THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Platform Lifecycle Support using Set-Based Concurrent Engineering

CHRISTOFFER E. LEVANDOWSKI

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2014

Platform Lifecycle Support using Set-Based
Concurrent Engineering
CHRISTOFFER E. LEVANDOWSKI
ISBN 978-91-7597-084-4

© CHRISTOFFER E. LEVANDOWSKI, 2014

Doktorsavhandling vid Chalmers tekniska högskola
Ny serie nr. 3765
ISSN 0346-718X

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 graphically portrays the philosophy of Systems Theory. It shows a system of systems with input and several outputs, controls and mechanisms, and how it all inevetably affect each other. It portrays both the solutions and problems adressed in this thesis.

Printed by
Chalmers Reproservice
Gothenburg, Sweden 2014

Abstract

Product development companies strive to provide their customers with high quality products, more quickly than their competitors, using as few resources as possible. One way of managing all three aspects at the same time is to reuse old, quality assured designs and knowledge in new products. A common way to do that is to create a platform with designs that are reusable in many different products.

Traditionally, research on platforms has focused on finding ways to provide manufacturing with a low number of parts to be able to increase utilization of expensive production equipment. However, reuse of parts does not benefit all businesses, especially those where customer requirements continuously change. To *cut development lead-time*, other types of reuse are necessary. The use of platforms based on core technologies and re-configurable systems as platform elements may provide the necessary support. They enable reuse on a more abstract level, reusing technologies, requirements and concepts rather than ready designed parts. This thesis elaborates on support for working with the type of platforms that are integrated across the lifecycle of a product.

The studies in this thesis show that platform approaches in literature today do not cover the need to support holistic platform development across all stages of a lifecycle. As a solution, configurable system elements are used to model platforms and the links between the lifecycles. The development processes and models may be further infused with *set-based concurrent engineering* to provide a framework for efficient development. These principles are integrated into the models and the processes to enhance the ability to manage the complex relationships within and between parts of the platform throughout the lifecycle.

Further, development platforms may be supported by a Product Lifecycle Management (PLM) architecture for engineering-to-order configuration, but it can also serve as a tool to learn about the knowledge gaps that need to be filled to get a product that meets requirements.

Keywords: product development, platform-based development, set-based concurrent engineering, product lifecycle management, configurable components.

Acknowledgements

This thesis was carried out at the department of Product and Production Development, Chalmers University of Technology in Gothenburg, Sweden. I would like to express my sincere thanks to my supervisor Professor Hans Johannesson who gave me the opportunity to pursue my own ideas, providing me with guidance when I needed it. Also, thank you Professor Johan Malmqvist, my co-supervisor, for your eye-opening comments and for posing the difficult questions.

I would like to thank my co-authors Marcel Michaelis, Dag Raudberget, Anders Forslund, Dag Bergsjö and Daniel Corin Stig for the moments of inspiration and creativity and for sharing the moments of pure pain. To Roger Jiao at Georgia Institute of Technology and Kevin Otto at Singapore University of Technology and Design, thank you for accommodating me and letting me propel my research through your guidance.

Funding for these visits, and my research in general comes from the Wingquist Laboratory VINN Excellence Centre under the umbrella of the Production Area of Advance at Chalmers. The Centre is supported by VINNOVA, the Swedish Governmental Agency for Innovation Systems. This support is greatly appreciated.

I would further like to direct my thanks to the people from our industrial partners, which I have had the pleasure to make acquaintances with during our studies. Thank you everybody at GKN Aerospace, Volvo Cars and Emerson Rosemount Tank Radar who has participated in our studies and assisted in making them happen, especially Ola Isaksson who helped to provide intriguing spot on cases for testing my work.

The time at PPU would not have been the same without my lovely colleagues. The quirky, intelligent, and inspiring discussions during coffee breaks have made it easier to get up in the morning. I am especially grateful to Anders Forslund, Marcel Michaelis and Ola Wagersten whom I have depended on heavily during the last few years. This would not have been possible without you.

I would like to thank my parents, siblings, my old friends and my new friends for supporting me through happiness, agony, success and failure. A special thanks goes out to my grand parents who always believed in me. Finally, my most sincere thanks go to Helena for keeping me straight when I had the world on my shoulders.

Christoffer Levandowski
Gothenburg, September 2014

Appended Publications

The following research papers form the foundation on which this thesis stands.

Paper A. Levandowski, C., Edholm, P., Ekstedt, F., Carlsson, J., Söderberg, R. and Johannesson, H., 2011, "PLM Architecture for Optimization of Geometrical Interfaces in a Product Platform", *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference - ASME IDETC/CIE 2011*, Washington DC, USA.

Paper B. Levandowski, C., Forslund, A., Söderberg, R., and Johannesson, H., 2012 "Platform Strategies from a PLM Perspective - Theory and Practice for the Aerospace Industry", 53rd Structures, Structural Dynamics, and Materials and Co-located Conferences - AIAA/ASME, Honolulu, HI, USA.

Paper C. Levandowski, C., Corin Stig, D., Bergsjö, D., Forslund, A., Högman, U., Söderberg, R., and Johannesson, H., 2013, "An Integrated Approach to Technology Platform and Product Platform Development," *Concurrent Engineering*, 21(1), pp. 65-83.

Paper D. Levandowski, C., Forslund, A., Söderberg, R., and Johannesson, H., 2013, "Using PLM and Trade-Off Curves to Support Set-Based Convergence of Product Platforms", *International Conference on Engineering Design – ICED 2013,* Seoul, South Korea.

Paper E. Michaelis, M.T., Levandowski, C., and Johannesson, H., 2013 "Set-Based Concurrent Engineering for Preserving Design Bandwidth in Product and Manufacturing System Platforms", *Proceedings of ASME IMECE 2013*, Paper No. 63624, San Diego, CA USA.

Paper F. Levandowski, C., Michaelis, M.T., and Johannesson, H., 2014, "Set-Based Development Using an Integrated Product and Manufacturing System Platform", *Concurrent Engineering,* 22(3), pp. 234-252.

Paper G. Levandowski, C., Raudberget, D., and Johannesson, H., 2014, "Set-Based Concurrent Engineering for Early Phases in Platform Development", *International Conference on Concurrent Engineering - CE2014*, September 8-10, Beijing, China.

Paper H. Levandowski, C., Jiao, R., and Johannesson, H., 2014, "A Two-Stage Model of Adaptable Product Platform for Engineering-to-Order Configuration Design", *Submitted to Journal of Engineering Design*.

The work on each paper was distributed among the authors accordingly:

Paper A: Christoffer Levandowski wrote the paper and created the PLM system architecture and process. Fredrik Ekstedt created the robustness optimization. All authors contributed in creating the case scenario. Johan Carlson, Rikard Söderberg and Hans Johannesson contributed with comments and feedback.

Paper B: Christoffer Levandowski did the literature analysis, wrote the paper and created the PLM system architecture and process. Christoffer Levandowski and Anders Forslund created the case scenario. Rikard Söderberg and Hans Johannesson contributed with comment and feedback.

Paper C: Christoffer Levandowski, Dag Bergsjö, Daniel Corin Stig and Anders Forslund and wrote the paper. Christoffer Levandowski led the writing, synthesized the theory and created the system architecture. Christoffer Levandowski, Daniel Corin Stig and Dag Bergsjö set up the case and did the analysis. Christoffer Levandowski, Dag Bergsjö, Daniel Corin Stig, Ulf Högman and Anders Forslund contributed to the empirical data. Rikard Söderberg and Hans Johannesson contributed with comment and feedback.

Paper D: Christoffer Levandowski wrote the paper and synthesized the theory. Christoffer Levandowski and Anders Forslund elaborated the case and Anders Forslund performed the engineering analyses. Hans Johannesson and Rikard Söderberg contributed with comments and feedback.

Paper E: Christoffer Levandowski and Marcel Michaelis synthesized the theory and elaborated the example. They wrote, reviewed and edited the paper in close joint collaboration. Hans Johannesson contributed with comment and feedback.

Paper F: Christoffer Levandowski and Marcel Michaelis synthesized the theory and elaborated the example. Marcel Michaelis conducted the data collection for the industrial example. They wrote, reviewed and edited the paper in close joint collaboration. Hans Johannesson contributed with comments and feedback.

Paper G: Christoffer Levandowski and Dag Raudberget synthesized the theory and elaborated the example. They wrote, reviewed and edited the paper in joint collaboration. Hans Johannesson contributed with comments.

Paper H: Christoffer Levandowski wrote the paper and elaborated the example. Christoffer Levandowski and Jianxin (Roger) Jiao synthesized the theory. Jianxin (Roger) Jiao and Hans Johannesson contributed with comments and feedback.

Additional Publications

The following publications are related to the research presented in this thesis although not making a central contribution to the result.

Otto, K., Levandowski, C., Forslund A., Söderberg, R., and Johannesson, H., 2013, "Uncertainty Modeling to Enable Software Development Platforms that Can Automate Complex Mechanical Systems Design", *International Conference on Engineering Design – ICED 2013,* Seoul, South Korea.

Levandowski, C., Bokinge, M., Malmqvist, J., and Johannesson, H., 2012 "PLM as Support for Global Design Reuse - Long Term Benefits and Immediate Drawbacks", *9th International Conference on Product Lifecycle Management - PLM12*, Montreal, Canada.

Bokinge, M., Levandowski, C., Johannesson, H., and Malmqvist, J., 2012, "A Method to Identify Risks Associated with a PLM Solution", *The 9th International Conference on Product Lifecycle Management - PLM 12*, Montreal, Canada.

Forslund, A., Söderberg, R., Lööf, J. and Levandowski, C., 2012, "Robust Lifecycle Optimization of Turbine Components using Simulation Platforms", The *28th Congress of the International Council of the Aeronautical Sciences - ICAS 2012*, Brisbane, Australia.

Bokinge, M., Levandowski, C., and Tidstam, A., 2011, "PLM and International Product Development," Entering the Tiger's Cave, D. Bergsjö, ed., *Department of Product and Production Development, Chalmers University of Technology*, Gothenburg, Sweden, pp. 13-18.

Bengtsson, K., Michaelis, M. T., Levandowski, C., Lennartson, B., and Johannesson, H., 2010, "Towards Sequence Planning Based on Configurable Product and Manufacturing System Platforms," *Proceedings of the 8th International Conference - NordDesign 2010*, Gothenburg, Sweden.

Edholm, P., Levandowski, C., Johannesson, H., and Söderberg, R., 2010, "Applied CC configuration in PDM/CAD environment," *INTECH 2010*, Prague, Czech Republic.

Contents

List of Abbreviations

3D .... Three-dimensional
BOM.... Bill of Materials
C.......... Constraint
CAD .... Computer Aided Design
CAE ..... Computer Aided Engineering
CC........ Configurable Component
CCM.... Configurable Component Modeler
CE .... Concurrent Engineering
CI .... Control Interface
CS......... Composition Set
CTO..... Configure-to-Order
DRM.... Design Research Methodology
DS .... Design Solution
DSM .... Design Structure Matrix
EB......... Electric Beam
e.g......... exempli gratia (for the sake of example)
et al. ..... et alii (and others)
etc......... et cetera (and more)
ETO ..... Engineering-to-Order
F-M...... Function-Means
FEA...... Finite Element Analysis
FR......... Functional Requirement
IA .... Interaction
iaio .... is an implementation of
ICA ...... Information Contents Assessment
icb .... is constrained by
i.e.......... id est (that is)
IF.......... Interface
iib .... is influenced by
ipmb .... is partly met by
IPS........ Industrial Path Solutions
isb......... is solved by
IT .... Information Technology
iw .... interacts with
LPD...... Lean Product Development
LPT ...... Low Pressure Turbine
NASA .. National Space and Aeronautics Administration
OTM.... Over-turning moment
p .... page
PDM .... Product Data Management
PFMP .. Product Family Master Plan

PLM ..... Product Lifecycle Management
pp. .... pages
PPD...... Product and Production Development
rf........... requires function
RDnT ... Robust Design and Tolerancing
RMS ..... Root Mean Square
RQ........ Research Question
SAR...... Spiral of Applied Research
SBCE.... Set-Based Concurrent Engineering
SOA ..... Service Oriented Architecture
TD........ Technology Development
TEC...... Turbine Exhaust Case
TIG ...... Tungsten Inert Gas
TRS ...... Turbine Rear Structure
U.S........ United States
VP .... Variant Parameter
VPV ..... Variant Parameter Value

1

Introduction

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."
- Albert Einstein

Product development is all around us. The products that we use every day, such as the computers we are using in our work, the car that almost ran you over on your way to work, or the toaster that makes our bread crisp and delightful for breakfast are all results of product development. Often, companies have several thousand employees working with developing new products for the market; a market that is more or less unpredictable. The economic success of most companies depends on how well they are able to identify and interpret the needs of their customers and quickly create products that answer those needs and that can be manufactured at low cost (Ulrich and Eppinger, 2008).

In other words, companies strive to provide their customers with high quality products, faster than their competitors, using as few resources as possible. However, time, quality and cost have traditionally been seen as in conflict. In other terms, if you for example want to achieve high quality, you will most probably end up with a higher cost product that may take longer to develop. Similarly, if you want to achieve a low cost product you would have to compensate by lowering the quality of the product. One way of managing all three aspects at the same time is to reuse old, quality assured designs in new products (Duffy and Ferns, 1999). Formal approaches for design reuse have proven beneficial in providing designers with a baseline from which to build their designs.

1.1. Approaches to design reuse

There are multiple ways of viewing the benefits of design reuse. For example, design reuse equips organizations with the means to provide variety to their customers, while at the same time keeping the design effort at a manageable level. A common way to accomplish this is to create a platform with designs that are reusable in a multitude of products. The combination of different parts or modules creates a variety of products that fulfill a range of customer needs (customization), but still keeps the number of parts at a minimum (commonality).

Using platforms increases quality and reduces risk through the use of already developed and verified designs (Meyer and Lehnerd, 1997). It also lowers the cost of production since expensive manufacturing equipment can be used longer. For purchased parts, the volume will also go up, thus allowing for bulk discounts.

Perhaps even more interesting is that it is not only a way to gain the benefits of scale in production, but also development (Jiao et al., 2007b, Meyer and Lehnerd, 1997, Robertson and Ulrich, 1998). Platforms decrease the time it takes to develop new designs because they provide a head start; not all parts need to be redesigned and those that need to be redesigned are not done so from a clean slate. In product development, the single largest cost is engineering hours. Thus, by reducing the lead-time and engineering time for new designs, platforms enable companies to bring products to the market earlier *and* at a lower cost.

To be able to increase the utilization of expensive production equipment, traditionally, research in platforms has focused on finding ways to provide manufacturing with high volumes per part, i.e. low number of different parts. However, to cut lead-time and gain first mover advantages, platforms need to support efficiency in the product development phase as well (Wheelwright and Clark, 1992). Platforms based on mixing parts in different configurations alone do not provide the support that product development needs in order to attain such efficiency (Gedell, 2011).

The physical building blocks that constitute these platforms are created with a fixed set of customer requirements in mind. Therefore, they are sub-optimal for businesses where customer demands change a lot, resulting in large or frequent changes to the product, for example products with low manufacturing volumes. To address this, there are other ways to keep the efficiency over time. For example, reuse could incorporate more than physical parts.

A designer needs more information than just the physical form of a design, for example why a subsystem looks the way it does and what function it realizes in order to reuse a design. This is extra apparent in engineering-to-order companies (Meyer and Lehnerd, 1997). Alblas and Wortmann (2009) suggest reuse on a higher level using function platforms. These platforms enable reuse of functions and the possibility to generate engineering variants. Their abstract character also allows for integration between product development, technology development and manufacturing. A functional platform may be combined with subsystems that are scalable, or re-configurable, to fit many different products while fulfilling the same function. These types of platforms are henceforth referred to as *technology-based configurable platforms* or *development platforms*.

Another design paradigm that adheres to design reuse is *set-based concurrent engineering (SBCE)*. Set-based concurrent engineering, or just set-based design, is a framework for efficient development (Holmdahl, 2010). It supports concurrent development of systems from different parts of the product and production system (Liker et al., 1996). It does so by defining sets of possible solutions to design problems, rather than just suggesting one solution. The designers eliminate parts of these design spaces by considering the constraints other neighbouring systems impose on the systems. Within the remaining design space, the designers try to find areas where all systems intersect. As the constraints are gradually narrowed, so is the feasible design space, until only one solution remains (Sobek et al., 1999). Solutions that are discarded due to infeasibility are stored (together with the information on which the decisions rest) and can be reused in future projects where the requirements are different and the technology more mature. Hence, SBCE also provides a baseline for design, which is well above what a blank sheet offers. In contrast to platform-based development, SBCE prescribe a set of principles, rather than models and processes. There are studies of successful implementations of several SBCE principles in industrial design processes (Raudberget and Sunnersjö, 2010), yet examples that implement all principles are rare. Further, there is no consensus on how these processes and the underlying principles can help to support other types of design processes, such as those applied in platform-based design.

1.2. Supporting Development throughout the Lifecycle

The benefits of design reuse are rigorously examined in research (in for example Duffy and Ferns (1999), Jiao et al. (2007b) and Kennedy et al. (2013)). To attain the benefits, theories of design reuse need appropriate implementation in processes, models and tools.

For many years, IT-tools have been used to manage knowledge and knowledge reuse (Abramovici, 2002). However, no software tool perfectly manage all business processes of a company, e.g. technology development, product development and manufacturing system development. For example, in technology development there are no established or commonly used data management systems similar to those used in product development and production. On the product side, the rapid growth of product requirements in both number and complexity demands increasingly accurate analyses to be able to decide if a concept is feasible or not. Furthermore, the complex processes of configuring products require vast databases that can store and mine complex relations, much different from the flexible structures required for technology development. On the bright side, the combined capabilities of different tools may, on the other hand, very well satisfy the needs (Burr et al., 2003).

Similar to the split between technology development and product development, each part of the platform lifecycle requires different processes yet they need to align to finally produce a feasible product. For example, a platform is designed and then used to produce a product family. These two stages are vastly different and require different processes, models and tools. However, if these supporting elements fail to integrate between the two stages, serious issues may arise. The benefits of doing platform-based design may be outweighed by the rework needed due to mismatches between lifecycle stages. It is apparent that different stages of the platform lifecycle require different support, yet there needs to be an integrated approach on how to leverage the knowledge created throughout the lifecycle.

Product Lifecycle Management (PLM) is a business approach that aims to integrate the business processes of an organization, as well as managing the information generated during the lifecycle. Integrating software tools to make the engineering design work run smoothly is also a major part of PLM, all aiming at supporting the system needs of the system users (Stark, 2005). Managing complex platforms requires a great deal from the business and PLM has proven useful to many. PLM is about giving all members in an organization *the right information, in the right context, at the right time* (Dutta and Wolowicz, 2005).

1.3. Research Focus

The research presented in the thesis elaborates on platforms and how to support design reuse throughout the platform lifecycle. Special consideration has been given to set-based concurrent engineering as a means to provide an over-all set of principles for efficient development. The research will focus on how set-based concurrent engineering can help enhance the *processes* and *models* necessary to carry out efficient platform development in all lifecycle stages. Product Lifecycle Management will be considered as an integrator of processes, models and tools that are used throughout the platform lifecycle, thereby providing a coherent support.

The research will have a particular focus towards means to gain efficiency in situations where customer requirements change a lot. The hypothesis throughout the thesis is that *development platforms* are a possible solution. In summary, the foci of the research presented in this thesis are the following:

- Processes and models for a new paradigm of platform-based development based on re-configurable systems, i.e. development platforms, supported by SBCE principles.
- Supporting platform-based development that is integrated across the lifecycle of a product.

1.3.1. Industrial Goals

In order to survive in the competitive environment in which manufacturing companies reside, efficiency in producing new products is a necessity. Platform-based design is the reality of many companies, yet their tools and processes are often best suited to single product development. The results from this thesis aim to provide manufacturing companies with insights into what is required to fully utilize the potential of platform-based design, especially in environments where customer requirements change a lot. Further, in terms of software tools and methods, the goal is to provide industry with demonstrators of what is possible with platform-based development, exceeding the tools that are used today.

More specifically, the goal is to provide models for design reuse from early stages of concept development coherently linked to models used in the detailed design of both product and manufacturing system platforms.

1.3.2. Scientific Goal

Though set-based concurrent engineering and platform-based design share the goal of design reuse, there are few examples where set-based design is used to design a product family rather than a single product. One of the scientific goals with this thesis is to explore the possibilities to use the acclaimed principles as a framework for efficient platform design.

PLM as a strategy for platform-based development has, from a scientific point of view, been studied previously. The focus has, however, been on PDM, rather than on the full spectra of PLM, and has further supported a platform paradigm where the platform is viewed as a set of physical parts or assemblies that are combined in different ways, resulting in different product variants. The aim of this thesis is to elaborate on design support for platforms based on core technologies and generic configurable systems that contain knowledge other than solely the physical representation of a part – by instead focusing on subsystems that fulfill functions. Finally, another goal is to present support for platform-based development conducted in an environment that is integrated across the lifecycle, creating a closer connection between technology development, product development, and manufacturing.

1.3.3. Research questions

To put the research focus into concrete words, three research questions are posed below. The research goals are met by answering these questions. The context of these questions is referred to earlier in this chapter.

RQ1: What models, processes and tools can be used to support development platforms, and how can they be improved to better suit them?

RQ2: How can models, processes and tools be integrated across the platform lifecycle to support a holistic reuse approach?

RQ3: How can set-based-concurrent engineering be implemented to improve processes and models for development platforms?

1.4. Delineations of the research

This thesis will focus on the technical aspects of supporting platform-based development. For example, the thesis will elaborate on technical possibilities of defining bandwidth based on core technologies. However, it will not prescribe ways to scope the platform to fit certain markets, including determining the bandwidth of parameters based on market requirements. It is here assumed that the scoping of a platform can be done as a separate upstream activity prior to the development of the platform.

The term lifecycle does not imply the pace in which derivative products are deployed to the market. Such strategic decisions are better suited for research in marketing. Instead, the focus is to reduce lead-time to unshackle the market department to do what they want – if products are developed fast, it is easier to release them when the market is most beneficial.

This thesis addresses knowledge reuse throughout the lifecycle. The primary objective of doing so is to improve efficiency in product development. While this thesis partly address development of production systems, the notion of *process* will always refer to *engineering processes*. When referring to manufacturing, focus is on the *production system*, i.e. the physical machines and their functionality. I do not consider the development and optimization of the production *processes* per se.

There are several mathematical approaches to platform-based design and SBCE, for example configuration through optimization and Constraint Satisfaction Problem solving. Because this is a well-developed area, this thesis will contribute elsewhere.

This thesis addresses the issue of change management from a technical point of view.

Change management refers strictly to the engineering processes in which changes to the design or manufacturing system are managed. The organizational change, which is needed to implement new processes and models, will not be elaborated on.

The earlier papers upon which this thesis builds were written before the term development platform was coined. The readers are asked to condone the various terms in which development platforms are referred to.

2

Frame of Reference

This chapter brings attention to literature on which the results and reasoning of this thesis rest. Further, this chapter explains important concepts, phenomena and the context to which the research in this thesis relate. As Isaac Newton expressed it, while citing Bernard of Chartres:

> *"If I have seen further, it is by standing on the shoulders of giants".*

2.1. Engineering Design

Engineering design has been a vital part of our civilization for as long as anyone can remember. Traditionally, skilled craftsmen developed and manufactured products based on solving everyday problems. As for the consumer market, blacksmiths and shoemakers, for example, would craft their products based on what their customers needed, producing unique products to fit each and every customer. At best, they would use drawings to describe their products. Often, it was sufficient to describe a product with a list of parts (Claesson et al., 2001).

The industrial revolution (1770-1800) was the start of a new era during which goods and consumer products were mass produced, rather than customized to a specific customer. By 1850, it was the dominant manufacturing principle in the U.S. Products were described as hierarchical structures. Efforts were made to customize products by changing parts, but perfect interchangeability was never achieved (Duguay et al., 1997).

As products grew increasingly sophisticated, more advanced ways of describing both the products, and the variability of the products emerged (more about that in section 2.4). More advanced products, and the requirement for efficiency resulted in large organizations dispersed across the world.

To achieve efficiency across the lifecycle, it is now common that several business processes run in parallel (Prasad, 1996). This type of concurrent engineering (CE) allows for example technology development, product development and manufacturing development to start sooner than in pure sequential development, thus shortening the lead time. On the other hand, it requires integration of teams, tools and product information. Having reusable digital product and process models, such as seen in a platform is a way to facilitate concurrency (Prasad, 1996).

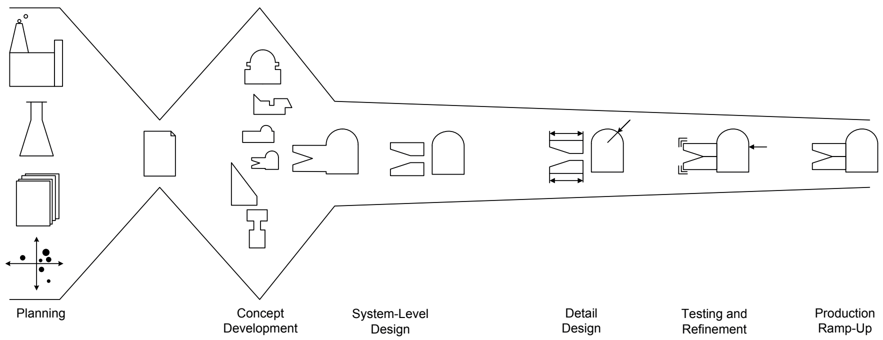

Figure 1: The development funnel, as proposed by Ulrich and Eppinger (2012)

2.1.1. The engineering design process

Numerous researchers throughout the years have studied the process of designing a product. A common elements in several of the approaches is that they prescribe a model for how to proceed; a product development process (Hubka and Eder, 1988). Pahl and Betiz (1988) as well as Ulrich and Eppinger (2008) contribute to the area of systematic design, describing processes and methods to manage complex product development. Both are examples of approaches that have been revised (Pahl and Beitz, 1996, Ulrich and Eppinger, 2012) to encapsulate new areas of product development, including for example considerations of additional lifecycle stages.

Ulrich and Eppinger (2008) describe the different steps in a typical product development process as depicted in Figure 1.

- *Planning* usually precedes the actual start of the project, and includes outlining strategies and approving investments in new technology. Here, the goals for the project are established.
- *Concept development* identifies market needs and targets and creates alternative product concepts. A concept can be described by function, form, and features, usually accompanied by a set of specifications.
- *Detail design* includes the complete specification of geometry, material selection and tooling design for each part of a product. Issues such as production cost and robust performance are managed in this phase.
- *Testing and refinement* focuses on development of prototypes and preproduction series. The design is verified against customer requirements, such as performance and reliability.
- *Production ramp-up* initiates the production of the final design. The purpose of this phase is to train the production work force and work out any remaining errors in the production process before proceeding to full-scale production.

2.1.2. Concurrent engineering

Concurrent Engineering (CE) addresses the issues with organizations split into functions and therefore loosing the lateral connections necessary to prevent mismatches between for example product development and production. Wheelwright and Clark (1992)

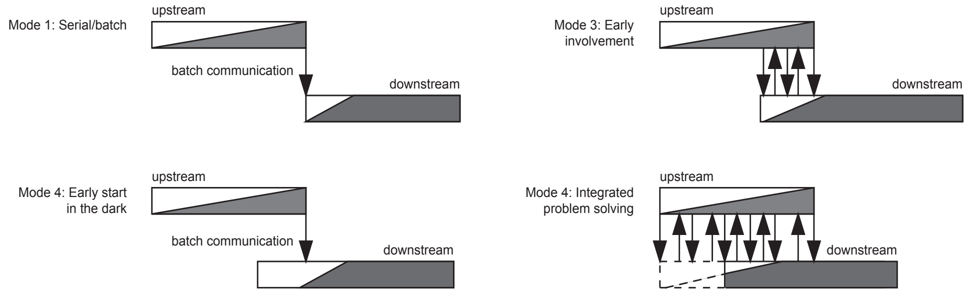

Figure 2: Four modes of upstream-downstream communication (adapted from (Wheelwright and Clark, 1992))

present four modes in which cross-functional communication can be conducted (Figure 2). Mode 4 illustrates the communication patterns of concurrent engineering.

The intent with concurrent engineering is to parallelize otherwise sequential activities to reduce lead-time, while simultaneously integrating them. According to Hauptman and Hirji (1996), concurrent engineering is

> *"the integrated and parallel design of products and their related processes, including manufacturing, test and support."*

Basically, CE implies sharing information that may affect downstream activities earlier, when the information is still preliminary. Though concurrent engineering is often portrayed as an organizational phenomenon, the notion of concurrency is also reflected in how designs are modeled. For example, to achieve concurrent engineering between sub-systems in the same product, e.g. the engine and the gearbox of a car, modularization of the product is a necessity (Erixon, 1998, Gershenson et al., 2003, Gu and Sosale, 1999). Modularization as a method for modeling variety is discussed in section 2.4.2.

2.2. Set-based concurrent engineering

Set-based concurrent engineering is a design approach that facilitates concurrent engineering in actual design work. It is praised for its superiority over conventional design paradigms (Liker et al., 1996). SBCE originated from a study of the Japanese automotive manufacturer Toyota's excelling development methods as a part of the Lean Product Development philosophy (Liker et al., 1996, Womack et al., 2007).

2.2.1. Point-based and set-based development

Sobek et al. (1999) summarize *set-based concurrent engineering* as engineers and product designers "reasoning, developing and communicating about sets of solutions in parallel and relatively independently." SBCE addresses issues with regular product development by considering a broad range of alternative design solutions that are systematically narrowed down by eliminating undesirable solutions (Malak Jr et al., 2009). This contrasts to point-based development where one solution is iteratively modified until it fits the specification (Morgan and Liker, 2006).

Important design decisions are made as late as possible, at the exact point at which they are required. Up to that point, the development is focused on building knowledge about the concepts, systems or components. These are defined, using ranges of parameter values and not as point solutions. This knowledge forms a base on which to found subsequent design decisions. Consequently, problems stemming from for example gut-feeling decisions, unintended loop-backs and sub-optimal solutions may be avoided (Holmdahl, 2010).

Few studies involve comprehensive development processes for SBCE. Rather, SBCE literature advocates the use of set-based design principles. Sobek et al. (1999) propose the following principles:

1.	Map the design space	– Define feasible regions – Explore trade-offs by designing multiple alternatives – Communicate sets of possibilities
2.	Integrate by intersection	– Look for intersections of feasible sets – Impose minimum constraint – Seek conceptual robustness
3.	Establish feasibility before commitment	– Narrow sets gradually while increasing detail – Stay within sets once committed – Control by managing uncertainty at process gates

These principles are concretized by Bernstein (1998) as depicted in Figure 3. The three systems, or functional groups, solve different parts of a product development problem. Each part represents a set of alternative solutions, rather than a point solution. The solution spaces are expanded until a small region of overlap can be identified (1). This region is then cooperatively expanded (2) and solutions outside of the region are eliminated (3). The intersecting region is then narrowed down to one final solution (4).

Step one in the above accounted for process implies that the solutions need to be modeled as a set, i.e. they need to be equipped with variety. There are several ways of modeling variety of products, for example parameterization and modularity. Another approach is to create a range of discrete solutions that satisfy different parts of the requirements range. Methods for modeling product variety are accounted for in more de-

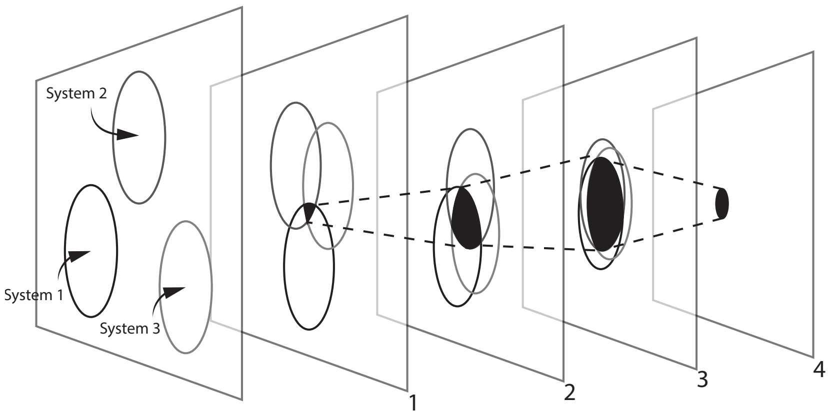

Figure 3: The set-based narrowing of design spaces (adapted from Bernstein (1998)).

tail in 2.4.

The narrowing down of a set, i.e. (3) and (4), is based on facts. Consequentially, a solution within the set can be eliminated for example if it is *proven* inferior to other solutions, incompatible with other partial solutions or that the technology realizing it is too immature to be reliable. Both virtual and physical tests are performed to attain enough information to be sure to make the right decision. Eliminated solutions are stored with the reason to why they were eliminated. In doing so, the company builds a solid knowledge base to reuse when developing the next generation of products.

Similar to platform-based design, SBCE targets design reuse and ranges of solutions. However, while platform-based design aims to keep the range of solutions longer to produce a product family, SBCE aims to produce one single product (Sobek et al., 1999). Table 1 summarizes the differences between point-based and set-based design.

Table 1: Comparison of point-based and set-based approaches (redrawn from Bernstein (1998) as adapted from Sobek (1997))

FUNCTION	POINT-BASED APPROACH	SET-BASED APPROACH
Search: How should solutions be found?	Iterate on existing ideas. Brainstorm new ideas.	Define feasible regions
Communication: Which ideas are communicated to others?	Communicate the best idea.	Communicate sets of possibilities.
Integration: How should the system be integrated?	Pass the idea among the team for critique.	Look for intersections.
Selection: How is the best idea identified?	Formal schemes for selecting the best alternative. Make prototypes to confirm that the solution works.	Design in parallel on each alternative until it is not worth pursuing. Look for low cost tests to prove infeasibility.
Optimization: How should the design be optimized?	Analyze and test the design. Modify the design as necessary to achieve objectives and improve performance.	Design in parallel on each alternative until it is not worth pursuing. Look for low cost tests to prove infeasibility.
Specification: How should you constrain others with respect to your own subsystem design?	Maximize constraints in specifications to assure functionality and interface fit.	Use minimum control specifications to allow optimization and mutual adjustment.
Decision risk control: How should one minimize the risk of "going down the wrong path"?	Establish feedback channels. Communicate often. Respond quickly to changes.	Establish feasibility before commitment. Pursue high-risk and conservative options in parallel. Seek solutions robust to physical, market, and design variation.
Rework risk control: How should one minimize damage from unreliable communications?	Establish feedback channels. Communicate often. Respond quickly to changes.	Stay within sets once committed.
Management: How should the process be controlled?	Review designs and manage information at transition points.	Manage uncertainty at process gates.

2.2.2. Trade-off curves

In any design scenario, there are multiple goals that need to be achieved (Vincent, 1983). These goals are often contradictory, for example safety and fuel consumption of a car. In such cases, the designer's task is often to assess how the goals are related and pick parameter values that enable the design to satisfy the combination of these goals (Otto and Antonsson, 1991). A trade-off curve is a graph that shows one performance criterion on the Y-axis, and another performance criterion on the X-axis. A curve is then plotted to illustrate the relation between the two criteria (Morgan and Liker, 2006). Trade-off curves as a means to store and model knowledge about designs is not specific to SBCE. Yet, it is an integral part, which is why it has its own section here. There are several historical examples of trade-off curves. For example, the Wright brothers used trade-off curves to assure the success of the world's first airplane before building it. Among other graphs, they used airfoil diagrams created through over two hundred wind tunnel tests with scale models (NASA, 2010). Today, virtual simulation complements extensive physical testing. Ideally, these are performed in a knowledge creation process separate from the regular product development to serve several different projects (Ward, 2007).

Trade-off curves play an important role in realizing all the principles of SBCE. To *map the design space*, trade-off curves are used to model the design in terms of what the current design can or cannot do. Typically, two requirements are mapped against each other and a curve establishes the relationship between them (Sobek et al., 1999). In this way, many different design alternatives are considered simultaneously as the curve represents the collected characteristics of all solutions of a subsystem. If the desired point on the plot is within the feasible region, e.g. above the curve, it is safe to assume that the design can cope with the requirements.

A trade-off curve may serve as a communication device for *integrating by intersection*. If modeled properly, they can illustrate the trade-off between two requirements, as shown in Figure 4, between parameter values of different sub-systems or parameters and requirements within each sub-system. As feasible regions of designs are mapped out, trade-off curves serve as a tool to keep track of the design space as the requirements are tightened.

2.3. Platforms

This section aims to account for what a platform is and how it supports development of complex systems. Specifically, different takes on what a platform may incorporate are discussed, as well as different views on the lifecycle of a platform.

2.3.1. Defining a platform

Using a platform as a means of reusing knowledge has been receiving a lot of attention over the past decade (Jiao et al., 2007b). The common view of a product platform is as a collection of different parts that can be combined into a variety of products, such as for example Lego. In literature, the term platform is comprehensive, essentially incorporating any form of reuse of design and manufacturing knowledge.

Meyer and Lehnerd (1997) define a product platform as *"a set of subsystems and in-*

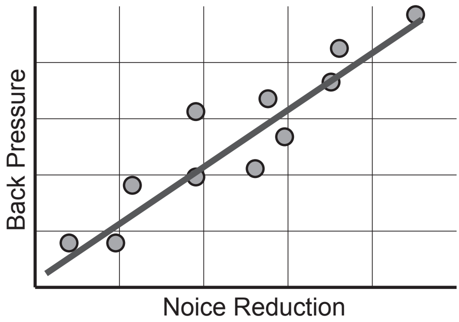

Figure 4: Multiple tests help understand the design space (adapted from Morgan and Liker (2006))

terfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced".

Robertson and Ulrich (1998) take an even wider grip on the term and define a platform as *"the collected assets shared by a family of products"* – an asset being a component, a subsystem, manufacturing equipment, a process or even an individual.

Based on this reasoning, the next step is to start looking at technology platforms as a basis for the systematic reuse of solutions found in technology development projects. McGrath (2001) defines a technology platform as "a set of initiatives organized around a macro-level functionality that helps to manage and optimize technology investments across multiple product platforms". Similar definitions of technology platforms as bases for different products and markets have also been proposed by, for example, Meyer and Lehnerd (1997), Shapiro (2006), and Jolly and Nasiriyar (2007) and the concept of a technology platform has similarities to its core competencies and dynamic capabilities (Jolly and Nasiriyar, 2007, McGrath, 2001).

2.3.2. The platform design process

The design process is the driver for all types of reuse throughout the lifecycle (Inns and Neville, 1998). The design process for platform-based design will therefore look different from the process of single product development. Further, an essential principle in the design process is that designers with the proper understanding of the design context make better decisions (Hansen and Andreasen, 2002). The same applies to platform-based design. The fundamental difference is that the basis for several product variants are created simultaneously, making the design context immensely more complex to overview (Pedersen, 2009) – elements of the platform can be both generic and varying, and trade-offs must be made that would not have been necessary in single product development. As opposed to designing single products, platform-based design is about modeling a wide set of potential solutions. These solutions need to share characteristics, but at the same time be distinct (Pedersen, 2009). On a higher level, design processes such as Ulrich and Eppinger (2008) and Pahl and Beitz (1996) apply, but the design philosophy behind these methodologies differ a great deal. Matters such as identifying the proper bandwidth of a subsystem and making designs reusable are inferior in single product development, but crucial to platform-based development. Further, it is within platform-based design, necessary to distinguish between developing a platform, and using an existing platform as a means from which to derive product variants.

Shahin et al. (1999) use an established process similar to the one presented by Ulrich and Eppinger, but more explicitly target product platform preparation and execution. They identify crucial steps where design reuse is facilitated. They mean that reuse can be performed on different levels and that reuse needs to be actively considered. For example, modeling the functions of a product during the concept phase allows for reusing of function carrying concepts at a later stage.

Pedersen (2009) identifies the presence of "intangible elements such as activities or organs" as a challenge in modeling *and* mindset. There are implications for product complexity and efficiency in the links between lifecycle phases, and in particular to the production setup. According to Tseng and Jiao (2001), "the main challenge for design methodologies is to support these multiple viewpoints to accommodate different modelling paradigms within a single, coherent and integrated framework".

2.3.3. The platform lifecycle

The platform is both a concept and a design template – thus the design template has to be designed and thereafter derivative products are designed. Depending on what development philosophy is adopted, a platform is then retired or lives on through evolution and updates. Consequently, the platform lifecycle is separate from the lifecycles of the products and of the manufacturing system (Wortmann and Alblas, 2009). Following that principle, there are several possible views of the platform lifecycle. For example, Pedersen (2009) defines three fundamental phases in the development of a platform: *platform preparation*, *platform execution*, and *platform maintenance*. Roughly, most design decisions are made during the preparation phase.

While *preparing* the platform, a first step is often scoping the platform (Pedersen, 2009). As a part of the scoping segmentation is utterly important (Meyer and Lehnerd, 1997) and often time consuming. It incorporates the early phases of market segmentation and mapping the future product platform to customer needs, or in other ways identified market segments. The methods for scoping platforms are many, some of which use information from previous products to improve the accuracy of the mapping between market and product (Meyer and Utterback, 1992). The result of the scoping is often a set of requirements, which can be used as bases for concept development and later detailed development of product and production system platforms.

The subsequent *execution* phase is where the product variants are actually generated – something that should require considerably less effort if the preparation is done correctly. Based on the platform, variants are configured. The bandwidth of the platform allows multiple variants serving different requirements to be generated from the same set of designs. Depending on the strategy, the configured variants may be mature enough for manufacturing or require further testing before start of production.

Upgrades and maintenance are conducted throughout the *maintenance* phase during which issues are addressed and the platform is kept updated. Updates could be induced as customer requirements become higher rendering the current bandwidth insufficient.

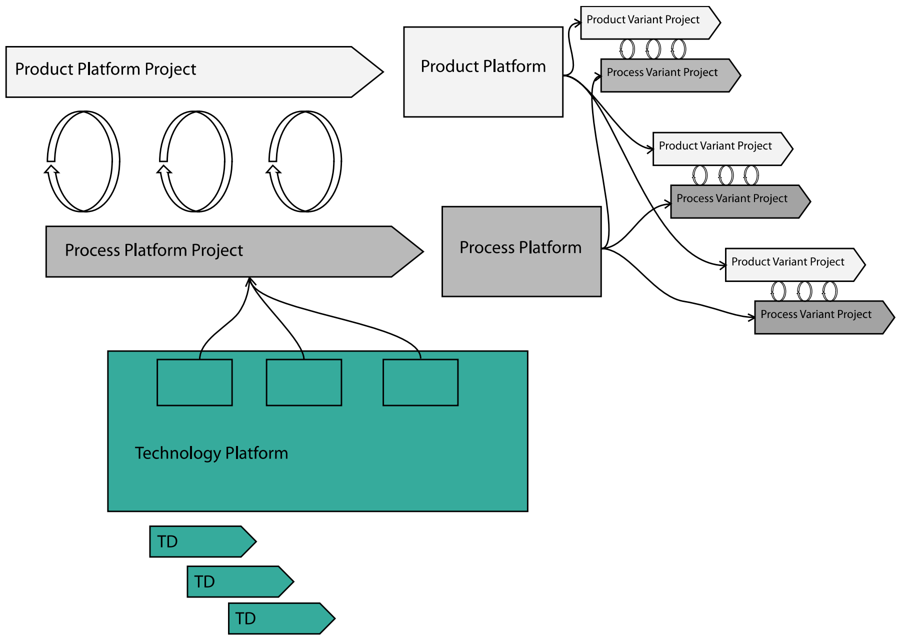

Figure 5: Platform-based development, redrawn from Bergsjö (2011).

The *maintenance* of a platform is defined by Alblas and Wortmann (2012) as

> *"the process of performing changes to predefined platform attributes (functionality, performance, interfaces, technologies, etc.)."*

Generally, a company runs platform development parallel to derivative and technology development (Wheelwright and Clark, 1992). Even though the development is run in parallel, a technology platform, when *executed*, acts as a basis for developing the product platform. In the same way, *executing* the product platform acts as the basis for configuring a product variant. On that note, Berglund et al. (2008) and Bergsjö (2011) present a platform development framework, which incorporates technology platforms, product platforms and production platforms, thus introducing a second dimension to the platform lifecycle. Figure 5 illustrates the different components of the platform development process. A distinction is made between platform projects that deliver knowledge to the product and process platforms and derivative projects (also called variant projects) that generate product variants for the market.

What Bergsjö (2011) illustrates is that a technology platform feeds the product and process platform projects with knowledge developed during previous technology development efforts. This stands in contrast to single product development where technology development projects feed the derivative projects directly (Jolly and Nasiriyar, 2007). When it comes to technology platforms, platform execution focuses on implementing technologies into a product platform (Bergsjö, 2011). Högman et al. (2009) explore reusability in four dimensions for an aerospace company, concluding that generic technologies are the only entities that are reusable across different Applications, Customers,

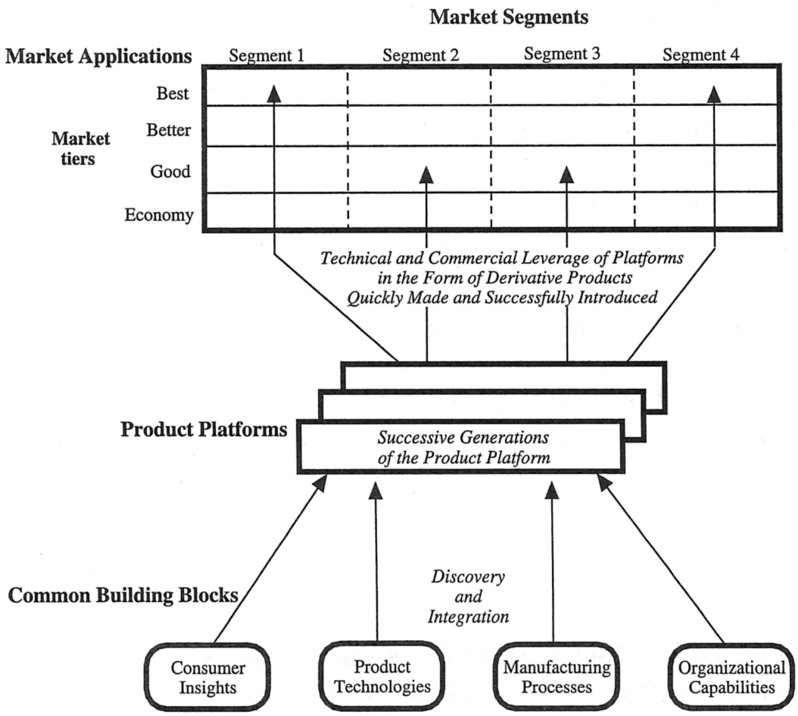

Figure 6: Different levels of a platform, according to Meyer and Lehnerd (1997)

Generations and Sizes.

Even though both Meyer and Lehnerd (1997) and Robertson and Ulrich (1998) base their view of a platform on the assumption of a *product* platform, they manage to encompass more in their definitions – perhaps the former pair of authors even more than the latter. Meyer and Lehnerd (1997) identify three different levels in a platform (Figure 6): the common building blocks; the product platform itself according to their definition; and variants generated from the product platform that together constitute a product family. Conclusively, it is reasonable to talk about two different dimensions of the platform lifecycle, i.e. technology-product-manufacturing and preparation-execution-maintenance.

2.3.4. Approaches covering the platform lifecycle

The approaches to platform-based development are many. Together, they form a patchwork of models, processes and tools. An example of a more holistic platform approach that covers large areas of the lifecycle includes the Product Family Master Plan (PFMP) by Mortensen (1999). The PFMP has been further developed (Mortensen et al., 2008) to comprise not only product modeling, but also capabilities for scoping the platform (i.e. platform preparation) based on, for example, sales volume for each variant, as well as the need for production system investments.

Other approaches target specific parts of the lifecycle. Mesehovic & Malmqvist

(2004) describe how a product platform can be used as a basis for sales configuration (i.e. execution), combining parts into different product variants, while Haug et al. (2012) describe system solutions and how to manage them. Harlou (2005) deals with strategic fundamentals and how to achieve the business goals of a product platform.

Others authors focus on *technology platforms* and supporting its creation (preparation) and use (execution). For example Corin Stig & Bergsjö (2011) describe a Wiki solution for managing reusable technology knowledge (Högman, 2011) prescribes a technology development stage-gate process specifically aimed at the aerospace industry. Bergsjö (2011) uses a platform framework, consisting of technology, product and production platforms to elaborate on the implementation of a technology platform and accompanying IT tools. In terms of managing technologies (i.e. maintenance) and reusing them (i.e. execution), Meyer & Utterback (1992) present their work on core competences and the effects of managing them well.

Production platforms in various forms are discussed by Erixon et al. (1996) as well as Michaelis (2011). The former uses modularization of the product and the production system as a way to increase the efficiency of development and production. The latter, Michaelis, describes how co-development of the product and production platforms can be performed. Gedell et al. (2011) speak of a unified product and production platform. Michaelis et al. (Michaelis (2013), Michaelis et al. (2014)) also describe the use of functional models for representing production system platform (preparation), and how these can be linked to the product platform using operations as connecting elements. Koren et al. (1999) suggests a reconfigurable production system (execution), which accommodates the variety of a product family. The configuration serves to quickly adjust to changing customer requirements, while flexibility of the system itself serves the product family variation.

2.3.5. Change management in platform design

Many studies on platform development address a situation where all requirements are known. The studies that include platform scoping build upon the idea that the result of such a scoping is true throughout the entire development of the platform (Jiao and Chen, 2006). In reality, customer requirements change constantly during the development. Furthermore, late design changes induced by late changes in customer requirements are costly (Loch and Terwiesch, 1999). The alternative, i.e. stick with the current design even though the customer requirements have changed, risks having a severe impact on sales volume, or in the case of a OEM-supplier relation completely eradicate the compatibility with the rest of the product. As powerful as platforms are, they increase the number of products that are affected by a change – the affected system is very likely to be found in several different products (Jarratt et al., 2011).

Change management is a well-explored field. Reidelbach (1991) and Rouibah and Caskey (2003a) focus on change management in multi-tier collaboration, while others focus on how to assess the impact a change might have. The interdependencies between systems in a product and between product and production system causes changes to propagate (Rouibah and Caskey, 2003b). Therefore, the propagation of changes needs to be evaluated and managed. Eckert et al. (2001) argue that a model with the most impor-

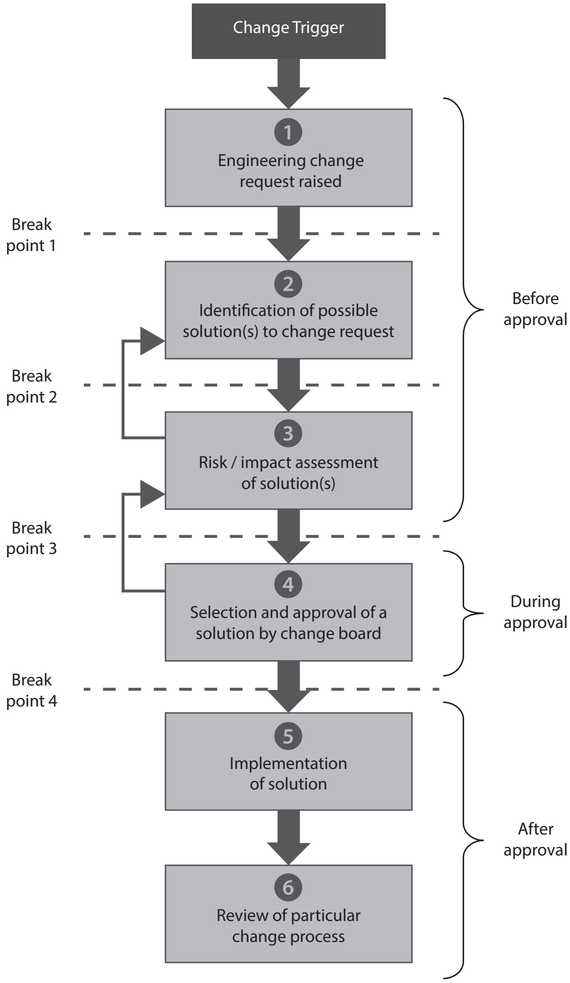

Figure 7: A general change process (redrawn from (Jarratt et al., 2005))

tant systems and their interdependencies will help in predicting change impact. Eger et al. (2007) also stress the importance of being aware of the effects of a change and provide a method for change propagation assessment while still in the design phase. Clarkson et al. (2004), presents a method for change prediction for mature designs. It uses a product model broken down into sub-systems. The connections between the sub-systems are represented in a Design Structure Matrix (DSM). Using values for impact and likelihood, the risk associated with a change can be calculated. Jarratt et al. (2005) propose a generic change process focusing both on the managerial aspects, such as the approval process, and the engineering side. Such considerations include finding a solution, assessing the impact on for example design, manufacturing and suppliers, as well as implementing the solution (see Figure 7). Ahmad et al. (2013) present a model capable of tracking change propagation across many different domains. They conclude that modeling the interactions between the domains is key in change impact assessment.

While the methods above provide support for single product and production development, little is know about their usefulness in platform development. Wortmann and Alblas (2009) and Alblas and Wortmann (2012) claim that platforms and derivatives alike should be put under change control, yet separate change control. Alblas and Wortmann (2012) further excavate the area of platform lifecycles and conclude that platforms may, if changes are managed correctly, aid lean production. Few researchers actually describe what the engineering change process is, but Leech and Turner (1985) describe it as a highly constrained miniature design process.

2.4. Modeling Platforms and Variety

This section introduces a number of models and approaches for representing designs throughout the lifecycle. Some of the models are tied to a specific stage in the lifecycle, while others encompass more than one stage. In some cases, models can be used as interfaces between lifecycle stages.

2.4.1. Models in engineering design

Models are scaled down representations of reality. Because they are simpler than the real things they represent, they can be used to analyze and synthesize reality but only for what the model is intended to do: a model has a purpose.

Models are used to represent many different facets of the design. They may for example represent customer requirements, physical parts or manufacturing processes. The *Chromosome Model* by Andreasen (1992) presents several different design elements from an number of domains. While these domains often belong to separate parts of the organization, they are inevitably linked. The model connects concrete parts used to create product structures, i.e. parts and subassemblies, to abstract product representations, i.e. functions, processes and organs. The original version includes a Function domain, which the later versions (Mortensen, 1999) has merged into the other domains. The chromosome model is illustrated in Figure 8.

In platform development, a common purpose is to model variety of products, production systems and technologies. For example, a platform is designed to meet a range of customer requirements. This range is created during the concept development and narrowed down to a desirable size in the concept screening. The range may be referred to as bandwidth. Berglund and Claesson (2005) introduce bandwidth as a systems flexibility, which allows it to be used in a variety of products. Thus, bandwidth may consider the physical and functional properties of a product, such as the range of engines that a car can have, which can be linked to the fulfillment of the range of customer requirements. Consequently, there is a bandwidth of the requirements, and on the design solutions that solves the requirements (Wahl and Johannesson, 2010). These examples describe models that encompass several of the domains identified by for example, Andreasen.

2.4.2. Modularization and scalability

Two fundamental approaches to build in variety into a platform are scalability and modularization. In scalable platforms, the design can be expanded and contracted

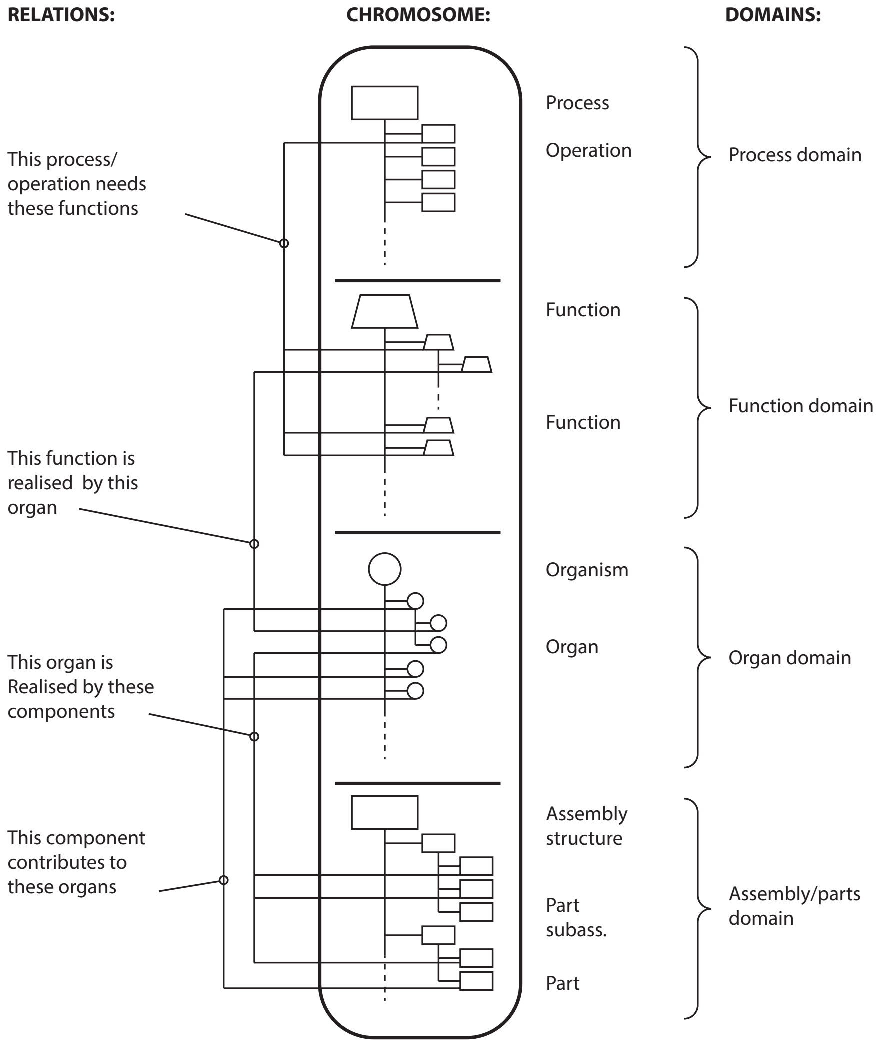

Figure 8: The chromosome model (adapted from Andreasen (1992))

to fit specific customer requirements (Simpson, 2004). This is achieved by manipulating parameters of the design such as the length of a piston or the size of the piston head to create different configurations. Using parameters instead of fixed values is a way to keep the design open longer.

It is common to define rules describing the relationships between parameters. In doing so, some parameters may become primary, and others secondary. Setting primary parameters automatically sets the secondary. For example, a rule can be defined as: a bolt hole cannot be smaller than the bolt. If the diameter of the bolt is primary, then the hole will have to adapt its diameter. The relationships can be defined using for example equations, data sheets or logical expressions. The rules are defined to prevent that changing one parameter generates a solution that fails or cannot be built.

A module-based platform consists of a set of interchangeable modules. By changing one module for another, different properties are achieved (Gonzalez-Zugasti and Otto, 2000). For example, changing the lens on a camera may give different focal length without having to change anything on the camera body. Modularization has many benefits. For example, it facilitates carry-over from old products and provides variation in functionality through changing modules for others (Gu and Sosale, 1999). Further, modularization provides the means for concurrent engineering by creating independent chunks through carefully managing interfaces between modules (Gershenson et al., 2003). Erixon (1998) describes modularization as

> *"decomposition of a product into building blocks with specified interfaces, driven by company specific reasons".*

Claesson (2006) adopts a more abstract approach and describes a module as a group of design solutions, which can be reused in several products. Much like an organ, these modules fulfill a certain function. A common view is that having fixed or standard interfaces between modules is a condition for success (Schilling and Steensma, 2001, Takeishi, 2002). However, locking the interfaces reduces the design flexibility drastically, and may limit the compatibility with new innovative solutions.

Modular and scalable platforms present two distinct options to leveraging of the market. Modularity enables horizontal leveraging, i.e. to be able to offer products in different product categories while still sharing components. Scalable platforms also focus on offering a wide range of functionality to different customer groups but has the ability to provide vertical leveraging, i.e. to offer low-cost and high-cost options within the same product category (Simpson et al., 2001).

The examples of modular product platforms are countless. On the production side, Michaelis and Johannesson (2011) describe a modular approach to robot cells in manufacturing. They compare a modular approach to an integrated approach by using function-means modeling. Apart from flexibility, the modular approach introduces a less complex layer in the system structure with fewer interactions that need to be managed.

A platform is in most cases either scalable or modular (Du et al., 2014), which may be a result of the prerequisites for execution. The execution of a modular platform assumes that the parameters of each module are fixed, i.e. the geometrical and physical properties of each module are constant (Fujita et al., 1999). On the other hand, a scalable platform assumes that the architecture is fixed, i.e. no modules will be switched out. There are studies where scalable and modular execution is performed simultaneously (Du et al., 2014), which does however require post-embodiment models.

2.4.3. Adaptable design

As an approach originally developed to manage changing environments, such as changing customer requirements, Adaptable Design is similar to platform-based design. The underlying philosophy is to adapt to changing environments while reusing the design or parts of the design (Gu et al., 2004).

Adaptable design may be divided into two different categories: design adaptability and product adaptability. Design adaptability arises from designs, i.e. the CAD drawings, conceptual drawings etc., which can easily be adapted to fit new requirements on functionality. Product adaptability concerns modifying existing products to incorporate new functionality (Gu et al., 2009).

By applying functional modeling, adaptable design has the ability to support product family design (Ong et al., 2006, Xu et al., 2006, Xu et al., 2007). Through applying information contents assessment (ICA), the information content of each design can be calculated, which creates an alternative way for designers to reuse the design information.

Li et al. (2008) describes how FRs and DPs from axiomatic design can be used to create design adaptability on several levels of the design. The methods build upon modularity, and interface design to work. Gu et al. (2004) proposes a process, which for adaptable design consists of: definition of adaptable design objectives, design of the product/system (conceptual and configuration), design of the adaptable product, life cycle considerations and evaluation. This process uses a fixed architecture.

Li et al. (2008) further proposes a high level process for creating an adaptable design. Roughly, the process steps are: product planning, studying modularity, establishing product architecture and adaptable interface design. This process builds upon a modular approach, and does not include how variants can be generated.

Few processes consider how the design information is going to be reused. Xue et al. (2012)/ describes how different configurations can be created, focusing on modular configuration. Wiendahl (2009) suggests an approach for creating adaptable manufacturing systems that will be responsive to market changes. He presents a framework including several layers of configuration, e.g. parameter configuration and functional logic and workflow. Though there is support for creating a product family, there are no processes for using a product platform to configure product variants in an adaptable manner.

2.4.4. Generic product and process structures

Product structures are the elements and their interrelationships that describe how a product is built up (Svensson and Malmqvist, 2002). These are often represented by Bills of Materials (BOMs) compromising of parts and assemblies used to build the product. Van Veen (1991) approaches variability by means of generic BOMs, which constitute a configurable platform model. With the generic BOMs it is possible to describe entire product families using one unified model, as opposed to describing individual products (Erens, 1996). Zhang and Rodrigues (2009) use the generic BOM concept as basis for their generic process trees. They present a method for creating manufacturing process families matching the variety of products in a product family. Also addressing process platforms, Jiao et al. (2007a) defines the operations as the reusable objects that link product and process platform.

Similar to the generic BOM, Männistö et al. (2001) present their *master BOM*, which is an integrated generic description of several product variants that together constitute the platform. However, in a response to the inabilities of those early platform models to manage a number of different aspects – for example a variety of multiple design parameters, determination of the number of product variants and determination of the platform extent – other researchers (Hernandez et al., 2003, Williams et al., 2007) propose approaches that use continuous variable design parameters. Yet another way of incorporating variability in a product platform and managing the above mentioned problems is the Configurable Component (CC) concept, first developed by Claesson (2006) and further refined through several studies (Edholm et al., 2010, Gedell et al., 2011). The approach uses configurable subsystems to compose a product architecture based on input parameters, such as customer demands. The CC concept is more thoroughly described in section 2.4.6.

2.4.5. Modeling reusable technologies

It is difficult to use the same models for technology platforms and product and production system platforms. Typically, technology development has the fuzzy goal of building knowledge or demonstrating feasibility, while product development has the more concrete goal of resulting in a commercial product (Nobelius, 2002). Also, technology development is hard to plan, and the requirements are often impossible to elicit because of the long time-frame (Högman and Berglund, 2007). Even though compared to product platforms, technology platforms capture a larger range of elements, including both physical and non-physical elements (Shapiro, 2006), and do not lend themselves to the building block modules and interface structures of product platforms (McGrath, 2001).

In these cases, other methods may be applied. For example, Corin Stig et al. (2011) concludes from a company case study that test results are a major carrier of reusable technology knowledge. These test results may, as discussed earlier be condensed into trade-off curves to illustrate relationships between parameters within the technologies and between technologies and designs. Thereby, using trade-off curves is a way to model technology bandwidth, which when executed into a product and production system platform enables the bandwidth of these systems.

A common practice in product development is to use digital knowledge repositories. The knowledge is often expressed in plain text in documents. However, technologies strong affiliation to other elements, such as assemblies or manufacturing processes, make them easy to structure and codify (Granstrand, 1998).

Corin Stig and Bergsjö (2011) structure technology information using a standard template common in Lean Product Development. The A3 method prescribes a number of headlines under which the most important information about the parts of the technology is summarized. These *sheets* also contain links to people responsible for the technology or who have participated in implementation projects.

2.4.6. The Configurable Component concept

A platform described using the CC concept consists of several autonomous systems, each described by a CC object. CC objects can *use* other CC objects to compose themselves. The Configurable Component concept has a great deal in common with a modular product platform, as described by for example Prasad (1996), allowing for concurrency while developing the different modules or subsystems. However, it aims to support a platform approach based on the idea of subsystems and concepts, rather than reusing parts. Each subsystem is configurable to fit a variety of contexts and fulfill the same function in each context. Hence, a CC platform applies both scalability and modularity to create variety.

A CC object may represent, for example, an entire car, a front door or a rear view

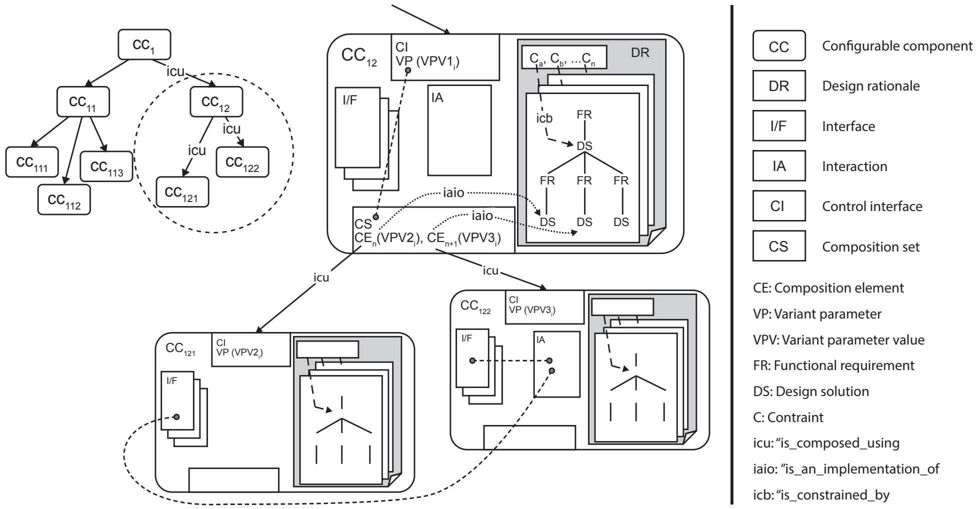

Figure 9: The composition of configurable components with encapsulated elements and relationship types (as drawn in Michaelis (2013), adapted from Claesson (2006)).

mirror. Essentially, CC objects do not represent merely one type of car door, but rather every door in a product platform – being a model of a system family. Depending on the parameters inserted different variants can be composed and as a result the door will look or behave differently.

The building blocks of the CC object can be seen in Figure 9. The connections with other subsystems are realized through an *interface* (IF) object. The interfaces are configurable and can be geometrical, electrical or logical, etc. An interface cannot be configured independently of its surroundings, thus the *interaction* (IA) object serves as communicator between interfaces. The *Control Interface* (CI) acts as an input of parameters, for example from other CCs or external models. The *Composition Set* (CS) determines what other CCs are used to further define and realize the functionality of the considered CC. The *Variant Parameters* (VPs) define variable parameters in the CC which can be used to configure the CC and all its inbound components. This is done by setting the *Variant Parameters Values* (VPVs).

Traditionally, interchangeability between modules has been assured by locking the interfaces. The CC concept goes quite the opposite way and declares a bandwidth within which platform elements, including interfaces, may vary. Thus, the interfaces are co-configured to fit each other, which allows for keeping design flexibility intact in the development process.

So far, two types of geometrical interfaces have been defined within the CC framework (Edholm et al., 2010, Edholm et al., 2009), i.e. *Locating Schemes* and *Mating Geometry.* Locating Schemes are used to fix a physical part to a coordinate system, locking all six degrees of freedom (Edholm et al., 2010, Edholm et al., 2009). A locating scheme, represented to the left in Figure 10, is typically made up of six 3D points in a Computer Aided Design (CAD) model and is often realized physically by a pin on one part match-

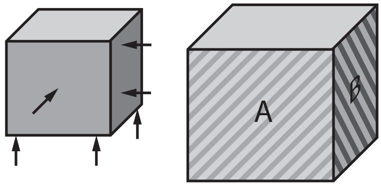

Figure 10: Locating scheme (to the left) and mating geometry (to the right).

ing a hole on a facing part.

Mating geometries, to the right in Figure 10, are the geometries facing the surroundings (Edholm et al., 2010, Edholm et al., 2009). As these geometries constitute the outer boundaries of the subsystem they are crucial for packing and other functions. Mating geometry also plays a major part in realizing functional requirements, such as transferring torque in a gearbox (the surfaces of the gears need to be in contact). The interfaces are essentially *design solutions* (DSs), but have been given a separate class due to their importance in the execution of the model. The configurable DSs represent all solutions within a CC that help solve the function of the CC.

The *design rationale* describes *why* the design looks and behaves the way it does. It does so by modeling the design using an enhanced function-means (F-M) tree. The function means-method is a systematic way of identifying solutions to functional requirements (Johannesson, 2014). It starts out with a primary function of the product to which a solution is identified, or in the case of platforms: *several* solutions are identified. This solution will have several sub-functional requirements, which in turn will get solutions. Consequently, alternating between FRs and DSs, the F-M tree grows. Conclusively, the tree relates the *functional requirements (FRs)* to the *design solutions* in a hierarchical breakdown, in which moving up the tree answers the question *why* a solution is needed, and going down answers the question of *how* a function is solved. Each DS may also be affected by a set of *constraints (Cs)*, representing requirements that are non-functional (Schachinger and Johannesson, 2000). The model also describes the lateral relationships connecting the different branches of the F-M tree.

Each functional requirement has a bandwidth within which it can vary. To answer to the bandwidth of the FR, the DS in itself has a bandwidth within which it can vary. To cover the entire bandwidth of the FR, it is sometimes necessary to switch between different design solutions. In other words, the FRs have a bandwidth (a parameter range) which is met by a set of design solutions (a concept range). Each design solution within this set also has a bandwidth (a parameter range) (Wahl and Johannesson, 2010). The design rationale is shown in more detail in Figure 11.

2.5. Support for design reuse

The development of information technology for engineering applications has resulted in numerous process supporting IT tools over the past decades. At the same time, new engineering methods and processes are introduced, possibly at an even greater pace, leaving a gap between processes and process support (Burr et al., 2004). Design reuse tools shall support the design process either by supporting the designer in retrieving and

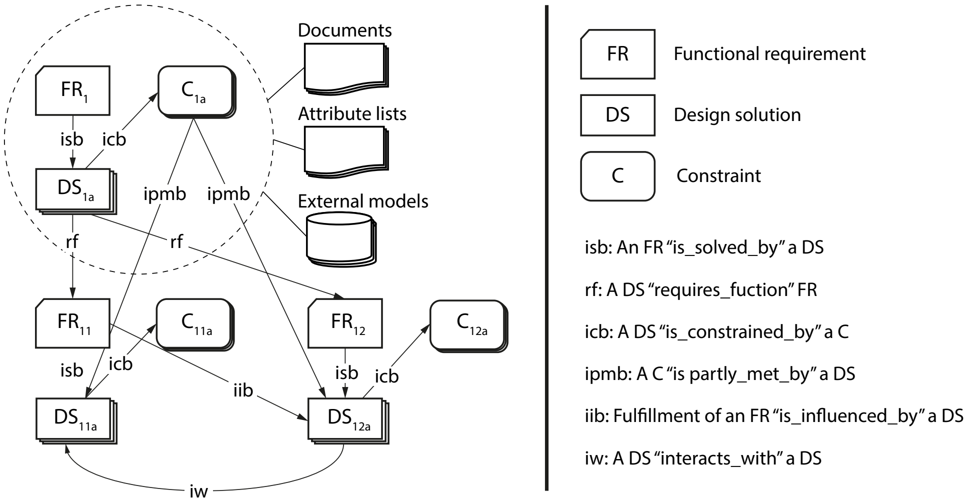

Figure 11: The enhanced function-means tree with the possibility to attach external models and documents describing the Design solutions, functions and constraints.

reusing knowledge, or by providing the means to capture knowledge generated during the design process (Baxter et al., 2007).

Frameworks and IT-support for capturing and retrieving knowledge have been researched extensively (e.g. Catic (2011), Hicks et al. (2002), Matsumoto (2005), and Wood Iii and Agogino (1996)) both by using existing support and by developing customized support. This section introduces some of the tools used to support design reuse and platform-based design.

2.5.1. Computer Aided Design and Computer Aided Engineering

Computer Aided Design (CAD) is used to create a geometrical model of a product. A cornerstone in contemporary design reuse through CAD is parametric design. A set of key parameters of the design are identified (for example the length of a shaft) and all other measurements are defined in relationship to these parameters (Baxter et al., 2007). Andrews et al. (1999) present two ways (a generative method and a variant method) to store and reuse knowledge in CAD. They conclude that the result depends on the designer's ability to store and retrieve designs systematically.

Full support can only be achieved through the integration of engineering systems (Burr et al., 2004). CAD systems are in general well integrated with the PDM system and have thus access to product meta data (Abramovici, 2002), thereby facilitating design reuse by sharing 3D models across the organization.

Computer Aided Engineering (CAE) systems is the collected name for a number of different systems aiding the virtual development of products. Typically, they are used for analysis and synthesis using CAD representations of a design. Finite Element Analysis (FEA) is used extensively for, for example, stress analysis and flow analysis. The analysis results may be used to represent design spaces (Yannou et al., 2003) or on which to base elimination decisions. However, there are no satisfying examples of integration of CAE systems for analysis or synthesis out-of-the-box. Information is rather transferred manually or in some cases integrated in one direction alone (Abramovici, 2002, Burr et al., 2004).

There are, however, examples within research of successful integration of CAE systems through customized solutions. Inoue et al. (2010) argue that the reuse of design through Set-Based Concurrent Engineering may be supported by integrating 3D-CAD and CAE systems. They present an approach to optimize structures within a set of design solutions, arriving at the optimal solution within the bandwidth through the use of integrated CAE and CAD.

2.5.2. Design spaces and set-based design

There are tools specifically aimed at management of design spaces. Apart from the immense collection of engineering optimization tools, there are tools specifically targeting set-based design. Almost exclusively, they aim at narrowing down a design space, rather than creating it.

Qureshi et al. (2010) suggests a tool for combing set-based design and robustness by using a parameterized design and a component library. Also using constraint modeling, Canbaz et al. (2014) suggests a method for controlling the convergence process using Wellbeing indicators. Using their method, they are able to narrow down a set of multiple subsystems to find a feasible solution for a clutch. However, these types of solutions require mature design information, which is not available in the early stages of development – these types of analyses are possible only *post-embodiment*. Particularly, in *pre-embodiment* phases, elimination is conducted using abstract models and knowledge from previous projects. For example, Finch and Ward (1997) use quantified relations, part catalogues and equations to model a technical system. Based on that, they eliminate solutions using constraint satisfaction techniques. Yvars and Duhau (2012) introduce functional configuration as a Constraint Satisfaction Problem. By modeling the functions of a product and the lateral constraints between them, certain infeasible solutions can be excluded. Even simpler methods are applicable, such as Pugh's elimination method (Pugh, 1990). The common denominator is that decisions are based on facts. The requirements act as drivers in the elimination process. These are defined as ranges that are gradually narrowed to reduce the design space.

2.5.3. PLM architectures

Product Lifecycle Management is widely recognized as a business approach to achieve fast and efficient product development (Grieves, 2006, Ming et al., 2005, Stark, 2005). CIMData (2010) defines PLM as a strategic business approach that supports the collaborative creation, management, and use of product definition information, spanning from concept to end-of-life of a product or plant, integrating people, processes, business systems, and information. Thus, PLM can be used to tie these varying aspects together.

Contrary to common belief, PLM is not solely an IT system. Stark (2005) argues that there are several additional parts of PLM, such as engineering methods and processes, the organization, the product and product information and IT systems which all need to be considered and coordinated. Svensson et al. (1999) share the same perspective, stating four views: *processes*, *information*, *systems* and *roles*.

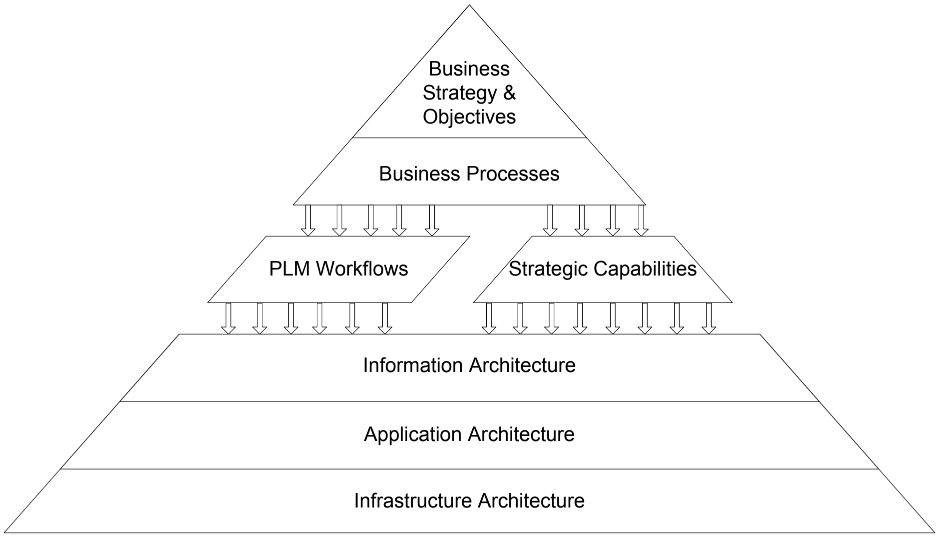

Figure 12: Describing PLM architectures, adapted from Zimmerman (2008) by Catic (2011).

Zimmerman (2008) defines a PLM architecture as an IT-centric enterprise architecture, comprised of several different layers or sub-architectures. These layers are described in Figure 12. Describing PLM architectures, adapted from Zimmerman (2008) by Catic (2011).. In more detail, the levels may be described as the follows:

- *Business Strategy and Objectives* includes the visions and goals of the architecture. It answers the question "What is to be realized by the company?"
- *Business Processes* constitutes the collective processes of the organization and their relationships, which are needed to realize the business strategy and objectives
- *PLM Workflows* are descriptions of how engineers will work with the PLM solutions in order to support the processes
- *Strategic Capabilities* make up basic functionality of the PLM system and are used in the workflows, helping to realize the business processes.
- The *Information Architecture* describes how the information that is used in the PLM workflows and by the strategic capabilities is structured and modeled
- The *Application Architecture* describes the relationship between application and task
- The *Infrastructure Architecture* constitutes the hardware that stores data and runs applications (Catic, 2011).

2.5.4. Wiki solutions

Wiki solutions for capturing abstract generic knowledge such as used in a technology platform concept has proven feasible within software development (Decker et al., 2005), construction (Dave and Koskela, 2009) and mechanical product and production development (Corin Stig and Bergsjö, 2011). A wiki is a website structure consisting of an indefinite collection of pages in which a registered user can add or edit pages in a built-in editor. The best-known example is Wikipedia, an encyclopedia that has grown organically based on small contributions of information from users worldwide. The main ways of finding information in a wiki are through an efficient search engine and through links between pages. A version tracker is normally present to prevent any loss of information from improper use.

For supporting a technology platform, the content of the pages in a wiki may include basic information about how technology works and where it is used, the current maturity for different applications, the company's plans for future use of the technology and contact information to experts in the field.

Due to the loose structure of a Wiki, a process is needed to guide a designer in creating knowledge according to a structure. Also, knowledge must be a clear deliverable in the process (Catic and Malmqvist, 2010). The loose structure is also what makes it powerful for use in early development phases, during which information is more text-based and where rigid relation databases of a PDM system fail to provide adequate support.

2.5.5. Systems integration and SOA

As a system solution, PLM is an integrator of tools and technologies to facilitate swift and accurate information flow throughout the product lifecycle (Terzi et al., 2010). Product Data Management (PDM) systems may well be one of the components of the PLM architecture (Abramovici, 2002), but are not considered to constitute the entire PLM strategy.

System integration is an essential issue in PLM. Burr et al. (Burr et al., 2003) introduce several ways of integrating systems in a PLM architecture. Two concepts are further refined (Burr et al., 2004): a*ll-in-one-integration* and *best-in-class-integration*. The first is an approach with only one database to which all the systems are connected. The latter, well suited for distributed sets of data, features multiple databases for each discipline with their respective expert tools connected. These databases are integrated through an integration software component.

Bergsjö et al. (2006) refer to Burr et al. but offer four different integration approaches: *best-in-class*; *all-in-one integration*; *one system as integrator* and *peer-to-peer integration*. One system as integrator is a mix between best-in-class integration and all-in-one integration in which some tools are connected to the main database, while others go through their respective expert databases. Finally, peer-to-peer integration is an approach whereby each software component has an individual interface to each and every other software component with which it needs to communicate.

Bergsjö et al. (2008) extends the different types of integration possibilities with Service Oriented Architecture (SOA) as a possible way to integrate distributed sources of information, such with platforms that are integrated across the lifecycle.

2.6. Conclusions of the frame of reference

There are a great deal of different ways to support efficient development. The aim of most tools and processes is to bring about a single design of high quality at low cost. However, few initiatives really consider the phenomena of design reuse. Those that do, focus on feeding manufacturing with a low number of different parts, which certainly saves manufacturing cost but has little effect on development time.

While set-based concurrent engineering and platform-based development share a great deal, such as design reuse and modular architectures, few studies have considered using set-based design throughout the platform lifecycle. Most tools and processes focus on execution through optimization, while the preparation stage is overlooked. In their paper from 2004, Burr et al. (2004) conclude that the contemporary engineering systems cannot fully support the emerging engineering methods. They refer to ways of working concurrently by linking different steps in the lifecycle. When it comes to design reuse, especially across the lifecycle, it is reasonable to assume that current support needs improvement to fit these engineering processes and methods.

The scope of the research presented in this thesis aims to address the issues above. How platform-based design that saves development time can be supported is of particular interest. Further, the thesis means to investigate current tools and processes that can be used to support set-based design for platforms, and how to achieve a coherent support throughout the product lifecycle. The problems will be addressed from an engineering-to-order point of view. Platforms which will aid developers of product and production systems will be favored over configuration and optimization of pre-designed building blocks.

3

Research Approach

Different disciplines have varying approaches to obtain credible research results. Traditionally, Engineering contrasts to Social Sciences when it comes to opinions on how to find accurate knowledge. This chapter describes the research approach, why the particular approach was chosen, and how it was adopted to fit the setting of this research.

3.1. Design and research in design

Engineering Design, or just *Design*, has many definitions, with a majority describing design as the process of bringing about a product based on a need, product idea or technology. The final result is the knowledge about how to manufacture the product to fulfill the perceived needs. It includes activities such as requirements specification, concept and detailed design, process planning and manufacturing systems design, and it often involves both individuals and enterprises. The final result may be a tangible object, but it may also be a service or a process.

Research within design is an area that has grown in importance over the past thirty years, resulting in a deep body of knowledge spanning a broad area of disciplines. The area relates to Engineering Science, which is broad in itself, connecting disciplines such as thermodynamics, materials and mechanics. The paradigm that forms design research began to grow with the increasing importance of managing complexity of designs in addition to economical aspects of engineering. The following was concluded:

> *"There is a big gap between scientific research and the engineering product, which has to be bridged by the art of the engineer."(Gibbons and Johnson, 1982).*

Thus, now design research is a field of its own, connecting to Engineering Sciences. There are also several other areas used in design research, for example Behavioral Science, which plays a pivotal role in understanding how engineers work, which is key to being able to support the design process. Further, designers rarely work alone, making the Social Sciences a neighboring field.

The definition of design research and how it differs from similar fields is ongoing (Blessing and Chakrabarti, 2009). Blessing and Chakrabarti (2009) contribute with their definition of design research by claiming two different views on design research: the development of *understanding* and the development of *support*. *Understanding* is to formulate and validate theories about design and related phenomena, including all connected facets (people, product, knowledge/methods/tools, organization, micro-economy and macro-economy). To develop *support* for design is to develop tools, methods and knowledge to support the design practice. Thus, although related to Engineering Science, design research must consider the human aspect of the design process (Michaelis, 2011).

As with any other research, there are several different methods and frameworks that help researchers to create legitimate and accurately obtained results.

3.2. Available frameworks and methods

On a high level, a research approach can be supported by a framework that defines the general process and that provides structure to the research. Closer to each activity within such a process, methods are used to aid the researcher in securing the quality of obtained results. Both the framework and methods used are accounted for below.

3.2.1. Research Frameworks

Design Research Methodology (DRM) is a framework developed by Blessing and Chakrabarti (2009) and focuses not only on aiding the process of providing understanding of design, but also to aid in the creation of support for conducting better design. Figure 13 depicts the prescribed p*rocess* with the m*eans* and o*utputs.* The research clarification stage aims at defining worthwhile and realistic goals for the research, using literature as the main source of information.

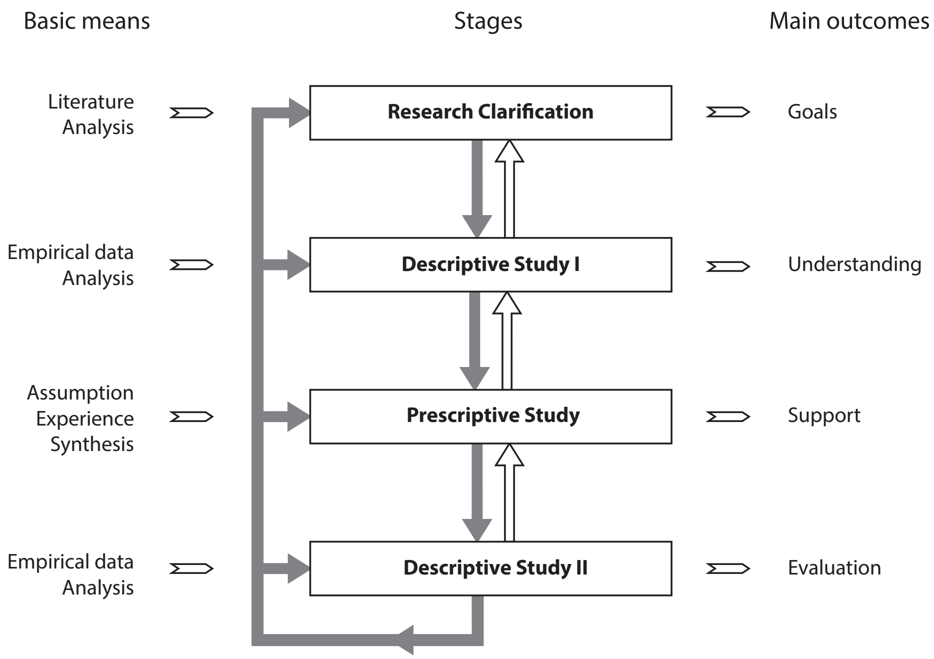

Figure 13: The DRM framework, redrawn from Blessing and Chakrabarti (2009)

In the Descriptive study I stage, the researchers describe the current situation, and the factors that could be addressed to improve the situation. The output is a better understanding of the situation; an as-is description. The Prescriptive Study that follows addresses those factors depicting h*ow* to affect them to improve the as-is state into the d*esired state*. Descriptive Study II aims at evaluating the true effects of the support implemented.

Each stage contains a selection of activities and deliverables to aid the researcher. Furthermore, the stages are not designed to be executed in a strict sequential manner, but loopbacks are instead encouraged. Taking into account that research projects are different in their character, Blessing and Chakrabarti define seven types of research within their framework that differ by the research emphasis placed on different stages.

Another approach that considers understanding design, and supporting it is *the Spiral of Applied Research* by Eckert et al. (2004). Their spiral model consists of four activities, any one of which can initiate a research project:

- Empirical studies of design behavior
- Development of theory and understanding
- Development of tools and procedures
- Introduction of tools and procedures

The white boxes in Figure 14 represent these activities. Together with intermediate evaluation activities they form eight different research objectives. All eight activities use available insights, information and requirements and create new ones.

Jørgensen (1992) presents an approach for applied research (see Figure 15). Applied research has two starting points: the problem base, typically a phenomenon observed in reality, for example an industrial need in addition to a theory base, where the knowledge

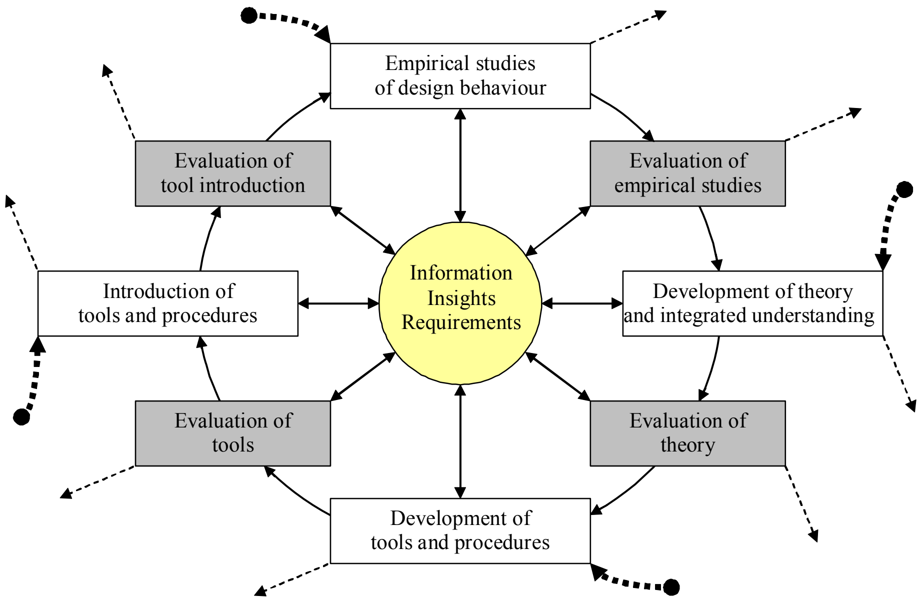

Figure 14: The Spiral of Applied Research: the eight types of research objective

gap is established by studying the knowledge base in literature.

The theory base is synthesized into models that are either descriptive or prescriptive. These models are tested and validated against analyzed results from the problem base and may result in new scientific acknowledgements. Though the process seems sequential, it is not as the work of analyzing, synthesizing and synchronizing between the two tracks is highly iterative (Jørgensen 1992). At the end of the research, the new scientific acknowledgements come closer to implementation in industry.

3.2.2. Methods for collecting data

There are numerous approaches for collecting data within design research. The type of results generated depends largely on the approach. *Case studies* rank as one of the more common approaches within design research. According to Yin (2003), a case study

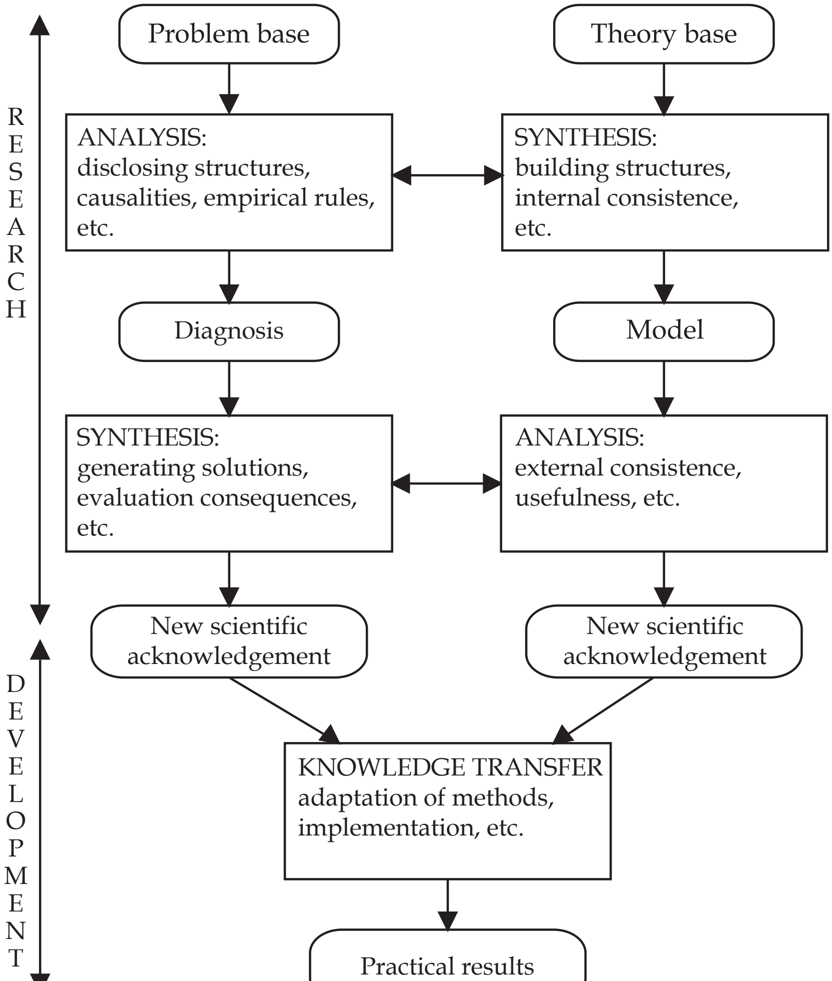

Figure 15: Framework for applied research, redrawn from Jørgensen (1992)

> *"investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident".*

This definition has two distinct components. First, a case study is about studying something contemporary as in that it is occurring now so that it can be observed, or that there are people alive that can tell of it. It contrasts with studying historical events, which would require another research approach.

Second, the boundaries between the phenomenon we want to study and the context in which we find it are not very clear. In a setting where a phenomenon can be isolated and studied without interference from its context, experiments would be preferred. However, if relevant parameters cannot be manipulated and phenomenon distinguished from the context, case studies are preferred. However, a case study is not a data collection method but rather a setting in which data can be collected.

Numerous methods for collecting data within the setting of a case study are available, one of the more common being the interview. Interviewing as a research method usually involves the researcher posing questions to a respondent, who hopefully gives answers. Robson (2002) and Blessing and Chakrabarti (2009) agree on three different classes of interviews. The *fully structured interview* is characterized by questions that are exactly worded and asked in a predefined order. The *semi-structured interview* gives more room for improvisation in that the questions are predefined, but the phrasing and order of questions can vary to get a better flow in the interview. Questions can also be given an explanation or excluded if found irrelevant for a particular interviewee. Finally, the *unstructured interview* is a conversation-like interview where the interviewer has a general area of interest that is discussed with the interviewee.

3.2.3. Methods for building theory and models

In the endeavor to create a supporting design processes it is common to build models of reality to condense the phenomena to be supported. These *phenomena models* describe aspects of reality essential to the design situation at hand. The phenomena models can, when appropriate, be developed into formal *information models* and later be developed into *computer models* and tools to prescribe a way of conducting design (Duffy and Andreasen, 1995).

It is important to notice that implementation of any model or tool will affect reality. Therefore, models continually evolve. Each model may at any given stage be evaluated against previous models (or *alien models* – models on which the present model is not based, but that if implemented might still affect reality) to enhance the understanding of reality, but also to enhance the models themselves. Figure 16 describes the development of models in design research. The loop-backs represent the mutual relationship between model and reality.

3.2.4. Validating and verifying the results

Validity within research refers to "the correctness, or credibility of a description, conclusion, explanation or interpretation" (Maxwell, 1996). Obtaining validity in research

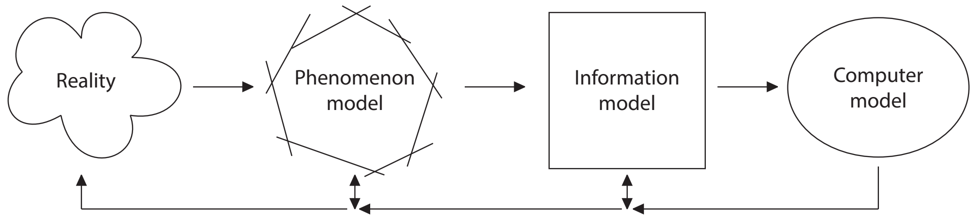

Figure 16: Design modeling research approach according to Duffy and Andreasen (1995)

is not about finding absolute truth, but rather providing grounds to distinguish credible results from those that are not credible. Within design research, verification differs from validation, in that verification focuses on the question *"Did we do it in the right way?"* whereas validation focuses on the question "*Did we do the right thing?*"

Buur (1990) expresses it differently by presenting two verification approaches within engineering design. One approach, *verification by acceptance* focuses on having new scientific contributions (e.g. axioms, theorems, models, methods or tools) accepted by experts within the field.

Another approach, *logical verification*, focuses on the internal consistency and completeness of the results. The results need to be consistent in the sense that there are no conflicts between individual elements. Completeness is achieved when all phenomena previously observed can be explained or rejected based on the results. Logical verification is also to establish agreement with known methods and theory. A design method must also be able to support specific design problems.

Olesen (1992) adds that research can be considered to be true when it can explain phenomena found in reality too, not only phenomena found in theory. Further, besides being accepted by research community and industrial practitioners, the research has to be applicable in a real industrial setting. The research result has to have newness, i.e. has to provide new approaches or new realization.

3.3. The applied approach in this research

The research has been performed at the Wingquist Laboratory Excellence Centre. The centre has several industrial participants where research results can be tested and evaluated. The participants contribute with their time and expertise, opening up their organization as a laboratory for researchers within the centre. In return, they benefit from the results of the research.

The context in which the research is performed is important to the design of the research approach. In this thesis, the context is characterized by the consideration of both a *research challenge* and an *industrial opportunity*. With reference to Jørgensen, real life industrial observations have created the problem base whereas the theory base builds upon research performed within the Wingquist Laboratory Excellence Centre, in connection to the industrial participants, as well as state-of-the-art research within the field.

3.3.1. A framework for this research

DRM as a framework for this research works well because it does not only focus on understanding the problem, but also on finding support for design. The purpose of the research thus matches the framework to support it. Though DRM will support on a high level, other approaches are applied to structure the results of the research.

The SAR as devised by Eckert et al. (2003) advocates seizing the opportunities that arise in large research projects, even though they align perfectly with the initial goals of the project. This approach is crucial due to the industrial collaboration serving as source for empirical studies. The majority of the research has been carried out in close collaboration with the industrial partners of the Wingquist Laboratory over several years during which new information, insights and requirements have arisen. As the collaborating partners were involved long before the research presented in this thesis started, there is a great knowledge base imprinted in publications by, and minds of, the individuals who have been working, and are working, in the research projects, both from academia and industry. Thus, comprehensive descriptions of the as-is state exist. As a consequence, Descriptive Study I can be based on studying already published literature, in combination with unstructured interviews with participants of the Wingquist Laboratory - the Descriptive Study I can be review based, allowing more focus to be placed on the Prescriptive Study.

The DRM framework describes seven different types of research projects (Figure 17). The types are designed to match a variety of research projects. According to previous reasoning, Type 3 would match the character of the research presented in this thesis.

3.3.2. Case studies with interviews for information collection

Since most research questions in this thesis relate to the design process, aiding the designer in creating better designs, it is feasible to adopt methods from the Social Sciences, rather than solely applying a quantitative approach, as is so often done in the Engineering Sciences.

As the research questions stated in the beginning of this thesis are exploratory rather than quantifying, case studies have been adopted to serve as the main source of informa-

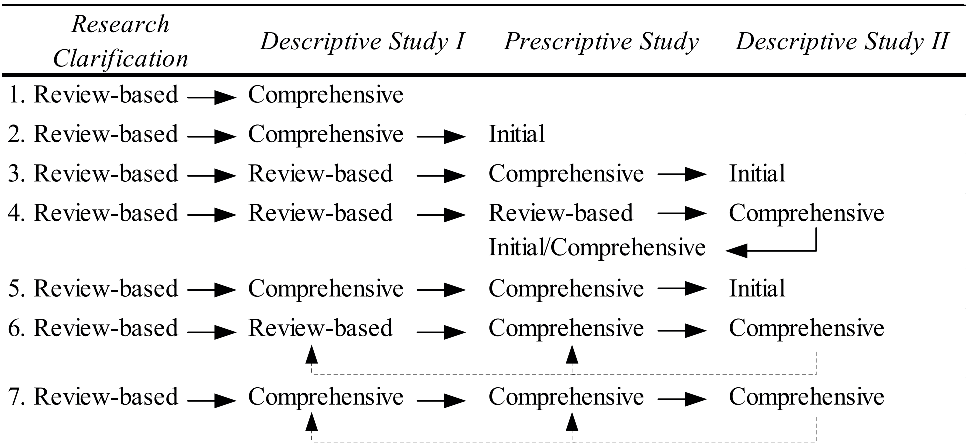

Figure 17: Types of design research projects and their main focus, redrawn from Blessing and Chakrabarti (2009).

strategy	form of research question	requires control over behavioural events?	focuses on contemporary events?
experiment	how, why	yes	yes
survey	who, what, where, how many, how much	no	yes
archival analysis	who, what, where, how many, how much	no	yes/no
history	how, why	no	no
case study	how, why	no	yes

Figure 18: Different research strategies are used for different research situations (Yin (2003)).

tion for the left leg of Jørgensen's framework. Case studies are excellent vehicles by which why and how questions may be answered. A variant of the how question is "how much", which in contrast to the original "how" may favor surveys. However, from a wider perspective where the "how" or "what" questions are more explanatory than quantifying, case studies are preferable (Yin, 2003). These strategies are outlined in These strategies are outlined in Figure 18.

Semi-structured interviews are widely used in qualitative research when trying to form an understanding of a particular situation (Robson, 2002). This research adopts the concept of semi-structured interviews for all stages. In the Research Clarification stage and the review based Descriptive Study I, interviews are used to create an understanding of the problems that are faced by today's industry, as a complement to the literature study.

In the Prescriptive Study, demonstrators are created to illustrate cases from industry. The demonstrators are based on the perceived needs of industry and realized through prescribed methods and tools; they are used as mediating objects in verifying and validating the results through verification by acceptance.

3.3.3. Validating the results in applied research

The concept of internal verification and external acceptance concurs with Jørgensen's (1992) model of the problem base and the theory base. Discussing and sharing the analyzed results with experts helps to achieve the external acceptance. More explicitly, all papers have been undergoing peer reviews as part of the publication process.

Papers A, B, D, E, and G have been presented at conferences where experts within the field had opportunities to express their opinions about the results. Further, as a part of the external verification process, the results have been presented to the industrial partners of the Wingquist Laboratory at workshops, at result days, and as full papers for peer review prior to submitting them for publication. Demonstrators were used as mediating objects in the verification process.

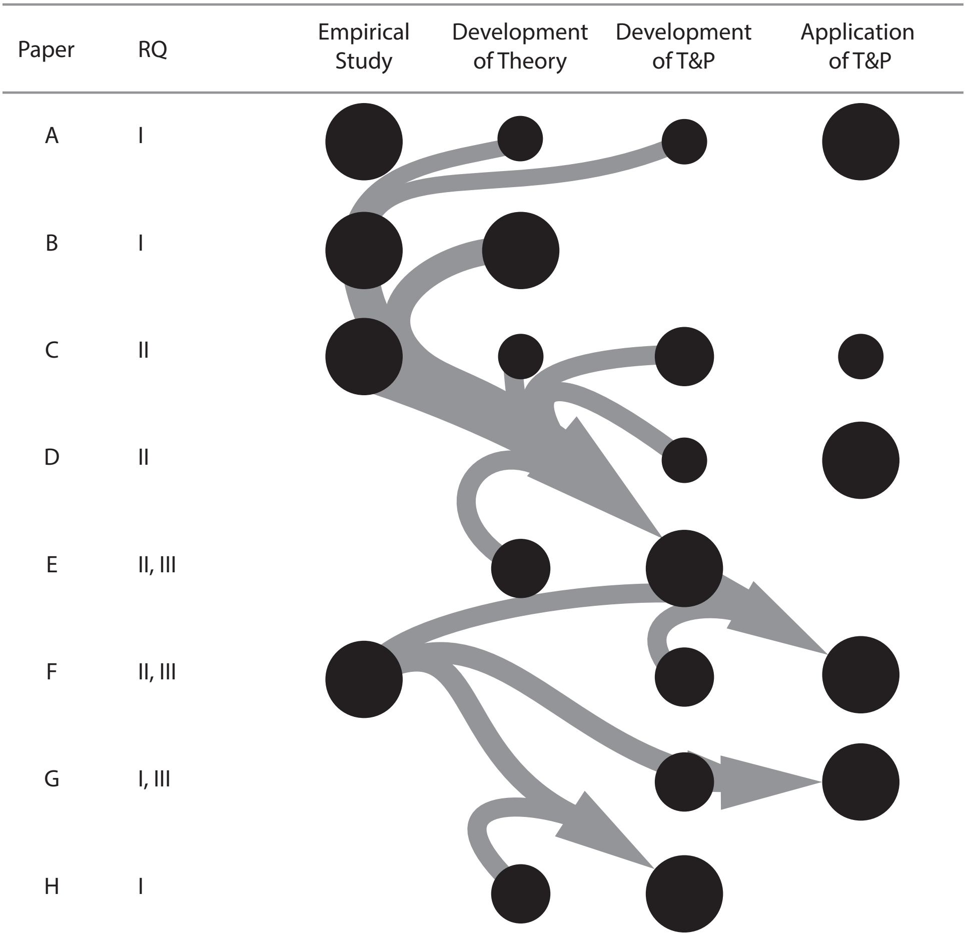

Figure 19: A schematic picture of how the result of the research relates to different research objectives and the research questions. T&P stands for tools & procedures.

The literature base on which this research rests is accounted for in Chapter 2. There is a large base of state-of-the-art in IT- and process support for product development, in addition to the domain of Engineering Design. Known IT- and PLM architectures models were used. Phenomena models, and in some cases information models, exist for several of the concepts used in this research. In those cases, the focus was on creating useful computer models in agreement with existing phenomena- and information models. The research was also about creating new phenomena models in which case these were verified to be complete and consistent through researching earlier studies and building upon known models.

3.3.4. Structuring the results of this research

As proposed by Eckert et al. (2003) the SAR must be adapted to each research project. On that note – and inspired by Michaelis (2013) –, four activities are defined in which the results of the research are structured. These categories are *Empirical studies of design* *behavior, Development of theory and understanding, Development of tools and procedures,* and *Application of tools and procedures*. They correspond to the objectives of Eckert et al., except for the last objective. Application is used instead of Introduction because the studies have focused on applying tools and procedures in demonstrators, rather than evaluating the implementation in daily work. *Tools and procedures* are here interpreted as artifact models, IT tools and engineering processes and methods.

The relationship between these phases, the papers and research questions are illustrated in Figure 19. The flow of gray arrows illustrates how the results from the papers have cascaded throughout the project. Note that the results from a*pplication of tools and procedures* are not fed back. The reason is that application of models has been used to verify results rather than produce new ones.

4

Results

This chapter summarizes the papers that are appended in the back of this thesis, the same papers on which all results of this thesis are built. The summary presented in this chapter focuses on the results gained throughout the studies, leaving less room for how the results were obtained and the theoretical context in which they reside. The full descriptions can be found in the appended papers in the back of this thesis.

The papers each contribute with pieces to answer the research questions. Paper A addresses RQ1 by applying current tools to development platforms. Paper B addresses the same research question by applying a similar set of tools to a new case, and to draw concussions from theory as to what the current tools lack to fully support development platforms. Paper C addresses RQ2 by applying the tools from Paper A and B to cross-lifecycle platform development. It also suggests procedures for integrating technology and product platform development. Paper D addresses the same research question and completes the reasoning in Paper C. It does so by suggesting a representation of the technology platform, which can easily integrated with the product and manufacturing system platforms.

Paper E addresses both RQ2 and RQ3. It does so by taking the theory and procedures from Paper C and D and enhancing it with set-based concurrent engineering. Paper F verifies the reasoning in Paper E in a real case study and develops the procedures and tools further to cover a larger area of the lifecycle.

Paper G addresses RQ1 and RQ3 by completing the previously developed procedures and tools with yet another part of the lifecycle. Here early platform development stages are brought to attention using set-based concurrent engineering and the previously developed artifact model. While Paper A and B focused on the tools for platform configuration, Paper H adds to RQ1 by presenting a detailed systematic process to how the artifact model can be used. This procedure draws from the case in Paper F.

4.1. Paper A – Platform execution using PLM

The paper presents a case study performed in cooperation with a Swedish car manufacturer. During the study, a PLM architecture was established to support *execution of a product platform* by configuration of product variants. The execution was based on a product platform, while some manufacturing aspects were included in the evaluation of the generated variants.

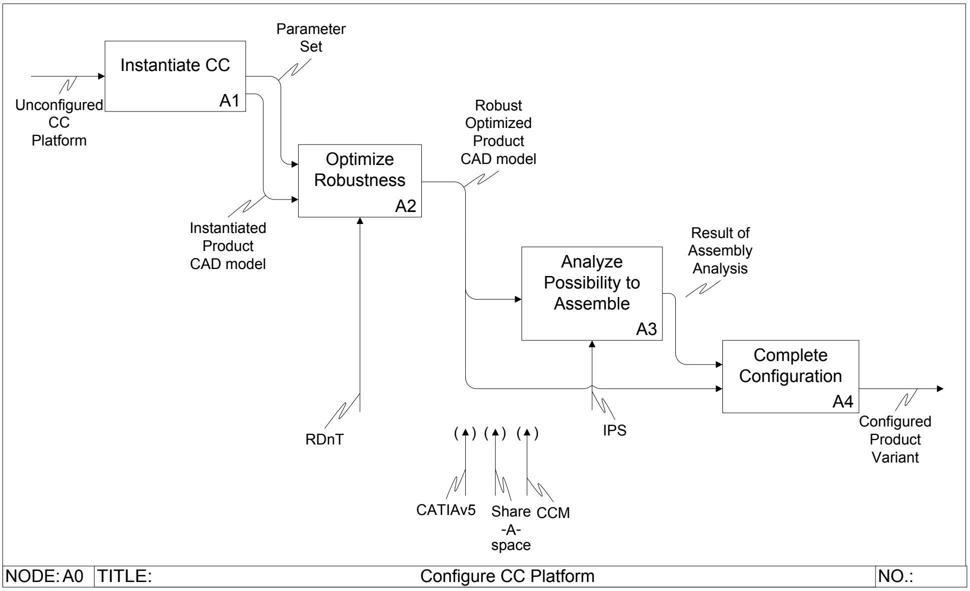

Figure 20: The configuration process presented in IDEF0. The process is based on the requirements that product needs to fulfil (Paper A)

The platform consisted of configurable system models as reusable platform elements, modeled using the Configurable Component concept. The full extent of the CC concept was not used – the study was instead aimed at configuring geometrical system interfaces (mating geometries and locating schemes).

The aim was to let the PLM architecture do the configuration and calculations necessary to ensure a functioning product variant, presenting any design decisions to the designer together with the proper amount of information to enable them to make informed decisions. Two geometrical requirements were assessed, resulting in the use of two different CAE tools to optimize and evaluate the configured product variant.

To be able to create a PLM architecture in its entirety, several different aspects need to be considered. Svensson et al. (1999) stress *Processes*, *Information*, *Systems* and *Roles* as factors to be considered in a PLM architecture. The process for going from an un-configured product platform to a product variant that fulfills both requirements is shown in Figure 20. It is this process that the PLM *information architecture* as well as the PLM *application architecture* shall support. The designer's *role* is to make any qualified design decision required to complete the configuration.

The *information architecture* consisted of a meta model of the car door, modeled with CCs and a CAD 3D model of the generic concept of the car door consisting of three parts (see Figure 21), all parameterized and configurable. The *application architecture* consisted of a configurator and modeling tool (CCM), a PDM system (Share-A-space) to store meta data and CAD models, a CAD tool to manage CAD models and two CAE tools to perform optimization and analysis of the parts. The tools were integrated using a point-to-point approach.

The contribution of Paper A to RQ1 can be summarized as:

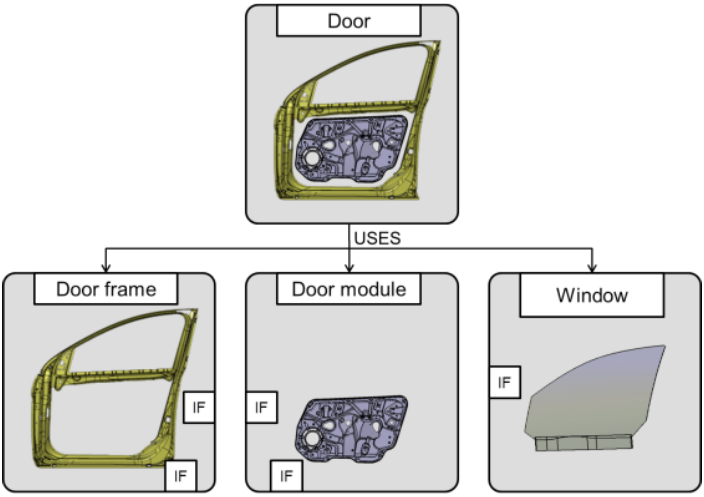

Figure 21: Graphic model of the platform modeled using CCs (Paper A)

- It is possible to create a PLM architecture that supports automatic configuration of geometrical system interfaces of platform elements modeled as configurable components.
- The requirements put on the product highly influence the configuration process and thus, the needed PLM architecture components.
- A point-to-point solution for integrating the software is beneficial for a case study, but when the requirements change and new analyses are needed, e.g. for a product aimed at different market, point-to-point integration would be too rigid. A more general integration approach, such as a SOA, would be preferable.

The case study has a clear aim towards the execution of the platform, i.e. creating product variants based on an already defined platform. A certain amount of platform preparation was necessary to perform the subsequent execution, but in order to prescribe any processes or support for platform preparation, more studies are needed.

4.2. Paper B – Platform lifecycles

The study that provides the backbone for paper B is based on two sources, a literature study reviewing literature on platform-based design in addition to a case study of a supplier to the aerospace industry, Swedish Volvo Aero Corporation (now GKN Aerospace Sweden AB).

The aim of the literature study was to find areas where platform-based design could be applied but where there was yet no support. The study uses three different dimensions: lifecycle stage, PLM architecture level, and platform abstraction level. In these three dimensions there are gaps, e.g. there are no studies on the connection between abstract platforms – such as suggested for the aerospace industry – through platforms represented by generic concepts, to more concrete platforms represented as readily designed components and subsystems.

The second part of the study addresses the gaps found in the literature study that matches a need of the company studied. The case company has low volume production and cannot benefit from reusing parts. The more abstract alternatives, such as reusing technologies, are instead more feasible. However, abstract representations of products and technologies cannot support product development in later stages (e.g. detailed design) during which systems need to be increasingly defined. At the final stage, the product representation has to be concrete enough to be produced. Thus, the case study elaborates on the possibilities of bringing aspects of the Volvo Aero product platform to a less abstract level to better be able to reuse design in more lifecycle stages than technology development. In this way, current PLM support could be utilized, rather than new customized systems to support abstract models. In order to do so, a PLM architecture was established and the product modeled as a platform using the configurable systems to allow configuration using the CC concept.

The contribution of Paper B to RQ1 can be summarized as:

- The PLM architecture from Paper A is verified in a real case proving the suggested tools, model and configuration process suitable for configuration of platforms, yet leaves much to wish for to support the actual development work.
- The platform approaches in today's literature do not cover the need to support holistic platform development across all stages of the lifecycle and for all levels of abstraction. They fail to do so because there are gaps in support of specific areas, for example in the stage of service and there are no studies on how platform approaches can be combined into covering the entire lifecycle.
- The study gives a first insight into the fact that platforms based on configurable system elements, modeled with the CC concept, can be used as a bridge between abstract descriptions of platforms (e.g. technology platforms) and concrete descriptions (e.g. part- and module-based platforms)

4.3. Paper C – Integrated technology and product platform

This paper proposes an integrated approach to using technology platforms and product platforms. The approach provides a holistic perspective on how to leverage benefits from reuse as well as leveraging strategic investments from early phases of technology development to the late product variant configuration phase. It addresses technology platform execution as a knowledge source for product and production system platform preparation. Finally, the prepared development platform is executed to create a set of product variants.

4.3.1. A holistic approach

A *technology based integrated platform* (i.e. a development platform) is defined as a product platform based on configurable systems elements – in this case modeled using the Configurable Component concept created using a technology platform with generic technology descriptions as platform elements which may also be referred to as a *development platform*.

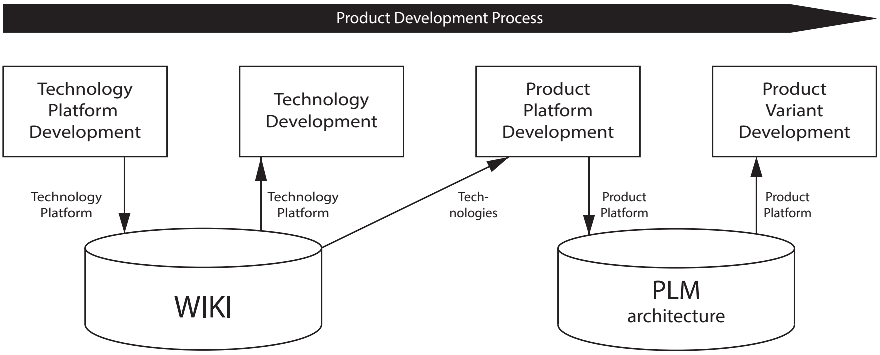

Figure 22: A simplified product development process using platforms and suggested support. In reality, the different parts are developed concurrently and more complex relations exist (Paper C)

The processes for creating and using a technology platform are different from creating and using a product platform. As a consequence, they require different support. Figure 22 shows a simplified picture – in reality there are loopbacks and iterations – of the relation and information flow between the processes. The cylinders represent what type of support that is suitable for the respective processes.

The technology platform is supported through a Wiki solution (Figure 23) in which descriptions of the technologies and their application areas can be found, together with trade-off curves from technology development, contact information to the technology owner, etc. The product platform is supported by a PLM architecture that is partly derived from the technology platform depending on the requirements posed on the product.

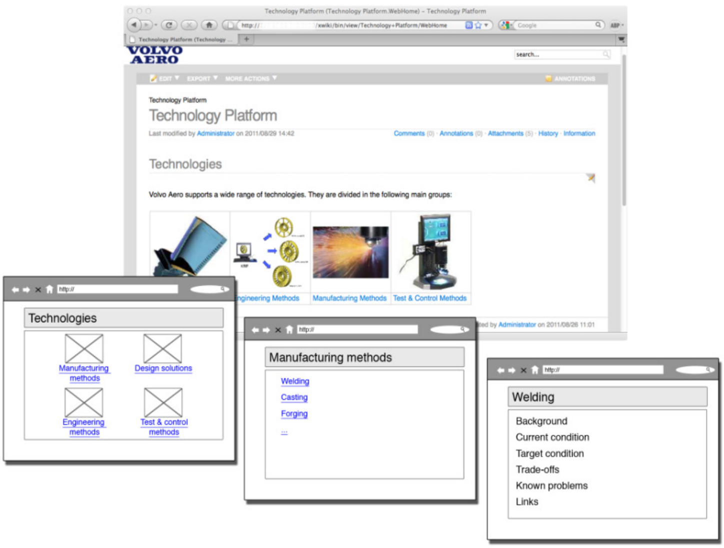

Figure 23: The Wiki solution contains generic technology knowledge and supports the technology platforms (Paper C).

The proposed process below is used to *prepare* the product platform, using the technology platform Wiki (see Figure 23). The result is a product platform based on configurable system elements modeled using the CC concept, and a PLM architecture able to generate product variants and automatically analyze requirements imposed on the product. The steps of the preparation process are:

1. Model the products as Configurable Components
    a. Model the functional requirements and constraints of a concept.
    b. Find design solutions (organs and physical parts) to match the functional requirements.
    c. Use trade-off curves, limits and possibilities from technologies to define CC bandwidth.
    d. Use technologies to define knowledge gaps that need to be filled. Identify technology development efforts required to bring the product to market.
2. Define activities
    a. Define the solution space by selecting parameters to vary.
    b. Match requirements to analysis activities required to analyze the fulfillment of the requirements.
3. Define the configuration process
    a. Find supporting activities, such as data transfer and storage.
    b. Find a feasible sequence of activities and define data flow.
4. Create systems architecture

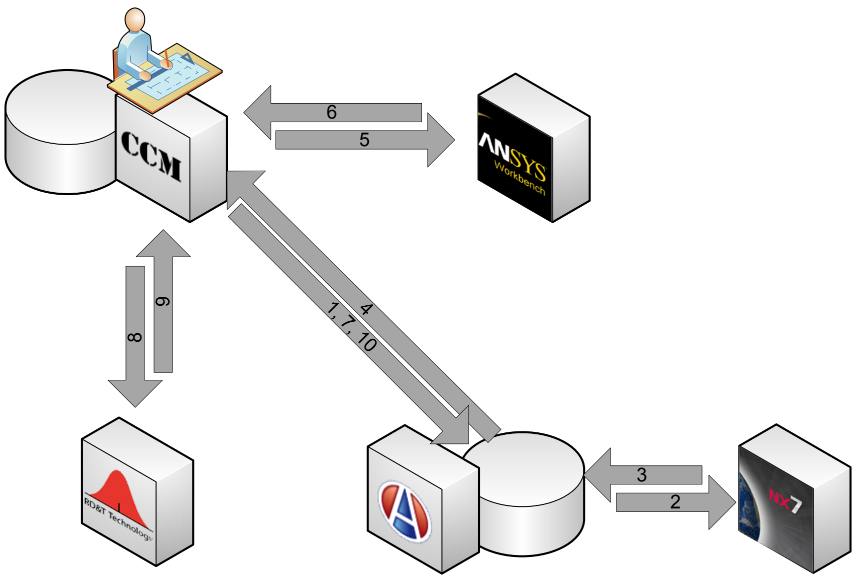

Figure 24: The PLM architecture (Paper C)

a. Map activities to IT tools or combinations thereof using engineering technologies from the technology platform.

b. Define interfaces between tools.

4.3.2. A case study of technology-based integrated platforms

The company involved in the case study, Volvo Aero Corporation in Trollhättan, Sweden, is a manufacturer of turbofan engine framing. As a sub-contractor in the aerospace industry the Volvo Aero Corporation provides sheet metal goods for several major jet engine manufacturers. One of their business goals in implementing platforms is to quickly be able to answer quotes from their customers with high precision. If they manage to accomplish this, they can quote a low enough price to win the deal, while at the same time staying on the right side of their margins.

The study starts with a product platform prepared according to the above suggested process. The goal is to arrive at an estimate of the ability of the current design to meet customer requirements. This is achieved by generating a number of product variants and testing them with CAE tools. A PLM architecture is created based on the requirements that need to be assessed, resulting in a configuration process and the application architecture seen in Figure 24.

The configuration is done automatically, and initiated by the configurator CCM, based on the generic concept modeled using the CC concept. The two parameters chosen for this case are varied so that six different configurations are generated. These concepts are then created in the PDM system (1), and the correct CAD files are generated, based on the generic concept model (2). The respective CAD files are connected to concepts in the PDM system (3). The PDM system then sends back the six different CAD models to CCM (4).

The first analysis activity is done by Ansys, which upon request (5) uses the CAD files

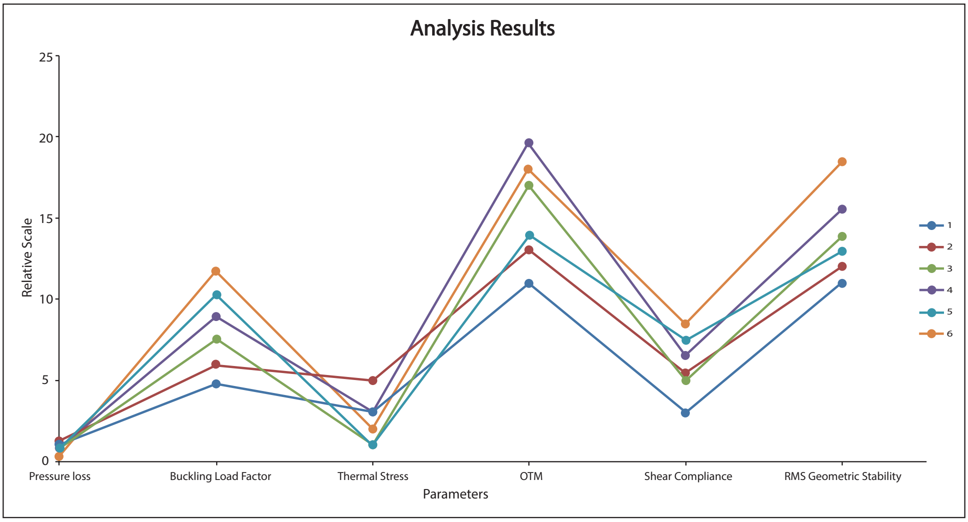

Figure 25: the results are displayed to the designer as graphs, each graph marking a concept (Paper C).

of the different concepts to perform a multi-criterion analysis. These analyses comprise *pressure loss*, *buckling load factor*, *thermal stress*, *over-turning moment (OTM)* and *shear compliance* for all six concepts. As the results are returned (6), they are both stored in the CCM database for later displays, as well as in the PDM system (7) for reuse on other occasions.

Subsequently, the second analysis is performed. The RD&T robust design tool carries out the flush analysis to determine how well the concepts perform if manufacturing variations are considered. The analysis is called for (8), and the result is returned (9) and stored in both CCM and Share-A-space (10).

The result of the analyses is displayed in Figure 25. Each concept is represented by a polygon. The set of concepts may be narrowed down by eliminating a concept that performs too poorly on any requirement, or a concept that is out-performed by another concept on all points (e.g. blue versus red concept in Figure 25).

The contribution of Paper C to RQ1 and RQ2 can be summarized as:

- Platforms are both prepared and executed. This paper presents a PLM architecture for execution, uses the CC artifact model for both preparation and execution, and outlines a process for preparation completed with the execution process to generate product variants.
- The paper also presents a Wikipedia solution to accommodate a technology platform. The technology platform is used as input, thereby being executed, in the preparation of the product platform. A holistic approach is manifested by integrating the lifecycle stages of technology and product by means of the preparation process, the artifact model used in the product modeling and the tools used to prepare the platform.
- For a company like Volvo Aero, the product platform configuration process may be used not only as an engineer-to-order tool, but also as a tool to find performance gaps of the products and technologies. It can be used to identify customer requirements that they cannot yet fulfill, areas for expansion of the product bandwidth or where new technologies are needed.
- With a better overview of both technologies and product platforms there will be opportunities for more informed decisions on introducing technology and the risks of deploying them for different applications. This will in turn facilitate concurrency between technology development and product development.

4.4. Paper D – Trade-off curves as integrator

Paper D elaborates on the connection between a technology platform and the product and manufacturing system platform. From a PLM perspective, this link may be represented by trade-off curves. Numerical representations of trade-off curves may be used and integrated into a PLM architecture similar to CAE systems. Consequentially, part of the knowledge in the technology platform may be used directly in the execution of the product and manufacturing system platform.

The trade-off curves in Figure 26 and Figure 27 are the result of virtual testing

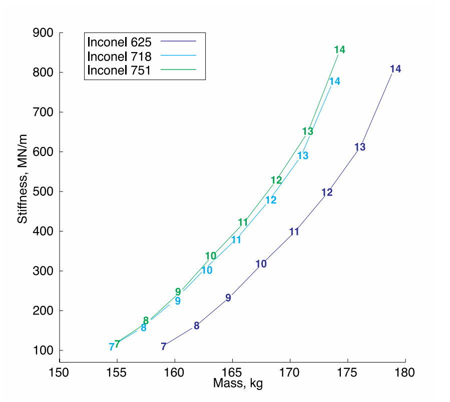

Figure 26: Trade-off curve showing a trade-off between mass and stiffness with current available technologies. The different lines represent different material (Paper D).

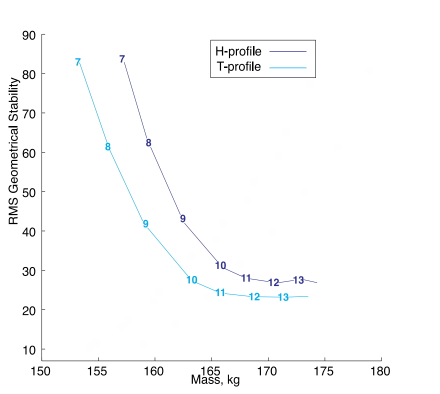

Figure 27: Trade-off curve showing a trade-off between Geometric Stability, Expressed in RMS and Mass, using current manufacturing technologies. The different lines represent different Manufacturing Concepts (Paper D).

of a range of different concepts. The current material technologies allow the concepts a bandwidth of performance. The information previously acquired by analyzing each concept (for example in Paper A) using CAE tools is instead retrieved by studying these trade-off curves. The curves are implemented as columns and rows in an excel database which is connected to a PLM architecture. The architecture is used for *execution* of the product platform of a Turbine Rear Structure (TRS), seen in red in Figure 28. The output from execution of the platform using these trade-off curves is similar to the output produced in Paper C. The proposed IT architecture does show that it is possible to create an IT support that aids the designer in the process of

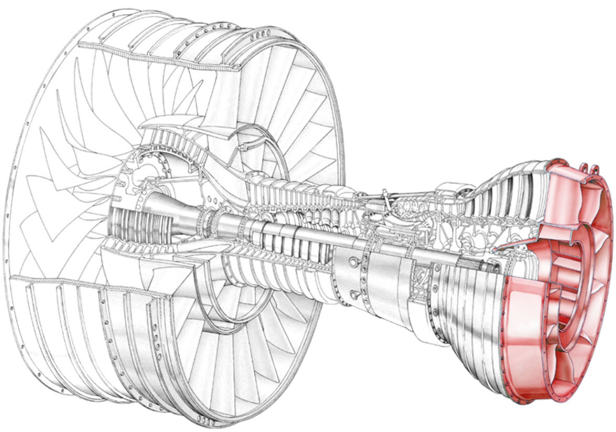

Figure 28: A jet engine with the Turbine Rear Structure (TRS) highlighted in red (Paper D)

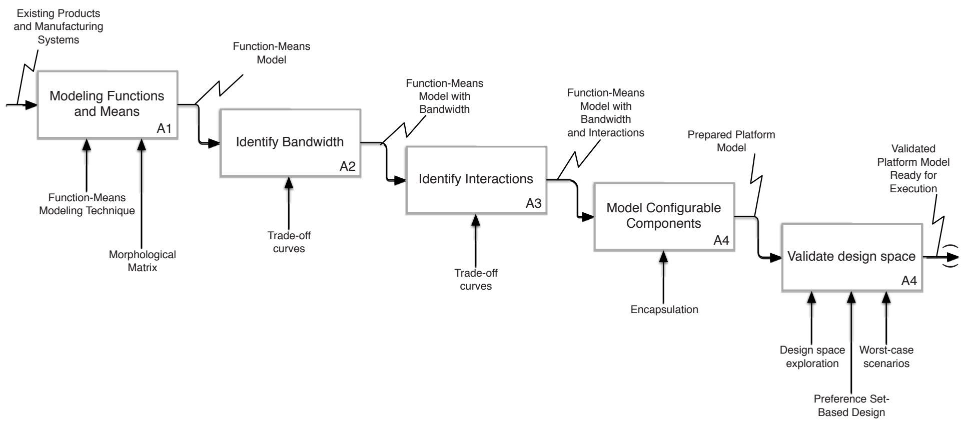

Figure 29: A process for platform preparation using the CC concept and trade-off curves to model the platform (Paper E).

converging designs towards a feasible solution. Through combining trade-off curves and CAE tools, in comparison to just using extensive analyses, the lead-time for analysis can be substantially reduced. However, though IT-systems supposedly do a lot of the work here, the use of platforms with trade-off curves will require a whole new way of working. Besides the implementation of new IT systems, using platforms requires the processes to be front loaded, preparing platforms in order to later be able to harvest the fruits of them.

The contribution of Paper D to RQ2 can be summarized as:

- This paper adds to RQ2 by proposing a way of representing parts of the technology platform as trade-off curves. Thereby, it is possible to integrate this knowledge in the product and production platform. It can be used in a preparation phase or directly in the configuration process while producing new product variants.
- Trade-off curves may serve as integrator between a technology platform and the product and manufacturing system platforms. The existences of such information depend on the deliverables of the engineering processes. For example, trade-off curves need to be defined as a deliverable in the technology development process, thereby aiding the lifecycle integration.

4.5. Paper E – Process and model for preparation and maintenance

Paper E addresses *preparation* and *maintenance* of a platform. It also addresses the integrated modeling of product and manufacturing system platforms.

As customer requirements progress over time, one might find that the bandwidth of the platform is insufficient to satisfy the need of the targeted market. The paper presents a process and a model for identifying a change and its impact on different sub-systems in the design, and a process for preparing the platform to accommodate the new requirement. Pure execution to generate a product variant is referred to as Mode I, and changing the platform to accommodate new requirements is referred to as Mode II. The process is an implementation of the principles of set-based concurrent engineering. The artifact model of the CC concept is used as a backbone, combing it with trade-off curves to model design bandwidth. Figure 29 shows the proposed process. Though not all of the goals of both approaches coincide, several similarities can be identified between the outlined steps and SBCE principles.

The first four steps relates to function-means modeling in general, and particularly how they are modeled as a part of a CC. The last step is a cause of combining set-based concurrent engineering and platform-based design.

4.5.1. A process for platform preparation

The *modeling of functions and means* in platform-based development as proposed in this paper bears strong resemblance to the principles of *communicating sets of possibilities* and *exploring trade-offs by designing multiple alternatives*. Rather than selecting and pursuing one DS, each FR is solved by a *set of design spaces* comprised of several alternative DSs. These alternative DSs together cover the bandwidth available to solve the FR.

In *identifying bandwidth*, the DSs and FRs may be completed with modeling the bandwidth using trade-off curves. These describe both how DSs in a set together cover a performance bandwidth and how changing parameter values of DSs in the set influences the performance of the product. Trade-off curves and limit curves *define feasible regions* concerning both the geometry and the physics of the product.

Identifying interactions between design solutions is key to *looking for intersections of feasible sets.* As the bandwidth of each set is described using trade-off curves, two interacting sets will have two or more trade-off curves describing the set. There may be overlaps between these sets creating an intersection of feasible sets. There may also be sets interacting where no intersection can be found, especially as an interaction can occur between any types of set (e.g., a product design and manufacturing system). In such a case, new design solutions must be created.

In *model configurable components,* branches of the function-means tree are encapsulated into configurable components. This prepares the platform for working concurrently, while *communicating sets of possibilities*. Going from a common function-means structure to a network of systems enables each system to be treated as a separate unit, which interacts with other systems. This splits the design space and the sets into manageable chunks. Thus, this step also prepares the platform for later configuration using for example design automation tools.

The final step in the platform preparation is to *validate the design space*. Since the platform execution should be effortless in Mode I, fully automatic execution must be possible. This means that the set must be mature before the platform is ready to be used in Mode I. Thus, the design space must first be *validated*. Consequently, an extra step in which the platform is validated is needed to enable execution in Mode I, and to make the model as complete as possible in case Mode II. Possible methods are suggested, but this paper does not elaborate the details of such validation. Identifying interactions between design solutions is key to looking for intersections of feasible

Table 2: The steps in the platform preparation process and the addressed set-based principles. Several of the rows also have suggestions for methods (adapted from Paper E).

PREPARATION AND USE STEPS	SET-BASED CONCURRENT ENGINEERING PRINCIPLES AND POSSIBLE METHODS
Model functions and means	Explore trade-offs by designing multiple alternatives. Communicate sets of possibilities (Applicable methods: Morphological matrix adapted to SBCE (Raudberget, 2011))
Identify bandwidth	Define feasible regions (Applicable method: trade-off curves from earlier development)
Identify interactions	Look for intersections of feasible sets
Model configurable components	Communicate sets of possibilities
Validate the design space for Mode I	Establish feasibility before commitment (Applicable methods: design space exploration (2000), trade-off curves, multi-criteria optimization, physical testing and worst-case scenarios (Henia et al., 2005, Thiele et al., 2002, Thiele et al., 2001))
Platform execution (Mode I)	Establish feasibility before commitment (Applicable methods: trade-off curves and preference set-based design (Inoue et al., 2010))

sets. As the bandwidth of each set is described using trade-off curves, two interacting sets will have two or more trade-off curves describing the set. There may be overlaps between these sets creating an intersection of feasible sets. There may also be sets interacting where no intersection can be found, especially as an interaction can occur between any types of set (e.g., a product design and manufacturing system). In such Table 2 summarizes the proposed steps with the principles of SBCE and indicates applicable methods from literature.

4.5.2. Illustrating the use of the process

The process and artifact model is illustrated using hydraulic cylinder platform. It is a made-up case, yet represents a realistic situation. The product platform and the manufacturing system platform are modeled using F-M modeling and encapsulated in configurable components. The model also includes the lateral relationships between the different subsystems in the two platforms.

A change to the platform is needed as the customer requirements progress to exceed the level to which the platform originally was designed. The cylinder speed needs an upgrade, which cannot be accommodated using current design solutions. The current cylinder has connections to the manufacturing system as seen in the artifact model. The artifact model and the trade-off curves connected to each system show that the introduction of a new design solution: a dampening system, will in turn spawn a change in the manufacturing system platform and a need for a five-axis lathe. The new solutions are then modeled according to the suggested process. The expanded platform is illustrated in Figure 30.

Real industrial operations will require iterations in the platform preparation process proposed here. It is thus not to be understood as a linear process, but as a guide to struc-

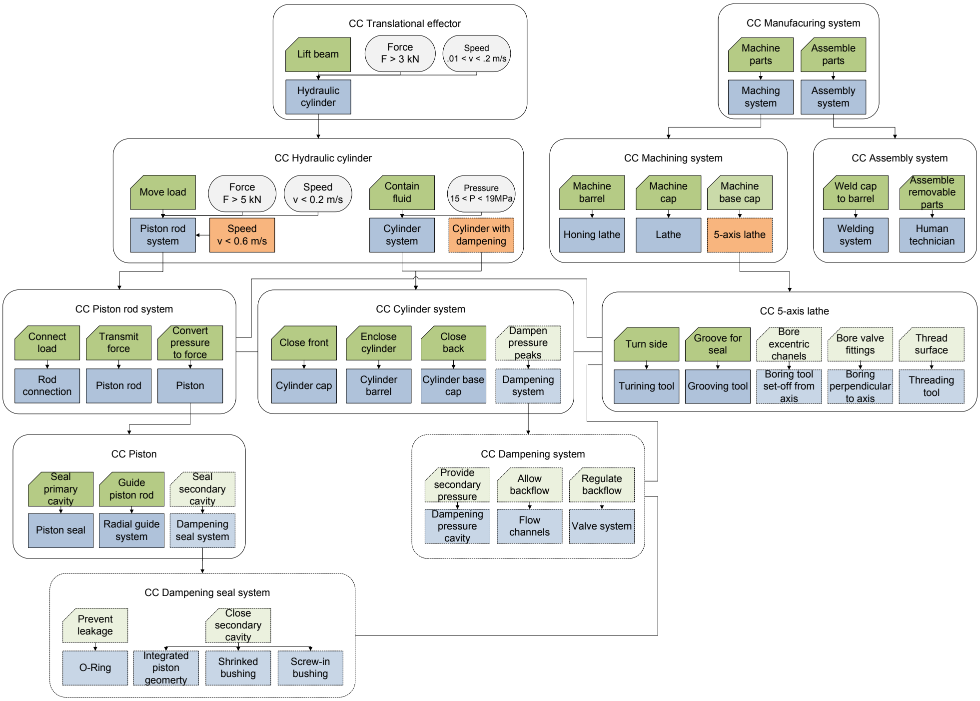

Figure 30: An example of a CC structure with extended bandwidth

ture overall work. Moreover, it must be seen in the context of other platform-related activities. The elicitation of customer requirements and strategic platform scoping help in specifying what bandwidth the platform must provide to solve the high-level functions. As preceding activities, they are input to the preparation process outlined here. However, on the lower levels, the bandwidths of FRs and DSs are elaborated in detailed design work and manufacturing planning. This is where the model provides support and helps connect the overall function with lower-level solutions.

The contribution of Paper E to RQ2 and RQ3 can be summarized as:

- The suggested process applies the CC artifact model for platform preparation. It combines the trade-off curves introduced in Paper D with a developed version of the process outlines in Paper C.
- The model illustrates the lifecycle meeting between product and manufacturing system platform. Using the here presented artifact model, product and manufacturing system platforms can be modeled using the same model. That allows engineers to make decisions on designs while still considering manufacturing issues. Specifically, decisions on design changes can be addressed from cross-lifecycle perspective.
- SBCE provides the framework and general philosophy for efficient platform-based development. Therefore, SBCE has been embodied by implementing the principles in the process and model.

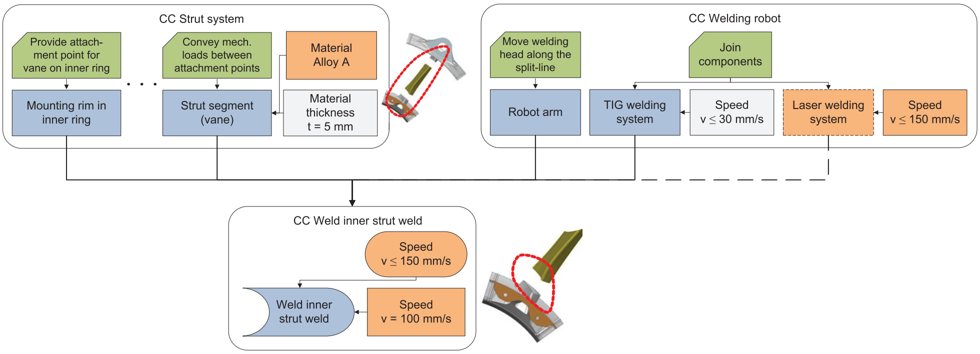

Figure 31: A schematic overview of a product and a manufacturing system platform modeled using the Configurable Components (Paper F).

4.6. Paper F – A model for product manufacturing integration

Paper F builds upon the results presented in paper E, expands the process and model, and applies them both in a case from the aerospace industry. The process is enriched with a detailed sub-process for identifying the impact of a change to a platform. It uses the lateral relationships in the CC model to find systems affected by a change and trade-off curves to assess the character of the impact. The model is expanded to use manufacturing operations as the interfacing object between the product platform and the manufacturing system platform.

4.6.1. Expanding the model

The artifact model from paper E was expanded to include manufacturing operations. This idea originates from a paper by Michaelis et al. (2014) in which manufacturing operations realizes the lifecycle meeting between product and manufacturing. The two different lifecycles are separately modeled using function-means modeling. Function-means modeling alternates between the functional domain and the solution domain, creating a tree of functional requirements and design solutions. Transforming functions as well as purpose functions may be used to represent the specific characteristics of the manufacturing system. The manufacturing operations are included in the platform model, linking the lowest levels of the two F-M trees. The operations represent the execution of the functions in the manufacturing system while producing the product. The F-M trees of the manufacturing system and the product, as well as the operations, are encapsulated in a CC to provide structure to the platform. Figure 31 illustrates part of an F-M-tree of a Turbine Rear Structure and its associated manufacturing system.

What the links truly represent is the link between two systems' functional requirements. These functional requirements may be modeled against each other in a trade-off curve. In cases where one trade-off curve is insufficient to represent the link, several can be used in combination.

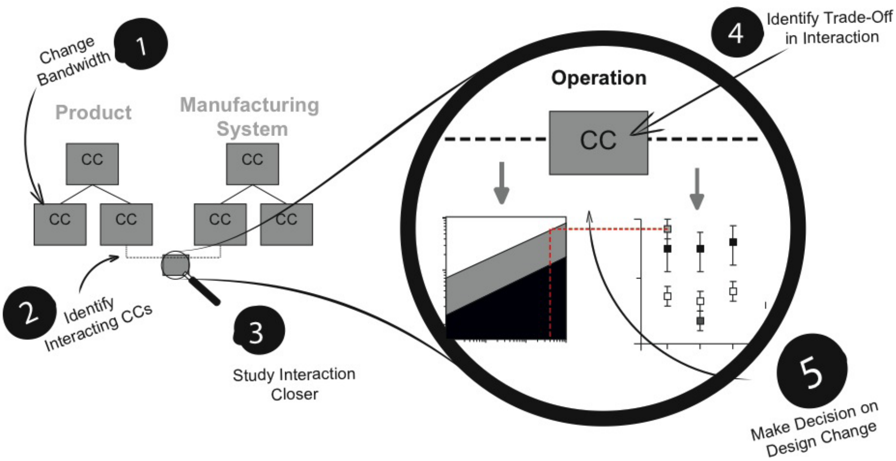

Figure 32: The proposed workflow for defining the change. The figure includes schematic pictures of how the elements in the model can be used to identify the change impact (Paper F).

4.6.2. Expanding the process

Using the enhanced artifact model, it is possible to assess the impact a change would have on the system. The link between product and manufacturing systems also makes it possible to address cross-lifecycle impacts. The initial step in the process of changing the platform to accommodate more functionality is to *define the change*. Figure 32 illustrates the workflow in defining the change. Since it is not being formally implemented in an IT tool, it has been stripped of formalism to the benefit of including elements of the model to illustrate how they are used to help identify the possible impact of a change. These five steps precede the process in Paper E thus complimenting the process for maintenance of an integrated product and manufacturing system platform.

In this paper, the TRS platform bandwidth is expanded (1) to reduce the welding

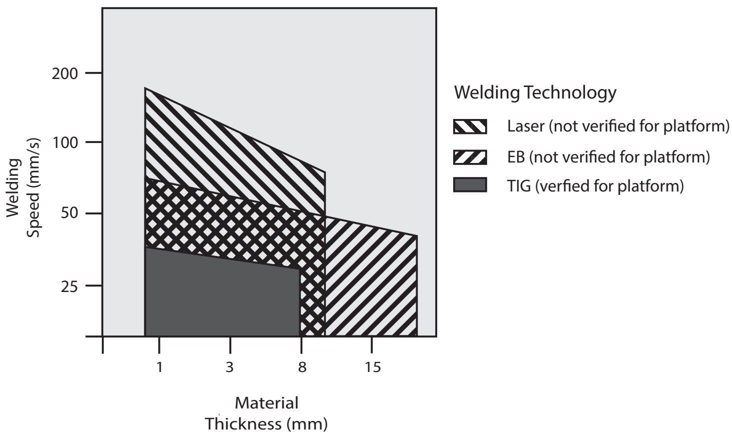

Figure 33: Trade-offs between welding speed and material thickness for different welding technologies (Paper F).

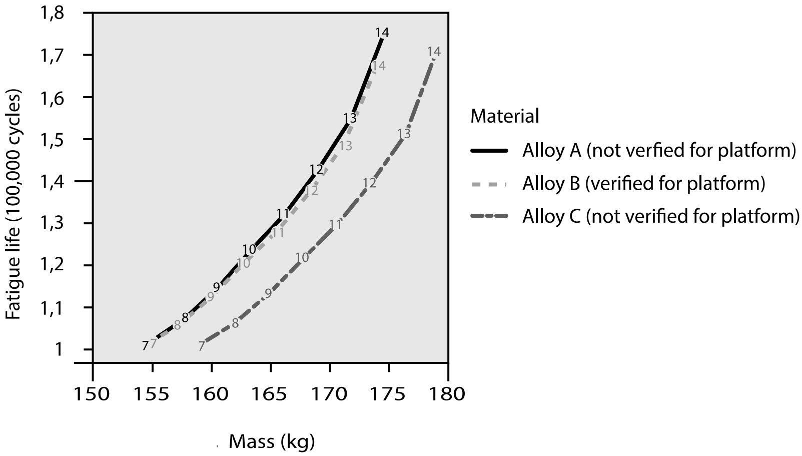

Figure 34: Trade-offs between fatigue life and mass of the TRS for three materials and different numbers of struts (Paper F).

time. Orange colored systems in Figure 31 using the Configurable Components (Paper F). represent systems that are affected (2) by the change. The interaction between the systems can be studied closer (3) using the trade-off curves for the two interacting systems (4) (see Figure 33 and Figure 34). Based on the information in the curves, it can be concluded (5) that expansion of the bandwidth can be accommodated if it is completed with laser welding. To maintain the performance of the product, a new material needs to be introduced.

To summarize, the model and the outlined processes accommodate changing customer requirements by redesigning and updating a limited number of systems in the platform. It does so using a process inspired by set-based design. The parameters that govern the interactions between these systems are, among others, represented by trade-off curves. With these elements and processes the platform approach allows making decisions on the expansion of functionality and performance of systems in the product and the manufacturing system.

The contribution of Paper F to RQ2 and RQ3 can be summarized as:

- The artifact model suggested in its most mature form in Paper F serves the purpose of reuse across the lifecycle using more than physical parts, ranging from preparation, through execution to maintenance. It shows how a platform can be updated while taking the change impact into consideration. The model uses lateral relationships between systems to connect objects from different lifecycle stages. The lifecycle meeting between product platform and manufacturing system platform is realized through manufacturing operations.
- The SBCE principles are fully integrated into the processes. The case shows how the principles manifest them selves into a feasible process, which considers design spaces, intersection of sets and to establish feasibility before commitment.

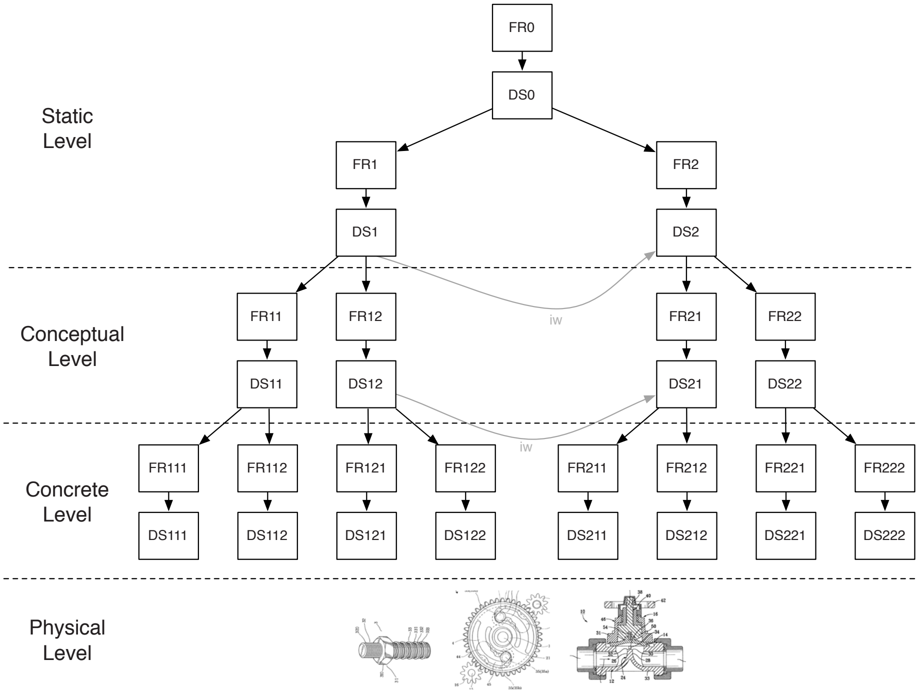

Figure 35: Levels in an F-M tree (Paper G)

4.7. Paper G - Platform concept development

Paper G proposes an approach for modeling platform concepts in early phases and eliminating undesired regions of the design space. It does so by applying SBCE to create sets of design solutions to cover the bandwidth of each functional requirement. It is illustrated with an example from the aerospace industry with information collected over several years' collaboration with the studied company.

The model used to support early platform preparation is based on the CC model. Using F-M-modeling to build up the CC objects from within, the design rationale is modeled in several levels. The levels are defined by the degree of detail, which also determines their proneness to change. Three levels are defined, the *static*, *conceptual* and *concrete* level, schematically depicted in Figure 35. The static level defines functions related to the core business and thus rarely changes. On the *conceptual level*, FRs and DSs define the range of possible solutions to the upper level requirements. Different branches in the tree represent different concepts. The *concrete level* is the last functional description before the allocation of physical components. An object on this level is close to the physical embodiment, and the functions are typically defining features. The physical level consists of physical part and is not a part of the F-M tree. To use the F-M tree in designing, a process is suggested involving four steps that are iterated for each level in the F-M tree. The steps are *generate and structure solutions*, which is supported by the F-M tree, *specify solutions*, supported by functional modeling and trade-off curves and *early narrowing of sets*.

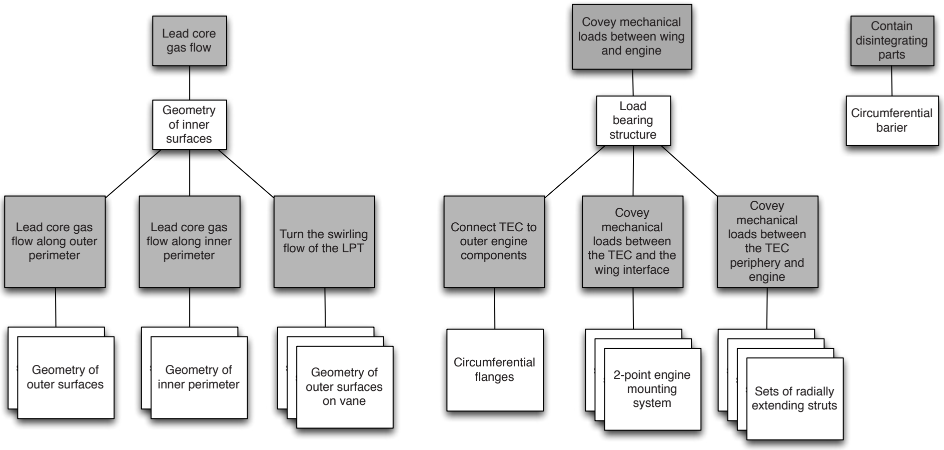

Figure 36: The static and conceptual level of a TEC from Paper G.

The approach advocates different ways of narrowing down the set depending on the level in the F-M tree and the amount of functional coupling between FRs and DSs. The static level does not incorporate any alternatives in design solutions, rendering elimination obsolete. For the other levels, elimination is possible based on a number of different reasons:

- Elimination based on fulfillment of Bandwidth
- DS elimination based on compatibility with the rest of the platform
- Concept elimination based on compatibility

There are two ways of using the *bandwidth to eliminate* bad or undesired solutions. If there are two interfering requirements, i.e. carry load and stall speed for an airplane, part of the bandwidth of one of the functional requirements may be reduced. This in turn will eliminate several design solutions that previously solved that part of the bandwidth. The second case in which the bandwidth may be used to reduce the design set is when design solutions are redundant in their coverage of the bandwidth.

The initial strategy for elimination is based on *compatibility between Design Solutions*. The solutions are assessed and those found incompatible with the overall system are eliminated. Eliminating a design solution at the Conceptual level effectively prunes a branch of underlying FR and DSs.

In theory, all possible total *concepts* in the platform are achieved by combining all design solutions on all levels. This creates a huge design space of which great parts will be infeasible. Practically, there are combinations that are not possible, for example using a fly-by-wire steering wheel with a mechanical steering system. Compatibility between solutions can be modeled using the *interacts with* relationship. Thus, these lateral relationships play a major role in defining concepts, and thus eliminating a great number of infeasible designs.

Figure 36 illustrates an F-M tree for a Turbine Exhaust Case (TEC) of a jet engine. By combining all possible solutions into concepts, 144 distinct variants can be identified.

Analyzing feasibility and performance using CAE tools would require extensive CAD work and analysis time. However, by using the elimination methods described above, 117 of the alternatives can be ruled out as infeasible. The pattern of *model and eliminate* is iterated on each consecutive level to produce a manageable design space.

There has been a void when it comes to detailed processes for efficiently designing platform concepts. There is a vast body of knowledge in platforms and the effects of using them for scalability in production, yet there are few who touch upon how to efficiently and accurately develop and select concepts for a platform. A design space can be reduced substantially by using the suggested model. The elimination of undesired design solutions is done by consulting the bandwidth (redundant solutions and solutions outside of the desired bandwidth are eliminated) and by addressing compatibility between design solutions. The elimination of undesired solutions helps to create design spaces manageable to ease further design efforts.

The contribution of Paper G to RQ1 and RQ3 can be summarized as:

- This paper covers yet another part of the platform lifecycle, i.e. the early phases of conceptual modeling before considering embodiment. Thus it improves the support for platform-based development in those phases. Even though the artifact model makes it possible to integrate the output of his phase with other lifecycle stages, this paper mainly addresses RQ1 and RQ3. The Process suggested fills a void in the platform development lifecycle, a process that is inspired and infused with Set-Based Concurrent Engineering as a basic framework. The abstract character of these early phases makes them an ideal candidate for design concept exploration before committing to a design.
- Function-means modeling was originally created for single product development, but has been expanded to encompass features for alternative designs. The method, supported by the F-M structure can be used to eliminate bad solution at early stages. These decisions are based on compatibility between DSs or how well a specific DS covers the desired bandwidth of an FR. This can be assessed using trade-off curves.

4.8. Paper H – A two-stage model for execution

Paper H addresses execution of product platforms from an engineering-to-order perspective. It proposes a two-step method that combines modular configuration with scalable configuration. By doing so, design flexibility is kept longer to counter late changes in customer requirements. At the same time, decisions at an architectural level can be made, which help progress the design work. The approach is schematically depicted in Figure 37.

The approach is illustrated with a case from the aerospace industry using a TRS as the example product. A TRS can be manufactured using T-sections or H-section. Further, the sections are joined through laser welding, TIG welding or EB welding. The first step is *module configuration*. This configuration decides the architecture of the product variant. The first step may result in selecting the H-concept because the customer has a high requirement on geometrical robustness (the H-concept is more robust) and Laser

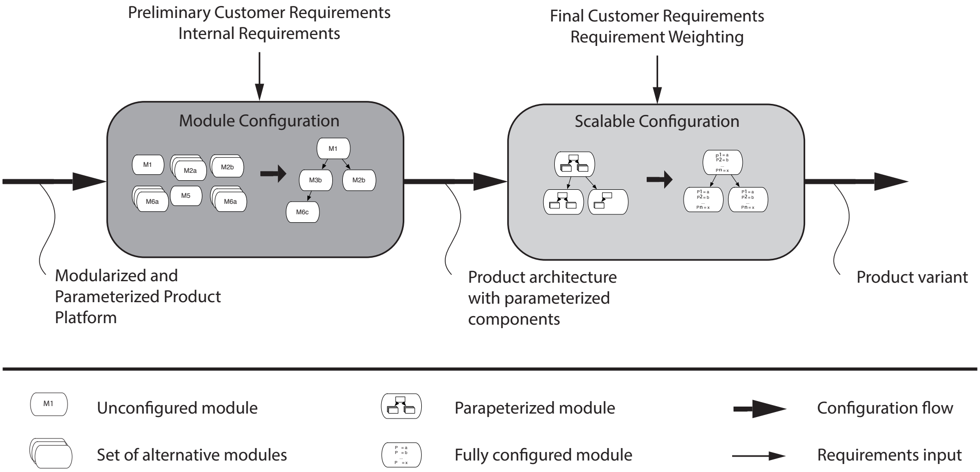

Figure 37: A schematic picture of the information flow in the suggested two-step approach to platform execution (Paper H)

welding because they want to weld it to their own cased parts (laser welding is fast even through thicker material) (see Figure 38). The bandwidth left after such configuration is described by the remaining DSs, namely the H-section, and the laser weld. The decisions are based on high-level information, such as preliminary customer requirements or technology maturity level.

The second step decides the parameters of each remaining design solution. This way, changing requirements can still be accounted for even though the architecture is set. The change in the case is defined in Table 3. By changing the material thickness in the TRS while also increasing the number of vanes, the new requirements are accommodated for. The decisions in the *scalable configuration* are based on trade-off curves, detailed analyses and design optimization based on final requirements.

The major conclusions from this paper are:

- The impact of changing customer requirements can be managed through keeping bandwidth on a scalable level to the very latest stages in development. In doing so, the design is still flexible enough to cope with changing parameter values while the design work can still be progressed on an architectural level. The Configurable

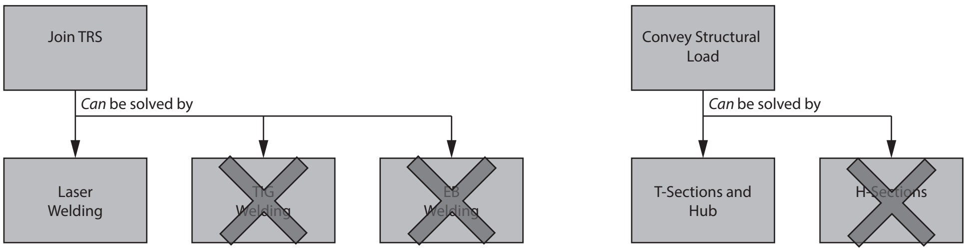

Figure 38: Different options for solutions to Join TRS and Convey Structural Load. Three DSs are eliminated in the architectural configuration.

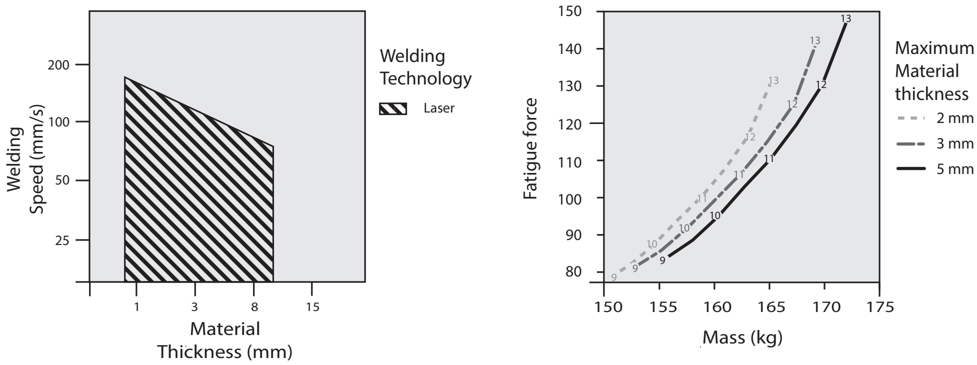

Figure 39: Trade-off curves describing the remaining bandwidth of the configured architecture. Welding system (to the left) plots material thickness and welding speed. The TRS (to the right) plots Fatigue force (% of normal fatigue force, giving 100,000 cycles in life) and weight. The numbers on the lines represent number of vanes.

Component artifact model enables this two-layer thinking (architectural layer and scalable layer) by modeling functional requirements with multiple possible design solutions, all of which are scalable and interchangeable.

- This paper presents a framework and a model for adaptable product platform design. It does not cover the organizational and managerial aspects, of implementing such an approach. Ultimately, this type of design support will need implementation in IT-tools.

4.9. Summary of the results

The platform approaches in today's literature do not cover the need to support holistic platform development across all stages of the lifecycle in detail. A platform consisting of configurable system elements may be used as a bridge between abstract descriptions of platforms (e.g. technology platforms) and concrete descriptions (e.g. part- and module-based platforms).

Within a development platform, the technology platform is executed while preparing the product platform. The reusable technology elements can aid in determining product bandwidth, finding means to meet functional requirements and defining constraints on configurable system elements of a product platform. Development platforms can be

Table 3: Old and new requirement specifications (adapted from Paper H)

OLD REQUIREMENT SPECIFICATION	NEW REQUIREMENT SPECIFICATION
Maximum weight 163 kg	Maximum weight 163 kg
Minimum life expectancy 100.000 cycles	Minimum life expectancy 100.000 cycles
Minimum force 100% of X	Minimum force 105% of X
Manufacturing time X	Manufacturing time X
Maximum weight 163 kg	Maximum weight 163 kg

realized through a Wiki solution supporting the technology platform, and a PLM architecture that supports the product platform. This holistic approach provides a better overview of both technologies and product platforms. That creates possibilities for more informed decisions on technology introduction regarding risks in their deployment for different applications. If the technology platform is partly represented with trade-off curves, these can be used in the execution of the product and manufacturing system platform, serving as the link between technology development and product and manufacturing development.

Modeling the product and manufacturing system platform using configurable system platforms has proven feasible as an integrator of these lifecycle stages. Again, trade-off curves play a big role as integrators together with the manufacturing operations that realize the lifecycle meeting between product and manufacturing.

Set-based concurrent engineering as a design approach can be used throughout the lifecycle of a platform. The principles of SBCE can be used in preparation of a product and production platform. SBCE facilitates structured elimination of undesired solutions for platform concept development. Thus, the design space is narrowed down to a manageable size at an early stage saving development time for later stages.

SBCE and platforms based on generic configurable concepts modeled as configurable components can be supported by a PLM architecture to configure product variants. Optimization tools can be used to set CC parameters, optimizing the systems to comply with requirements. CAE tools, as well as trade-off curves can be used to assess potential solutions within the solution space set by the bandwidth.

The suggested artifact model can be used to support an engineering-to-order approach by applying both modular and scalable configuration. In doing so, the design is adaptable to changing customer requirements, which reduces the need for changes. Changes can on the other hand be managed too, by using the artifact model to identify the impacted systems and to assess the magnitude and find solutions using trade-off curves.

5

Discussion

This section aims to give a clear answer to the research questions posed in the introduction of this thesis, and to discuss the quality of the results in relation to the research approach used.

5.1. Answering the research questions

RQ1: *What models, processes and tools can be used to support development platforms, and how can they be improved to better suit them?*

About processes

The business processes for developing single products compared to those for developing platforms are fundamentally different. Processes such as the ones established by Pahl and Beitz (1996) or Ulrich and Eppinger (2008) can provide support on a high level but fail to capture all detailed aspects of platform-based design. As platforms are prepared and then executed, special business processes are needed. In the preparation phase, knowledge is created with reuse in mind. In the execution phase, the processes focus on retrieving knowledge from the platform. More specifically within a development platform, the technology platform is executed while preparing the product and manufacturing system platform.

The processes for early phases in platform development are exceptionally immature. Instead, most processes take a configuration view on the development, starting after embodiment. The procedure for platform concept development suggested in Paper G aims to fill that void. The suggested process in paper E and F, puts forth an overall process for platform preparation. This process is heavily reliant on the Configurable Component artifact model, as well as trade-off curves for modeling of interactions between systems. In fact, processes for preparation rely more on the use of a good artifact model than processes for execution. The processes for execution are more general in their kind but put greater requirements on data integrity. The process suggested in paper C serves to ready the prepared platform for execution, including preparing a PLM architecture to support the execution.

The process for keeping the platform up to date, i.e. the change process, addresses the lifecycle phase of platform maintenance. This process focuses on the engineering side of change management, addressing issues such as identifying systems affected by a change and finding solutions to expanded requirements. The managerial aspects of change management, such as approval processes and supplier involvement in the change effort are not covered. However, being able to identify the affected systems and the nature of the needed change through the use of the artifact model and trade-off curves could alleviate such processes.

About models

The selection of models for pre-embodiment design for single-product development is vast, several of which have been adapted to platform-based design. Function-means modeling was created for single product development, but has been expanded to encompass features for alternative designs. Function-means modeling has proven feasible for platform development by enabling reuse of functions and design solutions rather than physical parts. This capability is key to providing support for engineering-to-order development. The two-stage execution in Paper H enables decisions on an architectural level while maintaining flexibility for changing customer requirements. Initially, design solutions are eliminated to produce a single variant. The remaining flexibility, or bandwidth, is realized through the scalability of each remaining design solution. The final configuration occurs when the customer requirements are considered stable enough to fully commit one design. Others have considered models with joint modular and scalable bandwidth, for example Du et al. (2014), yet provide support only for design optimization post embodiment.

As with the processes, traditional research in platform models has focused on supplying models for configuration of pre-designed blocks. The artifact model suggested in its most mature form in Paper F serves the purpose of reuse across the lifecycle using more than physical parts, ranging from preparation, through execution to maintenance. The model has been implemented in a PLM architecture (mainly Paper A-D), using various CAE-tools, including optimization routines to determine feasible parameter values.

This enables virtual design experiments that cover large design spaces and include CAE analyses using multiple CAD models. However, the resulting amount of data for large design spaces surpasses what is practically possible. Contrastively, using simple models, such as trade-off curves, enables engineers to incorporate more system complexity without producing an impractically large amount of data and to ultimately support the selection and analysis of real design cases.

About tools

The difference in processes compared to single product development impacts what tools that are appropriate to use. Development platforms can be supported by Wiki solutions carrying the technology platform easily due to their flexible structure and ways to store different kinds of knowledge. Current PLM architectures can support platform-based design, while neither PDM systems, CAD systems nor CAE systems are optimized for managing configurable function carrying platform elements; they rather provide support for single products. Inoue et al. (2010) agree on the questions surrounding current CAE and CAD tools and propose a customized tool which has not yet been commercialized nor tested on a large scale. There are other attempts at letting CAD and PDM systems manage functional carrying organs as a part of a product platform, for example Bruun and Mortensen (2012). However, they focus on the visualization aid for stages in which parts are already developed, at the expense of the actual development process.

In Paper C and D, a solution is proposed for being able to explore the design space and finding the optimal solution. However, this is still based on point-solutions rather than exploring the continuous design space because of the inability to do so by the CAD and CAE tools. Custom tools are needed to prepare and execute development platforms. Modeling system elements with CCM, by using the CC concept in combination with a Wiki seems feasible, with commercial tools supporting parts of the process. There are tools fitted for design space exploration, not only for post embodiment analyses, but also for functional analysis. There are optimization tools based on functional models, for example Yvars and Duhau (2012). However, this is a module-based approach which requires fixed parameters of all modules. Further, Wynn et al. (2010) present the Cambridge Advanced Modeller, which among other things can be used to represent and analyze multiple architectures from a functional point of view. However, neither of these are commercialized software which can be used out-of-the-box.

RQ2: *How can models, processes and tools be integrated across the platform lifecycle to support a holistic reuse approach?*

A holistic approach provides a comprehensive overview of technologies, products and manufacturing system platforms. Integrating these lifecycles creates possibilities for more informed decision-making. The risks in technology introduction can more easily be assessed, in addition to their readiness for different applications. The products will further be based on technologies that are tested and approved, which will reduce the need for technology development in the product development projects. That in turn will reduce the risk in product development and shorten lead-time for bringing new products to the market. A product platform that builds upon the core technologies at the forefront of technology development enables companies to respond quickly to new market demands and gain advantages vis-à-vis the competitors.

Products and manufacturing systems can be prepared for reuse, which provides better solutions in comparison to addressing reuse after the design is already set. Using the here presented artifact model, product and manufacturing system platforms can be modeled using the same model. That allows engineers to make decisions on designs while still considering manufacturing issues. Specifically, decisions on design changes can be addressed from cross-lifecycle perspective.

The model uses lateral relationships between systems to connect objects from different lifecycle stages. The lifecycle meeting between product platform and manufacturing system platform is realized through manufacturing operations. The relationship between the functional requirements of the systems, or the inherent parameters thereof, can be described by trade-off curves. Trade-off curves also serve as integrator between a technology platform and the product and manufacturing system platforms. They are used in the preparation phase to distinguish feasible solutions pre-embodiment, and in the maintenance phase to assess change impact and find knowledge gaps. The process for maintenance addresses the technical side of engineering changes. The managerial implications of integrating change processes across the organizations, which is often reality of integrating across the lifecycle, are covered by others, for example Pikosz and Malmqvist (1998) who conclude that PDM systems are capable of managing these processes. The process proposed here will latch on to these formal integrating processes for information sharing.

PLM serves development platforms as a business approach by which knowledge is integrated across the lifecycle (Grieves, 2006, Stark, 2005), yet PLM alone is no guarantee for success. By applying processes, information architectures *and* tools, it is possible to realize a strategy of platforms integrated across the lifecycle. However, it is only when all these layers are present that the business goals can really be fulfilled. The configurable system families can be represented using information models in modern tools. The information models of a PDM system are capable of representing both hierarchical structures and lateral relationship, which are used to relate systems to each other, and across the lifecycle. Yet there are limitations. Though PDM systems are good for integrating product development and manufacturing development, the capabilities for representing bandwidth in an efficient manner are lacking (Kovacs et al., 1998). The standards upon which many of these system build will have to represent each and every possible variant with separate information objects, which creates huge amounts of data that are impossible to manage (Männistö et al., 1998).

The characteristics of the processes, models and tools too will impact the ability to integrate. For example, the abstract information in a technology platform cannot be used directly in execution of a product and manufacturing system platform. However, the use of trade-off curves as an integrating object allows technology information to be used directly in the execution of the product and manufacturing system platforms. The existences of such information objects depend on the deliverables of the engineering processes. For example, trade-off curves need to be defined as a deliverable in the technology development process, thereby aiding the lifecycle integration.

RQ3: *How can set-based-concurrent engineering be implemented to improve processes and models for development platforms?*

The SBCE principles support the platforms approach presented in this thesis in several different ways. SBCE provides the framework and general philosophy for efficient platform-based development. Therefore, SBCE has been embodied by implementing the principles in the processes and models.

On the modeling side, function-means modeling is used to implement the notion of sets. The sets are modular by providing several different design solutions to one functional requirements, and scalable through parameterization of each design solution. These are encapsulated into configurable components. This design space is narrowed down to a desirable size using a collection of different approaches for different lifecycle stages. In early stages of platform preparation, i.e. concept development, how well a DS covers the desired bandwidth of an FR can be used to eliminate undesired concepts. This can be assessed using trade-off curves. By using the lateral relationships in the artifact model, concepts can also be eliminated based on their compatibility with other concepts. The *interacts_with* relationship serves as an indicator of interaction between two DSs – if the interaction is beneficial for requirements fulfillment or not is up to the designer to decide. Elimination on high levels in the F-M tree may prune entire branches of the solution tree, saving the time needed to elaborate that branch to lower levels.

A set-based approach differs from platform-based development at the point at which the narrowing down of the functional requirements occurs. In platform-based development, the bandwidth of the FR is frozen to preserve it, yielding a platform that can deliver variety rather than a point solution. The narrowing down is finalized in the execution of the platform. Here, one point in the bandwidth of the functional requirements is selected when one distinct variant is instantiated.

Considering this, the development platform, supported by the artifact model and a PLM architecture may be used as an engineering-to-order tool. For example, in a bidding process it is possible to narrow down the remaining design space to one variant, to check compliance to the new requirements before commitment. Thereby, it is possible to make more informed decisions on the possible design changes needed to address the new requirements. In doing so, it is possible to reduce the lead-time of the biding process, make a more accurate design effort, and reduce the risk in taking on a new project.

In practice, a lot of the development is driven by changes. By *communicating sets of possibilities,* the lifecycle meeting between product- and production can be realized and their mutual relationship can be considered while expanding the platform. This provides efficient means for example for assessing impacts from changes. To support this, the model is built up using both vertical and lateral relationships, thus providing means for assessing intersection between sets.

The main contribution to answering this research question comes from Paper E, F and G where processes and models are developed based on set-based principles. However, though not explicitly targeting SBCE, the PLM architectures, models and tools described in Paper A, B, C, D and H model the interfaces and information models necessary to implement SBCE on a large scale. The information produced using those tools is key to assessing the intersection of sets and upon which to base elimination decisions. PLM provides the infrastructure and interfaces between applications and interfaces and integration between models necessary to communicate sets of solutions. This is of particular interest in organizations dispersed around the world.

Many suggest mathematical methods for eliminating solutions. These methods are not considered here, yet they may contribute in later phases. Yvars and Duhau (2012) describe methods for eliminating solutions based on functional fulfillment. However, these methods require substantially more mature data, and work only with modular platforms.

Further, the abstract representation of technology platforms is not ideal for analysis in CAE systems. However, SBCEs way of embodying technology knowledge with trade-off curves provides the necessary numerical representation for a PLM architecture with CAE systems.

5.2. Discussing the results

As discussed in 3.2.4, verification of research results can be performed by Verification by acceptance and Logical verification. *Verification by acceptance* focuses on having new scientific contributions accepted by experts within the field. Research can be considered logically verified when it is *complete*, *internally consistent* and *externally consistent*.

Verification by acceptance

All papers have been subject to peer review. Papers A, B, D, E and G were submitted to conferences where the content is peer reviewed by experts in respective field. The results have also been subject of review in the presentations required to be published at each conference. As journal articles, experts within the field have reviewed Paper C and F. As for Paper H, is submitted to a journal and thereby undergoes the same peer review as paper C and F.

Further, the results of the papers have at numerous occasions been presented to the Wingquist Laboratory partner companies who have expressed their belief in the results. The artifact model has been implemented in an IT-tool – CCM, which have been used in several of the papers – for modeling and configuration. This tool is being tested and demonstrated in a EU-funded collaboration project between the leading aerospace companies in Europe. The tool and modeling strategy has been accepted as an integral part in efficient development of the next generation of airplanes.

External consistency

The results can be considered externally consistent if they agree with established literature. The research is based on known models and literature and is found to be in agreement with adjacent results. For example, using functional modeling in early stages of platform development, such as suggested in Paper E, F, G and H concurs with the views of other authors, for example Farrell and Simpson (2003). To analyze the functional coupling to determine the *goodness* of a design harmonizes with the work of for example Suh (1990).

The process and the artifact model is still missing an approach to relate requirements to actual scalable factors, which is essential to be able to configure based on customer requirements. Such a relationship could be modeled according to Ulrich and Eppinger's need-metrics matrix (Ulrich and Eppinger, 2008). Further, basing the bandwidth of a design on equations and design trade-offs concurs with the work of other authors (Fisher et al., 1999, Jiao and Tseng, 1999), but can also be accomplished by referring to customer demands (Pedersen, 2009).

SBCE for engineering-to-order design has been researched before. For example, SBCE as an approach for efficient design of low volume products is adopted by Singer et al. (2009) who apply it on ship design. Also, they too stress the importance of delaying detailed design decisions until enough is known about the design and requirements, as suggested in Paper H.

Investigating and solving a change (identify change impact) in Paper G bares resemblance to common problem solving techniques such as LAMDA (Look, Ask, Model, Discuss, Act) (Ward, 2007). As a product solving method aimed toward lean product development, it coincides with the proposed approach in aim and context.

It has been a hypothesis throughout this thesis that PLM, platforms-based development and set-based concurrent engineering share goals. Fielding et al. (2014) compares Lean Product Development (LPD), of which SBCE can be considered a part with PLM. LPD define people, processes, tools and technology as corner stones. In PLM, the same cornerstones are identified, also adding practices to the lot. Bernstein (1998) claims that platform-based design refers to the product strategy where as SBCE refers to the design strategy. Despite that difference, they have several elements in common. For example, having a modular product architecture is something that is a corner stone for both approaches. In both cases, modeling several alternatives at once is a way to achieve efficiency in development and to be able to provide customers with better quality (Gershenson et al., 2003, Ward, 2007).

Internal consistency

Internal consistency touches upon the definition of terms and how clear and non-conflicting they are. A few terms are coined here, for example Development Platforms. *Development platform* and *technology-based configurable platform* are used synonymously. The terms have been exemplified in several cases, for example in Paper C, to give a clear picture of what it means. The processes suggested to support them have been illustrated through case studies to give a better understanding of them. The examples give indications for what and how development platforms can be used, but are not meant to show the only uses. The modeling approach, for the product platform also agrees with established design theory and methodology, for example the theories presented by Andreasen (1980), Suh (1990) and Mortensen (1999).

5.3. Evaluation of the Research Approach

The long-going relationship with the companies studied provides a springboard for going forward to prescriptive studies. The vast body of research performed within the Wingquist Laboratory has sufficed for quickly leaping into prescriptive stages. Further, the research clarification did not only pertain to literature study but resembled the Jørgensen approach in which a research gap provided by a literature analysis and also an industrial need provided the setting in which the research was performed.

5.3.1. Motivation for using case studies

As for the form of case studies as a way to gather empirical data, the form was chosen for its appropriateness for answering *how* and *why* research questions (Yin, 2003). The industry is our lab and we must adopt research approaches that suit available means. For example, the setting we work in is highly contextualized and cannot be separated from the phenomenon. Therefore, we must resort to case studies to accomplish generalization. It is true that the case study, like the experiment, does not represent a sample, and thus cannot be used for *statistical generalization*, but rather for *analytical generalization*, which requires that the analysis of the results of the studies is in fact generalizing, not particularizing. This generalization is achieved through triangulating using different industries in the examples.

Some may argue that the research resembles experiments, which is not the case because it is not controlled from its context. Rather, we are aware of the context and use it for analytical generalization based on, for example, company size, business area, manufacturing volume etc.

5.3.2. Managing validity threats in qualitative research

Qualitative research does not, in contrast to quantitative research, have the luxury of being able to create controls for reducing validity threats, but it has to deal with this issue after the research has begun (Maxwell, 1996).

Both logical verification and verification by acceptance has its problems. Verification by acceptance poses a pedagogical problem (Buur, 1990). The acceptance of a design support highly depends on how the support is presented, and the knowledge of the subject to which the support is presented. As a way to manage these, the verification process included triangulation (as suggested by Maxwell (1996)) both of how the results were presented, and to whom. More specifically:

- the research was conducted at three different companies, manufacturing a range of products,
- the interviewees were from different departments and have had different roles in the company, being interviewed through workshops, tele-meetings and face-to-face individual interviews,
- apart from interviews, documents and drawings were studied,
- the result was presented in writing to peer-reviewing conferences and a journal, as well as to experts at workshops and presentations,
- the tool, artifact model and processes have been implemented in demonstrators for three different companies. They have also been tested as an integral part in an international project targeting future ways of developing European airplanes collaborating with multiple companies and universities.

6

Conclusions and Future Work

This chapter presents the core of the thesis in terms of results and future work. A general conclusion is that working with development platforms has a major potential, provided that companies are willing to change. Consider what will happen if they do not change: other companies will outrun those that still cling to the old ways:

> *"As the present now will later be past, the order is rapidly fadin'. And the first one now will later be last, for the times they are a-changin". – Bob Dylan*

6.1. Conclusions from the work

Being efficient in developing products is considerably easier with the right models, process and tools. The artifact model suggested in this thesis enables cross-lifecycle considerations through modeling the product and the manufacturing system using the same model. The model represents the systems and their interfaces from the abstract of early phases, using functions and design solutions, to the concrete in detail design, by for example linking the artifact model to CAD models with geometrical interfaces. These systems are modeled with a bandwidth consisting of interchangeable modules, and scalable parameters. The relationships between the systems are described with trade-off curves, detailing the relationships between parameters and requirements. This allows for configuration of variants that provide the market with a broad range of products. It also allows for the use of set-based concurrent engineering in the design phase.

Set-based concurrent engineering can be implemented to support platform-based development. This is apparent in the processes developed for preparing, executing and maintaining a platform. With the artifact model as basis, platform concepts can be developed. From a broad range of possible concepts, the bad ones can be eliminated using compatibility between design solutions. The bandwidth that is left after all bad solutions are eliminated constitutes the final platform. This can be executed to generate product variants for manufacturing, or to generate information on the development effort required to satisfy emerging customer demands.

Today's commercial tools are developed for single product development and do not have the ability to fully support development platforms. Wiki solutions may support the technology-portion of the development platform, and a PLM architecture may support the product- and manufacturing system-portion of the development platform. However, CAD and CAE tools are optimized to develop point-solutions and are not enough to represent and analyze entire design spaces, such as those used in a development platform. PDM systems suffer from the same issues. The modeling of configurable system elements, as a part of the development platform, must be conducted using customized third party software.

Although implementing IT-tools is a major issue, it is still minor compared to the organizational change that a company needs to undergo to revise their processes to conform to a platform-based thinking.

6.2. Future work

Future research will more deeply investigate the concepts explored in this thesis, verifying them in industrial contexts and extending them to new endeavors. Future work will also address the issues below.

- The cross-lifecycle change processes presented here manage product and manufacturing system platforms. However, a process that manages changes of the technology platform through updating the trade-off curves is needed to maintain the platform over time. It is still left to investigate the manifestation of such a process.
- The results show that it is possible to configure product variants based on modifying solution parameters. Future research could elaborate on configuring variants based on requirements, which would require developing a meta-model for linking requirements to measurements.
- The results implicitly require an organization adapted to platform-based design. The organizational roles and units needed to manage development platforms efficiently as well as the journey towards such an organization are subject to future elaboration.
- The lifecycle of a platform has here been limited to end with the manufacturing systems. It could be interesting to address other lifecycle stages, such as maintenance of manufactured products and termination thereof. The applicability of the models and processes for these lifecycle stages are yet to be investigated.

"We shall not cease from exploration,
and the end of all our exploring
will be to arrive where we started
and know the place for the first time."
- T.S. Eliot

References

Abramovici, M. (2002) 'Status and Development Trends of Product Lifecycle Management Systems', *International Conference on Integrated Product and Process Development*, November 21-22, Wroclaw, Poland.

Ahmad, N., Wynn, D. C. and Clarkson, P. J. (2013) 'Change impact on a product and its redesign process: a tool for knowledge capture and reuse', *Research in Engineering Design*, vol. 24, no. 3, pp. 219-244.

Alblas, A. and Wortmann, H. (2009) 'The need for function platforms in engineer to order industries', *Proceedings of ICED 2009*, 24-27 August, Stanford, California, USA.

Alblas, A. and Wortmann, H. (2012) 'Impact of product platforms on lean production systems: evidence from industrial machinery manufacturing', *International Journal of Technology Management*, vol. 57, no. 1, pp. 110-131.

Andreasen, M. M. (1980) 'Syntesemetoder på Systemgrundlag - Bidrag till konstruktionsteori' (In Danish), Ph.D. thesis, Lund University.

Andreasen, M. M. (1992) 'Designing on a 'designer's workbench' (DWB)', *Proceedings of the 9th WDK Workshop*, March, Rigi, Switzerland.

Andrews, P. T. J., Shahin, T. M. M. and Sivaloganathan, S. (1999) 'Design Reuse in a CAD Environment — Four Case Studies', *Computers & Industrial Engineering*, vol. 37, no. 1–2, pp. 105-109.

Baxter, D., Gao, J., Case, K., Harding, J., Young, B., Cochrane, S. and Dani, S. (2007) 'An Engineering Design Knowledge Reuse Methodology using Process Modelling', *Research in Engineering Design*, vol. 18, no. 1, pp. 37-48.

Berglund, F., Bergsjo, D., Högman, U. and Khadke, K. (2008) 'Platform Strategies for a Supplier in the Aircraft Engine Industry', *Proceedings of the ASME International Design Engineering Technical Conference and Computers and Information in Engineering Conference - DETC/CIE2008*, 3-6 August, New York, New York, USA.

Berglund, F. and Claesson, A. (2005) 'Utilising the Concept of a Design's Bandwidth to Achieve Product Platform Effectiveness', *Proceedings of ICED 2005*, 15-18 August, Melbourne, Australia.

Bergsjö, D. (2011) 'Process and IT Support for Technology Platform Development and Use', *Proceedings of IAMOT 2011*, Miami Beach, Florida, USA.

Bergsjö, D., Catic, A. and Malmqvist, J. (2008) 'Implementing a Service Oriented PLM Architecture Using PLM Services 2.0', *Proceedings of the 10th International Design Conference - DESIGN'08*, Issue 48, pp. 271-280, Seoul, Korea.

Bergsjö, D., Malmqvist, J. and Ström, M. (2006) 'Architectures for Mechatronic Product Data Integration in PLM Systems', *Proceedings of 9th International Design Conference - DESIGN 2006*, Dobrovnik, Croatia.

Bernstein, J. I. (1998) 'Design methods in the aerospace industry: looking for evidence of set-based practices', M.Sc. thesis, Technology and Policy Program, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Blessing, L. T. M. and Chakrabarti, A. (2009) *'DRM, a Design Research Methodology'*, London: Springer London.

Bruun, H. C. and Mortensen, N. H. (2012) 'Visual Product Architecture Modelling for Structuring Data in a PLM System', *9th International Conference on Product Lifecycle Management - PLM 12*, Montreal, Canada.

Burr, H., Deubel, T., Vielhaber, M., Haasis, S. and Weber, C. (2003) 'Challenges for CAx and EDM in an International Automotive Company', *International Conference on Engineering Design - ICED 03*, Stockholm, Sweden.

Burr, H., Vielhaber, M., Deubel, T., Weber, C. and Haasis, S. (2004) 'CAx/EDM integration - Enabler for methodical benefits in the design process', *Proceedings of the 8th International Design Conference - Design 2004,* vol. 1-3 pp. 833-840.

Buur, J. (1990) 'A Theoretical Approach to Mechatronics Design', Ph.D. thesis, Department of Engineering Design, Technical University of Denmark, Lyngby, Denmark.

Canbaz, B., Yannou, B. and Yvars, P.-A. (2014) 'Improving Process Performance of Distributed Set-Based Design Systems by Controlling Wellbeing Indicators of Design Actors', *Journal of Mechanical Design*, vol. 136, no. 2, pp. 021005.

Catic, A. (2011) 'Knowledge-based Engineering in Product Development Processes - Process, IT and Knowledge Management perspectives', Ph.D. thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Catic, A. and Malmqvist, J. (2010) 'Implementing a Wiki to Capture and Share Engineering Knowledge', *Proceedings of Proceedings of NordDesign 2010*, Gothenburg, Sweden.

CIMdata (2010) 'Product Lifecycle Management (PLM) Definition', Retrieved March 2, 2011, from http://www.cimdata.com/plm/definition.html.

Claesson, A. (2006) 'A Configurable Component Framework Supporting Platform-Based Product Development', Ph.D. thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Claesson, A., Gedell, S. and Johannesson, H. (2001) 'Platform Product Development: Product Model - A System Structure Composed of Configurable Components', *Design Engineering Technical Conferences and Computers and Information in Engineering Conference - ASME 2001 DETC/CIE*, September 9-12, Pittsburg, Pennsylvania, USA.

Clarkson, P. J., Simons, C. and Eckert, C. (2004) 'Predicting Change Propagation in Complex Design', *Journal of Mechanical Design*, vol. 126, no. 5, pp. 788-797.

Corin Stig, D. and Bergsjö, D. (2011) 'Means for Internal Knowledge Reuse in Pre-Development – The Technology Platform Approach', *International Conference on Engineering Design - ICED11*, August 15-18, Copenhagen, Denmark.

Corin Stig, D., Högman, U. and Bergsjö, D. (2011) 'Assessment of Readiness for Internal Technology Transfer–A Case Study', *INCOSE 2011*, June 20-23, Denver, USA.

Dave, B. and Koskela, L. (2009) 'Collaborative Knowledge Management - A Construction Case Study', *Automation in Construction*, vol. 18, no. 7, pp. 894-902.

Decker, B., Ras, E., Rech, J., Klein, B., Reuschling, C., Höcht, C., Kilian, L., Traphöner, R. and Haas, V. (2005) 'A Framework for Agile Reuse in Software Engineering Using Wiki Technology', *Proceedings of Conference Professional Knowledge Management - Experiences and Visions*, pp 411-414, Kaiserslautern, Germany.

Du, G., Jiao, R. J. and Chen, M. (2014) 'Joint optimization of product family configuration and scaling design by Stackelberg game', *European Journal of Operational Research*, vol. 232, no. 2, pp. 330-341.

Duffy, A. and Ferns, A. (1999) 'An analysis of design reuse benefits', *Proceedings of the 12th International Conference on Engineering Design - ICED '99*, pp 799-804.

Duffy, A. H. B. and Andreasen, M. M. (1995) 'Enhancing the Evolution of Design Science', *International Conference on Engineering Design - ICED '95*, August 22-24, Prague, Czech Republic.

Duguay, C. R., Landry, S. and Pasin, F. (1997) 'From Mass Production to Flexible/Agile Production', *International Journal of Operations & Production Management*, vol. 17, no. 11-12, pp. 1183-1195.

Dutta, D. and Wolowicz, J. P. (2005) 'An Introduction to Product Lifecycle Management (PLM)', *Proceedings of The 12th ISPE International Conference on Concurrent Engineering (CE2005): Research and Applications*, Dallas, USA.

Eckert, C., Stacey, M. and Clarkson, P. J. (2004) 'The Lure of the Measurable in Design Research', *Proceedings of the 8th International Design Conference- DESIGN 2004*, Dubrovnik, Croatia, pp 21-26.

Eckert, C., Zanker, W. and Clarkson, P. J. (2001) 'Aspects of a better understanding of changes', *International Conference on Engineering Design - ICED 2001*, August 21-23, Glasgow, United Kingdom.

Eckert, C. M., Stacey, M. and Clarkson, P. (2003) 'The Spiral of Applied Research: a Methodological View on Integrated Design Research', *Proceedings of the 14th International Conference on Engineering Design (ICED'03),* 19-21 August 2003, Stockholm, Sweden.

Edholm, P., Johannesson, H. and Söderberg, R. (2010) 'Geometry Interactions in Configurable Platform Models', *The 11th International Design Conference International Design Conference - DESIGN 2010*, May 17-20, Dubrovnik, Croatia.

Edholm, P., Wahl, A. L., Johannesson, H. and Söderberg, R. (2009) 'Knowledge Based Configuration of Integrated Product and Process Platform', *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2009*, August 30-September 2, San Diego, California, USA.

Eger, T., Eckert, C. M. and P, J. C. (2007) 'Engineering change analysis during ongoing product development', *International Conference on Engineering Design - ICED 2007*, August 28-31, Paris, France.

Erens, F. J. (1996) 'The Synthesis of Variety: Developing Product Families', Ph.D thesis, Enidhoven University of Technology, Eindhoven, The Netherlands.

Erixon, G. (1998) 'Modular Function Deployment - A Method for Product Modularisation', Ph.D. thesis, Depatment of Manufacturing Systems, The Royal Institute of Technology, Stockholm, Sweden.

Erixon, G., von Yxkull, A. and Arnström, A. (1996) 'Modularity - the Basis for Product and Factory Reengineering', *CIRP Annals - Manufacturing Technology*, vol. 45, no. 1, pp. 1-6.

Farrell, R. S. and Simpson, T. W. (2003) 'Product Platform Design to Improve Commonality in Custom Products', *Journal of Intelligent Manufacturing*, vol. 14, no. 6, pp. 541-556.

Fielding, E. A., McCardle, J. R., Eynard, B., Hartman, N. and Fraser, A. (2014) 'Product lifecycle management in design and engineering education: International perspectives', *Concurrent Engineering*, vol. 22, no. 2, pp. 123-134.

Finch, W. W. and Ward, A. C. (1997) 'A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty', *Proceedings of Proceedings of the 1997 ASME Design Engineering Technical Conferences*, September 14–17, 1997, Sacramento, California, USA.

Fisher, M., Ramdas, K. and Ulrich, K. (1999) 'Component Sharing in the Management of Product Variety: a Study of Automotive Braking Systems', *Management Science*, vol. 45, no. 3, pp. 297-315.

Fujita, K., Sakaguchi, H. and Akagi, S. (1999) 'Product variety deployment and its optimization under modular architecture and module commonalization', *Proceedings of Proceedings of the 1999 ASME design engineering technical conferences*, September 12-15, Las Vegas, Nevada, USA.

Gedell, S. (2011) 'Efficient Means for Platform-Based Development - Emphasizing Integrated, Information-Rich System Models', Ph.D thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Gedell, S., Michaelis, M. T. and Johannesson, H. (2011) 'Integrated Model For Co-Development of Products and Production Systems - A Systems Theory Approach', *Concurrent Engineering: Research and Applications*, vol. 19, no. 2, pp. 139-156.

Gershenson, J. K., Prasad, G. J. and Zhang, Y. (2003) 'Product Modularity: Definitions and Benefits', *Journal of Engineering Design*, vol. 14, no. 3, pp. 295-313.

Gibbons, M. and Johnson, C. (1982) 'Science, Technology, and the Development of the Transistor', in B. Barnes and D. Edge (ed.) *Science in Context*: MIT Press, Cambridge, MA.

Gonzalez-Zugasti, J. P. and Otto, K. N. (2000) 'Modular platform-based product family design', *Design Engineering Technical Conferences and Computers and Information in Engineering Conference - ASME 2001 DETC/CIE*, 10-13 September, Baltimore, Maryland, USA.

Granstrand, O. (1998) 'Towards a theory of the technology-based firm', *Research policy*, vol. 27, no. 5, pp. 465-489.

Grieves, M. (2006) *'Product Lifecycle Management: Driving the Next Generation of Lean Thinking'*, New York, New York, USA: McGraw-Hill.

Gu, P., Hashemian, M. and Nee, A. (2004) 'Adaptable design', *CIRP Annals-Manufacturing Technology*, vol. 53, no. 2, pp. 539-557.

Gu, P. and Sosale, S. (1999) 'Product modularization for life cycle engineering', *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 5, pp. 387-401.

Gu, P., Xue, D. and Nee, A. (2009) 'Adaptable design: concepts, methods, and applications', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, no. 11, pp. 1367-1387.

Hansen, C. T. and Andreasen, M. M. (2002) 'Two Approaches to Synthesis Based on the Domain Theory', in A. Chakrabarti (ed.) *Engineering Design Synthesis,* London: Springer-Verlag.

Harlou, U. (2005) 'Developing Product Families Based on Architectures: Contribution to a Theory of Product Families', Ph.D. thesis, Department of Mechanical Engineering, Technical University of Denmark, Lyngby.

Haug, A., Hvam, L. and Mortensen, N. H. (2012) 'Definition and Evaluation of Product Configurator Development Strategies', *Computers in Industry*, vol. 63, no. 5, pp. 471–481.

Hauptman, O. and Hirji, K. K. (1996) 'The influence of process concurrency on project outcomes in product development: an empirical study of cross-functional teams', *IEEE Transactions on Engineering Management*, vol. 43, no. 2, pp. 153-164.

Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K. and Ernst, R. (2005) 'System level performance analysis - the SymTA/S approach', *IEEE Proceedings-Computers and Digital Techniques*, vol. 152, no. 2, pp. 148-166.

Hernandez, G., Allen, J. K. and Mistree, F. (2003) 'Platform Design for Customizable Products as a Problem of Access in a Geometric Space', *Engineering Optimization*, vol. 35, no. 3, pp. 229-254.

Hicks, B. J., Culley, S. J., Allen, R. D. and Mullineux, G. (2002) 'A Framework for the Requirements of Capturing, Storing and Reusing Information and Knowledge in Engineering Design', *International Journal of Information Management*, vol. 22, no. 4, pp. 263-280.

Högman, U. (2011) 'Processes and Platforms Aligned with Technology Development: the Perspective of a Supplier in the Aerospace Industry', Ph.D thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Högman, U. and Berglund, F. (2007) 'Technology Management Challenges for a Sub-supplier in the Aerospace Industry', *16th International Conference on Management of Technology - IAMOT 2007*, May 13-17, Miami Beach, Florida, USA.

Högman, U., Bergsjö, D., Anemo, M. and Persson, H. (2009) 'Exploring the potential of applying a platform formulation at supplier level-The case of Volvo Aero Corporation', *Proceedings of the 17th International Conference on Engineering Design- ICED 2009*, August 24-27, Palo Alto, California, USA.

Holmdahl, L. (2010) *'Lean Product Development på Svenska'*, Gothenburg, Sweden: Lars Holmdahl.

Hubka, V. and Eder, W. E. (1988) *'Theory of Technical Systems – A Total Concept Theory for Engineering Design'*, Berlin, Germany: Springer.

Inns, T. G. and Neville, P. (1998) 'Establishing a Company-Level Design Process to Facilitate Design Reuse', *Proceedings of Engineering Design Conference '98 on Design Reuse*, Brunel University, UK.

Inoue, M., Nahm, Y. E., Okawa, S. and Ishikawa, H. (2010) 'Design Support System by Combination of 3D-CAD and CAE with Preference Set-based Design Method', *Concurrent Engineering: Research and Applications*, vol. 18, no. 1, pp. 41-53.

Jarratt, T., Clarkson, J. and Eckert, C. (2005) 'Engineering change', in *Design process improvement*, London: Springer.

Jarratt, T., Eckert, C., Caldwell, N. and Clarkson, P. (2011) 'Engineering change: an overview and perspective on the literature', *Research in Engineering Design*, vol. 22, no. 2, pp. 103-124.

Jiao, J., Lianfeng, Z. and Pokharel, S. (2007a) 'Process Platform Planning for Variety Coordination From Design to Production in Mass Customization Manufacturing', *Engineering Management, IEEE Transactions on*, vol. 54, no. 1, pp. 112-129.

Jiao, J. and Tseng, M. M. (1999) 'A Methodology of Developing Product Family Architecture for Mass Customization', *Journal of Intelligent Manufacturing*, vol. 10, no. 1, pp. 3-20.

Jiao, J. R. and Chen, C.-H. (2006) 'Customer requirement management in product development: a review of research issues', *Concurrent Engineering: Research and Applications*, vol. 14, no. 3, pp. 173-185.

Jiao, R. J., Simpson, T. and Siddique, Z. (2007b) 'Product family design and platform-based product development: a state-of-the-art review', *Journal of Intelligent Manufacturing*, vol. 18, no. 1, pp. 5-29.

Johannesson, H. (2014) 'Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms', in T. W. Simpson, J. R. Jiao, Z. Siddique and K. Hölttä-Otto (ed.) *Advances in Product Family and Product Platform Design*, New York: Springer.

Jolly, D. R. and Nasiriyar, M. (2007) 'Technology Platform Exploitation: Definition and Research Boundaries', *16th International Conference on Management of Technology - IAMOT 2007*, May 13-17, Miami, Florida, USA.

Jørgensen , K. A. (1992) 'Videnskablige arbejdsparadigmer ("Scinetific Work Paradigms", in Danish)', Institute for Production, Aalborg University Center, Aalborg, Denmark.

Kennedy, B. M., Sobek, D. K. and Kennedy, M. N. (2013) 'Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process', *Systems Engineering*, vol. 17 no. 3 pp. 278-296.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H. (1999) 'Reconfigurable manufacturing systems', *CIRP Annals-Manufacturing Technology*, vol. 48, no. 2, pp. 527-540.

Kovacs, Z., Le Goff, J. and McClatchey, R. (1998) 'Support for product data from design to production', *Computer Integrated Manufacturing Systems*, vol. 11, no. 4, pp. 285-290.

Li, Y., Xue, D. and Gu, P. (2008) 'Design for product adaptability', *Concurrent Engineering: Research and Application*, vol. 16, no. 3, pp. 221-232.

Liker, J. K., Sobek, D. K., Ward, A. C. and Cristiano, J. J. (1996) 'Involving suppliers in product development in the United States and Japan: evidence for set-based concurrent engineering', *Engineering Management, IEEE Transactions on*, vol. 43, no. 2, pp. 165-178.

Loch, C. H. and Terwiesch, C. (1999) 'Accelerating the Process of Engineering Change Orders: Capacity and Congestion Effects', *Journal of Product Innovation Management*, vol. 16, no. 2, pp. 145-159.

Malak Jr, R. J., Aughenbaugh, J. M. and Paredis, C. J. (2009) 'Multi-attribute utility analysis in set-based conceptual design', *Computer-Aided Design*, vol. 41, no. 3, pp. 214-227.

Männistö, T., Peltonen, H., Martio, A. and Sulonen, R. (1998) 'Modelling generic product structures in STEP', *Computer-Aided Design*, vol. 30, no. 14, pp. 1111-1118.

Männistö, T., Peltonen, H., Soininen, T. and Sulonen, R. (2001) 'Multiple Abstraction Levels in Modelling Product Structures', *Data & Knowledge Engineering*, vol. 36, no. 1, pp. 55-78.

Matsumoto, I. T., Stapleton, J., Glass, J. and Thorpe, T. (2005) 'A Knowledge-Capture Report for Multidisciplinary Design Environments', *Journal of Knowledge Management*, vol. 9, no. 3, pp. 83-92.

Maxwell, J. A. (1996) *'Qualitative Research Design : an Interative Approach'*, Thousand Oaks, California; London: Sage.

McGrath, M. E. (2001) *'Product Strategy for High-Technology Companies: Accelerating Your Business to Web Speed'*, New York ; London: McGraw-Hill.

Mesihovic, S. and Malmqvist, J. (2004) 'A Process-Oriented Approach for Management of Product Configuration Models', *Proceedings of ASME 2004*, pp. 691-701, Salt Lake City, UT, USA.

Meyer, M. H. and Lehnerd, A. P. (1997) *'The Power of Product Platforms: Building Value and Cost Leadership'*, New York, New York, USA: Free Press.

Meyer, M. H. and Utterback, J. M. (1992) 'Core Competencies, Product Families and Sustained Business Success', MIT Sloan School of Management International Center for Research on the Management of Technology, Sloan WP# 3410-92, Cambridge, Massachusetts, USA.

Michaelis, M. T. (2011) 'Co-Development of Products and Manufacturing Systems - Applying a Configurable Platform Approach', Licentiate thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Michaelis, M. T. (2013) 'Co-Development of Products and Manufacturing Systems Using Integrated Platform Models', Ph.D. thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Michaelis, M. T. and Johannesson, H. (2011) 'Platform Approaches in Manufacturing: Considering Integration With Product Platforms', *Proceedings of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - IDETC CIE ASME 2011,* August 28-31, Washington DC, USA.

Michaelis, M. T., Johannesson, H. and ElMaraghy, H. A. (2014) 'Function and Process Modeling for the Co-Development of Products and Manufacturing Systems', *Journal of Manufacturing Systems*.

Ming, X. G., Yan, J. Q., Lu, W. F. and Ma, D. Z. (2005) 'Technology Solutions for Collaborative Product Lifecycle Management - Status Review and Future Trend', *Concurrent Engineering: Research and Applications*, vol. 13, no. 4, pp. 311-319.

Morgan, J. M. and Liker, J. K. (2006) *'The Toyota Product Development System: Integrating People, Process And Technology'*, New York: Taylor & Francis.

Mortensen, N. (1999) 'Design modeling in a designer's workbench–contribution to a design grammar', Ph.D thesis, Department of Control and Engineering Design, Technical University of Denmark, Copenhagen, Denmark.

Mortensen, N. H., Harlou, U. and Haug, A. (2008) 'Improving Decision Making in the Early Phases of Configuration Projects', *International Journal of Industrial Engineering-Theory Applications and Practice*, vol. 15, no. 2, pp. 185-194.

NASA (2010) 'Overview of the Wright Brothers Invention Process', Retrieved May 26, 2014, from http://wright.nasa.gov/overview.htm.

Nobelius, D. (2002) 'Managing R&D Processes - Focusing on Technology Development, Product Development, and their Interplay', Ph.D. thesis, Operations Management and Work Organization, Chalmers University of Technology, Gothenburg, Sweden.

Olesen, J. (1992) 'Concurrent Development in Manufacturing - based upon dispositional mechanisms', Ph.D thesis, Insitute of Engineering Design, Technical University of Denmark, Copenhagen.

Ong, S., Xu, Q. and Nee, A. (2006) 'Design reuse methodology for product family design', *CIRP Annals-Manufacturing Technology*, vol. 55, no. 1, pp. 161-164.

Otto, K. N. and Antonsson, E. K. (1991) 'Trade-off strategies in engineering design', *Research in Engineering Design*, vol. 3, no. 2, pp. 87-103.

Pahl, G. and Beitz, W. (1996) *'Engineering Design: a Systematic Approach'*, New York: Springer.

Pahl, G., Beitz, W. and Wallace, K. (1988) *'Engineering Design : a Systematic Approach'*, New York: Springer-Verlag.

Pedersen, R. (2009) 'Product Platform Modelling - Contributions to the discipline of visual product platform modelling', Ph.D. thesis, DTU Management, Technical University of Denmark, Lyngby, Denmark.

Pikosz, P. and Malmqvist, J. (1998) 'A comparative study of engineering change management in three Swedish engineering companies', *Proceedings of ASME 1998 Design Engineering Technical Conference*, vol 1, Atlanta, GA, USA.

Prasad, B. (1996) *'Concurrent Engineering Fundamentals'*, Upper Saddle River, N.J.: Prentice Hall PTR.

Pugh, S. (1990) *'Total Design: Integrated Methods for Successful Product Engineering'*, Wokingham: Addison-Wesley.

Qureshi, A. J., Dantan, J.-Y., Bruyere, J. and Bigot, R. (2010) 'Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm', *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1173-1186.

Raudberget, D. (2011) 'Enabling Set-Based Concurrent Engineering in Traditional Product Development', *Proceedings of ICED 2011*, 15-18 August, Copenhagen, Denmark.

Raudberget, D. and Sunnersjö, S. (2010) 'Experiences of Set-Based Concurrent Engineering in Four Product Developing Companies', *Proceedings of the TMCE 2010*, 12–16 April, Ancona, Italy.

Reidelbach, M. A. (1991) 'Engineering change management for long-lead-time production environments', *Production and Inventory Management Journal*, vol. 32, no. 2, pp. 84-88.

Robertson, D. and Ulrich, K. (1998) 'Planning for Product Platforms', *Sloan Management Review*, vol. 39, no. 4, pp. 19-31.

Robson, C. (2002) *'Real World Research: a Resource for Social Scientists and Practitioner-Researchers'*, Madden, Mass. ; Oxford: Blackwell Publishers.

Rouibah, K. and Caskey, K. (2003a) 'A workflow system for the management of inter-company collaborative engineering processes', *Journal of Engineering Design*, vol. 14, no. 3, pp. 273-293.

Rouibah, K. and Caskey, K. (2003b) 'Change management in concurrent engineering from a parameter perspective', *Computers in Industry*, vol. 50, no. 1, pp. 15-34.

Schachinger, P. and Johannesson, H. (2000) 'Computer Modelling of Design Specifications', *Journal of Engineering Design*, vol. 11, no. 4, pp. 317-329.

Schilling, M. A. and Steensma, H. K. (2001) 'The Use of Modular Organizational Forms: An Industry-Level Analysis', *Academy of Management Journal*, vol. 44, no. 6, pp. 1149-1168.

Shahin, T. M. M., Andrews, P. T. J. and Sivaloganathan, S. (1999) 'A Design Reuse System', *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, vol. 213, no. 6, pp. 621-627.

Shapiro, A. R. (2006) 'Measuring Innovation: Beyond Revenue from New Products', *Research-Technology Management*, vol. 49, no. 6, pp. 42-51.

Simpson, T. W., Maier, J. R. and Mistree, F. (2001) 'Product platform design: method and application', *Research in engineering Design*, vol. 13, no. 1, pp. 2-22.

Singer, D. J., Doerry, N. and Buckley, M. E. (2009) 'What Is Set-Based Design?', *Naval Engineers Journal*, vol. 121, no. 4, pp. 31-43.

Sobek, D. K. (1997) 'Principles that Shape Product Development Systems: a Toyota-Chrysler Comparison', Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, USA.

Sobek, D. K., Ward, A. C. and Liker, J. K. (1999) 'Toyota's principles of set-based concurrent engineering', *Sloan Management Review*, vol. 40, no. 2, pp. 67-83.

Stark, J. (2005) *'Product Lifecycle Management: 21st Century Paradigm for Product Realisation'*, London, UK: Springer.

Suh, N. P. (1990) *'The Principles of Design'*, New York, New York, USA: Oxford University Press.

Svensson, D. and Malmqvist, J. (2002) 'Strategies for Product Structure Management at Manufacturing Firms', *Journal of Computing and Information Science in Engineering*, vol. 2, no. 1, pp. 50-58.

Svensson, D., Malmström, J., Pikosz, P. and Malmqvist, J. (1999) 'A Framework for Modelling and Analysis of Engineering Information Systems', *ASME Design Engineering Technical Conferences*, September 12-15, Las Vegas, Nevada, USA.

Takeishi, A. (2002) 'Knowledge Partitioning in the Interfirm Division of Labor: The Case of Automotive Product Development', *Organization Science*, vol. 13, no. 3, pp. 321-338.

Terzi, S., Bouras, A., Dutta, D., Garetti, M. and Kiritsis, D. (2010) 'Product Lifecycle Management – from its History to its New Role', *International Journal of Product Lifecycle Management*, vol. 4, no. 4, pp. 360-389.

Thiele, L., Chakraborty, S., Gries, M. and Kunzli, S. (2002) 'A framework for evaluating design tradeoffs in packet processing architectures', *Proceedings of the 39th Design Automation Conference 2002*, 10-14 June, New Orleans, Louisiana, USA.

Thiele, L., Chakraborty, S., Gries, M., Maxiaguine, A. and Greutert, J. (2001) 'Embedded Software in Network Processors — Models and Algorithms', in T. Henzinger and C. Kirsch (ed.) *Embedded Software,* Berlin, Germany: Springer.

Tseng, M. and Jiao, J. (2001) 'Mass Customization', in G. Salvendy (ed.) *Handbook of Industrial Engineering, Technology and Operation Management, 3rd Edition*: New York, NY: Wiley.

Ulrich, K. T. and Eppinger, S. D. (2008) *'Product design and development'*, New York: McGraw-Hill Higher Education ; London : McGraw-Hill [distributor].

Ulrich, K. T. and Eppinger, S. D. (2012) *'Product design and development'*, New York: McGraw-Hill/Irwin.

van Veen, E. A. (1991) 'Modeling Product Structures by Generic Bills-Of-Material', Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

Vincent, T. L. (1983) 'Game theory as a design tool', *Journal of Mechanisms, Transmissions and Automation in Design*, vol. 105, no. 2, pp. 165-170.

Wahl, A. L. and Johannesson, H. (2010) 'Managing Design Change in Configurable Component Based Product Platforms', *Proceedings of the 8th Biannual Conference NordDesign 2010*, 25-27 August, Gothenburg, Sweden.

Ward, A. C. (2007) *'Lean Product and Process Development'*, Cambridge, Massachusetts, USA: Lean Enterprise Institute.

Wheelwright, S. C. and Clark, K. B. (1992) *'Revolutionizing Product Development : Quantum Leaps in Speed, Efficiency, and Quality'*: Free Press: New York, NY, USA.

Wiendahl, H.-H. (2009) 'Adaptive Production Planning and Control - Elements and Enablers of Changeability', in H. A. ElMaraghy (ed.) *Changeable and reconfigurable manufacturing systems*: Springer.

Williams, C. B., Allen, J. K., Rosen, D. W. and Mistree, F. (2007) 'Designing Platforms for Customizable Products and Processes in Markets of Non-Uniform Demand', *Concurrent Engineering: Research and Applications*, vol. 15, no. 2, pp. 201-216.

Womack, J. P., Jones, D. T. and Roos, D. (2007) *'The machine that changed the world: The story of lean production--Toyota's secret weapon in the global car wars that is now revolutionizing world industry',* New York: Simon and Schuster.

Wood Iii, W. H. and Agogino, A. M. (1996) 'Case-Based Conceptual Design Information Server for Concurrent Engineering', *Computer-Aided Design*, vol. 28, no. 5, pp. 361-369.

Wortmann, H. and Alblas, A. (2009) 'Product platform life cycles: a multiple case study', *International Journal of Technology Management*, vol. 48, no. 2, pp. 188-201.

Wynn, D. C., Wyatt, D. F., Nair, S. M. and Clarkson, P. J. (2010) 'An introduction to the Cambridge advanced modeller', *Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes (MMEP 2010)*, July 19-20, Cambridge, UK.

Xu, Q., Ong, S. and Nee, A. (2006) 'Function-based design synthesis approach to design re-use', *Research in engineering design*, vol. 17, no. 1, pp. 27-44.

Xu, Q., Ong, S. and Nee, A. (2007) 'Evaluation of product performance in product family design re-use', *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4119-4141.

Xue, D., Hua, G., Mehrad, V. and Gu, P. (2012) 'Optimal adaptable design for creating the changeable product based on changeable requirements considering the whole product life-cycle', *Journal of Manufacturing Systems*, vol. 31, no. 1, pp. 59-68.

Yannou, B., Simpson, T. W. and Barton, R. R. (2003) 'Towards a conceptual design explorer using metamodeling approaches and constraint programming', *Proceedings of ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, September 2–6, 2003, Chicago, Illinois, USA.

Yin, R. K. (2003) *'Case Study Research: Design and Methods'*, Thousand Oaks, California, USA: Sage.

Yvars, P.-A. and Duhau, Q. (2012) 'Managing system configuration through constraint propagation in function and value analyses of product families', *Journal of Engineering Design*, vol. 23, no. 5, pp. 361-377.

Zhang, L. L. and Rodrigues, B. (2009) 'A tree unification approach to constructing generic processes', *IIE Transactions*, vol. 41, no. 10, pp. 916-929.

Zimmerman, T. (2008) 'Implementing PLM Across Organisations', Ph.D thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.