at Chalmers University of Technology in Gothenburg, Sweden. It has received support by the Swedish Governmental Agency for Innovation Systems (VINNOVA) through Wingquist Laboratory VINN Excellence Centre, and the Strategic Programme for Production Research and Innovation in Sweden (Production2030) under the umbrella of the Production Area of Advance at Chalmers. I gratefully acknowledge their support.

I would like to send my deepest appreciations to my supervisor, Emeritus Professor Hans Johannesson, for bringing me into his research team. Your patience and valuable advice, and your support and belief in me have been tremendously encouraging. Thank you Hans. I also want to thank my co-supervisor, PhD Christoffer Levandowski, for all your support during my work, especially during my questioning times and in discussions about research. Much of my learnings in research are due to your patience and cheerful attitude, as well as a supervisor and friend. Thank you Christoffer. I also want to thank Professor Rikard Söderberg for making my research project possible through the infrastructure of the Wingquist Laboratory. Thank you Rikard.

A special thanks goes to all industrial collaborators, especially Professor Ola Isaksson and PhD Johan Vallhagen at GKN Aerospace Sweden AB, PhD Johan S Carlson and Jonas Kressin at Fraunhofer Chalmers Center (FCC). Thank you for great collaboration, valuable discussions, and more to come.

I also want to thank my co-authors for the great discussions and eye-opening moments: Daniel Stenholm, Dag Bergsjö, Dag Raudberget, Konstantinos Stylidis, Peter Edholm, Magnus Andersson, David Renborg, and Kristina Wärmefjord. Thank you all.

Brilliant colleagues, thank you for your inspiring discussions during coffee breaks and lunches. I am especially grateful for the supportive and cheerful company of Anders Forslund, Daniel Corin Stig and Julia Madrid. Thank you for intriguing conversations during lunches, support in common courses, and when supervising students.

And to my friends and family, thank you for your successive support, in good times and in bad times. Your support provides me with strength.

Jonas Landahl
Gothenburg, March 2016

APPENDED PUBLICATIONS

The following publications are used to underpin the research presented in this thesis.

Paper A. Landahl, J., Bergsjö, D., Johannesson, H. (2014). *Future Alternatives for Automotive Configuration Management*. Procedia Computer Science. 12th Annual Conference on Systems Engineering Research – CSER 2014, March 21-22, Redondo Beach, CA; United States.

Work distribution

Jonas Landahl wrote most of the paper, but both Dag Bergsjö and Hans Johannesson complemented with some parts. The interview study was prepared and executed by Jonas and Dag. Jonas analyzed the data, with support from Dag. Hans provided comments and advise.

Paper B. Johannesson, H., Landahl, J., Levandowski, C., Raudberget, D. (2016). *Platform Systems Engineering Design: Theory and Methodology*. Journal of Intelligent Information Systems (JIIS), Special Issue on Personalization and Mass Customization. (*Submitted*)

Work distribution

Jonas Landahl, Christoffer Levandowski and Hans Johannesson synthesized the theory and wrote the paper. Jonas and Christoffer described the aerospace sub-system case. Hans and Dag Raudberget described the vehicle seat case. Jonas described the electromagnetic contactor case.

Paper C. Edholm, P., Andersson, M., Landahl, J., Renborg, D., Levandowski, C., Johannesson, H. (2016). *Platform Systems Engineering Design: Software Tool and Industrial Case Studies*. Journal of Intelligent Information Systems (JIIS), Special Issue on Personalization and Mass Customization. (*Submitted*)

Work distribution

Jonas Landahl, Christoffer Levandowski and Hans Johannesson synthesized the theory and wrote the paper. Jonas and Christoffer developed the aerospace sub-system case in collaboration with GKN Aerospace Sweden AB. Hans, Peter Edholm and Dag Raudberget worked in collaboration with automotive supplier, and developed the vehicle seat case. David Renborg, Peter and Hans developed the electromagnetic contactor case in collaboration with the company from the power industry. Peter and Magnus Andersson developed the software, CCM, in collaboration with the researchers, to reflect the methodology presented in Paper B.

Paper D. Landahl, J., Levandowski, C., Johannesson, H., Söderberg, R., Wärmefjord, K., Carlson, J. S., Kressin, J., Isaksson, O., Vallhagen, J. (2016). *Using Product and Manufacturing System Platforms to Generate Producible Product Variants*. 6th

CIRP Conference on Assembly Technologies and Systems – CIRP CATS 2016, May 16-18, Gothenburg; Sweden.

Work distribution

Jonas Landahl and Christoffer Levandowski synthesized the theory, wrote the paper, and developed the PLM architecture and process. All authors developed the case in close collaboration, during workshops. Hans Johannesson, Rikard Söderberg, Krisitna, Johan S Carlson, Ola Isaksson and Johan Vallhagen provided comments and feedback.

ADDITIONAL PUBLICATIONS

The following publications are related to the research presented in this thesis, but do not fully contribute to its results.

Stenholm, D., Landahl, J., Bergsjö, D. (2014). *Knowledge Management Life Cycle: An Individual's Perspective*. 13th International Design Conference – DESIGN 2014, May 19-22, Dubrovnik; Croatia.

Landahl, J., Raudberget, D., Johannesson, H. (2015). *Assessing System Maturity of Interacting Product and Manufacturing Alternatives Before Early Technology Commitment*. Annual International Association for Management of Technology Conference – IAMOT 2015, June 8-11, Cape Town; South Africa.

Stylidis, K., Landahl, J., Wickman, C., Johannesson, H., Söderberg, R. (2015). *Structuring Perceived Quality Attributes for use in the Design Process*. 20th International Conference on Engineering Design – ICED15, July 27-30, Milan; Italy.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AM – Additive Manufacturing
3D – Three Dimensional Space
BOM – Bill of Materials
C – Constraint
CAD – Computer Aided Design
CAE – Computer Aided Engineering
CAx – Computer Aided Technologies
CC – Configurable Component
CCM – Configurable Component Modeler
CE – Concurrent Engineering
CI – Control Interface
CS – Composition Set
DP – Design Parameter
DR – Design Rationale
DS – Design Solution
DfA – Design for Assembly
DfM – Design for Manufacturing
DSM – Design Structure Matrix
e.g. – exempli gratia (for the sake of example)
et al. – et alii (and others)
EF-M – Enhanced Function-Means
F-M – Function-Means
FMS – Flexible Manufacturing Systems
FR – Functional Requirement
GBOM – Generic Bill of Materials
IA – Interaction
iaio – is an implementation of
icb – is constrained by
i.e. – id est (that is)
IF – Interface
iib – is influenced by
ipmb – is partly met by
isb – is solved by
IT – Information Technology
iw – interacts with
PDM – Product Data Management
PLM – Product Lifecycle Management
PMC – Platform Modeling and Configuration
rf – requires function
RFQ – Request for Quotation
RM – Requirements Management
RMS – Reconfigurable Manufacturing Systems
RQ – Research Question
SBCE – Set-Based Concurrent Engineering
SE – Systems Engineering
ToD – Theory of Domains
TRS – Turbine Rear Structure
TTS – Theory of Technical Systems
VP – Variant Parameter
VPV – Variant Parameter Value

LIST OF FIGURES

I

"Everything around you that you call life was made up by people that were no smarter than you."

– Steve Jobs

INTRODUCTION

The world around *us* is ever-changing; politics, climate, economy, socio-culture, technologies and legislation. If *you* are changing, the world is changing. Change is a precursor for global trends, where driving forces within *you*, such as creativity and curiosity, can be harnessed for the good of the world. Suppose *you* were given three tasks; building a house, chopping vegetables, and holding coins – what would *you* do? One way of solving each task is by using a tool, which *you* can utilize to extend *your* body to aid a given purpose. For example, if *you* build a wooden house, *you* can use a hammer and nails to fix wooden planks together to form the structure of the house. When *you* chop vegetables, a knife is suitable. And, when *you* decide to keep the change, *you* can use a pocket to hold it. The hammer, the nails, the knife and the pocket are all products or design solutions destined to solve certain functions, which add value to *you* and *your* needs.

Products are well thought out structures, which for example have carried humans to the moon – the space ship, and down to the bottom of the deep blue sea of the Mariana Trench – the submarine. At a given point in time, all products have been produced. When *you* think of it, a product is quite often used to produce another product, such as when using the hammer and nails to build a wooden house, or a knife when *you* chop vegetables. But remember that if *your* house is constructed using bricks, the hammer and nails will be replaced by for example a spatula and a mortar. Or if *you* want to chop vegetables into tiny pieces, the knife can be replaced with a domestic appliance. Even though the main function of a product is the same, the choice of solution may vary.

Our needs do not only reflect the functionality *we* seek in products. They also drive global competition. *You* may for example compare product alternatives based on price, performance and quality before *you* decide to buy the solution that best fits *your* needs.

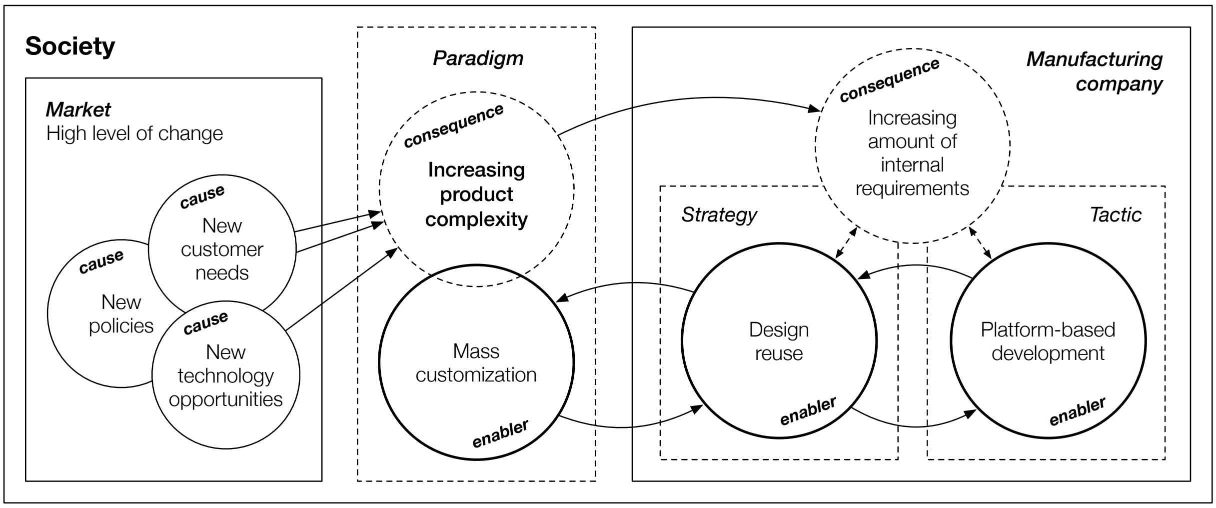

Figure 1. Causes of changing market conditions and their consequences for products, relation to the established paradigm of mass customization, and possible actions of manufacturing companies

Nevertheless, for a product to be bought, it is, first and foremost, necessary to ensure that the product can be produced. Otherwise, it would never see the light of day. As a customer *you* might think less about the producibility of a product, and most likely care about its functionality, performance, quality and price. The challenge of satisfying *your* needs is rather in the hands of those who develop and produce the products – the manufacturing companies. They desire to develop and produce affordable high quality products quickly, to attract customers and increase market shares.

A true challenge for manufacturing companies today is to master the act of responding fast to changing market conditions. These changes are partly driven by new technologies that are continuously implemented into products, the fact that customers constantly want improved product functionality (Wheelwright and Clark, 1992), and that new policies are put into effect (Blanchard and Fabrycky, 1990), such as climate goals – e.g. the EU has committed to reduce emissions by 80-95% below 1990 levels before 2050 (European Commission Climate Action, 2015). For these reasons, products are becoming increasingly complex (Bhise, 2013, Stevens, 1998). To keep up-to-speed with adopting new functionality, not only the increasing complexity alone needs to be taken care of. For manufacturing companies to be truly competitive and react quickly to market needs, the product development process needs to be equally effective and efficient. A proven paradigm model to support swift responses to meet changing requirements is *mass customization* (Ferguson et al., 2013, Jiao et al., 2007, Simpson, 2004, Pine, 1993, Tseng and Hu, 2014). Another approach, or strategy, proven to have significant impact on product development efficiency is *design reuse* (Ong et al., 2008). However, solely adopting these approaches does not guarantee fast responses to changing requirements. Inside the walls of manufacturing companies, people are planning, designing products, designing manufacturing systems, and preparing manufacturing processes to reach the common goal of delivering products that satisfy customers. The time-pressure in product development is typically great, and the flow of information that propel development activities is seldom seamless. Therefore, highly changing requirements come with a risk of slowing down the development process by increasing the amount of design rework. For example, design engineers can design a product with high performance, but it can very well turn out to be inferior in the production phase. In such a case, the product has to be modified iteratively to accommodate true manufacturing capabilities. A reputed tactic serving both *mass customization* and *design reuse* with high potential of serving the interchange of information among and across disciplines and systems is *platform development*.

1.1 BACKGROUND

Traditional research on *platforms* and *design reuse* often focuses on economies of scale for manufacturers of consumer goods in the production phase (Robertson and Ulrich, 1998). The well-reputed approach prior to *mass customization* was mass production. Mass production provides scale benefits based on high production volumes used to finance investments in manufacturing equipment, tooling, engineering and training (Jiao et al., 2003). In the beginning of the 20th century, Ford's mass production system dictated that identical products should be produced over and over again to reduce the overall cost of manufacturing. In recent years, a paradigm of customized products has evolved. Mass customization shall serve future markets without predicting future needs. The concept challenges the contradiction between mass and customization. In this paradigm, manufacturing companies have the potential of meeting customer needs while maintaining efficiency in production over time (Tseng et al., 1996).

As mass production provides scale benefits in production, mass customization aims at providing scale benefits in development, by reducing development cost and lead-time. With the increasing flexibility built into modern manufacturing systems, companies with predicted low to medium production volumes can gain an edge over competitors by implementing mass customization (Jiao et al., 2003). Zipkin (2001) contrasts mass customization to mass production and argues that mass customization requires richer information flows and more stringent requirements for process flexibility. Consequently, challenges of implementing mass customization still remain. Zipkin (2001) continues by stating four such challenges: 1) investigating the potential mass market for customized products, 2) extracting customer needs, 3) developing highly flexible production technology, and 4) implementing direct-to-customer logistics. Ferguson et al. (2013) made a thorough literature review of 130 papers, published since the year 2000, related to mass customization and the product development process. They conclude that more research is needed on requirements management, information flow between disciplines and systems, and methodologies for concept generation and evaluation. They explicitly state, *"the expansion of the mass customization paradigm is dependent on developing rigorous models and tools that support designers throughout the mass customization product development process."*

1.1.1 Platforms – Means of Achieving Mass Customization

Product *platforms* can be used as enablers to achieve mass customization and satisfy a wide variety of customers needs (Jiao et al., 2007). However, researchers have different views of product platforms and how they relate to concepts such as product families, modules and brands (Halman et al., 2003). Simpson et al. (2001) define product platforms as a set of parameters, features and/or components which remain constant for several products within a given product family. In this definition, product families are described as groups of related products that share common features, components and subsystems, all of which can be combined into products to satisfy various market niches. By combining common components with unique components, distinctive products can be configured. However, these platforms have their shortcomings. They are inflexible in development. Components used in such platforms include elements of standardization and are therefore less flexible and less prone to change when there is a need for modifications. Also the inclusion of other valuable resources such as manufacturing capabilities, as well as methods and IT systems, is problematic. When studying platforms, a wider scope is exemplified by Robertson and Ulrich (1998), who describe a platform as a collection of assets, components, processes, knowledge, people and relationships that are shared by a set of products. Another view of platforms is architectures controlled by design (Gershenson et al., 2004). These are characterized by common structures, scaled variables and variable structures, which can support more than one product. This view articulates the need to exchange parts or components and also scaling products to suit certain customer segments. Another view focuses on basic architectures that comprise subsystems or modules with interfaces in-between (Meyer and Lehnerd, 1997). Here, the need for interfaces between interacting systems is emphasized. According to Jiao et al. (2007), platforms are designed for either functional variety or technical variety. Functional variety aims of satisfying diverse customer needs, whereas technical variety aims to reduce in-house variety. Each approach requires its own strategy to address the two divergent advantages searched for in platform development, i.e. the variety to enable customization or the reduction of unique parts to gain economies of scale.

Researchers have also deliberated upon other types of platforms, such as manufacturing platforms, discussed by Erixon et al. (1996), as well as Michaelis (2013). The former researchers use modularization of the product and manufacturing system as a way of increasing the efficiency of development and manufacturing. The latter, Michaelis, describes how co-development of the products and manufacturing systems through integrated platforms can be achieved. Gedell et al. (2011) speak of a unified product and manufacturing system platform. Michaelis and Johannesson (2011) describe the use of functional models for representing the manufacturing system platform and how these functional models can be linked to the product platform using manufacturing operations as linking systems. Closely related to this, Koren et al. (1999) suggest a reconfigurable manufacturing system that accommodates the variety within a product family. Configuration aims at quickly adjustments to changing customer requirements, whereas flexibility of the system itself serves variation within the product family.

Use of Platforms

Researchers have proposed different frameworks, methods, and mathematical tools to define and make use of platforms in various industrial settings, e.g. Jose and Tollenaere (2005), Simpson (2004) and Simpson et al. (2006). A well-known industrial example from Black and Decker is reported by Meyer and Lehnerd (1997) and Simpson (1998). Another industrial example described by, for example Prencipe (1998), is from Rolls Royce. (Pirmoradi et al., 2014) state that platforms are commonly used to support design reuse. Standardization and modularization are ways of increasing design reuse aimed at economies of scale. A number of publications by authors such as Baldwin and Clark (2000) and Ericsson and Erixon (1999) discuss different techniques for standardization and modularization. All these platform approaches are based on parts, or physical elements. Madni (2012) identifies some risks inherent in part-based platform approaches. Such approaches may reduce an the ability of an organization to evolve, because of increased uncertainty regarding the future demand for variants. They also involve a greater technical risk because an error in platform design and architecture will permeate all variants. To reduce late design modifications, cut development lead-time, and gain first-mover advantages, platforms need to support efficiency, not only in manufacturing, but increasingly during the development phases (Wheelwright and Clark, 1992). Platforms based on mixing parts into different arrangements as a way of reusing design, alone do not provide the support needed by that design engineers to improve the development process (Gedell, 2011). This is particularly apparent in engineer-to-order companies (Brière-Côté et al., 2010) where the reuse of physical parts alone may prove insufficient to satisfy a large variety of customer needs. One proven approach to support efficiency in the development of complex products is through abstract design objects as opposed to physical parts.

Design Reuse Flexibility

Majority of products that are developed are not designed from scratch (Bhise, 2013), but involve some kind of design reuse. Design reuse stretches across disciplines, and applies to various engineering roles related to product development, such as mechanical, electrical, software and manufacturing engineering (Ong et al., 2008). There are several ways to accomplish design reuse (Ong et al., 2008). Shahin et al. (1999) advocate four different forms of design reuse: 1) a list of functions and basic requirements representing a concept, 2) an F-M tree representing functions and solutions, 3) a parts tree representing the embodiment design, and 4) a set of drawings or CAD models representing the detailed design. Pahl and Beitz (2013) write about two kinds of embodiment, preliminary embodiment and detailed embodiment. Preliminary embodiment, or overall layout, is represented as a draft or configuration of shapes that through iterative steps will form a detailed embodiment. Soon to be parts, for example represented by a Bill of Materials (BOM), with parametric formulas and other design relationships between them are typically sealed in design software, such as Computer Aided Design (CAD) systems. This affect the flexibility of the design reuse. For example, a function sealed into a product feature as a *function carrier* (Pahl and Beitz, 2013), in a discipline-specific software such as CAD, is less flexible and will restrict the chance of efficient design reuse, or the interchange of information between disciplines (Ong et al., 2008). Hou et al. (2011) suggest a Generic Bill of Materials (GBOM) to increase design reuse flexibility. However, to serve flexible design reuse in the early phases of development, design engineers need information about the functionality of systems and subsystems, how they interact, their limitations and possibilities, and how they came into existence. Modeling such information is a complex process and may require a variety of knowledge sources and modeling methods (Gaines and Shaw, 1992).

The use of formal representations and the reuse of design concepts in the early phases of development are rare. Some examples exist, such as design reuse through function-based design synthesis (Xu et al., 2006), or by the use of functional platforms (Alblas and Wortmann, 2009). Functional platforms enable the reuse of functions and the generation of engineering variants. Their abstract character also allows for the integration of product and production development. A functional platform can make use of subsystems that are scalable, or re-configurable, to fit many different products while fulfilling the same functions. These models need to represent the concepts in such a way that they support the design decisions that propel the development work. The models need to provide a basis upon which credible decisions can be made to find feasible design alternatives. To determine the feasibility of a design in early phases, the model needs to be adaptable and analyses made when the design is immature.

Requirements and Change

Requirements are often used to propel design activities in product development. The truth is that requirements seldom stay constant; they must be changed to meet customer or other stakeholder needs (Almefelt, 2005). Different definitions of requirements have been posed. Harwell et al. (1993) state, *"if it mandates that something must be accomplished, transformed, produced, or provided, it is a requirement – period."* Regarding the increasing complexity in engineering, as well as in manufacturing (ElMaraghy et al., 2012), the extensive outsourcing and increase in platform collaboration, managing requirements has become increasingly difficult (Almefelt, 2005). Management of requirements regards the whole lifecycle of a product, from customer needs to disposal. Software systems have long been utilized to manage requirements (Stevens, 1998), such as Product Data Management (PDM) systems and Product Lifecycle Management (PLM) systems.

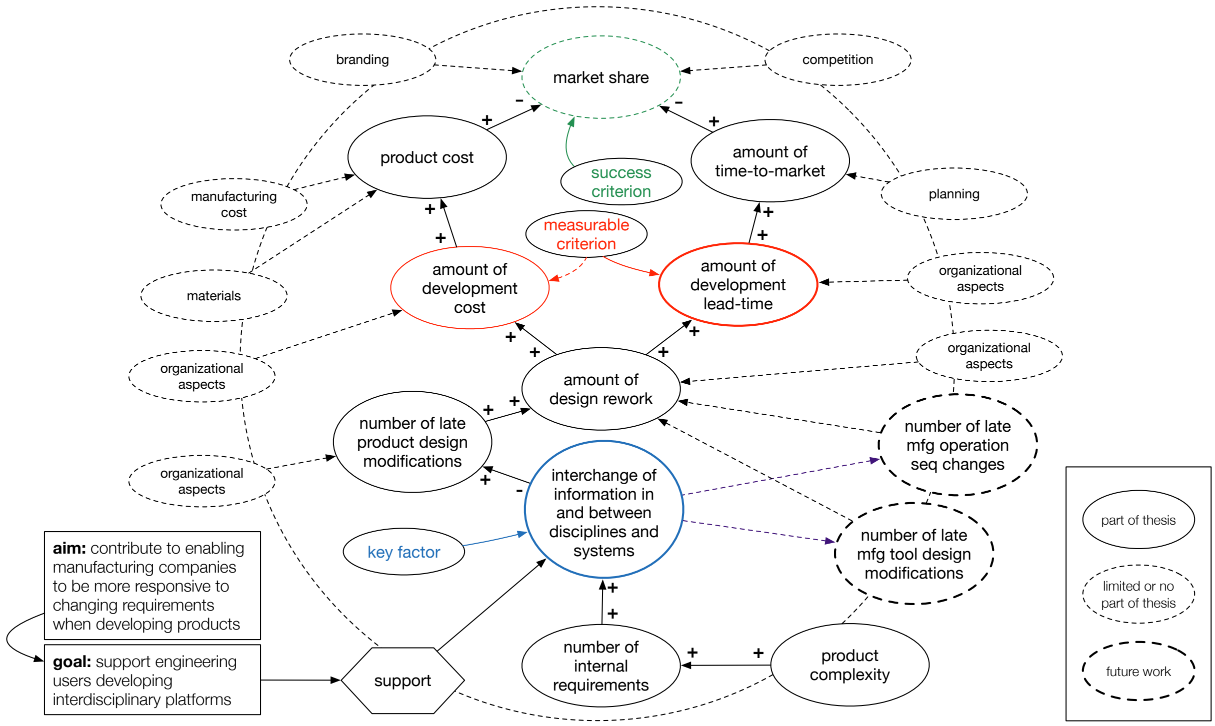

Figure 2. Reference model used to illustrate the network of consequences due to high or low influence of key factor, which is the main studied phenomenon (applied from Blessing and Chakrabarti (2009))

1.2 THE SCIENTIFIC MISSION

The scientific mission has been introduced above. In the following Section, the mission is further broken down into aim and goals. To support the reasoning behind such aim and goals, a reference model (Blessing and Chakrabarti, 2009) has been created; see Figure 2. The reference model is used to illustrate the network of consequences due to the high or low influence of an identified key factor: *the interchange of information among and across disciplines and systems*. The key factor is dependent on the assumption that more product complexity leads to a greater number of requirements. In this thesis, requirements related to products, manufacturing systems and the process needed to develop a producible product family, derived from a platform, have been taken into account.

The possible outcome of the design support is to influence the key factor to accomplish a lesser number of late design modifications, and less design rework. Preferably, these effects can lead to the main measureable criterion, or business objective: *shorter lead-time in development*. The ultimate success criterion or business aim is: *increased market share*. This success criterion is beyond the scope of this thesis.

1.2.1 Aim and Goal

The overall aim of the research conducted is to enable manufacturing companies to be be more responsive to changing requirements during the development of products, whether it is due to new legislation, or due to increased product functionality, for e.g. highly demanding customers. The aim originates in an industrial problem where early made design decisions to increasingly complex products increase development cost and lead-time when the need for late design modifications arises. The way forward, as presented in this thesis, is driven by how theory and practice on *mass customization*, efficient *design reuse* and *the interchange of information among and across disciplines and systems* can be combined to counteract time delays and costly design rework.

Phenomenon

The main phenomenon in this thesis is *the interchange of information among and across disciplines and systems.*

Main Hypothesis

The hypothesis of this thesis is that *increasing product complexity can be managed by engineering users through the development of interdisciplinary platforms*.

1.2.2 Definition of Interdisciplinary Platforms

Manufacturing companies regularly encounter interdisciplinary challenges in product development, such as managing increasing and changing requirements and capabilities. Interdisciplinary entails: among and across planning, design, and manufacturing, as well as among and across a multitude of technical systems, including products, manufacturing equipment and tools, and IT systems.

Requirements have to be taken into account during the development phases to ensure that the product stays within budget, schedule and performance. The product also has to satisfy customer needs and be producible. Compromises between requirements continuously need to be managed. Even though a product, in development, can deliver high performance, it may still prove inferior in manufacturing. Such a product inevitably has to undergo several loops of design modification before producibility can be assured, not seldom by compromising some of the performance of the product.

In this thesis, an interdisciplinary platform, see Figure 3, is regarded as a common denominator for a collection of different disciplines, technical systems and software, encompassed within the scope of product development. The relevant disciplines are those related to the product and its lifecycle. The different disciplines can be studied from various levels of abstraction, such as within a technical system, e.g. *structural*, *aerodynamic*, *thermodynamic*, or *electrical*. It can also relate to external technical systems, and the performance and capabilities of those, such as when manufacturing equipment and tools are used to, in a process, materialize a product.

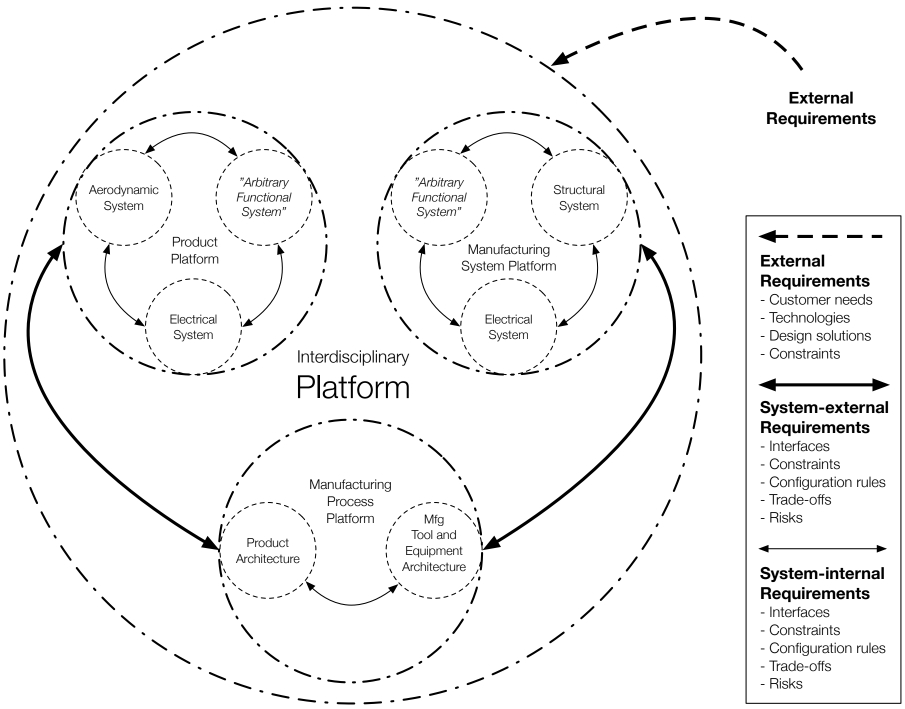

Figure 3. The interchange of information among and across systems of an interdisciplinary platform

1.2.3 Research Questions

Based on the scientific mission, the following research questions were formulated to propel the research work.

RQ 1) What main challenges in the development of complex products exist, and what approaches, models, methods and tools are suitable for meeting these challenges to achieve efficient interdisciplinary platform development?

The answer to *RQ 1* is first and foremost based on a comprehensive literature review; however, both descriptive and prescriptive studies have been conducted to create an increased understanding of industrial and academic needs.

RQ 2) How can interdisciplinary platforms be developed and by whom, to ensure the efficient interchange of information among and across disciplines and systems and what are the implications of such implementation?

The answer to *RQ 2* is first and foremost prescriptive; however, descriptive elements can be found in the reasoning of results to establish a clear link to the industrial and academic needs.

1.3 RESEARCH SCOPE AND DELIMITATIONS

The research work presented in this thesis focuses on the development of complex products and takes a strong direction towards interdisciplinary means. With an interdisciplinary framework in mind, the thesis specifically regards the interchange of information among and across disciplines and systems. This scope might be regarded as quite broad in terms of literature coverage and depth. Interdisciplinary design requires intensive processes and activities supported by methods and tools among which more than one discipline or system are interacting.

The thesis explicates how things are designed but some attention is also paid to how things ought to be designed. In designing, people are obviously involved, which is why a design support is aimed at specific users. Thus, interdisciplinary design also suggests the collaboration between multiple engineering users. Apart from this, other organizational aspects do not receive much attention in this thesis. The design support, or methodology, presented in this thesis comes along with a software system supporting the practical use of the methodology. This research is delimited from the development of this software system, it is rather a means towards validating the methodology.

1.4 OUTLINE OF THE THESIS

The thesis is divided into six chapters for the purpose of providing industrial and academic practitioners with findings driven by the research questions within the scope and delimitations stated above. The content of the chapters is described below. The structure of the thesis is illustrated in Figure 4.

Chapter 1 is an introduction to the industrial need and an analysis of existing design practice and its limitations. Key areas for developing better processes, methods and tools have been identified to improve the product development process. Based on this, the research is justified by clarifying the scientific mission, main phenomenon studied and an hypothesis. Two main research questions have been posed to propel the research work forwards.

Chapter 2 postulates an assortment of theories relevant to the work presented in this thesis. You will be provided with a summary of the most prominent research findings within platform-based development, suitable application areas, as well as other research fields worth considering. By explicating this, the research gap, addressed in this thesis, will be further addressed.

Chapter 3 provides you with the applied research approach, framework and methodology. How the research was carried out and how the results are validated are transparent. The choices of methods are justified, and the use thereof, to meet the scientific mission addressed in this thesis is explained.

Chapter 4 essentially provides you with the findings with regard to the scientific mission stated in this thesis. The studies conducted are clarified as far as content, execution, and participant roles are concerned.

Chapter 5 is dedicated to provide you with a discussion regarding the stated industrial challenges, scientific mission, research approach and findings. It mainly concerns the quality of the findings, the consistency and validity of the research.

Chapter 6 provides you with a brief summary of the findings and their reliability and validity. The scientific value and industrial relevance of this research are ultimately proposed. It also points to the future of platform-based development and provides you with new impetus for further research opportunities, as well as future directions within the scope of the scientific mission.

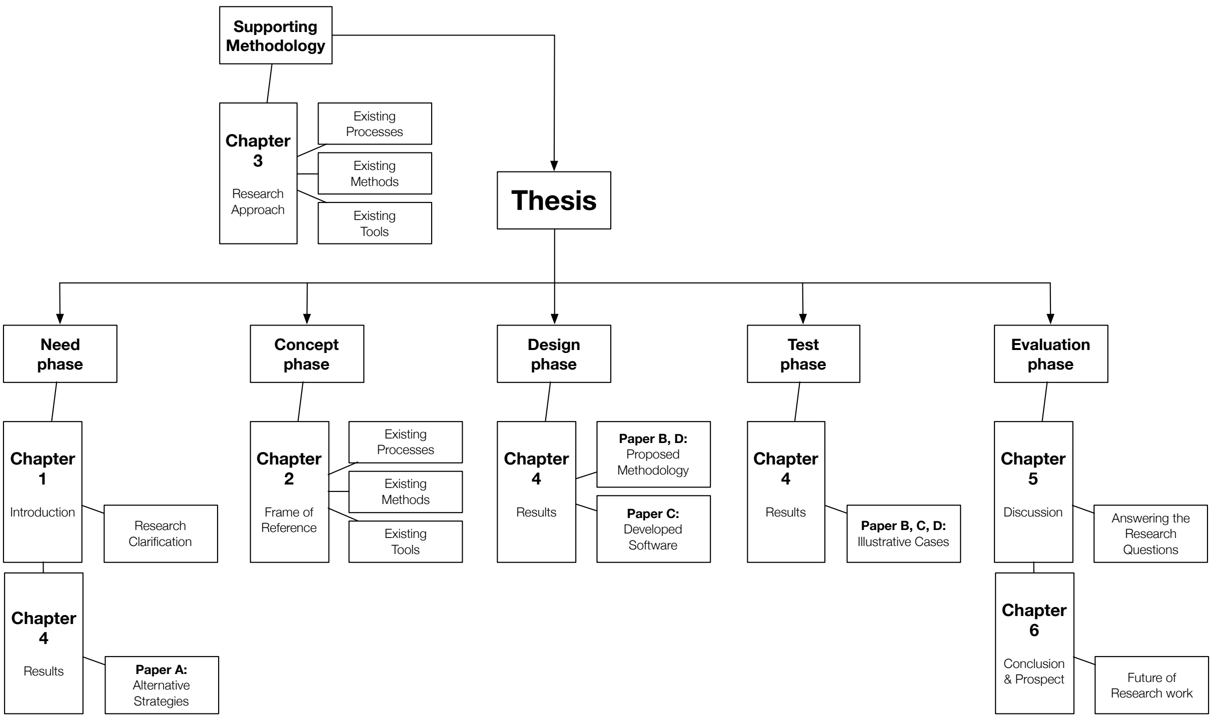

Figure 4. The structure of this thesis

2

"A wise man can learn more from a foolish question than a fool can learn from a wise answer."

– Bruce Lee

FRAME OF REFERENCE

This chapter postulates an assortment of theories relevant to the work presented in this thesis. You will be provided with a summary of the most prominent research findings within platform-based development, suitable application areas, as well as other research fields worth considering. By explicating this, the research gap, addressed in this thesis, will be further addressed.

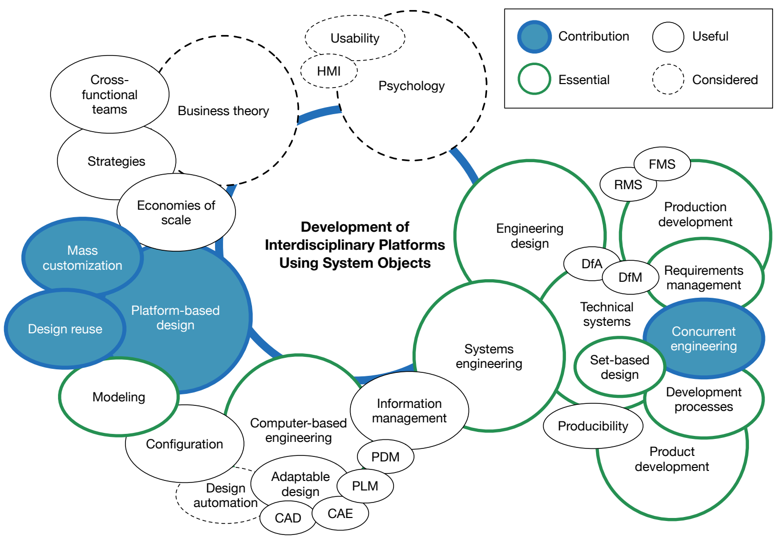

Figure 5. Areas of Relevance and Contribution (ARC) diagram (applied from Blessing and Chakrabarti (2009))

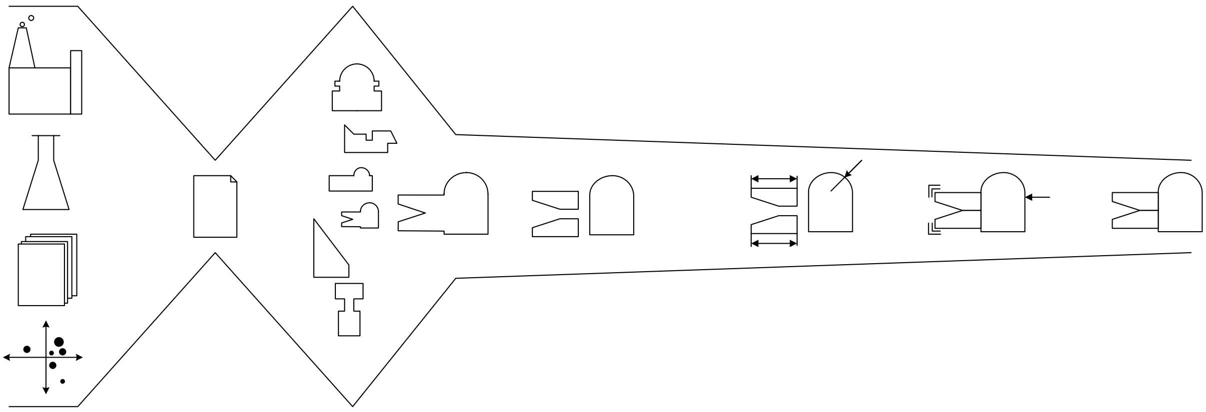

Figure 6. Generic product development process (Ulrich and Eppinger, 2012)

The main contribution of this thesis includes such fields as, 1) *platform-based development*, 2) *mass customization*, 3) *design reuse*, and 4) *concurrent engineering*. It largely contributes to *engineering design* and *systems engineering* and overlaps the two fields. A visualization of the areas of relevance and contribution is provided in Figure 5. The contribution is further discussed in Chapter 5.

2.1 INTRODUCTION TO PRODUCT DEVELOPMENT

Product development is a highly complex and iterative process involving a multitude of stakeholders and requirements. Ulrich and Eppinger (2012) write, *"product development is a set of activities beginning with the perception of a market opportunity and ending in the production, sale, and delivery of a product."* Three main iterative processes based on the numerous underlying processes within product development used to design and produce products are i) developing and managing requirements for products, manufacturing equipment and tools, ii) understanding the constituent systems of a design, iii) analyzing trade-offs of performance and capabilities among and across these systems (Bhise, 2013). The main attention of this thesis is paid to the development of complex products, and specifically the increasing complexity of products. A product can be considered to be complex if it consists of many systems and components in which a wealth of interrelationships, or interfaces, and requirements need to be taken into account (Bhise, 2013).

The generic product development process illustrated in Figure 6 is solely one approach to product development. Following the scope of this thesis, approaches to interdisciplinary product development, or rather platform development in the paradigm of mass customization, are emphasized.

2.2 APPROACHES TO MASS CUSTOMIZATION

Mass customization is not likely to work for every company (Ferguson et al., 2013). Depending on the strategy and products they develop, the need for customized goods differs. Some products are less suitable to mass customization, but instead more suitable for standardization and mass production. Different examples of products and their relative level of complexity are illustrated in Figure 7. The strategy and products of a company also affect its suitable production setting. In the mass customization paradigm, companies no longer need to forecast products in a make-to-stock setting with the risk and investments inherent in maintaining stocks, such as over-stocking, under-stocking, and writing off stock due to obsolescence (Kratochvíl and Carson, 2005). In this thesis, mass customization is regarded

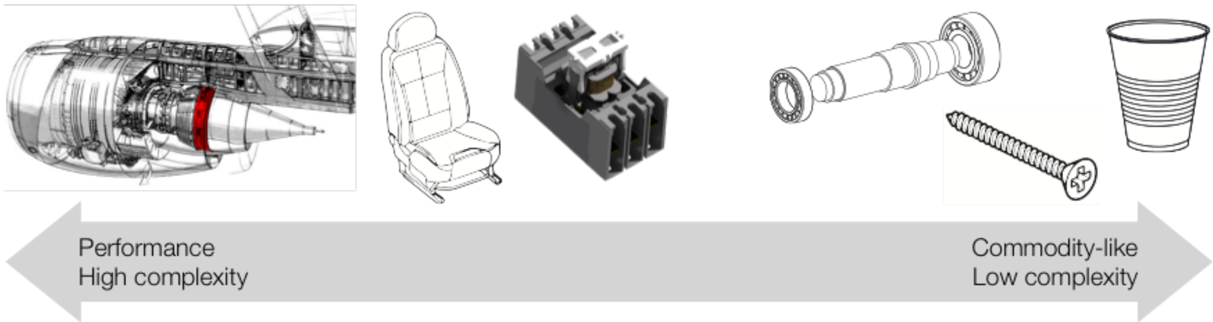

Figure 7. Examples of products and their relative level of complexity

in the same way as defined by (Ferguson et al., 2013): a product is not manufactured until the customer places an order. This definition includes manufacturing settings such as make-to-order (Hvam et al., 2008), configure-to-order, and engineer-to-order (Brière-Côté et al., 2010). Some of the different views of production are illustrated in Figure 8. To achieve mass customization, companies need to implement strategies, practices, and technologies that allow flexibility, modularity, configurability and reconfigurability during the entire lifecycle of a product. *"This eliminates the definition of a predefined product family, but allows for platform-based customization."* (Ferguson et al., 2013)

There are several approaches to increased flexibility in manufacturing needed to meet mass customization. A dedicated manufacturing system is typically optimized for a single product and is most suitable to mass production as opposed to mass customization. The concept of flexible manufacturing systems (FMS) was developed in the 1960s to meet changes in work orders, production schedules, part programs, and tooling. Two well-known approaches that serve mass customization are 1) reconfigurable manufacturing systems (RMS) (Koren et al., 1999), where the flexibility is confined within the product family (Tseng and Hu, 2014), and 2) additive manufacturing (AM) (Reeves et al., 2011), which is fully flexible to meet customers on-demand (Tseng and Hu, 2014, Fogliatto et al., 2012). Even though some recognize assemble-to-order and build-to-order are within the scope of mass customization (Ferguson et al., 2013), they are disregarded in this thesis due to the assumption that the product architecture, in such settings, would be pre-defined. In this thesis, early support for mass customization is the main target.

Table 1. Characteristics of dedicated, flexible, reconfigurable manufacturing systems (adapted from Koren and Shpitalni (2010))

	Dedicated	***FMS***	***RMS***
System structure	Fixed	Changeable	Changeable
Machine structure	Fixed	Fixed	Changeable
System focus	Part	Machine	Part family (around a part family)
Scalability	No	Yes	Yes
Flexibility	No	General	Customized
Simultaneous operation tools	Yes	No	Possible
Productivity	Very high	Low	High
Cost per part	Low (for full utilization)	Reasonable (several parts simultaneously)	Medium (parts at variable demand)

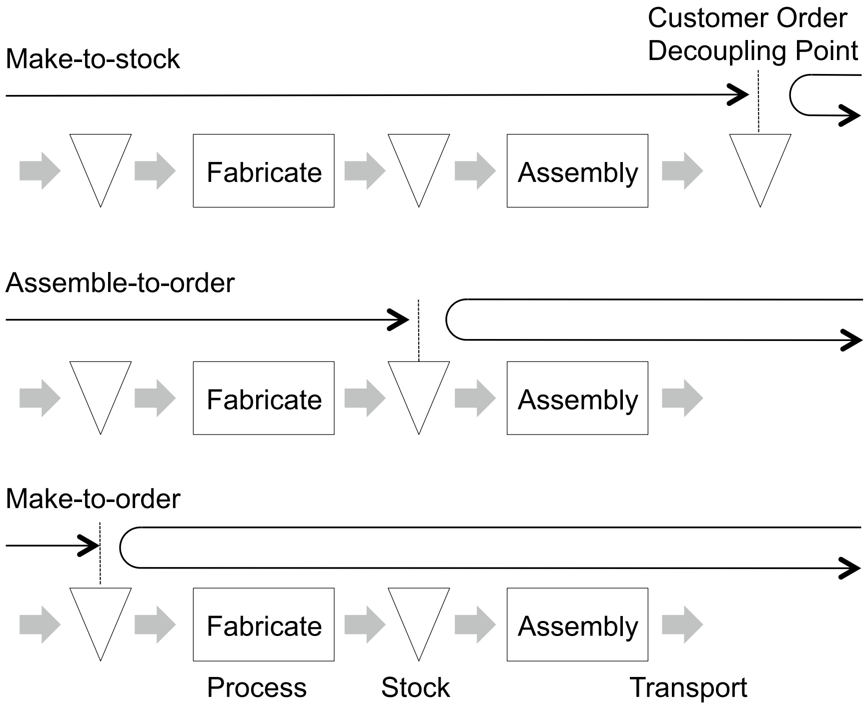

Figure 8. Different views on production and customer order decoupling points respectively (as drawn by Michaelis (2013), adapted from Hvam et al. (2008))

The high number of Information Technology (IT) systems, used by engineers from different disciplines, can also be a barrier to the design of mass customized goods. The IT systems have to be structured with the appropriate flexibility, and the IT architectures need to be capable of the effective and efficient interchange of information between disciplines (Ferguson et al., 2013).

This thesis is using the tactic of *platform-based development* as an enabler for *mass customization* and *design reuse*. To further manage increasing product complexity, the improved interchange of information among and across disciplines and systems is needed. To study this phenomenon various approaches to interdisciplinary development, as well as models, methods and tools used to support the interchange of information in the early phases of development, have been reviewed.

2.3 APPROACHES TO INTERDISCIPLINARY DEVELOPMENT

Developing platforms includes planning for a wide range of customer needs, and the product architecture that is decided upon shall deliver differentiated producible variants. This process is inherently difficult to manage, because of the coordination required between disciplines, such as marketing, design, and manufacturing. When designing a new platform, the disciplines that focus on the customer features of a product are often in conflict with groups that care about parts and manufacturing processes (Robertson and Ulrich, 1998). Different approaches have been developed to accommodate the difficulties in managing interdisciplinary development. A well-reputed approach is integrated product development, by Andreasen and Hein (1987), which considers the development process as a means of intertwining marketing, design, and manufacturing; see Figure 9.

There are numerous other approaches regarding interdisciplinary development (Loureiro, 1999). Some of them, regarded as fundamental to this research, are referenced below.

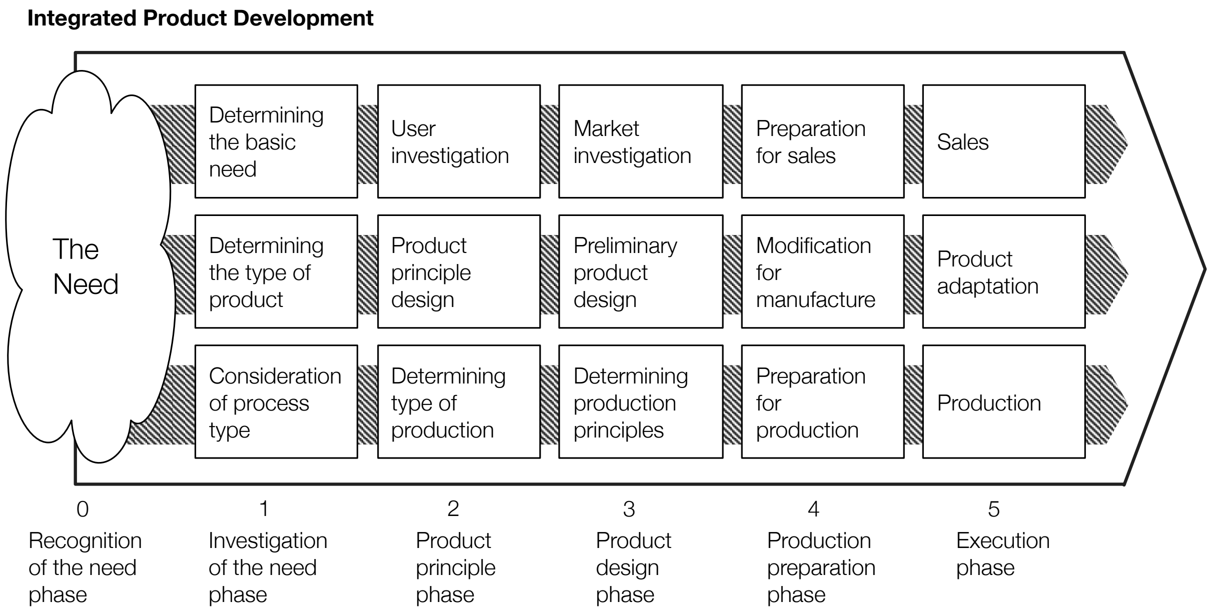

Figure 9. Integrated product development (redrawn from Andreasen and Hein (1987))

2.3.1 Concurrent (Simultaneous) Engineering

Concurrent Engineering (CE) is a systematic approach to the interdisciplinary development of products, manufacturing systems and supporting processes (Loureiro, 1999). CE is mainly seen as an organizational approach; however the concurrency can also be reflected in how designs are modeled (Levandowski, 2014). CE offers potentially faster development of producible quality products by, among other things, emphasizing cross-functional integration, improved communication with customers and suppliers early in development, and the concurrent design of manufacturing systems and processes (Swink et al., 1996). CE reflects all phases of the product lifecycle, from concept through disposal, including the requirements from both customers and users, and between functions within design, such as structural functions or aerodynamic functions, and across disciplines such as from manufacturing to product design (Loureiro, 1999). CE primarily emphasizes the early interchange of information that affect downstream activities, when the information is still uncertain. Four different modes for the interchange of information are contrasted by (Wheelwright and Clark, 1992), and is illustrated in Figure 10. To achieve CE in design, some research advocates modularization of the product (Gershenson et al., 2004, Erixon, 1998, Gu and Sosale, 1999).

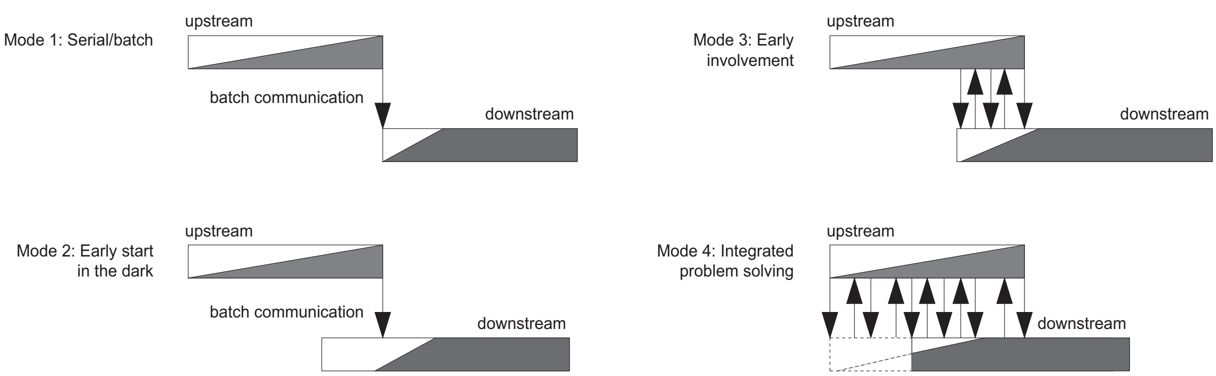

Figure 10. Concurrent Engineering and the interchange of information
(as drawn in Levandowski (2014), adapted from Wheelwright and Clark (1992))

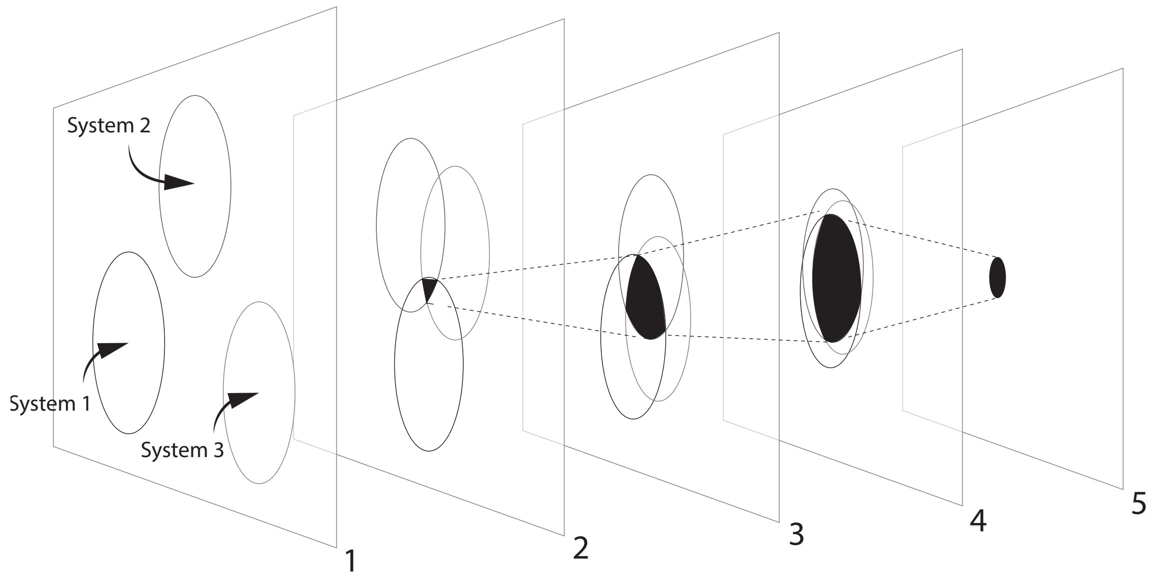

Figure 11. Set-based Concurrent Engineering
(slightly modified from Levandowski (2014), adapted from Bernstein (1998))

Set-based Concurrent Engineering

Set-based design, often combined with Concurrent Engineering, has proven an efficient strategy for design (Sobek et al., 1999). In the early phases of development, contrary to point-based design, set-based design applies extensive design space exploration, rather than selecting an arbitrary solution. Set-based design builds on three principles, 1) *mapping the design space*, 2) *integrating by intersection*, and 3) *establishing feasibility before commitment*. These principles advocate a sound depiction of a design and how it may vary due to changing functionality and requirements. Sobek et al. (1999) summarize *Set-Based Concurrent Engineering* (SBCE) as "*reasoning, developing and communicating about sets of solutions in parallel and relatively independently.*" SBCE can be applied to explore a broad range of alternative design solutions that are systematically narrowed down by eliminating unfeasible solutions (Malak et al., 2009). An example of three parallel working disciplines, or functional systems, converging to a confined solution is provided by Bernstein (1998). The five steps below are also illustrated in Figure 11:

1. The disciplines, or systems, are expanding the number of options individually
2. Integrating a small region of overlapping design solutions
3. Expanding the overlapping region by increasing the number of solutions which will satisfy the inherent requirements of the three disciplines
4. Eliminating options to further converge an overlapping region
5. Narrowing the solution space until only one or a few feasible solutions remain

2.3.2 Systems Engineering

Systems engineering (SE) is an interdisciplinary and collaborative approach that affects engineers from various disciplines, such as mechanical, electrical, aerospace, and computer science (Bhise, 2013). It focuses, among other things, on defining customer needs, mapping functionality, documenting requirements, synthesizing design, and validating systems (Blanchard and Fabrycky, 1990). The process of SE reflects the transformation of customer needs into designs with the performance, size and configuration of meeting these needs. SE advocates a top-down approach used to analyze a product as a whole and decompose it into various levels, such as systems, subsystems, and components. In this process, system architects play a key role. The system architect is lead in the development of the system architecture, including definition of requirements and interfaces, evaluating trade-offs between conflicting requirements, and harmonizing technical risk between systems (Bhise, 2013). SE supports the whole product lifecycle through planning, designing, training, testing, manufacturing, and disposal. Product Lifecycle Management (PLM), used to manage the entire product lifecycle, is further reviewed below.

Product Lifecycle Management

Product Lifecycle Management (PLM) is a widely recognized management approach to efficient product development (Stark, 2015), which is part of SE. All phases of development have to be studied and coordinated, from customer needs to product disposal (Stark, 2015). PLM can be seen as an integrator of tools and technologies to facilitate the interchange of requirements throughout a product lifecycle (Terzi et al., 2010). PLM, as a software system, is used to cope with ever-changing requirements, however PLM also encompass the organization, collection of engineering processes, methods, and tools, as well as the product and its related data and information (Abramovici, 2007). Software can be used to manage such product data, often referred to as Product Data Management (PDM) software, and may very well be one of the components of a PLM architecture (Abramovici, 2002), but do not constitute the entire PLM strategy. Svensson et al. (1999) agree that there are different views of PLM, other than software systems, and emphasize such aspects as processes, information, systems and roles, the frameworks of which can be used to model PLM architectures.

2.3.3 Manufacturing Requirements and Capabilities in Design

The incorporation of manufacturing requirements and capabilities early in development to shorten lead-time has been studied for decades (Boothroyd, 1994). There are several approaches to include manufacturing in design, such as Design for Manufacturing (Manufacture, or Manufacturability) (DfM), and Design for Assembly (DfA) (Boothroyd, 1994). These approaches provide design engineers with guidelines on how to design products to be producible. Vallhagen et al. (2013) define producibility as *"the capability to produce the product in a robust and efficient way to meet the design specifications for functions and reliability of the product"*. Producibility advocates a strong link to product functions, characteristics and performance (Vallhagen et al., 2013). There are numerous variables that can be used to characterize producibility (Hadley and McCarthy, 2011), for example, geometrical robustness (Wärmefjord et al., 2014), accessibility in the assembly process (Hadley and McCarthy, 2011) and process quality (Vallhagen et al., 2013), which all relate to the producibility of manufactured products. The integration of manufacturing in product platforms, or vice versa products in manufacturing systems platforms, has been explored and studied by Gedell (2011) and Michaelis (2013) who emphasize the co-development paradigm. They regard both products and manufacturing systems as technical systems, and model them using function-means (F-M) modeling to increase reuse among the two technical systems. F-M modeling is reviewed in Section 2.4.3.

2.3.4 Cross-functional Teams

While organizational aspects are not the main focus of this thesis, some of the aspects are considered too important to be overlooked.

Many manufacturing companies have turned to cross-functional team solutions to deal with growing product complexity and to manage requirements. Zimdars (2003) state that cross-functional teams are most effective in fast-changing markets because less time can be spent on gathering information from the various functional areas of a company, such as marketing, design, and manufacturing (Henke et al., 1993). Even representatives from suppliers can be part of a cross-functional team (Parker, 2003). An strong advantage of integrating suppliers into cross-functional teams is to communicate the variety of operating system interfaces and mutual production processes (Henke et al., 1993). Due to the heavy demand for a wide variety of information from various disciplines in the early phases of development, Wheelwright and Clark (1992) emphasize the importance of using cross-functional teams when developing platforms.

2.4 MODELS, METHODS AND TOOLS

Several models, methods and tools used in product development have been studied and improved to serve platform development (Michaelis, 2013, Levandowski, 2014). A focal model, to support design, in product development is the *product architecture*, with its composition of physical components (Ulrich and Eppinger, 2012, Ulrich, 1995). In the context of architecture, some practitioners define a product as a system and a component as a sub-system (Stevens, 1998, Hood et al., 2007). In this thesis, a *system* is equivalent to this definition; however, a system also encompasses more abstract entities of an architecture, such as an arbitrary *cooling system*. How the entities of the product architecture, or system architecture, are defined is crucial for their flexibility or proneness to be modified according to changing requirements.

2.4.1 The Architecture of a Product

Product architecture is defined as *"the scheme by which the function of a product is allocated to physical components"* (Ulrich, 1995). Fixson (2007) writes that this scheme includes the arrangement of functional elements, mapping from functional elements to physical components, and the specification of interfaces. Wheelwright and Clark (1992) emphasize that manufacturing requirements and capabilities need to be combined with design requirements to lay down the architecture of a product. Contrasting the definition of a pure product architecture of physical components, a definition of system architecture is provided by Crawley et al. (2004): *"system architecture is an abstract description of the entities of a system and the relationships between those entities."* The product architecture is intricate to manage, when satisfying customers with various needs and demands that change over time. As product complexity is increasing, some companies have adopted a platform strategy to serve several products or brands and prolong the life of products and make it upgradeable in the future. A platform strategy affects the structure of the product architecture, and can be introduced to increase the reuse of components through the commonality among products (Robertson and Ulrich, 1998). Therefore, the product architecture also affects the essential economies of a product (Reinertsen, 1997). Product architectures are often categorized into one of two archetypes: integral or modular (Fixson, 2007). Ulrich and Eppinger (2012) further note that the most important characteristic of the product architecture is its modularity. Differences between an integral and a modular architecture are described by Ulrich (1995). These differences are summarized in Figure 12. Modularization is a way to achieve commonality and economies of scale. Put differently, the main objective of modularization is to find a maximum of distinct product variants, with minimal internal product variation. Baldwin and Clark (1997) write that modularization is about *"building a complex product or process from smaller subsystems that can be designed independently yet function together as a whole."*

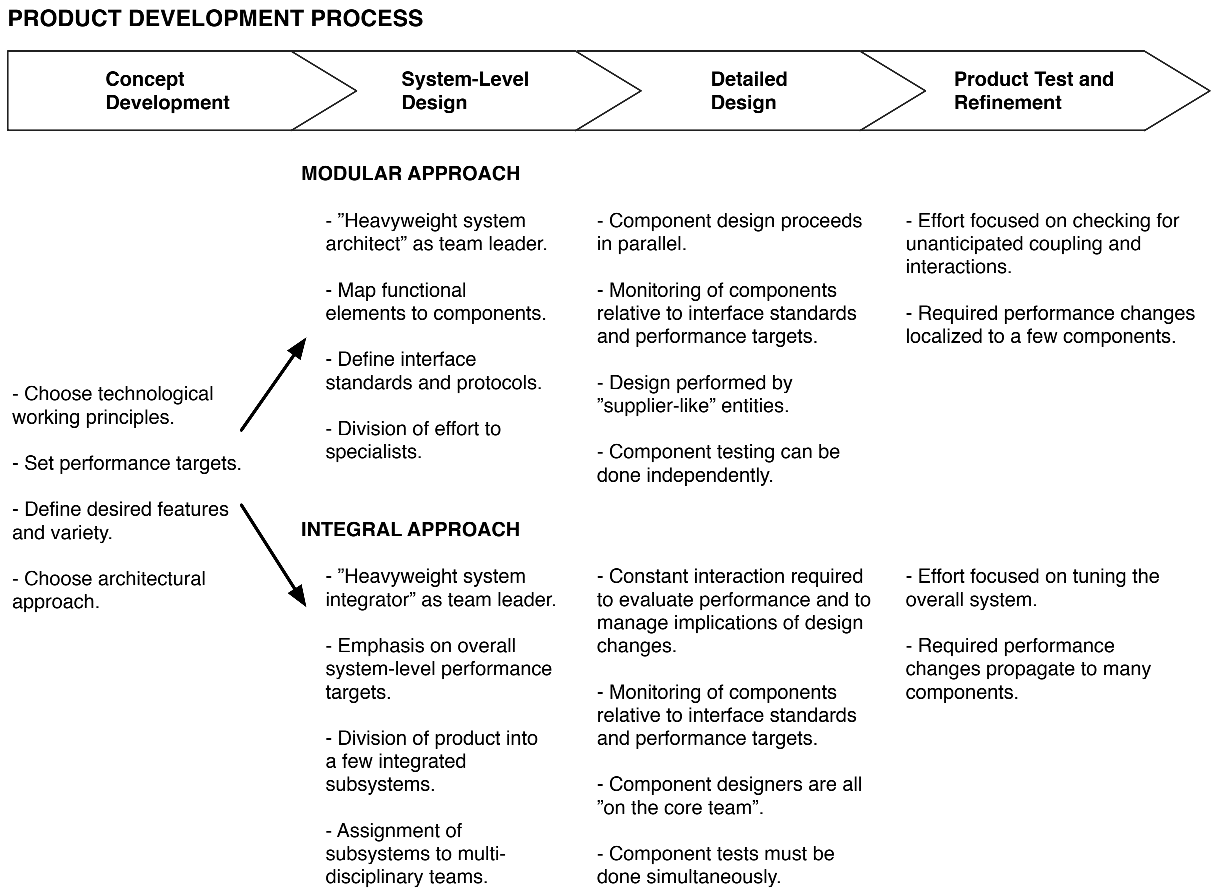

Figure 12. Differences between a modular and an integral architectural approach (redrawn from Ulrich (1995))

A focal characteristic of modularization is standardized interfaces (Crawley et al., 2004). Due to standardized interfaces, modules can be studied separately from the rest of the architecture. Therefore, the modular architecture is more flexible than the integral architecture. However, to meet future requirements, customer needs and increasing product complexity, there is a need for increased flexibility of the product architecture. Crawley et al. (2004) state, *"the process of architecting requires a model of future usage of the system, including an understanding of uncertainties in the environment, competition, regulations, and future user needs."* They also emphasize that increased flexibility may include such activities as over-designing and adding extra interfaces, which may be time-consuming. The transition of moving from an integral to a modular architecture includes substantial changes in products and development processes (Persson and Åhlström, 2006). In any case, increased flexibility of the architecture may pave the way for increased distinctiveness as well as commonality (Muffatto and Roveda, 2000). This hypothesis is illustrated in Figure 13. Gu et al. (2004) speak about increased flexibility of designs to accomplish functional independence and advocate adaptable modules and adaptable interfaces as means of achieving those objectives (Gu et al., 2004).

2.4.2 Systems Theory

A system is an example of an entity where the whole is more than the sum of its parts. More specifically, the behavior of a system depends on its sub-systems and their interactions or emergent properties (Checkland, 1981) cannot be attributed to any specific

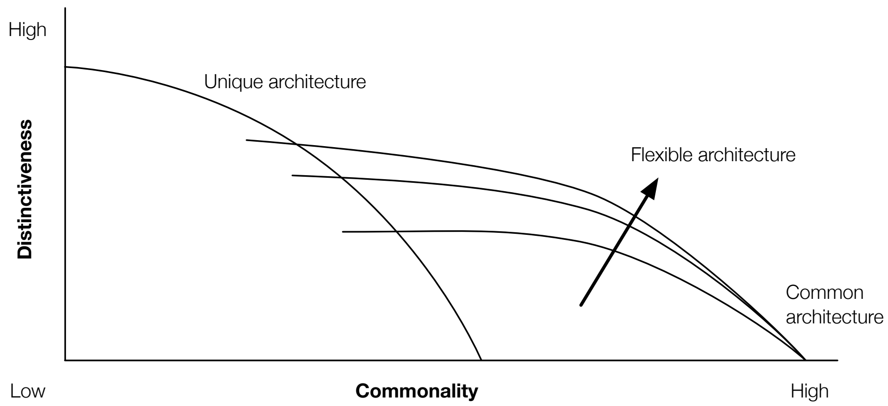

Figure 13. Different architectures and the effect on product distinctiveness and commonality (redrawn from Muffatto and Roveda (2000), adapted from Sheriff (1998))

part of the system. Rather, they emerge only when the system as a whole is considered. Hitchins (2003) expresses, "*the properties, capabilities, and behaviors of a system derive from its parts, from interactions between those parts, and from interactions with other systems.*"

A key factor in successfully managing complex systems is to deal with a limited number of parts of a system at a time, instead of the complete system at once. Decomposition of systems is a way of limiting the task. However, it has its drawbacks. Hitchins (2003) describes how decomposition will make the parts lose their interactive abilities, thereby losing their context. Elaboration and encapsulation, on the other hand, look at the parts in isolation and in a context, thus maintaining their interactions. Elaboration refers to moving down in such a hierarchy, focusing on details. Encapsulation can be compared to moving up in the hierarchy while placing containers around sets of entities. The container does not cut the interactions, as decomposition does. Instead, they remain intact. Encapsulation conceals unnecessary details and reduces the perceived complexity of the system.

In 1988, Hubka and Eder (1988) presented the *Theory of Technical Systems* (TTS). At its core, the theory aims at providing a comprehensive theory of classifying and categorizing the information of technical systems into ordered sets of statements. The TTS concept of transformation and the different domains used to represent different aspects of a technical system are especially interesting. Technical systems are the principal means by which a transformation is achieved. The technical system exists only to realize a transformation from input to output. A combination of input and the internal states of the system will define its output, as well as the internal state that it will adopt. Hubka and Eder (1988) presents five abstract models of technical systems: *purpose*, *process structure*, *function structure*, *organ structure* and *component structure*. Andreasen (1991) proposed the *Theory of Domains* (ToD), based on Hubka and Eder's Theory of Technical Systems. The model consists of four domains: *process domain*, *function domain*, *organ domain*, and *component domain*. Following ToD, Mortensen (1999) developed the chromosome product model as a generic structure of these domains.

2.4.3 A System Object Model

An approach to describe more abstract platform entities, opposing other approaches of design reuse through physical parts, was proposed by Claesson (2006). The *configurable component (CC)* approach is based on systems theory principles (Hitchins, 2003) and design theory (Hubka and Eder, 1988, Andreasen, 1991). The CC contains a system description of an entire system family. It contains information about both the system solution itself, the means

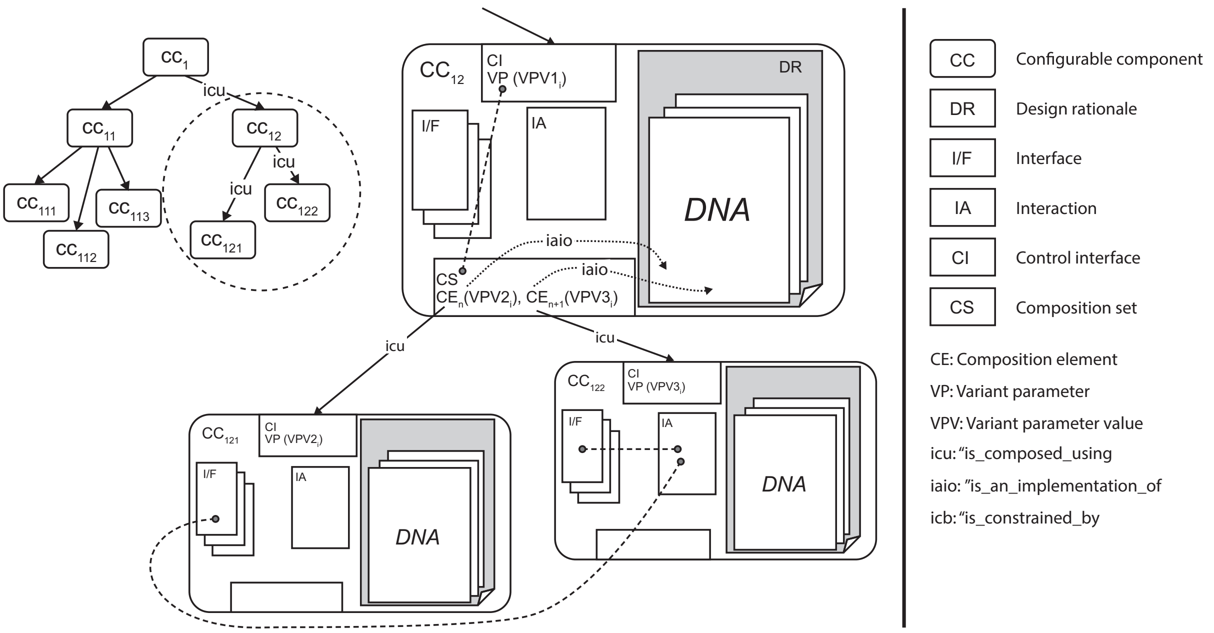

Figure 14. A system object, *configurable component*, including the *DNA* and its objects and relations (slightly modified from Michaelis (2013), adapted from Claesson (2006))

of composing system variants, as well as its underlying requirements and motivations, i.e. its *design rationale*, or the *DNA* of the system. Besides the *DNA* of the system, a CC object contains a *composition set* (CS) and a *control interface* (CI). The CS object and CI object are used to communicate hierarchically across CC objects. By interchanging *variant parameters* (VPs) hierarchically from the CS of one CC to the CI of another CC. The CC objects and their links are incorporated in governing CC objects as *composition elements* (CEs). Design rules, needed to model a CC object are implemented as formulas. CC objects also contain certain configurable *interfaces* (IF) to communicate laterally. The lateral interchange across CC objects is incorporated into an *interaction* (IA). The system object model, *configurable component*, is illustrated in Figure 14.

The DNA of the System Object Model

There have been several approaches to capture *design intent*, *design rationale* (DR) and *design history*. Andersson (2003) presents how these concepts are interrelated. Design intent forms the underlying reason why a certain object exists. Design rationale includes the justifications for why it exists; alternatives, trade-offs, and argumentation (Lee, 1997). Design history includes the recorded process of the design, describing how the object came into being. One method of including such information is function-means (F-M) modeling. F-M modeling is a systematic way of finding *design solutions* (DSs) that fulfill *functional requirements* (FRs). An FR is defined as what a product, or an element of a product, actively or passively shall contribute to a certain purpose by creating internal or external effects. The FRs motivate the downright existence of a specific solution. The means, organs, or DSs, are tangible objects, for example components or features, or non-physical, for example service or software, entities that can possibly fulfill a specific FR. An F-M model is an hierarchical model of a particular system, which is decomposed into subordinate sub-systems. The F-M tree is also a representation of *Hubka's law*, which states that: *"the primary functions of a machine system are supported by a hierarchy of subordinate functions, which are determined by the chosen means (organs)"*. The model was originally developed by Tjalve (1976) and Andreasen (1980), and has evolved over time. In an analogy to the F-M tree, axiomatic design

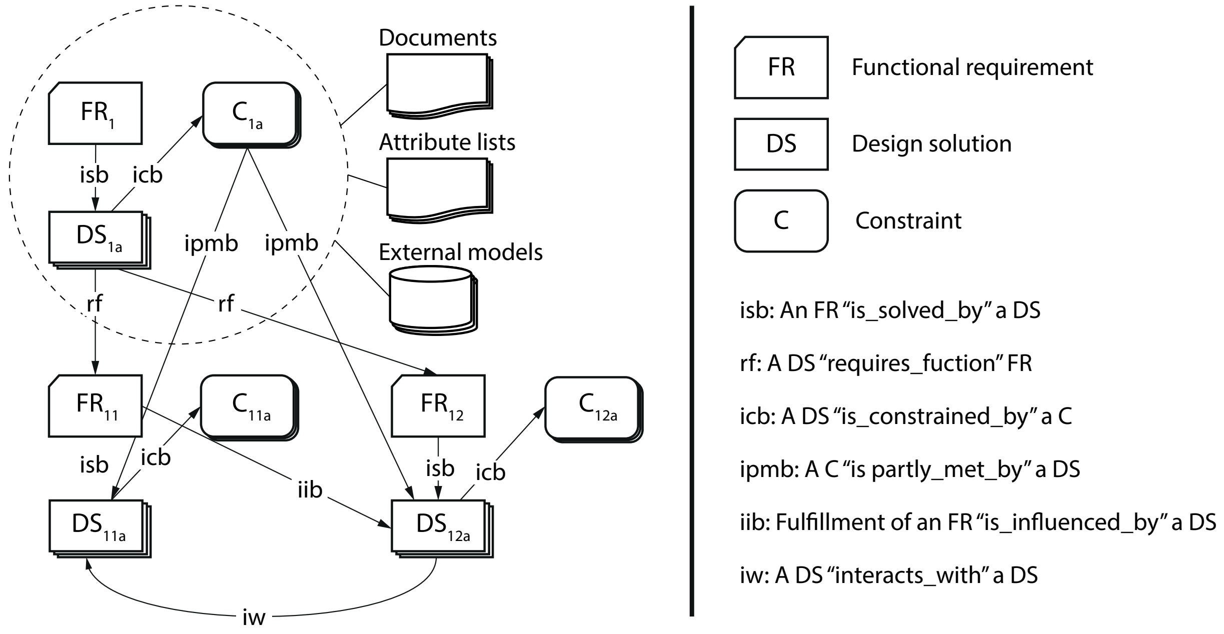

Figure 15. Enhanced function-means tree
(as drawn in Levandowski (2014), adapted from Johannesson and Claesson (2005))

Suh (1990) describes the zigzagging between *functional requirements* (FRs) and *design parameters* (DPs). The zigzagging points out the fact that a requirement cannot be decomposed into other requirements without identifying intermediate solutions (i.e. DPs). Schachinger and Johannesson (2000) enhanced the function-means method by adding the abilities of describing additional types of relationships and of separating functional requirements from non-functional requirements, constraints (C); see Figure 15. The purpose of the Cs is to delimit the allowed design space for DSs.

The *DNA* of the system object model is based on *enhanced function-means* (EF-M) models. It is used as a formalized description of a technical system's specification. The EF-M model is used as an underlying hierarchical object model, which can be decomposed top-down. By starting with an overall functional requirement, a number of different design solutions that provide this functionality can be modeled. Also, a number of constraints, restraining these design solutions, can be modeled. There are some underlying rules to this type of modeling. The cardinality of an FR and a DS is always 1↔1. This relation is denoted as *is solved by* (isb). And the cardinality between a DS and its constraining Cs is 1↔n. This relation is denoted as *is constrained by* (icb).

A DS can be decomposed into lower hierarchical levels, following *Hubka's law*. In this process, two or more sub-functional requirements are formulated. This relation between DS and FR is denoted *requires function* (rf). The constraints at this level can be partly divided through a number of DSs, for example if a weight constraint is evenly distributed between them. This relation is denoted *is partly met by* (ipmb).

Apart from the pure hierarchical relations, a semi-lateral and a lateral dependency can be modeled. The first one, the semi-lateral, is a relation between a DS and the FR of another DS. It indicates that the main DS is influenced by another function. This relation is denoted *is influenced by* (iib). The second one, the lateral, is an interaction between two DSs at the same hierarchical level. This relation is denoted *interacts with* (iw). Both relations can be used to perform matrix-based analyses of the structure model, such as analyses and evaluations using Design Structure Matrices (DSMs) and axiomatic couplings. Also additional information, such as attributes, external documents and other external models, can be linked to relations and objects in the model.

2.4.4 Accommodated Design Flexibility of System Objects

The definition of the system objects accommodates design flexibility. This is governed through *modular* and *scalable bandwidth*.

Modularity and Scalability

The CC object is a model of an entire system family of variants, consisting of FR, DS and C objects. These objects accommodate design flexibility in terms of bandwidth, which means that functional and constraining properties, as well as the design solution characteristics, can vary within predefined ranges. It means that a design solution is parameterized so that a set of variant defining parameters (VPs), within the system bandwidth, can be configured in accordance with stated requirements.

Most design problems have both modular and scalable solutions. Typically, module-based platforms consist of a set of interchangeable modules. By changing a module for another, different properties are achieved (Gonzalez-Zugasti and Otto, 2000). These concepts can be managed through bandwidth, *modular* and *scalable*. Modular bandwidth can be accomplished by accommodating multiple design alternatives each solving one function (Wahl and Johannesson, 2010). Scalable bandwidth can be accomplished by building in value ranges in parameters (Berglund and Claesson, 2005). Both Michaelis et al. (2013) and Levandowski (2014) continued to elaborate on these bandwidths. Modular bandwidth can be illustrated by, for example, the number of bearings used in a car. Bearings are, above all, used for the wheel axis, as well as in other applications, such as in the gearbox. The main function of a bearing is to convey mechanical loads, both axially and radially. Thus, a bearing system, and the module character thereof can be reused for the wheel axis and the gearbox. Even though the design and size of it may change, depending on the mechanical loads it will convey, the fundamentals of a bearing system can be reused. In scalable platforms, the design can be stretched and shrunk to fit specific customer requirements (Simpson, 2004). Thus, for the bearing example, also scalability can be achieved. As the size of a bearing is driven by the need to convey mechanical loads, parameters will need to accommodate the full range of sizes for different loads, used in for example a car. The modular and scalable bandwidths for bearings can be seen in Figure 16.

Adaptable Interfaces

Bhise (2013) writes that an interface is the connection where two entities are linked together to serve certain functions. For entities to be compatible, the interfaces must carry the same values as their shared parameters (Bhise, 2013). The interchange between interfaces, or the interaction between them, can involve four different types: 1) *physical space*, 2) *energy*, 3) *material*, and/or 4) *data or information* (Pimmler and Eppinger, 1994). Interfaces must be identified in the early development phases to be properly analyzed and managed in the following, detailed, design activities. Engineers must know how and what an interface shall interchange to make the entities work together and fulfill functions intended (Bhise, 2013).

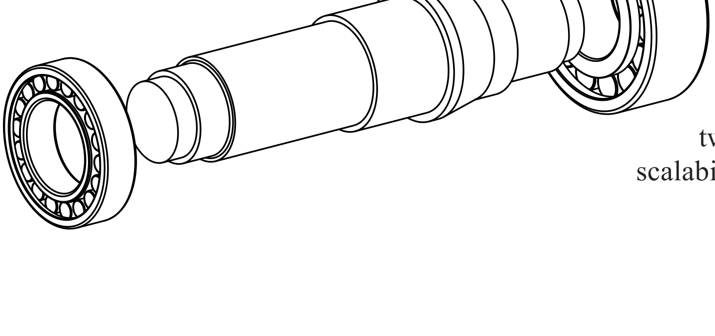

Figure 16. Illustrating modularity: two instances of a bearing system, and scalability: the different size of the bearings

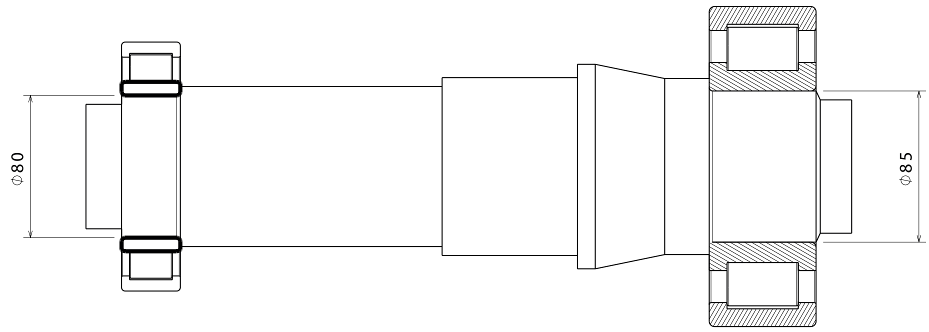

Figure 17. Two different sizes of bearings, and their diameter interface to an axis

Sosa et al. (2007) studied the link between component modularity and component redesign. They found that the relationship depends on the interfaces connecting these components. Due to the various types of design dependencies, a design modification in one component is likely to propagate to other components. Based on the definition of the system objects provided above, this redesign can be managed through adaptable interfaces, i.e. an interface, with its requirements and constraints, can be efficiently interchanged with another interface. Trying to concretize, let say that the inner hole of a bearing, typically mounted on an axis, can convey loads that account for a certain axis diameter of 80 millimeters. Then, if the loads were to increase, the axis need to be reinforced, and the interface might change to a larger diameter. Thus, the bearing has to be modified to meet these new requirements; see Figure 17. Although this is a trivial system, which is easy to manage in as a closed system, any increased complexity, for example a car that holds numerous interfaces between multiple systems, the interfaces become intricate to manage without the seamless interchange of information between the interacting systems. Sosa et al. (2007) suggest further research into software tools that facilitate the documentation of design dependencies and interfaces in complex products.

2.4.5 Generating Design Alternatives

In complex products, the number of systems that interact are numerous. Interchangeable systems, solving a specific function, can be combined into suitable concepts. To be able to explore feasible concepts of such products, the design space has to be systematically explored. Methods to generate concepts exist. The morphological matrix, or combination table (Ulrich and Eppinger, 2012), is considered an important step in the conceptual design process (Pahl and Beitz, 2013), by combining design solutions into a number of concepts (Weber and Condoor, 1998). This process becomes typically difficult to manage in complex products. However, the combinatorial explosion can be efficiently and systematically managed following the SBCE principles (Levandowski, 2014, Sobek et al., 1999): 1) *mapping the design space*, 2) *integrating by intersection*, and 3) *establishing feasibility before commitment*. Some examples of generating product concepts using an implemented morphological matrix in software exist, however most of them focus on optimization (Ölvander et al., 2009), even though some regards conceptual exploration (Strawbridge et al., 2002). However, when combining solutions, problems of ensuring physical and geometrical compatibility may arise (Pahl and Beitz, 2013). Here adaptable interfaces, discussed in Section 2.4.4, and the interchange of information among and across disciplines and systems may be key enablers.

Another problem with combining solutions, as discussed by Pahl and Beitz (2013), is finding technically and economically feasible concepts.

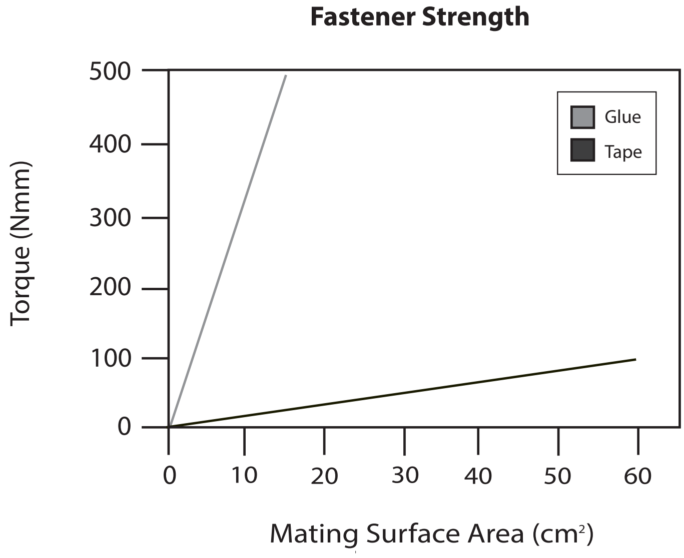

Figure 18. Trade-off curves, providing knowledge about the fastener strength of glue and tape (redrawn from Ward and Sobek II (2014))

2.4.6 Understanding and Evaluating Design Alternatives

To find technically and economically feasible concepts and to establish feasibility before commitment, different evaluation techniques can be applied. Two viable tools considered useful for the evaluation and comparison between alternatives are *design structure matrices* (DSMs) (Clarkson et al., 2004) also called *interface matrices* (Bhise, 2013), axiomatic coupling analysis, as well as *trade-off curves* (Blanchard and Fabrycky, 1990, Ward and Sobek II, 2014) and. A DSM provides a structure of interactions, or interfaces, between entities of a system architecture. With the increasing complexity of products, it becomes cumbersome to managing a large system of interrelations, which is why the value of the DSM increases with the increasing complexity of the products (Eppinger and Browning, 2012).

Trade-off Curves

In designing products, many conflicting requirements need to be incorporated. One way to explicating known trade-offs is by illustrating them through trade-off curves. A trade-off curve is a graph that shows two criteria, one on each axis. Another type of trade-offs can be illustrated by response surfaces, where more than two critera are traded. In this way, design engineers can better harmonize performance vs. capabilities. Ward and Sobek II (2014) write that trade-off curves can be used for design reuse and emphasize their use as a means of increased design understanding. Trade-off curves are a vital part of SBCE as a way to mapping the design space, evaluating design alternatives, and eliminating inferior designs based on proper knowledge (Levandowski, 2014). An example of a trade-off, providing knowledge about the fastener strength of two design alternatives, glue and tape, is illustrated in Figure 18.

2.4.7 Computer Aided Technologies

Computer Aided Technologies often referred to as CAx systems include, among other systems, Computer Aided Design (CAD), Computer Aided Engineering (CAE), Product Data Management (PDM), and Product Lifecycle Management (PLM).

CAD systems have had a huge impact on design and the activities of design engineers (Pahl and Beitz, 2013). CAD systems can be used to model and visualize three-dimensional geometries and create drawings as bases for manufacture (Ulrich and Eppinger, 2012). In the mass customization paradigm, enabling responsiveness to meet changing requirements, flexibility needs to be built into geometry models. This flexibility can be achieved through parameterization of CAD models. Parameterization is a way of governing the parameters that define design and create design alternatives. A variety of systems to model flexible designs have been developed over the years; however, most of them entail detailed design phases and deepen into multi-disciplinary design optimization, such as La Rocca and Van Tooren (2007).

CAE systems are typically complementary to CAD. CAE systems can be used to analyze CAD models through multipurpose simulations (Ulrich and Eppinger, 2012), such as the finite-element analysis of thermal stress distribution, robust design through geometry assurance, or motion planning for robots in manufacturing. La Rocca and Van Tooren (2007) state that current design activities, including CAE tools, are time-consuming and repetitive, and that too little time is spent on investigating additional product alternatives and better exploit the skills and creativity of design engineers. To make time for this, an increased automation of processes may be implemented.

Systems Integration and PLM Architecture

To make PLM work sufficiently, system integration is essential. CAD systems are in general well integrated inot the PDM system, and thus have access to product metadata (Abramovici, 2002). There are few satisfying examples of the integration of CAE systems for the interdisciplinary analysis or synthesis during early phases of development. However, one promising example provided by Zweber et al. (1998), shows the feasibility of an automated analysis of structural performance and manufacturing cost for a number of aircraft wings, and demonstrates the trade-offs of different alternatives. In most cases, however, information is manually transferred, or in some cases integrated in one direction alone (Abramovici, 2002, Burr et al., 2008).

Bhise (2013) identifies some areas where software capabilities, to meet increasingly complex systems, are needed. Some of those include support for system decomposition, graphical modeling and managing of components, systems, functions, requirements, and interfaces, for import/export capability for many CAx tools, and document generators.

3

"Plans are nothing, planning is everything."

– Dwight D. Eisenhower

RESEARCH APPROACH

This chapter provides you with the applied research approach, framework and methodology. How the research was carried out and how the results are validated are transparent. The choices of methods are justified, and the use thereof, to meet the scientific mission addressed in this thesis is explained.

As a research field, design regards the development of products, including services and processes, with respect to various stakeholder needs, regulations, ideas and technology advancements. To ensure that products function as intended, while being exposed to various users and variations in their environment, engineering knowledge is a vital part of design. Engineering knowledge has grown over the years, and various engineering disciplines with detailed knowledge and specialized skills are working together in business organization to make use of an aggregated vast body of engineering knowledge. Theoretically, engineers go through sequenced activities such as specifying requirements, developing concepts, designing them in detail, planning manufacturing processes, and designing manufacturing systems. In reality however, these activities are highly iterative. During these activities, various methods and tools are used to accomplish specific design tasks.

According to (Creswell, 2013), there are three different approaches to research: 1) *qualitative research* – an approach to explore and understand phenomena, 2) *quantitative research* – an approach to test theories by examining the relations between variables, and 3) a mix between the two – *mixed methods research*. The work presented in this thesis follows a qualitative research approach. The research is based on real industrial needs, emphasizing holistic rather than atomistic viewpoints. The scope involves the development and improvement of processes and methods to facilitate design practice and support engineers in their early development activities. Because researchers and industrial practitioners have created knowledge in collaboration, the scientific mindset is constructivistic. To entitle some validity to the work, empirical studies have been conducted. Considering the level of prescriptive contribution, logic through both deduction and induction. Deduction account for the process of testing a theory based on theoretical premises, through operationalizing variables and measuring them, whereas induction is the process of gathering information from participants, analyzing the data and making generalizations in accordance to theory and past experiences (Creswell, 2013).

3.1 RESEARCH IN DESIGN

Research in design can be regarded as a type of meta-development. It involves the development of models, methods and tools to support designers in developing products, through various design activities, more effective and efficient. Horvath (2001) define research in design as *"generating knowledge about design and for design."*

The research presented in this thesis focuses on design methodologies. Horvath (2001) continues to propose that design methodology research involves 1) the methodological systematization of design processes, 2) exploration of the mechanisms of design decision-making, and 3) the improvement of design modeling, representation, analysis, simulation, evaluation, and/or physical testing techniques.

3.2 RESEARCH FRAMEWORK

The research presented in this thesis has been strongly influenced the Design Research Methodology (DRM) framework, introduced by Blessing and Chakrabarti (2009). The framework encapsulates at least two purposes. The first purpose regards the understanding of design – descriptive: *how things are*. The second framework regards the creation or development of support for designers to make better design – prescriptive: *how things ought to be*.

The framework comprises a four-step methodology: 1) *research clarification*, 2) *descriptive study I*, 3) *prescriptive study*, and 4) *descriptive study II*. The framework also suggests *basic means* to support each step towards the main outcome. The DRM is not intended to be linear, as illustrated by the arrows linking back to previous steps (Blessing and Chakrabarti, 2009). The DRM framework is illustrated in Figure 19.

According to DRM, research can be of different types, and therefore be directed towards various outcomes and contributions, to add to the design research community.

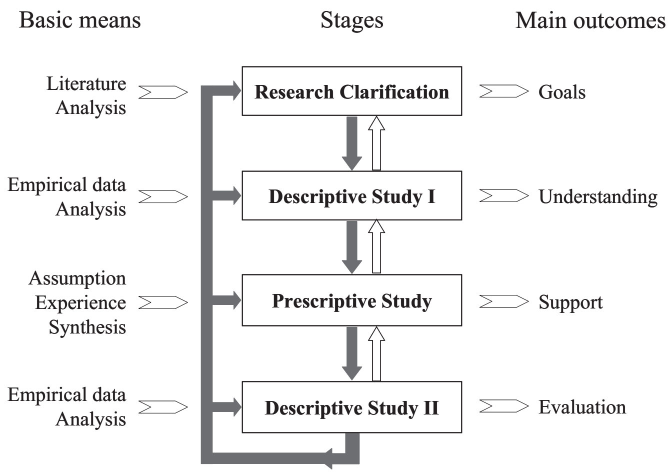

Figure 19. Design Research Methodology (DRM) framework (Blessing and Chakrabarti, 2009)

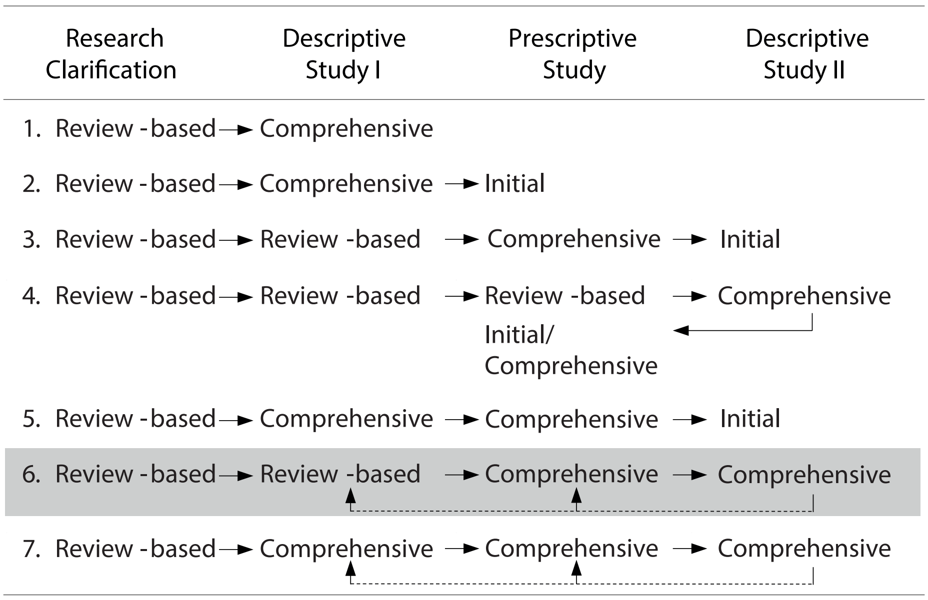

Figure 20. Types of design research and the main research focus of this thesis accentuated in grey (redrawn and applied from Blessing and Chakrabarti (2009))

3.2.1 Type of Research

The positioning, or focus, in the DRM framework is core to the planning and execution of the research, and affects the research path and consistency of the results derived from the research process. Therefore, the research focus is clarified using the DRM structure for different types of research (Blessing and Chakrabarti, 2009). The applied research type is illustrated in grey in Figure 20. This research type *(type 6)* suggests the development of design support and the comprehensive evaluation of this support. The understanding of the existing situation is obtained from literature, which means that *descriptive study I* is primarily review-based. This material reviewed, through searching for internal and external sources (Ulrich and Eppinger, 2012), is destined to be sufficient to develop the support, referred to in the *prescriptive study*. Even though descriptive elements, aimed at understanding design through the evaluation of the design support, are emphasized for *descriptive study II*, studies have also been conducted to understand design better, as referred to *descriptive study I*.

To clarify the research methodology, methods suggested by Blessing and Chakrabarti (2009) have been applied. To give purpose and direction to the research, aim and goals of the research have been clarified, and the phenomena studied have been stated; see Section 1.2.1. To propel the research work, research questions has been developed; see Section 1.2.3. Conceptualization is made through the development of assumptions using the *reference model* method to illustrate a network of consequences due to high or low influence of the studied phenomenon, as illustrated in Figure 2. Through this, concepts that interrelate are disclosed using the *Areas of Relevance and Contribution* (ARC) diagram method, as illustrated in Figure 5. A large part of the understanding of design is deduced through the analysis of literature to reveal design challenges. However, empirical studies, by interrogating industrial practitioners, have been conducted to reveal industrial challenges and to create an aggregate understanding of the phenomena. The industrial challenges are both review-based and empirically induced in collaboration with industrial partners.

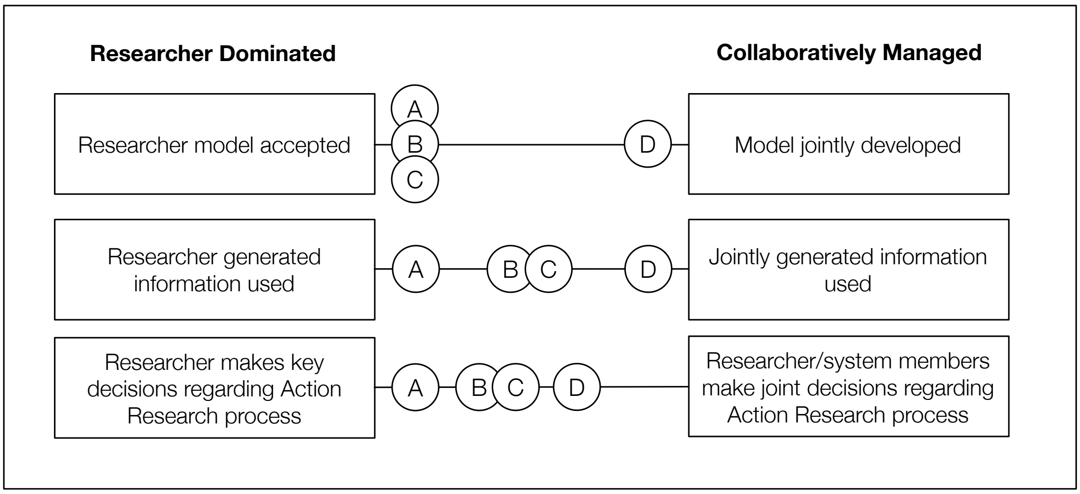

Figure 21. The researcher role in the action research and the research approach for Papers A, B, C, and D accordingly (redrawn and applied from Chisholm and Elden (1993))

3.3 COLLECTION OF DATA

There are various ways whereby data within design research may be collected; however, it highly depends on the research approach and type. The research presented in this thesis is qualitative in nature. Bryman and Bell (2007) state that qualitative research may be used to validate theories, not solely creating them. This thesis combines the two approaches, however there is a slight emphasize on validation rather than creation. In qualitative research, there are multiple ways of collecting data to gain increased understanding of phenomena, such as through observations, interviews, and examining documents (Creswell, 2013).

3.3.1 Observing, Interviewing and Examining Documents

Observations are properly executed in the field, through taking notes of the behavior and activities of individuals (Creswell, 2013). Creswell (2013) also writes about the use of interviews and how they can be structured, as a means of qualitative data collection. Interviews are carried out face-to-face, and are either *unstructured* – do not prescribe precise questions, *semi-structured* – precise and open-ended questions, or *structured* – precise questions with pre-defined multiple-choice answers (Blessing and Chakrabarti, 2009). Focus group interviews typically engage six to eight people, and can be applied to extract views and opinions from participants by posing a few unstructured and open-ended questions (Creswell, 2013). Examining documents and products are other forms of data collection. Blessing and Chakrabarti (2009) emphasize the use of products, drawings, notes, and meeting minutes to understand design. Documents can be internal, reflecting the organization and processes (Blessing and Chakrabarti, 2009), but may very well be public, such as official reports (Creswell, 2013). Opposed to observing and interviewing, participating in design and still research is advised through action research.

Action Research

Action research was developed to increase the relevance of research results and to produce sufficient solutions to society's problems (Blessing and Chakrabarti, 2009). The action research approach is based on both action and research. Blessing and Chakrabarti (2009) state, *"through cycles of action and research a better understanding is obtained."* A main characteristic of action research is the role of the researcher, being a co-participant in the activities of the study objects (Chisholm and Elden, 1993).

3.3.2 Applied Data Collection Methods

The type of research, presented in Section 3.2.1, reflects the data collection methods posed in this thesis, as well as in the appended papers. With this in mind, the papers that underpin this thesis are mapped into a model; illustrating the role of the researcher in the research conducted; see Figure 21.

Data Collection: Paper A

Paper A aims at exploring and understanding the industrial challenges of interdisciplinary development and the configuration of complex products. An interview study was conducted at a prominent car manufacturer in Sweden. Ten semi-structured interviews and two workshops were held with ten senior managers, who had near 30 years in the business, operating in different organizational disciplines, such as market, development, manufacturing, sales and after-market. All participants had been working with configuration aspects of design. The interviews were transcribed, and then sent to the interviewees for verification.

Data Collection: Paper B, C and D

For Papers B, C and D, the primary method used to collect data is based on Jørgensen (1992), as shown in Figure 22. The method has dual starting points, 1) a problem base that is found in collaboration with the industrial partners, and 2) a theoretical foundation based on primarily *engineering design* and *systems engineering*.

Three real-life cases from the aerospace, automotive, and power industries are illustrated. The case studies performed are carried out using a software system, which has been developed in tandem with the research process. The development of the software is however not part of this research work; rather it is used as a means of validating a methodology.

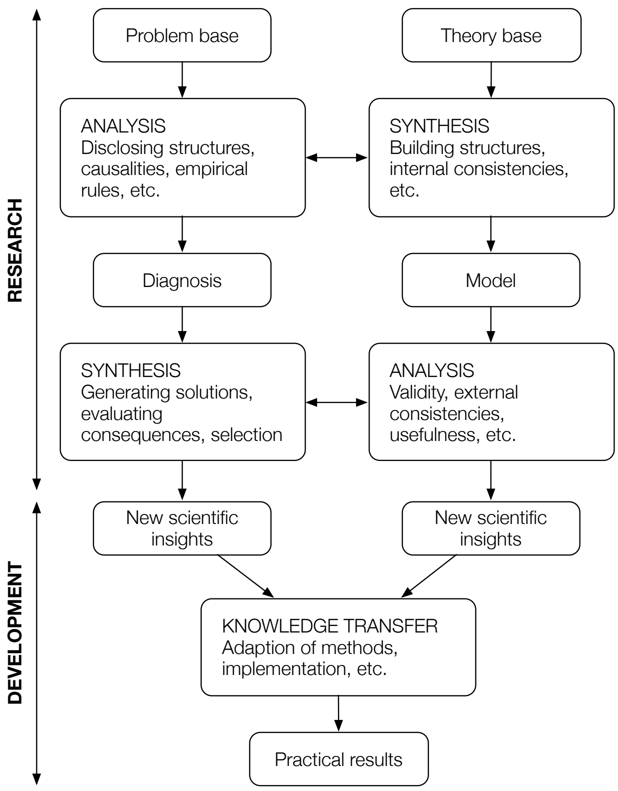

Figure 22. Research method applied in Papers B, C, and D
(redrawn from Jørgensen (1992))

Platforms of aero engine sub-systems, vehicle seats and electro-mechanical components have been modeled in retrospect by industrial system specialists and the researchers. The purpose has been to obtain in-depth system knowledge in terms of the functions that systems and their parts are aimed to provide and their constraints, how this has been manifested in terms of design solutions, and why a particular solution has been chosen. This has been conducted by means of iterative processes where: 1) the specialists have provided in-depth knowledge about the systems, supported by relevant documentation, during interviews and workshops, 2) the researchers have interpreted the information provided and prepared the platform models according to the theory, and 3) the proposed models have been evaluated together by the specialists and researches during feedback sessions where the models have been explained and demonstrated. Considering the feedback and the new knowledge gained during each feedback session 4) the platform models have been revised and refined.

3.3.3 Methods for Developing Design Models

This research focuses on the information that is handled during the development of platforms and how that information can describe a *reality* represented by models. Besides delivering scientific contributions, there is also an explicit goal to deliver results relevant to industry. Such relevance is met if it is possible to implement proposed models in computer-based software to support aimed at specific users. Duffy and Andreasen (1995) describe four different modeling steps to illustrate this, see Figure 23. To describe reality, they claim that *phenomena models*, *information models*, and *computer models* are needed.

To be meaningful, and to support the design work, all models are to represent reality while gradually increasing their level of detail, as well as their level of concretization. A phenomenon model describes basic constructs identified when observing and analyzing reality. The phenomenon model can be further refined, or detailed, into an information model where each piece of information is structured using, for example, classes and attributes. Such a model can then be implemented in the software. This thesis is taking the journey from reality, phenomenon model and information model. The work has been accompanied by a computer model; however, it is not a part of the actual research work, but has rather been developed externally and is utilized to do research.

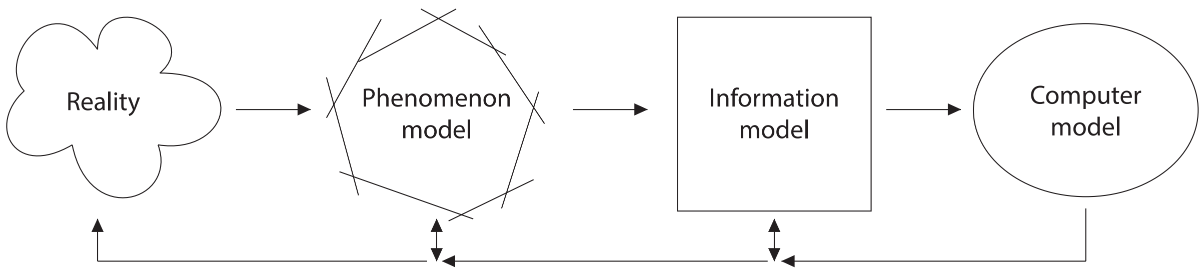

Figure 23. Research approach for design modeling (Duffy and Andreasen, 1995)

3.3.4 Validation of Design Methodologies Using Case Studies

A vital focus of this thesis is to validate a methodology for developing platforms, i.e. to support the preparation through system design, and the execution through generation of system variants for industrial application. To accomplish this, four case studies have been outlined. Yin (2003) defines four quality criteria in case study research that are stated in Table 2 and described below.

Construct validity (1) in this study is defined by the organizational and user acceptance. This also coincide with the definition of verification by acceptance as stated by Buur (1990).

Verification by acceptance implies that experts accept the contributions. The internal and external validity is of foremost concern to ensure such acceptance. Bryman and Bell (2007) define internal validity (2) as if there is a good match between the observations of the researchers and the theoretical ideas they develop. Internal validity is closely related to logical verification as explained by Buur (1990) as the completeness and consistency of research results. Completeness refers to when the results are in agreement with established theory, whereas consistency is when the terminology is clear and conforming. Since extensive efforts are needed to gain an in-depth understanding of the phenomena studied, internal validation carries strengths in case study research. Bryman and Bell (2007) also define external validity (3) as if the findings can be generalized across different settings, and to what degree. In case study research, external validity is a weakness in contrast to internal validity (Bryman and Bell, 2007). According to Yin (2003), case study research relies on analytical generalization, whereas quantitative research rely on statistical generalization. In quantitative research, the test sample is the basis for external validity. In defense of external validity in case study research, Yin (2003) continues to argue that the case studies can be seen as the test sample. Reliability reflects the reproducibility of measurements (Blessing and Chakrabarti, 2009). The research presented in this thesis applies elements of action research, which is why the reliability (4) of the research can be somewhat questioned. Action research suffers from a high degree of bias, as the researchers have been interacting with and possibly been influencing their study objects.

Table 2. Four different kinds of validity explained by Yin (2003)

1. *Construct validity* Establishing correct operational measures for the concept being studied
2. *Internal validity* Establishing a causal relationship and false relationships, whereby certain conditions are shown to lead to other conditions
3. *External validity* Establishing the domain to which the findings of a study can be generalized
4. *Reliability* Demonstrating that the iterations of a study, such as data collection procedures, can be repeated gaining the same results

4

"However beautiful the strategy, you should occasionally look at the results."

– Winston Churchill

RESULTS

This chapter essentially provides you with the findings with regard to the scientific mission stated in this thesis. The studies conducted are clarified as far as content, execution, and participant roles are concerned.

The integral part of the results presented in this thesis is based on the appended papers. The focus of communicating the results is on the overall findings, related to the phenomenon *interchange of information among and across disciplines and systems*, gained throughout the studies conducted. Full descriptions of the results can be found in the appended papers at the back of this thesis.

Each paper contributes to specific parts of the research questions. Paper A addresses *RQ 1* by a broad study of the configuration management strategies and different architectural approaches to the increasing complexity of products. Paper B addresses *RQ 1* by applying current processes, methods, and tools to form a methodology for platform development. Three industrial examples are provided to partly validate the methodology. Paper C addresses *RQ 1* by applying the methodology by using a complementary software system. Three industrial cases are used to primarily validate the methodology, but also the software system. Paper D addresses *RQ 2* by applying and improving the methodology slightly from Papers B and C, in an interdisciplinary platform context, using a case of early producibility assessments.

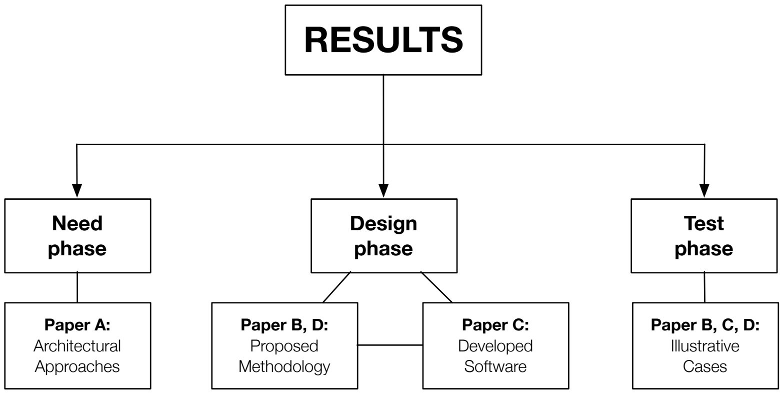

Figure 24. Structure of results

The overall structure of the results presented in this chapter is as follows:

- Clarifying the industrial need by building upon the theory with fresh industrial insights, and proposing possible strategies for managing increased product complexity
- Combining and explaining established design models, methods and tools into a platform development methodology
- Briefly describing a custom-made software tool to reflect the platform development methodology
- Providing four illustrative cases, each reflecting a distinctive design scenario, to validate the methodology for the development of interdisciplinary platforms:

  Scenario 'a') Design space exploration and extension
  Scenario 'b') Design space exploration through producibility assessments
  Scenario 'c') Supply-chain collaboration
  Scenario 'd') Configure-to-order

The main results of the research presented below are a collection of models, methods and tools that form a methodology. The methodology is implemented in a software system aimed to support suitable engineering users completing design tasks when developing platforms. The view on design support applied in this thesis is illustrated in Figure 25.

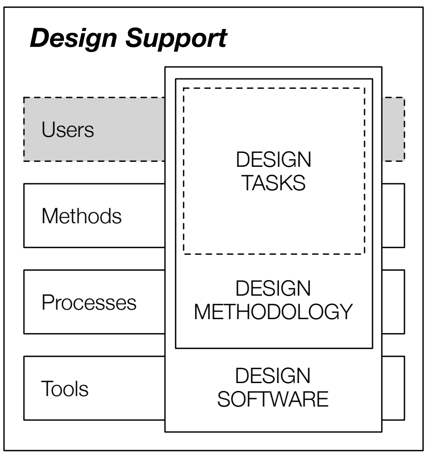

Figure 25. Levels of Design Support

4.1 PAPER A: MANAGING OR LIMITING PRODUCT COMPLEXITY

Paper A takes a closer look into the architecture of a product, and deals with the challenge of configuring products while meeting changing customer needs. Today, internal costs increase with increasing product complexity. The difference between limiting and managing complexity is contrasted. Four approaches are discussed, limiting complexity short-term and long-term, or managing complexity short-term and long-term; see Figure 26. Limiting complexity, of the product architecture, primarily regards standardization, optimization of configuration rules, and reducing variants. Managing complexity, of the product architecture, instead allows variability, yet controlling it. Two approaches used to manage complexity of the product architecture are discussed: 1) the modular approach with restricted well-defined interfaces, and 2) a fully flexible approach with adaptable interfaces. It is concluded that a step towards highly customized products requires flexibility of the product architecture. In some parts of the product architecture, the modular approach is suitable, whereas in other parts, an even more flexible approach is feasible and necessary. Flexibility must be introduced in areas of the product architecture where redesign is commonplace and persistent. And to emphasize design reuse through commonality, it is viable to introduce flexibility in subsystems that recur in the product architecture. Likewise, there is no need to force full flexibility into a standardized system – such as standardized screws. However, in business where change constantly need to be incorporated, flexibility of the product architecture can be used to prepare for change and to manage complexity over time. A transition from integral product architecture can be implemented stepwise by introducing flexibility into a single system, or module, at a time, to verify benefits on a smaller scale. Then, if benefits can be proven, additional systems can be gradually transformed. Flexibility of the product architecture imposes a lengthy platform life, for many generations of products, which can motivate the high initial cost of such an investment.

	Short-term	Long-term
Manage product complexity	**DESIRED SITUATION** ***Existing architectural approach*** *Offer customized products* *Sell premium products* *Economies of scale* *Commonality* *(no cost)* **NOT VIABLE**	**ALTERNATIVE SITUATION** ***Changed architectural approach*** *Offer customized products* *Sell premium products* *Economies of scale* *Commonality* **FEASIBLE** *(high perceived cost)*
Limit product complexity	*(low perceived cost)* **POSSIBLE** ***Existing architectural approach*** *Offer standard products* *Aim for premium segment* *Optimize configuration rules* *Reduce variants* **AS-IS**	**POSSIBLE** *(high hidden cost)* ***Existing architectural approach*** *Offer standard products* *Risk of loosing premium segment* *Optimize configuration rules* *Reduce variants* **UNDESIRABLE SITUATION**

Figure 26. Strategic positioning regarding different architectural approaches

4.2 PAPERS B AND D: A PLATFORM DEVELOPMENT METHODOLOGY

Paper B and Paper D introduce a methodology for the development of platforms, comprising methods for both *platform preparation*, and *platform execution*. Platform preparation includes the creation of a configurable platform model with design information and knowledge. Platform execution implies the configuration of feasible product variants within a defined bandwidth. The platform methodology is initiated by a change, such as a design decision or a new market input. It can be a new requirement, such as added functionality or increased performance or manufacturing capability, or new opportunities such as the implementation of new technology. The methodology can be divided into three different levels, according to the maturity of the platform model: the *functional level*, the *system level*, and the *detailed level*. Dependent on these levels, different engineering users are more or less suitable.

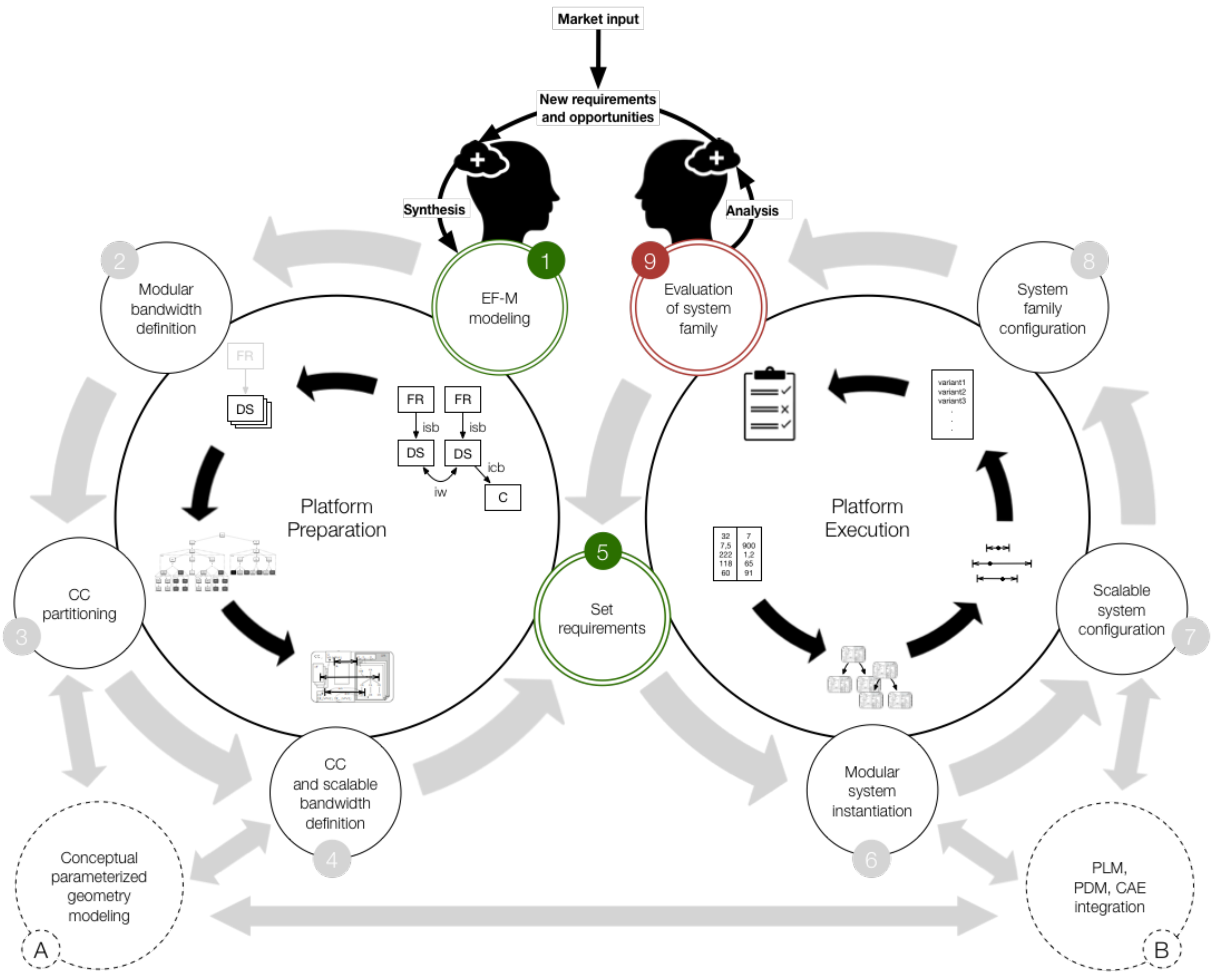

Figure 27. The proposed methodology for development of platforms

4.2.1 The Platform Preparation Process: Functional Modeling (Paper B)

Paper B describes the steps of the *functional level* of the platform. At the functional level, expert knowledge about design is necessary to prepare for design reuse. Therefore, design engineers are the most suitable users to engage during these process steps.

EF-M Modeling

The first step *(1)* of the platform preparation process, *EF-M modeling* as described in Section 2.4.3, ideally starts with the formulation of an overall functional requirement (FR) for the whole system family. This FR addresses the aggregated functionality of a system design. The solution to this function is an envisioned design, rather than a design expressed in detail. The envisioned design is represented by a system family, and is modeled as a design solution (DS) object. The functionality of this DS is decomposed into an arbitrary number of functional sub-requirement objects. Capturing a design as a design matures during the development is a recurring modeling activity. The more mature the design is, the richer and in more details the EF-M model can be modeled.

Modeling the Modular Bandwidth

The second step *(2)*, *modular bandwidth definition*, involves finding alternative design solutions (DSs) to functional requirements (FRs), thus defining the modular bandwidth of the platform. This step is strongly linked to the first principle of SBCE, creating alternative DSs as part of mapping the design space. Finding alternative DSs is an extension of EF-M modeling. As new technologies, design concepts or both are formed; they must be represented as alternative DSs and can assume different maturity levels, i.e. a DS with high maturity has been developed before, and can be fully reused in a new setting, whereas a DS with low maturity is very conceptual and might be dependent on a technology with low maturity level. The modular bandwidth is also described in Section 2.4.4. The modular bandwidth is used to generate modular variants, such as combining solutions into distinct system concepts as described in Section 2.4.5.

4.2.2 The Platform Preparation Process: System Modeling (Paper B)

Paper B also describes the steps in the platform preparation process regarding system modeling. Due to the holistic and architectural nature of the *system level*, system architects are primarily suitable to be engaged in these process steps.

Partitioning of CC objects

The third step *(3)*, *CC partitioning*, includes partitioning the EF-M tree into encapsulated system objects, or configurable components (CCs), as described in Section 2.4.3. These system objects are closely related to the modules described in Section 2.4.1, which can be used for design reuse, in a system architecture. A CC is created from a DS in the structure, with its sub-FRs and DSs. This partitioned structure is now the embryo to the *DNA*, or DR, in a CC object.

Modeling the Scalable Bandwidth of System Objects

The fourth and last step *(4)* of the platform preparation process, *CC and scalable bandwidth definition*, includes the detailed design of each partitioned CC and DR with its FRs, DSs and Cs objects. This involves the parameterization of properties and characteristics of the system objects, definition of variant parameters (VPs) that shall govern the configuration of each CC, and the development of design rules that define the composition of the system and generate the output parameter values determining the configuration and simulation results. The scalable bandwidth, or parameter value ranges, shall reflect the functional capabilities of the system design envisioned. The scalable bandwidth is used to generate scalable variants, such as combining parameters into distinct system variants as described in Section 2.4.5.

4.2.3 The Platform Preparation Process: Detailed Modeling (Papers B and D)

Paper B describes two CAx steps proposed in the methodology. The *detailed level* includes the design of physical models with more detailed features than the functional level and system level, which is why design engineers are suitable for engagement in the process steps.

The first CAx step *(A)* implies *conceptual geometry modeling* and includes the creation of parameterized geometric models, using CAD software. The parameterization of the CAD models needs to reflect the parameters and bandwidth modeled for the system objects.

The second CAx step *(B)* implies the *integration of the CAx software*. It involves the creation of analysis and simulation models that the system object models need for their configuration and performance simulation. The governing parameters required for the activities are interchanged between software, and stored in PDM software. The parameterized models can thus be used in external CAE systems where simulation models are developed. The system object models are linked to CAE models arranged into a PLM architecture. Paper D is taking a closer look at the modeling of PLM architectures and design activities related to the interchange of information between CAx software. Case D is presented in Section 4.3.4.

4.2.4 The Platform Execution Process (Papers B and D)

Platform execution is a systematized process for generating configured and feasible system variants, or for exploring the design space within the modular and scalable bandwidths of the platform as a means to mature the platform model through the development.

The first step *(5)* of the platform execution process, *set requirements*, includes setting input parameter values in the control interface of the CCs. The methods applied in the execution process will in the end return a feasible system variant if the input parameter values are within platform bandwidth. Both functional property and solution descriptive parameters can be used as input. In the second case, functionality is explored by specifying ranges of functional property values as input in the first execution step. If the ranges specified are within the bandwidth, the execution process will return feasible system variants.

In the second execution step *(6)*, *modular system instantiation* reflects the actual generation of alternative system solutions based on alternative design solutions as described in Section 2.4.5. The top node CC and the following used CCs will employ their composition elements (CEs) to link CCs. The design rules are executed starting from the top node CC and continuing down through the composed CC structure. Depending on the availability of matching CC and DS alternatives, different functionality based system architectures will be generated as a result.

In the third execution step *(7)*, *scalable system configuration*, reflects the actual generation of alternative system solutions based on alternative parameters, as described in Section 2.4.5. Again, the top node CC and the following used CCs will use their composition elements (CEs) to link CCs. The resulting parameter values from each CC are transferred to their underlying CCs. The information transfer between CCs goes from the composition set (CS) of the using CC to the control interface (CI) of the used CC; see Figure 14. Based on the scalable bandwidth defined, system architectures, parameterized system variants or both will be generated as a result. The forth execution step *(8)*, *system family configuration* treats the complete system model, thus the configuration of a system family compiling the variant results from the modular instantiation and scalable configuration. The compatibility of the composed architectures is checked and the complete DR of each architectural option is re-established employing the DRs of the CCs used in each composition. To compare the different architectural options, DSMs and axiomatic coupling matrices can be generated.

The fifth and final step *(9)* of the platform execution process, evaluation of system family, forms the basis for various assessments and relies heavily on the interaction of system architects and designers to evaluate and decide upon feasible variants depending on the scope of the platform. The evaluation can be based on DSMs, axiomatic coupling matrices and trade-off curves. The platform execution steps are illustrated in Figure 27, and in Figure 28.

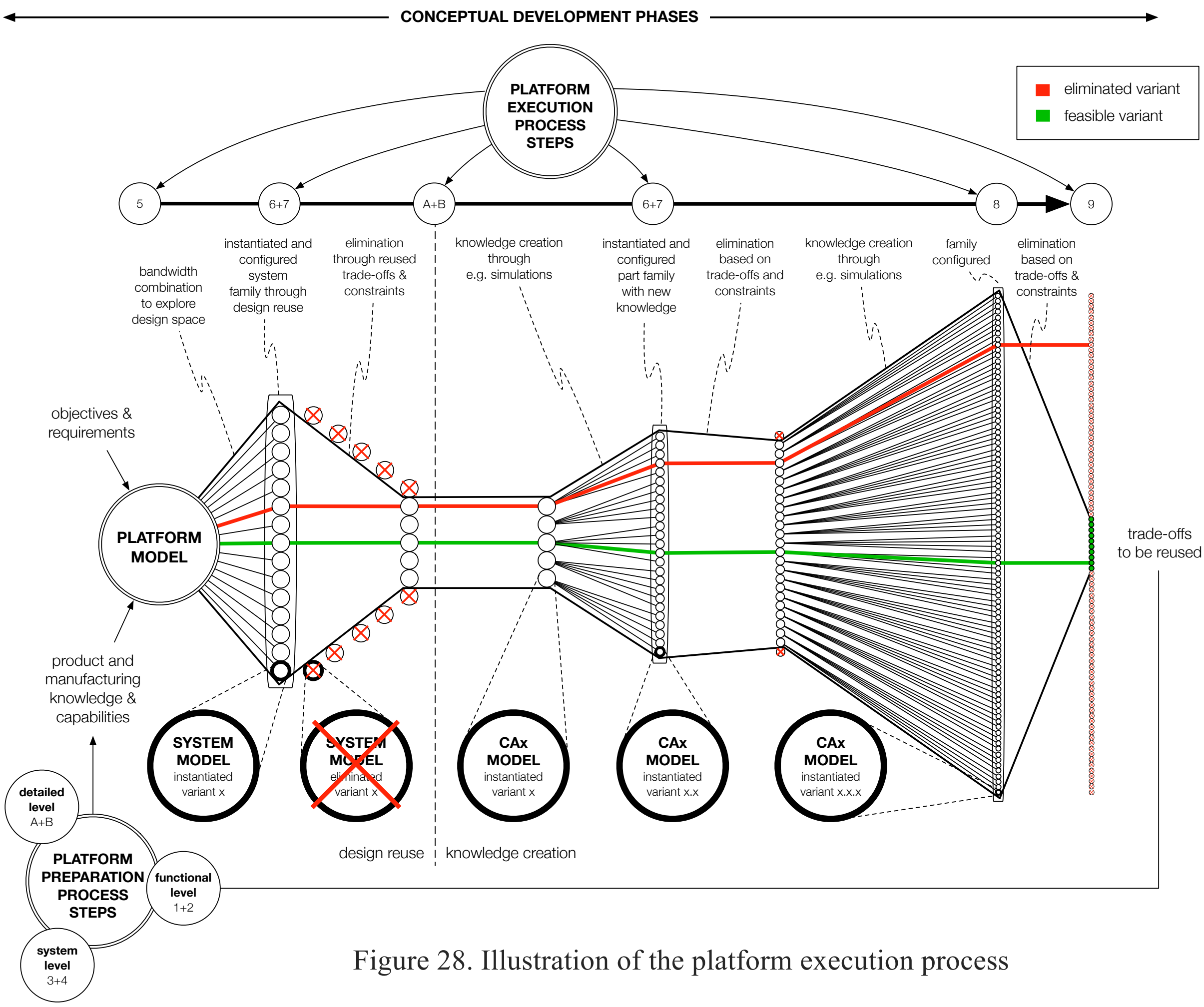

Figure 28. Illustration of the platform execution process

4.2.5 Paper C: A Software for Development of Platforms

Paper C describes a software system for development of platforms – the Configurable Component Modeler (CCM). CCM supports all steps, processes and methods presented in the platform development methodology described in Figure 27 and in Figure 28. CCM is a means of implementing the methodology in practice.

A PLM architecture, including CCM, can be modeled to achieve capabilities of platform modeling and configuration (PMC), requirements management (RM), product data management (PDM), and the integration to tools for advanced computer aided design (CAD) and engineering (CAE). The software developed is based on an object-oriented approach, which enables the development of flexible platforms.

The software system, CCM, is not a main contribution in itself but rather a means of further validating the methodology and enabling software integration to ensure the efficient interchange of information among and across disciplines and systems.

4.3 ILLUSTRATING THE METHODOLOGY USING CASE STUDIES

The approach, including the proposed platform development methodology and the complementary software, is aimed at supporting conceptual phases of development. Four case studies are presented, to validate different parts of the methodology. In

Figure 29, the full context for the methodology and case studies is framed, illustrating conceptual phases through the *functional*, *system*, and *detailed levels*. The steps of the platform development methodology and the two methodological modes, *platform preparation* and *platform execution*, are illustrated. The methodology comprises nine steps and an additional two. These steps are listed below:

Platform Preparation

- *Functional level*
    1. *EF-M modeling*
    2. *Modular bandwidth definition*
- *System level*
    3. *CC partitioning*
    4. *CC and scalable bandwidth definition*
- *Detailed level*
    A. *Parameterized geometry modeling (CAD)*
    B. *PLM, PDM, CAE integration*

Platform Execution

5. *Set requirements*
6. *Modular system instantiation*
7. *Scalable system configuration*
8. *System family configuration*
9. *Evaluation of system family*

The steps of the methodology described above are related to eight theoretical concepts. These concepts are cross-references below:

1. *SBCE principles:* reviewed in Section 2.3.1
2. *Platform characteristics:* reviewed in Section 1.1.1
3. *System decomposition:* reviewed in Section 2.4.2
4. *Bandwidth type:* reviewed in Section 2.4.4
5. *Requirements management:* reviewed in Chapter 2
6. *Product architecture:* reviewed in Section 2.4.1
7. *Design representation:* reviewed in Section 2.4
8. *Design reuse flexibility:* reviewed in Section 1.1.1

Each case represents a specific design scenario: a) *design space exploration and extension*, b) *supply-chain collaboration*, c) *configure-to-order*, and d) *design space exploration through producibility assessments*. The four cases are plotted into the map of Figure 29 according to the methodology steps and concepts used to illustrate the purpose of each case. To prepare a platform, the *functional*, *system*, and *detailed levels* can be chosen iteratively depending on the maturity of the platform and the changed conditions accordingly. These conditions can for example be the need for new functionality, altered constraints, or a new technology. For reasons like these, new systems need to be developed or reused to accommodate the new capacity and meet the new conditions.

CONCEPTUAL DEVELOPMENT PHASES

	Platform Preparation				Platform Execution					CAx Steps	
	1	2	3	4	5	6	7	8	9	A	B
	a, b, c, (d) **EF-M modeling**	a **Modular bandwidth definition**	b, d **CC partitioning**	b, c, d **CC and scalable bandwidth definition**	b, c, d **Set requirements**	a, b **Modular system instantiation**	c, d **Scalable system configuration**	a, b, c, d **System family configuration**	a, b, c, d **Evaluation of system family**	c, d **Parameterized geometry modeling (CAD)**	c, d **PLM, PDM, CAE integration**
SBCE principles	a, b, c, (d) map the design space	a map the design space	b, c, d map the design space	b, c, d map the design space	b, c, d integrate by intersection	a, b integrate by intersection	c, d integrate by intersection		a, b, c, d establish feasibility before commitment		
Platform characteristics		a modular	b, c, d modular	c, d scalable				a, b, c, d system family generation		c, d part family generation	
System decomposition	a, b, c, (d) elaboration	a elaboration	b, c, d encapsulation	b, c, d encapsulation/ elaboration							
Bandwidth type		a modular bandwidth		c, d scalable bandwidth		a, b modular bandwidth	c, d scalable bandwidth				
Requirements management	a, b, c, (d) functional requirements & constraints	a platform boundaries		b, c, d platform boundaries	b, c, d technical requirements						c, d requirements database
Product architecture	a, b, c, d interactions and interfaces			b, c, d interactions and interfaces		a, b architectural options		a, b, c architectural options			
Design representation	a, b, c, (d) design solution	a alternative design solutions	b, c, d configurable component	b, c, d parameter ranges	b, c, d parameter values	a, b variants based on modules	c, d variants based on parameters	a, b, c, d family of variants	a, b, c, d DSMs, trade-off curves, & architectures	c, d CAD models	c, d CAE models
Design reuse flexibility	a, b, c, (d) very high	a very high	b high	b, c high	b, c medium					c low	c low

FUNCTIONAL LEVEL — SYSTEM LEVEL — DETAILED LEVEL

[a] design space exploration and extention [b] supply-chain collaboration [c] configure-to-order [d] early producibility assessments

Figure 29. Schematic map of the platform development methodology, showing links between combined and improved processes, models, methods and tools serving *platform preparation* (*1*)-(*4*), *platform execution* (*5*)-(*9*), plus two *CAx steps* (*A*) and (*B*), and the allocation of four illustrative cases, a) *design space exploration and extension*, b) *supply-chain collaboration*, c) *configure-to-order*, and d) *design space exploration through producibility assessments*, to support *conceptual level*, *system level* and *detailed level*

4.3.1 Paper B: Design Space Exploration and Extension

To illustrate the *functional level* of the methodology for platform development in a *design space exploration and extension* scenario, an example from the aerospace industry is presented. The case company is a component supplier responsible for the mechanical design and manufacturing of static parts for aero engines. The product studied, Turbine Rear Structure (TRS), is located at the rear of an engine, and is illustrated and highlighted in red in Figure 30. Each TRS is manufactured at a yearly volume of approximately 400 units and is customized for different customer requirements. Due to its location, the TRS is exposed to high temperatures, which induce high thermal loads; see the right part of Figure 31. The relation between the methodology and theoretical concepts is illustrated in

Figure 29 (not: *a*).

A new requirement is introduced into an existing aero engine sub-system platform. In this new setting, the TRS component will be exposed to temperatures of 900°C instead of 700°C. This exposure drives the need for introducing a new functional requirement (FR) – reduce thermal loads. The preparation of the aero engine sub-system platform begins with expanding the modeled DR and its EF-M tree *(1)*. The modular bandwidth is defined creating alternative DSs – thermal matching, cooling system, heat shield, and a more thermally resistant material *(2)*. Interactions between DSs are modeled. The new requirement is set – from 900°C to 700°C. Based on the modular bandwidth defined, the platform is executed using modular instantiation to generate nine architectural options *(6)*. In this example, the nine architectural options represent the system family *(8)*. The modular instantiation is aimed at the exploration of available architectural options in the platform, and the identification of needs for further development. In this scenario, a DSM of each architectural option is generated. The nine DSMs are then used as bases for the analysis of, for example, change propagation (Raudberget et al., 2015) or DS clustering *(9)*. DS clustering can form the basis for a new scenario of design space exploration and extension, where such analysis can be used to partition CC objects. The process described is made possible by modeling interactions between alternative design solutions without designing any geometry model.

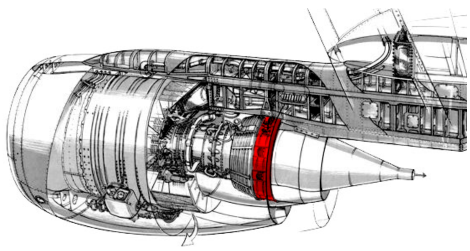

Figure 30. An aero engine with the TRS highlighted in red

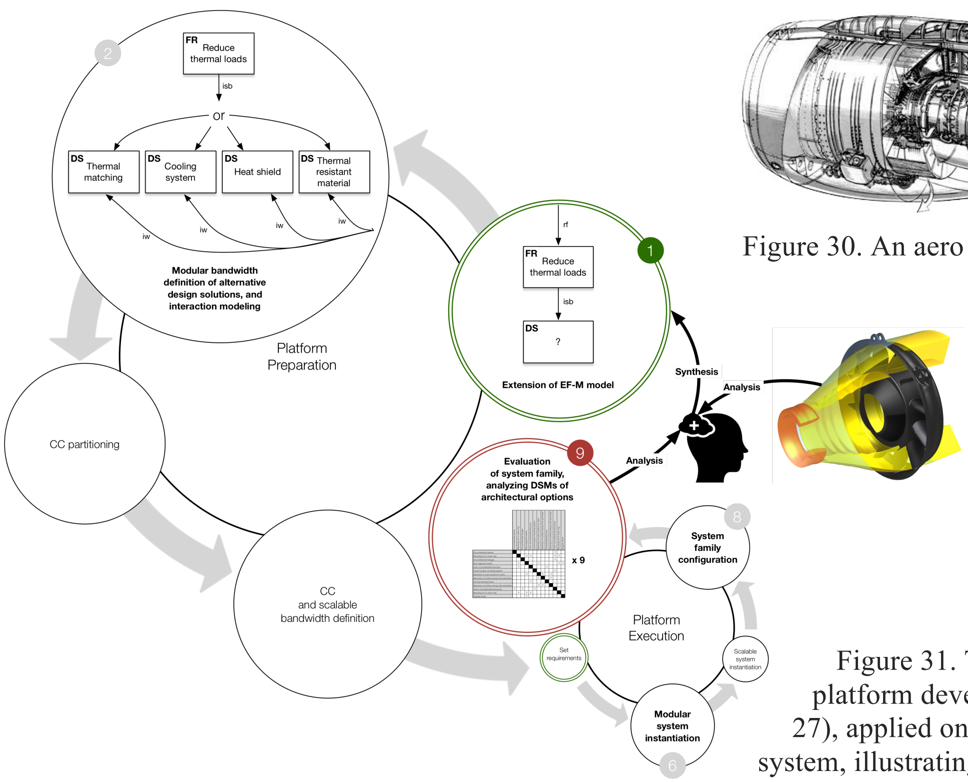

Figure 31. The methodology for platform development (ref. Figure 27), applied on the aero engine sub-system, illustrating the *functional level*

4.3.2 Paper C: Supply-Chain Collaboration

To illustrate the *system level* of the methodology for platform development in a *supply chain collaboration* scenario, the CCM software has been used. The relation between the methodology and theoretical concepts is illustrated in

Figure 29 (not: *b*). The system platform has been created together with a supplier of vehicle seats.

Vehicle seats are complex products with a great number of functions. These functions may vary between seats for different types of users and vehicles, as well as between different market segments. There are many common functions and design solutions that can be reused in different seat variants to provide scale benefits in development, as well as in production. This motivates the development of a platform for vehicle seats where the reuse the common features of different seats can be combined with specific features when variants are configured. The vehicle seat system, in this case, contains 44 subsystems as seen in Figure 32. Each of these systems are modeled as reusable and scalable system objects.

Platform preparation is initiated by modeling the *DNA*, or EF-M trees *(1)*, for each subsystem *(3)*. The alternative architectures, generated through the execution process, are governed by rules. These rules are applied to rf relations, for including or excluding functions, and isb relations, for including or excluding design solutions. Requirements of weight and cost are applied to the bottom level DSs in the DR model of each CC to enable weight and cost estimates. The vehicle seat platform presented has been prepared for different categories and market segments dependent on rules modeled *(4)*, type of vehicle, type of user, type of safety belt arrangement, and market segment.

Platform execution addresses the modular instantiation, which is based on the parameters for types and market segments, and generates 24 architectural options *(6)*. Thus, the system family has been configured *(8)*. The aim of the evaluation is to communicate different architectural options based on varying market segments together with suppliers *(9)*. In collaboration with the supplier, new functionality and new systems can be developed. The alignment of interfaces between the seat system and vehicle can be addressed early based on discussions of alternative architectural options. Early communication of feasible architectural options is valuable for system architects when planning for detailed design. It is also an efficient means of improving customer collaboration during the development.

Figure 32. A vehicle seat and 44 subsystems, modeled as system objects in the software CCM to generate different system architectures

4.3.3 Paper C: Configure-to-Order

To illustrate the *detailed level*, of the methodology for platform development in a *configure-to-order* scenario, the CCM software has been used. The relation between the methodology and theoretical concepts is illustrated in

Figure 29 (not: *c*).

A company from the power industry is presented. The case company is a supplier off mechatronic designs of parts for contactors. An electromagnetic contactor is a switch for 3-phase electricity and can be controlled, as well as remotely maneuvered. The contactor contains systems from various disciplines including mechanical, electro-mechanical and electronic hardware, as well as software. The main subsystems are prepared with their *DNA*: electromagnet, electronic control system, chassis for contact parts, chassis for operating parts, spring system, contact system, contact bridge and arc control device *(1+3)*. These subsystems are common between different variants. Scalable bandwidth, accommodating different geometries, is prepared *(4)*. The contactor is customized based on the input current and size. It is designed for high voltages, 400 Volt, and currents, 900 Ampere. The defined governing requirements include: the current of the circuit intended, the size of chassis that fits the application intended, an electromagnetic power constraint (<250 W), and a cost constraint (<330 SEK) *(5)*. The contactor system is instantiated *(7)*. CCM distributes data to CAx systems, completing the static analyses using *Maxwell* and the dynamic analyses using *Simplorer*.

A wide variety of product variants can be configured using this platform approach. The requirements can be used to communicate feasible regions through the generation of trade-off curves. It supports design engineers and system architects in eliminating unfeasible variants, thus contributing to mass customization. This evaluation holds 1000 possible product variants within the boundaries of the defined bandwidth.

As a basis for evaluation, a trade-off curve between electromagnetic power, generated by an electromagnet, and cost, based on the material cost, has been derived (9). In this way, feasible variants can be identified as a result of an electromagnet power requirement above 250 W, at a cost of no more than SEK 330. Hence, the feasible red area can be seen in Figure 33. To satisfy a set of customers, the requirements can be changed to suit the preferences of many customers.

The platform model can be used to configure contactor variant geometries within the platform bandwidth, harmonize between distinctiveness and commonality due to size, and explore feasible regions of the design space when new requirements are introduced.

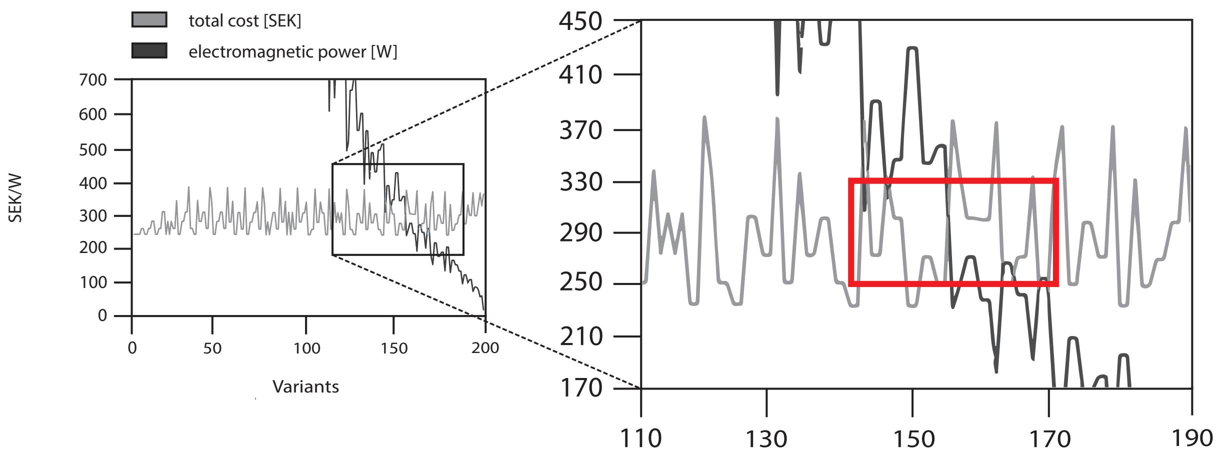

Figure 33. The trade-off curves can be used to harmonize between total cost and electromagnetic power of the configured variants

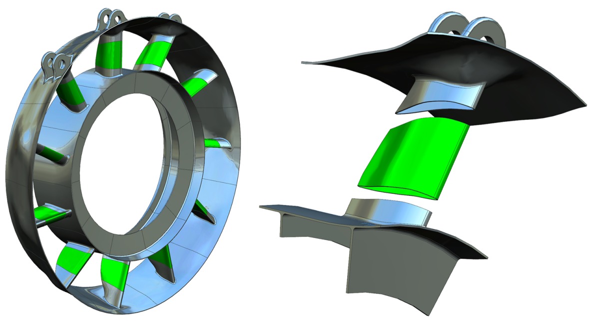

Figure 34. The aero engine sub-system TRS, on a detailed level

4.3.4 Paper D: Design Space Exploration through Producibility Assessments

To illustrate the *detailed level* of the methodology for platform development, in a scenario of *design space exploration through producibility assessments*, an example from the aerospace industry is presented. The product studied, Turbine Rear Structure (TRS), is located at the rear of the aero engine, illustrated in Figure 30. Each TRS is currently manufactured at a yearly volume of approximately 400 units and is customized for a few different customer requirements. However, an expected increase of new engine variants is imminent. The relation between the methodology and theoretical concepts is illustrated in

Figure 29 (not: *d*).

The aerospace industry is performance-driven and the level of complexity of product systems is high. To remain competitive on the market it is necessary to develop and manufacture products within a short timeframe while meeting increasingly challenging requirements. The case company has the ambition of reducing the time from a customer RFQ (Request for Quotation) to an offer of feasible conceptual alternatives from three months to three weeks. To be prepared for such a scenario, several phases of the product lifecycle needs to be assessed earlier than before, in the development process. Especially and typically complex manufacturing processes affect time and product performance, which is why it is precarious not to ensure the producibility a product before answering a customer RFQ.

The TRS can be manufactured in various ways and in different combinations, such as full cast, partial cast and partial welding, or partial cast, partial sheet metal pressing and partial welding. This case illustrates a welding assembly scenario, as the TRS is divided into segments, shown in Figure 34.

A PLM architecture is prepared, linking defined CAx systems to CCM, so that simulations can be applied to gain knowledge regarding producibility aspects of design, in the conceptual phases of development *(A+B)*. The CAx systems in this case involve CAD (SIEMENS NX), Geometry Assurance and Robust Design (RDnT), and Geometry and Motion Planning (IPS). The interchange of information between the software systems is illustrated in Figure 35. Based on the information collected, designs that are inferior in terms of producibility can be eliminated.

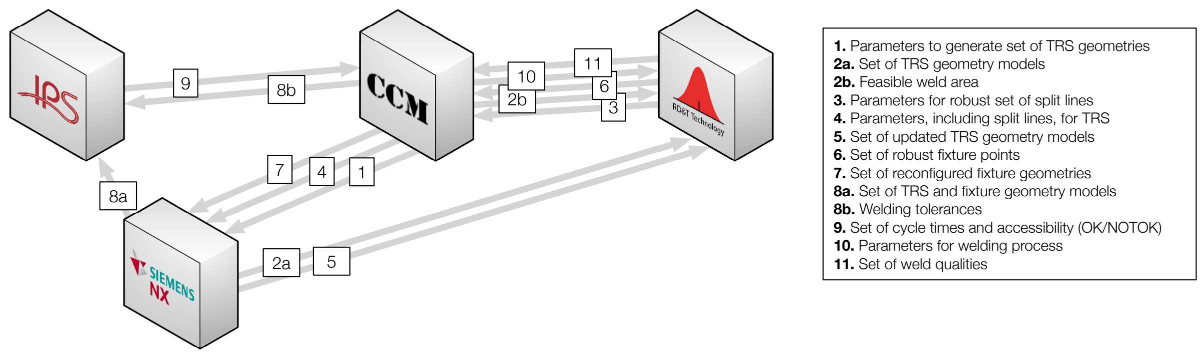

Figure 35. A PLM architecture illustrating the interchange of information between CAx systems

4.4 BRIEF SUMMARY OF FINDINGS

The approach proposed in this thesis is targeting interdisciplinary means. Interdisciplinary design requires intensive processes and activities supported by methods and tools where more than one discipline or system are interacting. It also suggests collaboration between multiple engineering users.

To develop interdisciplinary platforms, these users need to interact and structure product and manufacturing capabilities and design knowledge for the common use and reuse. A methodology to develop interdisciplinary platforms is proposed, capitalizing on existing theoretical approaches, mainly from *engineering design* and *systems engineering*. An assortment of useful models, methods, and tools to serve *mass customization* and *design reuse*, as well as the *interchange of information*, has been identified and reviewed.

The methodology has been validated in four real-life industrial cases, partly using a custom-made software system to incorporate support for the methodology. The development of interdisciplinary platforms has proven promising when generating product performance trade-offs (Case A and Case C), trade-offs between product and manufacturing, from a producibility perspective (Case D), and business-to-business communication of alternative product architectures (Case B). Through these cases, some rigor to the validation of the methodology, and software system, has been established. Due to the interdisciplinary nature of the platform, suitable users have been proposed as interacting during different phases of development. The methodology is aimed at providing a design support for design engineers and system architects, when interchanging information during the phases of planning, concept development and system-level design; see Figure 6.

All four cases are unique according to their specific design scenarios. The cases represent interdisciplinary challenges, such as dealing with market fluctuations and a vast body of requirements and capabilities. Below, the four cases are briefly summarized based on the industry, design scenario, and their interdisciplinary nature, respectively.

Design space exploration and extension: Aerospace Industry

The case of the *aero engine sub-system*, demonstrating the *design space exploration and extension* scenario, considers a product platform at the *functional level*, with its product-internal requirements, taking for example aerodynamics, structural, thermodynamic, and electrical functions and systems into account early in development. The case primarily focuses on demonstrating the methodology in terms of modeling modular bandwidth and generating alternative system architectures and *design structure matrices* (DSMs) as bases for evaluation and design decisions. By using the methodology for platform development, functional requirements and alternative solutions and interactions to interrelated solutions can be modeled to map the design space. This is the basis for execution of the platform to generate alternative architectural options to make early evaluations of very immature system solutions. These evaluations can be the basis for gradually eliminating inferior system architectures, and be valuable for system architects when continuing with system-level design activities.

Supply-chain collaboration: Automotive Industry

The case of the *vehicle seat system*, demonstrating the *supply-chain collaboration* scenario, considers a product platform at the *system level*, with the difficulties of dealing with business-to-business relationships. Communicating system architectures with its requirements, capabilities, functions, and systems is a vital part of development. The case primarily focuses on validating the methodology in terms of generating system architectures, using the software

Configurable Component Modeler (CCM), based on customer segmentation as a basis for evaluation and design decisions in business-to-business relationships. By using the methodology for platform development, functional requirements, solutions and rule-based categories are modeled: type of user, type of safety belt arrangement, and market segment. The platform is executed for generating alternative architectural options, which can be used for early evaluation of immature system architectures. These evaluations can be valuable as input for system architects to gradually eliminate unfeasible architectural options in early phases of development.

Configure-to-order: Power Industry

The case of the *electromagnetic contactor system*, demonstrating the design scenario *configure-to-order*, considers a product platform in the *detailed level*, with product-internal interchange of structural and electrical functions, requirements and capabilities. The configuration of product variants is made possible with the use of different CAx software to simulate product performance as a basis to generate alternative solutions. The case primarily focuses on validating the methodology in terms of linking system objects and CAD models so that trade-off curves can be generated as a basis for evaluation and design decisions. The methodology for platform development is used to model functional requirements, design solutions and constraints – in this case electromagnetic power and cost. The platform is executed to generate a family of contactors, which can be used for evaluation of different design alternatives based on trade-off curves.

Design space exploration through producibility assessments: Aerospace Industry

The case of the *aero engine sub-system*, demonstrating the *design space exploration through producibility assessments* scenario considers an integrated product and manufacturing system platform at the *detailed level*, with interacting product and manufacturing functions and systems, and conflicting requirements and capabilities. The execution process is made possible with the use of different CAx systems and the interchange of information between them. The case primarily focuses on validating the methodology in terms of modeling PLM architectures and demonstrating the interchange of information between CAx systems. The platform is executed for generating a family of producible product variants, which is the basis for generating sufficient knowledge to evaluate different design alternatives, for example through the use of trade-off curves.

5

"In questions of science, the authority of a thousand is not worth the humble reasoning of a single individual."

– Galileo Galilei

DISCUSSION

This chapter is dedicated to provide you with a discussion regarding the stated industrial challenges, scientific mission, research approach and findings. It mainly concerns the quality of the findings, the consistency and validity of the research.

5.1 ANSWERING THE RESEARCH QUESTIONS

The four appended papers in this thesis contribute to answering the two research questions, posed in Section 1.2.3, differently. The papers are mapped according to their contribution to the research questions in Table 3. The research questions are answered and discussed below.

Table 3. The four appended papers, A, B, C, and D,
are distributed according to their contribution to each of the two research questions

	Paper			
Research Question	A	B	C	D
RQ 1) What main challenges in the development of complex products exist, and what approaches, models, methods and tools are suitable for meeting these challenges to achieve efficient interdisciplinary platform development	⬤	●		●
RQ 2) How can interdisciplinary platforms be developed and by whom, to ensure the efficient interchange of information among and across disciplines and systems and what are the implications of such implementation?		⬤	⬤	•

RQ 1) What main challenges in the development of complex products exist, and what approaches, models, methods and tools are suitable for meeting these challenges to achieve efficient interdisciplinary platform development

The main challenges in the development of complex products are deduced from both literature and the four industrial case studies addressed in this thesis. This research takes the position that a wider range of customers can be served by a family of products, as opposed to a single product. Such an approach would serve the mass customization paradigm.

Developing a single product is different from developing a family of products, advised through platform development. The existing challenges of developing one product therefore increase when developing a family of product, at least initially, before the platform is created. The main challenges and suitable approaches, models, methods and tools to counteract them are provided below.

Approaches

First, instead of developing a single product, with the preferred functionality and performance expected by the customer, platforms need to accommodate the functionality and performance that fulfill a range of needs from different customers. In this way, platforms serve the mass customization paradigm. However, compared to the development of a single complex product, they come with the aggregated complexity of multiple products in a family. One product variant generated from a platform is not like any other variant generated. However, all variants can be built up by common elements, accommodated by the platform. The platform elements chosen that build up the different variants mostly depend on the description of the elements, and how these variants are configured into distinctive products. The way platforms and its elements are structured is therefore fundamental for the use of a platform, and how well different customer needs are met.

Second, in platform development, increased attention to the product lifecycle needs to be addressed, for example to assure producibility of a product. Bear in mind that simply to ensure producibility of a single product is time-consuming, due to the many iteration processes of prototyping and testing. Thus, ensuring producibility of a whole family of products is overwhelming. Early producibility assessments of a product family are therefore even more burdensome than ensuring producibility of a single product. However, by making use of knowledge about manufacturing systems and their inherent limitations, functionality and performance early on in the development process, this challenge can be met. Even making manufacturing part of the platform, rather than addressing it for the entire product family independently, would benefit the increased and needed integration of the two technical systems in platform development.

Third, too scant information about all elements of a product are typically available in the early phases of development for design engineers to make credible design decisions. This can be partly due to the organizational structure, which typically is separated by product functions, but also to the inefficient interchange of information among and across disciplines and systems. Design reuse, as a way of building new designs upon prior design knowledge, would not only increase the efficiency of product development, but also platform development, such as integrated platforms of products and manufacturing systems. To achieve design reuse across disciplines with sufficient interchange of information, the use of interdisciplinary platforms is advised. The scope of interdisciplinary platforms in this thesis involves all functions regarding technical systems in product development, including manufacturing systems and their processes.

Models and Methods

The increased integration of interrelated systems, which characterizes complex products and platforms, needs to be modeled to accommodate the range of different customer needs, to form products that fit the needs of individual customers. There are two fundamentally different ways of achieving design reuse to serve the process efficiency while still configuring a product family. The first one advocates design reuse through modules as physical parts, thus through modularization. The second one supports design reuse through elements, or system objects, equipped with explicit functionality, design solutions, and non-functional requirements. Design reuse through modules as physical parts uses standardized interfaces, which make them interchangeable among product variants. As opposed to designing a single product architecture as advised through single product development, platforms need modules that are more flexible than physical parts to manage the increased complexity of accommodating multiple architectures, advised through platforms. Therefore, there is a need for increased flexibility of the constituent elements of architectures in serving a range of different customer needs. This increased flexibility of architectural elements, to serve multiple architectures, is advised through the *configurable component* (CC) concept. A *configurable component*, as a system object, is designed to hold a system family with certain functionality, design solutions and requirements – its *DNA*. Following the CC concept, the *DNA* is modeled using *enhanced function-means* (EF-M) modeling, which is a method used to primarily capture why a design exists and the justifications for its existence. Such information has proven efficient in single product development, and is equally, or increasingly, important when developing a product family. Also, a function can easily be reused, whereas the way in which a function can be solved will evolve. Such design reuse flexibility can increase the commonality among products in a family to achieve scale benefits in development. Therefore, EF-M modeling is a suitable method for interdisciplinary platform development.

Regarding justifications of why a design exists, providing information about design alternatives and trade-offs, there are several methods that are suitable for interdisciplinary platform development. As advised through the CC concept, a range of customer needs can be met by the modular and the scalable bandwidth. Based on two different customer preferences, a design solution suitable for the first customer may very well differ for the second customer. Also, the range of sizes of a design can be provided based on the same principle. The bandwidth of the platform is fundamental for its scope and for the generation of a product family, as opposed to a single product. The bandwidth can be used to generate multiple architectural options through the instantiation of modules and the full product family through configuration. Also, the system objects all have adaptable interfaces, as opposed to standardized interfaces, which means that the interacting systems can adjust to each other better than a platform based on physical modules with standardized interfaces.

An apparent challenge in single product development is the lack of information in the early phases of development. The design is immature and therefore the design decisions are often poorly executed. Creating sufficient information about immature designs among a family of products and eliminating them in accordance with their feasibility is an even tougher challenge. With the use of interactions modeled across objects in an EF-M tree, *design structure matrices* (DSMs) and matrices for *axiomatic coupling analyses* can be generated for a family of systems and be used as decision support for immature designs. This is primarily viable without having a 3D model in this pre-embodiment stage.

To serve increased flexibility of the product architecture, necessary to develop a family of products rather than a single product, the models and methods described above may very well be suitable. These models and methods can provide support during platform development, with the flexibility needed to accommodate many generations of products.

Tools

Manufacturing companies, developing a single product, can make use of tools to store information (PDM systems), interchange information (PLM systems), model 3D shapes (CAD systems), and simulate such items as functionality and performance (CAE systems). These tools have not been developed to support the development of a product family, at least not without profound manual interventions. There are no tools for the proper creation of platforms to serve the development of a family of products as opposed to a single product. Nor are there any tools to sufficiently evaluate multiple product architectures, without having to create a 3D shape, or drawings. Rather, available tools are very powerful in creating a 3D shape, and multi-disciplinary optimization loops can be applied to find ideal shapes, or architectures, with defined functionality and performance. However, before 3D shapes or drawings are created, in a pre-embodiment stage when the designs are quite immature, tools to create proper basis for design decisions supportive of evaluation are lacking.

Also, the current tools often fail to prove sufficient bases when combining producibility and performance analysis. When developing a family of products, conceptual 3D models with increased adaptability, such as through parameterization, in combination with the design reuse flexibility of the system objects advised above, producibility aspects can be applied early to eliminate bad design alternatives while keeping the good alternatives.

RQ 2) How can interdisciplinary platforms be developed and by whom, to ensure the efficient interchange of information among and across disciplines and systems and what are the implications of such implementation?

Interdisciplinary platforms can support design reuse across disciplines, such as product and manufacturing, to generate a family of functional architectures (Case ‘a’), system architectures (Case ‘b’), part architectures (Case ‘c’), and producible product variants (Case ‘d’). This is advised through the use of a platform development methodology comprising two main processes: *platform preparation* and *platform execution*. Also, a software system that reflects the steps of the methodology has been developed to enable use of the methodology in practice. The methodology reflects and builds on the existing theoretical models, methods and tools addressed in *RQ 1*. The methodology serves three development levels, reflecting the level of design detail within the product family: 1) *functional level;* functions and alternative ways of solving them, 2) *system level:* system objects with design parameters, and 3) *detailed level:* conceptual 3D shapes. The three levels can be used iteratively, as the platform matures throughout the development process. In this way several design alternatives can be evaluated as they mature. The design space is narrowed down step-wise through the principles of SBCE – *map the design space*, *integrate by intersection* and *establish feasibility before commitment*. The knowledge gained can be then be reused to continuously support platform development; see Figure 28.

Functional Level

The functional level primarily comprises platform preparation steps *(1) EF-M modeling*, and *(2) modular bandwidth definition*. The modeled interactions between FRs, alternative DSs and Cs, define the functional system. The steps in the platform execution process are used to support the instantiation and configuration of functional variants, thus generating a function family. All variants based on the *DNA* can be compared and evaluated based on analyses such as using DSMs and matrices for axiomatic couplings as reviewed in Section 2.4.6. The evaluation of the functional level can be used to eliminate unfeasible design alternatives, thus narrowing down the design space based on reusing design. At the *functional level*, expert knowledge about designs is necessary, which is why design engineers are suitable users.

System Level

The system level is based on system descriptions of designs. CC objects, together with the defined *DNA*, constitute the total system. The *system level* comprises steps *(3) CC partitioning*, and *(4) CC and scalable bandwidth definition* of the methodology. The steps in the platform preparation process are used to develop reusable system objects and their interactions, which represent the system architecture. The steps during the platform execution process are used to support instantiation and configuration of system variants, thus creating a system family. All variants in the system family can be compared and evaluated based on analyses such as using DSMs, axiomatic couplings, as well as trade-off curves. The evaluation at the system level can be used to support elimination of unfeasible architectural options, thus further narrowing down the design space. Due to the holistic and architectural nature of the *system level*, system architects make suitable users.

Detailed Level

The detailed level comprises the CAx steps: *(A) the conceptual parameterized geometry modeling*, and *(B) the integration of CAx software*. The variants in the part family can be compared and evaluated based on CAE analyses. These analyses typically require a lot of time and effort to generate valuable data as a basis for evaluation, especially for multi-disciplinary optimization algorithms where several different CAE tools are involved. The evaluation, based on the detailed level, can be used to support communication of feasible regions using for example trade-off curves.

The detailed level can be applied to carry out a finer analysis by creating a richer knowledge base for decisions, and complement the less time-consuming analyses at the *system level* with geometry and performance specific analyses. The models are represented as parts, assemblies or both. The steps in the platform execution process are used to support the instantiation and configuration of part and assembly variants, thus the part family. The *detailed level* includes physical models with detailed features, which is why design engineers are the most suitable users.

The Process

The platform preparation process provides design engineers and system architects with processes and methods necessary to model interdisciplinary platforms, including functional and non-functional requirements, alternative designs solutions and systems. Although it may seem as though there is a strict sequence of the steps defined for the methodology proposed, this is not the case. As for any design process, it is truly iterative. The methodology is also selective, meaning that with respect to the maturity of the platform model and the purpose of the development of the platform, an appropriate selection of steps can be derived based on the three levels described above. The system objects, described in Section 2.4.3, support design reuse during the product lifecycle all the way from customer needs through production.

The platform execution is not intended as a means for complete design automation, but rather the process requires users to carefully engage by making design decisions throughout the platform development process. However, it may allow for less keyboard mashing and less design modifications when the platform has been created. Through the design reuse described above, it may also allow for scale benefits during development.

An interdisciplinary platform is not only a means of supplying a family of products, but also a family of manufacturing equipment and tools, which will be accounted for in future work.

5.2 EVALUATION OF RESEARCH

The research results presented in this thesis are used as a springboard towards the dissertation. Therefore, it is important to evaluate and verify the research results through *acceptance* and *logic*, as referred to in Section 3.3.4, to prove research reliability. It is equally important to evaluate the research approach that led to the research results.

5.2.1 Verifying Research Results

Verification by acceptance means that experts within the field accept the new academic contributions. Papers A and D are both conference papers that have been peer reviewed by experts, accepted for publication and presented at the conferences. Papers B and C are both submitted to journals, but have not yet been peer reviewed. The results have also been presented at the Wingquist Laboratory, where companies have shown their interest in the implementation possibilities.

Logical verification concerns research completeness and its consistency. The results show completeness if they fit into established theory, and consistency if the terminology is clear and conforming. Acknowledged theoretical approaches, as well as established models, methods and tools from primarily engineering design and systems engineering strengthen the research results presented.

For Paper A, interviewees were invited to workshops to discuss the results. Papers B, C, and D have been conducted in collaboration with an industrial company. Through this, internal consistency has been secured. Regarding the terminology, some might be partly new; such as *interdisciplinary platforms*, as to define the interaction between the two technical systems of products and manufacturing systems. Interdisciplinary platforms are defined in Section 1.2.2.

5.2.2 Evaluating Research Approach

The evaluation of the research presented in this thesis is based on the quality of studies performed, as well as the reliability of research results. Design research relies upon the application of rigorous methods in both the construction and evaluation of a design model (von Alan et al., 2004). The long-term collaboration with the companies provides a viable basis for further studies and emphasizes the merging of academic and industrial needs.

Paper A poses an overview of industrial challenges. Ten semi-structured interviews have been conducted with senior-level engineers and managers, with over 30 years of experience in the business, from a wide variety of disciplines, including market, product planning, design, production, sales and after market. Papers B, C and D, dig more deeply into the interdisciplinary challenges. Several unstructured interviews, as well as observations, and workshops in collaboration with design engineers and production engineers, have been conducted at GKN Aerospace Sweden AB. To further understand design compromises between product and manufacturing, observations has been continuously conducted in development facilities and in production facilities at GKN. Because of the action research approach, there is a risk for researcher bias. Biases have been somewhat excluded through verification by acceptance from experts within the companies. The results related to suitable users and how these can use the proposed methodology and software system to develop interdisciplinary platforms have been synthesized through literature reviews and collaborative work with professionals from the three industrial companies studied in this thesis.

6

"The balance of benefits and dangers from scientific and technological advance, and other forms of social change, is imponderable. We may need quite often to be bold rather than cautious in supporting scientific innovation or other forms of change."

– Anthony Giddens

CONCLUSION

This chapter provides you with a brief summary of the findings and their reliability and validity. The scientific value and industrial relevance of this research are ultimately proposed. It also points to the future of platform-based development and provides you with new impetus for further research opportunities, as well as future directions within the scope of the scientific mission.

To meet increasing product complexity, and especially the complexity of managing a wide range of customer needs through a system family, development of platforms is advised. A platform development methodology, applied to achieve *mass customization* through *design reuse*, is proposed. Two methodological modes have been identified and addressed, *platform preparation* and *platform execution*. Platform preparation provides design engineers and system architects with the methods and tools needed to design and model platforms. Platform execution is used to generate new knowledge about a set of system variants for design engineers and systems architects to analyze and evaluate to make credible design decisions based on product performance and manufacturing capabilities.

The methodology supports three development levels, reflecting the maturity of the platform model: 1) the *functional level;* enhanced function-means modeling – functional and non-functional requirements, as well as alternative design solutions and interactions 2) the *system level;* modeling reusable and scalable system objects, and 3) the *detailed level;* modeling CAx models, and linking them together through PLM architectures. To ensure the efficient interchange of information through the three development levels, a common platform shared by disciplines is advised – *interdisciplinary platforms*.

Through the approach presented, design reuse can be achieved, and new knowledge can be created to support evaluation of multiple design alternatives. By providing such design knowledge about products and manufacturing systems concurrently, design engineers and system architects can make credible design decisions early during the development process by striking a balance between performance and capabilities.

By developing interdisciplinary platforms, the co-development of products and manufacturing systems have proven feasible to ensure producible product variants during the conceptual phases of development. The platform methodology and the accompanying software support the interchange of information in the early phases of development, such as during product planning, conceptual design, and preliminary detailed design phase.

6.1 Future Work

Future work is aimed at improving the concurrent development of products and manufacturing systems even further. By including manufacturing planning of processes and resources as an integral part of the platform, the conceptual design of manufacturing equipment and tools may be achieved. Employing interdisciplinary platforms with well-defined interfaces enable uniform interchange of information among and across disciplines and systems. By these means, manufacturing companies may further increase their responsiveness to changing requirements with the result that less design modifications are needed; see the vision in Figure 36. This approach would serve the mass customization paradigm towards more practical implementations in manufacturing companies, as put forward by Ferguson et al. (2013).

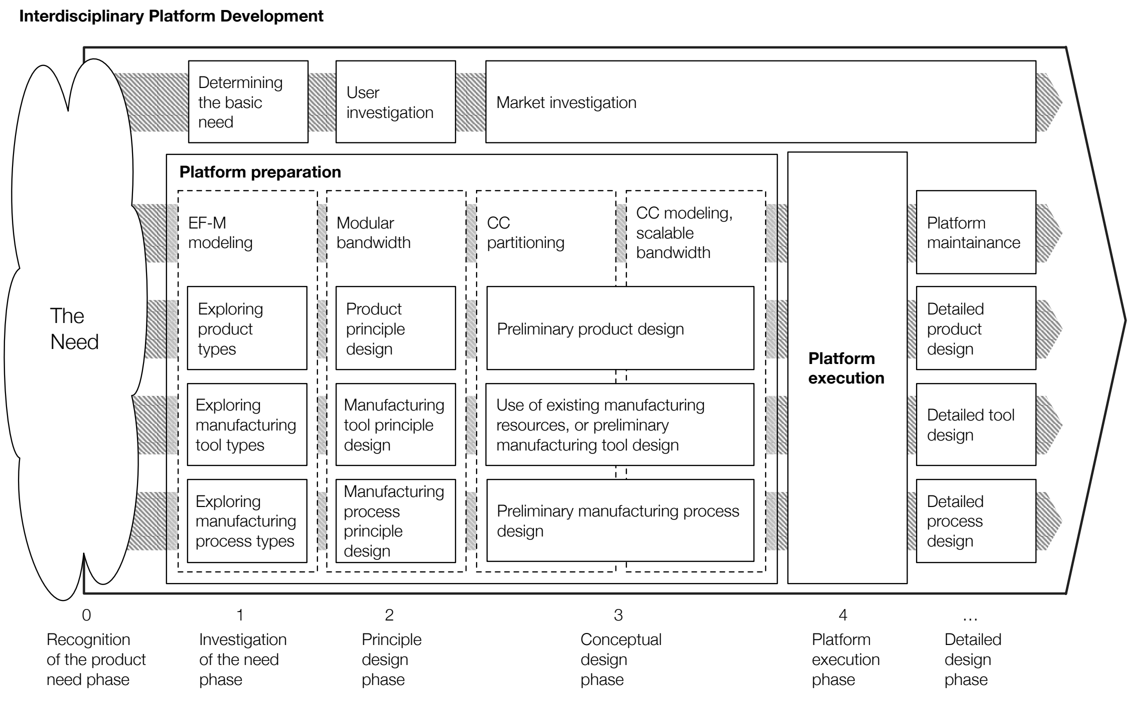

Figure 36. Potential of Interdisciplinary Platform Development
(inspired by Andreasen and Hein (1987), from Figure 9)

"Just as good product engineering involves up-front consideration of manufacturing issues, good platform planning requires up-front consideration of design and manufacturing issues."

– Robertson and Ulrich (1998)

REFERENCES

Abramovici, M. (2002). Status and Development Trends of Product Lifecycle Management Systems. *International Conference on Integrated Product and Process Development.* Wroclaw, Poland.

Abramovici, M. (2007). Future trends in product lifecycle management (PLM). *The future of product development.* Springer.

Alblas, A. & Wortmann, H. (2009). The need for function platforms in engineer-to-order industries. *17th International Conference on Engineering Design, ICED 09.* Palo Alto, CA, USA.

Almefelt, L. (2005). *Requirements-Driven Product Innovation-Methods and Tools Reflecting Industrial Needs.* Dissertation, Chalmers University of Technology.

Andersson, F. (2003). *The Dynamics of Requirements and Product Concept Management A Product Modelling Approach.* Dissertation, Chalmers University of Technology.

Andreasen, M. M. (1980). *Syntesemetoder pa Systemgrundlag - Bidrag til en konstruktionsteori.* Dissertation, Lund University.

Andreasen, M. M. (1991). The theory of domains. *Workshop on understanding function and function to form evolution.* Cambridge University, UK.

Andreasen, M. M. & Hein, L. (1987). *Integrated product development,* New York, USA, Springer.

Baldwin, C. & Clark, K. (1997). Managing in an Age of Modularity. *Harvard business review,* 75**,** 84-93.

Baldwin, C. Y. & Clark, K. B. (2000). *Design rules: The power of modularity*, MIT press.

Berglund, F. & Claesson, A. (2005). Utilising the Concept of a Design's Bandwidth to Achieve Product Platform Effectiveness. *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy.* Engineers Australia.

Bernstein, J. I. (1998). *Design Methods in the Aerospace Industry: Looking for Evidence of Set-based Practices.* Dissertation, Massachusetts Institute of Technology.

Bhise, V. D. (2013). *Designing complex products with systems engineering processes and techniques*, CRC Press.

Blanchard, B. S. & Fabrycky, W. J. (1990). *Systems Engineering and Analysis,* Englewood Cliffs, New Jersey, Prentice-Hall, Inc.

Blessing, L. T. & Chakrabarti, A. (2009). *DRM: A Design Reseach Methodology*, Springer.

Boothroyd, G. (1994). Product design for manufacture and assembly. *Computer-Aided Design,* 26**,** 505-520.

Brière-Côté, A., Rivest, L. & Desrochers, A. (2010). Adaptive generic product structure modelling for design reuse in engineer-to-order products. *Computers in Industry,* 61**,** 53-65.

Bryman, A. & Bell, E. (2007). *Business Research Methods*, Oxford University Press.

Burr, H., Vielhaber, M., Deubel, T., Weber, C. & Haasis, S. (2008). CAx/EDM integration: enabler for methodical benefits in the design process.

Buur, J. (1990). *A theoretical approach to mechatronics design.* Dissertation, Technical University of Denmark.

Checkland, P. (1981). Systems thinking, systems practice.

Chisholm, R. F. & Elden, M. (1993). Features of emerging action research. *Human Relations,* 46**,** 275-298.

Claesson, A. (2006). *A Configurable Component Framework Supporting Platform-Based Product Development.* Dissertation, Chalmers University of Technology.

Clarkson, P. J., Simons, C. & Eckert, C. (2004). Predicting change propagation in complex design. *Journal of Mechanical Design,* 126**,** 788-797.

Crawley, E., De Weck, O., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D. & Whitney, D. (2004). The influence of architecture in engineering systems.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*, Sage publications.

Duffy, A. H. B. & Andreasen, M. M. (1995). Enhancing the Evolution of Design Science. *International Conference on Engineering Design ICED 95.* Prague, Czech Republic.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology,* 61**,** 793-814.

Eppinger, S. D. & Browning, T. R. (2012). *Design structure matrix methods and applications*, MIT press.

Ericsson, A. & Erixon, G. (1999). *Controlling design variants: modular product platforms*, Society of Manufacturing Engineers.

Erixon, G. (1998). *Modular Function Deployment - A Method for Product Modularisation.* Dissertation, The Royal Institute of Technology.

Erixon, G., von Yxkull, A. & Arnstroem, A. (1996). Modularity–the basis for product and factory reengineering. *CIRP Annals-Manufacturing Technology,* 45**,** 1-6.

European Commission Climate Action (2015). Climate Change. In: Secondary European Commission Climate Action (ed.) Secondary Climate Change. DOI: 10.2834/34323.

Ferguson, S. M., Olewnik, A. T. & Cormier, P. (2013). A review of mass customization across marketing, engineering and distribution domains toward development of a process framework. *Research in Engineering Design,* 25**,** 11-30.

Fixson, S. K. (2007). Modularity and commonality research: past developments and future opportunities. *Concurrent Engineering,* 15**,** 85-111.

Fogliatto, F. S., da Silveira, G. J. C. & Borenstein, D. (2012). The mass customization decade: An updated review of the literature. *International Journal of Production Economics,* 138**,** 14-25.

Gaines, B. R. & Shaw, M. L. (1992). Integrated knowledge acquisition architectures. *Journal of Intelligent Information Systems,* 1**,** 9-34.

Gedell, S. (2011). *Efficient Means for Platform-Based Development–Emphasizing Integrated, Information-Rich System Models.* Dissertation, Chalmers University of Technology.

Gedell, S., Claesson, A. & Johannesson, H. (2011). Integrated Product and Production Model–Issues on Completeness, Consistency and Compatibility. *International Conference on Engineering Design, ICED'11.* Copenhagen, Denmark.

Gershenson, J. K., Prasad, G. J. & Zhang, Y. (2004). Product modularity: measures and design methods. *Journal of Engineering Design,* 15**,** 33-51.

Gonzalez-Zugasti, J. P. & Otto, K. N. (2000). Modular platform-based product family design. *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Baltimore, Maryland.

Gu, P., Hashemian, M. & Nee, A. Y. C. (2004). Adaptable Design. *CIRP Annals - Manufacturing Technology,* 53**,** 539-557.

Gu, P. & Sosale, S. (1999). Product modularization for life cycle engineering. *Robotics and Computer-Integrated Manufacturing,* 15**,** 387-401.

Hadley, J. R. & McCarthy, D. J. (2011). Producibility and confidence indices during defense acquisition. *NDIA Ground Vehicle Systems Engineering and Technology Symposium.*

Dearborn, Michigan.
Halman, J. I., Hofer, A. P. & Van Vuuren, W. (2003). Platform-driven development of product families: linking theory with practice. *Journal of Product Innovation Management,* 20**,** 149-162.
Harwell, R., Aslaksen, E., Mengot, R., Hooks, I. & Ptack, K. (1993). What is a Requirement? *INCOSE International Symposium.* Crystal City, VA.
Henke, J. W., Krachenberg, A. R. & Lyons, T. F. (1993). Cross-Functional Teams: Good Concept, Poor Implementation! *Journal of Product Innovation Management,* 10**,** 216-229.
Hitchins, D. K. (2003). *Advanced systems thinking, engineering, and management*, Artech House.
Hood, C., Wiedemann, S., Fichtinger, S. & Pautz, U. (2007). *Requirements management: The interface between requirements development and all other systems engineering processes*, Springer Science & Business Media.
Horvath, I. (2001). A contemporary survey of scientific research into engineering design. *Proceedings of the International Conference on Engineering Design, ICED 01.* Glasgow, Scotland.
Hou, S., He, L., Xie, H. & Liu, Y. (2011). Research on product family modeling method for design resource reuse. *Computing and Intelligent Systems.* Springer.
Hubka, V. & Eder, W. E. (1988). *Theory of technical systems: a total concept theory for engineering design,* Berlin, Germany, Springer.
Hvam, L., Mortensen, N. H. & Riis, J. (2008). *Product customization*, Springer Science & Business Media.
Jiao, J., Ma, Q. & Tseng, M. M. (2003). Towards high value-added products and services: mass customization and beyond. *Technovation,* 23**,** 809-821.
Jiao, J. R., Simpson, T. W. & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing,* 18**,** 5-29.
Johannesson, H. & Claesson, A. (2005). Systematic product platform design: a combined function-means and parametric modeling approach. *Journal of Engineering Design,* 16**,** 25-43.
Jose, A. & Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent manufacturing,* 16**,** 371-390.
Jørgensen, K. (1992). Videnskabelige arbejdsparadigmer. *Institute for Production, Aalborg University, Denmark of Technology*.
Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. & Van Brussel, H. (1999). Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology,* 48**,** 527-540.
Koren, Y. & Shpitalni, M. (2010). Design of reconfigurable manufacturing systems. *Journal of manufacturing systems,* 29**,** 130-141.
Kratochvíl, M. & Carson, C. (2005). *Growing modular: mass customization of complex products, services and software*, Springer Science & Business Media.
La Rocca, G. & Van Tooren, M. (2007). Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. *Journal of Design Research,* 5**,** 333-352.
Lee, J. (1997). Design rationale systems: understanding the issues. *IEEE intelligent systems***,** 78-85.
Levandowski, C. (2014). *Platform Lifecycle Support using Set-Based Concurrent Engineering.* Dissertation, Chalmers University of Technology.

Loureiro, G. (1999). *A systems engineering and concurrent engineering framework for the integrated development of complex products.* © Geilson Loureiro.
Madni, A. M. (2012). Adaptable platform-based engineering: Key enablers and outlook for the future. *Systems Engineering,* 15**,** 95-107.
Malak, R. J., Aughenbaugh, J. M. & Paredis, C. J. (2009). Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design,* 41**,** 214-227.
Meyer, M. H. & Lehnerd, A. (1997). *The Power of Product Platforms: Building Value and Cost Leadership,* New York, Free Press.
Michaelis, M. T. (2013). *Co-Development of Product and Manufacturing Systems Using Integrated Platform Models.* Dissertation, Chalmers University of Technology.
Michaelis, M. T. & Johannesson, H. (2011). Platform Approaches in Manufacturing - Considering Integration with Product Platforms. *Proceedings of ASME DETC.* Washington D.C.
Michaelis, M. T., Levandowski, C. & Johannesson, H. (2013). Set-based concurrent engineering for preserving design bandwidth in product and manufacturing system platforms. *ASME 2013 International Mechanical Engineering Congress and Exposition.* San Diego, USA: American Society of Mechanical Engineers.
Mortensen, N. H. (1999). *Design Modelling in a Designer's Workbench: Contribution to a Design Language.* Dissertation, Technical University of Denmark.
Muffatto, M. & Roveda, M. (2000). Developing product platforms:: analysis of the development process. *Technovation,* 20**,** 617-630.
Ong, S. K., Nee, A. Y. C. & Xu, Q. L. (2008). *Design Reuse in Product Development Modeling, Analysis and Optimization*, World Scientific Publishing Company.
Pahl, G. & Beitz, W. (2013). *Engineering Design: A Systematic Approach*, Springer Science & Business Media.
Parker, G. M. (2003). *Cross-functional teams: Working with allies, enemies, and other strangers*, John Wiley & Sons.
Persson, M. & Åhlström, P. (2006). Managerial Issues in Modularising Complex Products. *Technovation,* 26**,** pp. 1201-1209.
Pimmler, T. U. & Eppinger, S. D. (1994). Integration analysis of product decompositions.
Pine, B. J. (1993). *Mass customization: the new frontier in business competition*, Harvard Business Press.
Pirmoradi, Z., Wang, G. G. & Simpson, T. W. (2014). A Review of Recent Literature in Product Family Design and Platform-Based Product Development. 1-46.
Prencipe, A. (1998). Modular design and complex product systems: facts, promises, and questions. *Economic and social research council, CoPS Publication,* 47.
Raudberget, D., Levandowski, C., Isaksson, O., Kipouros, T., Johannesson, H. & Clarkson, J. (2015). Modelling and Assessing Platform Architectures in Pre-Embodiment Phases through Set-Based Evaluation and Change Propagation. *Journal of Aerospace Operations*.
Reeves, P., Tuck, C. & Hague, R. (2011). Additive manufacturing for mass customization. *Mass Customization.* Springer.
Reinertsen, D. (1997). *Managing the Design Factory,* New York, Free Press.
Robertson, D. & Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review,* 39**,** 19-31.
Schachinger, P. & Johannesson, H. L. (2000). Computer modelling of design specifications. *Journal of engineering design,* 11**,** 317-329.
Shahin, T., Andrews, P. & Sivaloganathan, S. (1999). A design reuse system. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,*

213, 621-627.
Sheriff, A. (1998). The evolving product platform in FIAT Auto. *5th International Product Development Management Conference.* Como, Italy.
Simpson, T. W. (1998). *A concept exploration method for product family design.* Citeseer.
Simpson, T. W. (2004). Product platform design and customization: Status and promise. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 18**,** 3-20.
Simpson, T. W., Maier, J. R. & Mistree, F. (2001). Product platform design: method and application. *Research in engineering Design,* 13**,** 2-22.
Simpson, T. W., Siddique, Z. & Jiao, R. J. (2006). *Product platform and product family design: methods and applications*, Springer Science & Business Media.
Sobek, D. K., Ward, A. C. & Liker, J. K. (1999). Toyota's principles of set-based concurrent engineering. *Sloan management review,* 40**,** 67-84.
Sosa, M. E., Eppinger, S. D. & Rowles, C. M. (2007). A Network Approach to Define Modularity of Components in Complex Products. *Journal of Mechanical Design,* 129**,** 1118.
Stark, J. (2015). *Product lifecycle management*, Springer.
Stevens, R. (1998). *Systems engineering: coping with complexity*, Pearson Education.
Strawbridge, Z., McAdams, D. A. & Stone, R. B. (2002). A computational approach to conceptual design. *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Montreal, Canada.
Suh, N. P. (1990). *The principles of design*, Oxford University Press New York.
Svensson, D., Malmström, J., Pikosz, P. & Malmqvist, J. (1999). A framework for modelling and analysis of engineering information management systems. *Proceedings of ASME DETC'99*.
Swink, M. L., Sandvig, J. C. & Mabert, V. A. (1996). Customizing concurrent engineering processes: five case studies. *Journal of Product Innovation Management,* 13**,** 229-244.
Terzi, S., Bouras, A., Dutta, D., Garetti, M. & Kiritsis, D. (2010). Product lifecycle management-from its history to its new role. *International Journal of Product Lifecycle Management,* 4**,** 360-389.
Tjalve, E. (1976). Systematisk Udformning af Industriprodukter—Værktøjer for konstruktøren [Systematic Design of Industrial Products—Tools for the Design Engineer]. *København: Akademisk Forlag*.
Tseng, M. M. & Hu, S. J. (2014). Mass customization. *Cirp Encyclopedia of Production Engineering.* Springer.
Tseng, M. M., Jiao, J. & Merchant, M. E. (1996). Design for Mass Customization. *CIRP Annals - Manufacturing Technology,* 45**,** 153-156.
Ulrich, K. (1995). The Role of Product Architecture in the Manufacturing Firm. *Research policy,* 24**,** 419-440.
Ulrich, K. T. & Eppinger, S. D. (2012). *Product Design and Development,* Boston, Massachusetts, USA, McGraw-Hill Education.
Vallhagen, J., Madrid, J., Söderberg, R. & Wärmefjord, K. (2013). An approach for producibility and DFM-methodology in aerospace engine component development. *Procedia CIRP,* 11**,** 151-156.
von Alan, R. H., March, S. T., Park, J. & Ram, S. (2004). Design science in information systems research. *MIS quarterly,* 28**,** 75-105.
Wahl, A. L. & Johannesson, H. (Year) Published. Managing Design Change in Configurable Component Based Product Platforms. 8th Biannual Conference NordDesign2010, 2010

Gothenburg, Sweden.
Ward, A. C. & Sobek II, D. K. (2014). *Lean product and process development*, Lean Enterprise Institute.
Weber, R. G. & Condoor, S. S. (1998). Conceptual design using a synergistically compatible morphological matrix. *28th Annual Frontiers in Education Conference FIE'98.* Tempe, AZ, USA.
Wheelwright, S. C. & Clark, K. B. (1992). *Revolutionizing product development: quantum leaps in speed, efficiency, and quality*, Simon and Schuster.
Wärmefjord, K., Söderberg, R. & Lindkvist, L. (2014). Form Division for Welded Aero Components in Platform-Based Development. *Journal of Aerospace Engineering*.
Xu, Q. L., Ong, S. K. & Nee, A. Y. C. (2006). Function-based design synthesis approach to design reuse. *Research in Engineering Design,* 17**,** 27-44.
Yin, R. K. (2003). *Case study research: Design and methods,* Thousand Oaks, CA, USA, SAGE publications.
Zimdars, L. (2003). The world of cross-functional teams. *Crossfunctional teams: Working with aliens, enemies, and other strangers***,** 1-11.
Zipkin, P. (2001). The limits of mass customization. *MIT Sloan Management Review,* 42**,** 81-87.
Zweber, J. V., Blair, M., Kamhawi, H., Bharatram, G. & Hartong, A. (1998). *Structural and manufacturing analysis of a wing using the adaptive modeling language*, Defense Technical Information Center.
Ölvander, J., Lundén, B. & Gavel, H. (2009). A computerized optimization framework for the morphological matrix applied to aircraft conceptual design. *Computer-Aided Design,* 41**,** 187-196.